



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
 Πρόγραμμα Μεταπτυχιακών Σπουδών
 «Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	LibrApp: Μία συνεργατική ψηφιακή βιβλιοθήκη βασισμένη στο Android Librapp: A collaborative Digital Library on the Android platform
Όνοματεπώνυμο Φοιτητή	Μιχαήλ Ρόδιος
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΣΠ/ 10021
Υπεύθυνος καθηγητής	Χρήστος Δουληγέρης, Καθηγητής
Επιβλέπων ερευνητής	Δρ. Σαράντης Μητρόπουλος

Ημερομηνία Παράδοσης **Απρίλιος 2014**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Δουληγέρη Χρήστο για την ανάθεση της μεταπτυχιακής διατριβής, καθώς και τον Δρ. Μητρόπουλο Σαράντη για την επίβλεψη, και την επιστημονικά άρτια καθοδήγησή του κατά την διάρκεια του σχεδιασμού και της ανάπτυξης αυτής της εργασίας, αλλά και για τις καίριες συμβουλές που μου έδωσε για την βελτίωση του συστήματος που αναπτύχθηκε.

Θα ήθελα ακόμα να ευχαριστήσω την οικογένειά μου, για την ευκαιρία που μου έδωσε να πραγματοποιήσω το συγκεκριμένο πρόγραμμα μεταπτυχιακών σπουδών, και την Μπαγλατζή Σοφία, για την στήριξή της και την υπομονή της καθ' όλη την εκπόνηση της εργασίας, καθώς και για την βοήθεια της στην επιμέλεια του τελικού κειμένου.

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή έχει ως αντικείμενο την ανάπτυξη ενός συστήματος ψηφιακής βιβλιοθήκης, το οποίο γίνεται διαθέσιμο στους χρήστες μέσω μίας εφαρμογής για έξυπνες συσκευές κινητών τηλεφώνων. Η εφαρμογή είναι προσανατολισμένη και γραμμένη για το λειτουργικό σύστημα Android, ένα από τα πιο πολυχρησιμοποιημένα συστήματα.

Το συγκεκριμένο σύστημα έχει ως στόχο την αύξηση της αποδοτικότητας και της συνεργατικότητας των χρηστών μίας παραδοσιακής βιβλιοθήκης ή ενός ιδρύματος που παρέχει λειτουργίες βιβλιοθήκης στους χρήστες/υπαλλήλους. Αυτό επιτυγχάνεται προσφέροντας υπηρεσίες όπως chat, διασύνδεση με άλλους χρήστες, συστήματα διαχείρισης κ.α.

Για την υλοποίηση της πλατφόρμας χρησιμοποιήθηκαν σύγχρονες τεχνολογίες υλοποίησης τόσο από την μεριά της εφαρμογής, όσο και από την μεριά της βάσης δεδομένων η οποία περιέχει το υλικό που διαχειρίζεται η εφαρμογή. Πέραν από το Android API που χρησιμοποιήθηκε για τον σχεδιασμό και την υλοποίηση της εφαρμογής έγινε χρήση του JSON για την ασύγχρονη επικοινωνία με τον απομακρυσμένο εξυπηρετητή και χρήση PHP 5.0 για την συγγραφή των web services που επικοινωνούν με την βάση (MySQL).

Η εφαρμογή προσφέρει εκτός από τις βασικές υπηρεσίες ψηφιακής βιβλιοθήκης (αναζήτηση και αποθήκευση βιβλίων), κλασικές συνεργατικές υπηρεσίες (σύνδεση με φίλους, chat, δημιουργία εργασίας σε wiki) και μία λειτουργία διαχείρισης καθηκόντων.

Τέλος, για να διερευνηθεί η χρησιμότητα της εφαρμογής, έγινε μία έρευνα με ερωτηματολόγιο, η οποία εξετάζει το πόσο ωφελεί τους χρήστες από άποψη αποδοτικότητας. Τα αποτελέσματα της έρευνας αναλύθηκαν βάση ενός μαθηματικού μοντέλου αποδοτικότητας και φαίνεται να είναι ενθαρρυντικά για την περαιτέρω ανάπτυξη και την χρήση μιας τέτοιας εφαρμογής.

Λέξεις – Κλειδιά: Android, Ψηφιακές Βιβλιοθήκες, JSON, Συνεργατικότητα, Task Management

Abstract

This MSc thesis aims to develop a digital library system, which is made available to users through an application for smart mobile devices. The application is oriented towards and written for the Android operating system, one of the most widely used systems.

This system's target is to increase the efficiency and the collaboration of a traditional library or an institution that provides library functions to its users / employees. This is achieved by offering services such as chat, interaction with other users, management systems etc.

For the implementation of the platform we used modern technologies both at the application side, and at the side of the database that contains the material that manages the application. Apart from the Android API used for the design and implementation of the application we used JSON for the asynchronous communication with the remote server and PHP 5.0 for writing the web services that communicate with the base (MySQL).

The application offers besides the main digital library services (search and store books), traditional collaboration services (connecting with friends, chat, create work in wiki) and a task management function.

Finally, to investigate the usefulness of the application, a survey was conducted which examines how users benefit in terms of efficiency. The survey results were analyzed based on a mathematical model for evaluating efficiency and provided encouraging results for the further development of the application.

Keywords: Android, Digital Libraries, JSON, Collaboration, Task Management

Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ	3
ΠΕΡΙΛΗΨΗ	4
ABSTRACT	4
ΛΙΣΤΑ ΠΙΝΑΚΩΝ	9
1. ΕΙΣΑΓΩΓΗ	10
1.1. ΣΚΟΠΟΣ ΤΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΑΤΡΙΒΗΣ	10
1.2. ΔΟΜΗ ΤΗΣ ΔΙΑΤΡΙΒΗΣ ΚΑΙ ΜΕΘΟΔΟΛΟΓΙΑ ΠΟΥ ΑΚΟΛΟΥΘΗΘΗΚΕ	11
2. ΨΗΦΙΑΚΕΣ ΒΙΒΛΙΟΘΗΚΕΣ	12
2.1. ΟΡΙΣΜΟΣ	12
2.2. ΔΙΑΚΡΙΣΗ ΑΠΟ ΤΙΣ ΚΟΙΝΕΣ ΒΙΒΛΙΟΘΗΚΕΣ	13
2.3 ΔΟΜΗ ΜΙΑΣ ΨΗΦΙΑΚΗΣ ΒΙΒΛΙΟΘΗΚΗΣ	14
2.3.1 Δομή της αποθήκης δεδομένων	15
2.3.2 Διεπαφές χρήστη	17
2.3.3 Δομή της πληροφορίας	20
2.3.4 Ψηφιακά ράφια και OPAC	21
2.3.5 XML, Marc και Μεταδεδομένα	21
2.4 ΟΙ ΨΗΦΙΑΚΕΣ ΒΙΒΛΙΟΘΗΚΕΣ ΤΩΡΑ	22
2.4.1. Γνωστές ψηφιακές βιβλιοθήκες	23
2.4.2 Συνεργατικότητα σε ψηφιακές βιβλιοθήκες	25
2.4.3 Μελλοντική Έρευνα πάνω στον τομέα των ψηφιακών βιβλιοθηκών	28
3. ΚΙΝΗΤΕΣ ΕΦΑΡΜΟΓΕΣ ΚΑΙ ΤΟ ΣΥΣΤΗΜΑ ANDROID	30
3.1 ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID	32
3.1.1 Η αρχιτεκτονική του Android	32
3.1.2. Εφαρμογές Android	35
4. ΑΝΑΛΥΣΗ ΤΩΝ ΛΕΙΤΟΥΡΓΙΩΝ ΚΑΙ ΤΩΝ ΤΕΧΝΟΛΟΓΙΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	44
4.1. ΠΕΡΙΓΡΑΦΗ ΛΟΙΠΩΝ ΤΕΧΝΟΛΟΓΙΩΝ	44
4.1.1. Το πρότυπο JSON	44
4.1.2. Η γλώσσα προγραμματισμού PHP	46
4.1.3. Βάση δεδομένων MySQL	46
4.1.4. Το JQuery Framework	47
4.2. ΜΟΝΤΕΛΟ ΠΕΛΑΤΗ/ΕΞΥΠΗΡΕΤΗΤΗ ΚΑΙ ΑΛΛΕΣ ΤΕΧΝΙΚΕΣ	48
4.2.1. Το μοντέλο πελάτη/εξυπηρετητή	48
4.2.2. Η αρχιτεκτονική MVC	49
4.3. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ	51
4.4. ΣΧΕΔΙΑΣΜΟΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	52

5.Η ΕΦΑΡΜΟΓΗ LIBRAPP	60
5.1.ΧΡΗΣΗ ΤΩΝ ΤΕΧΝΟΛΟΓΙΩΝ	60
5.1.1.Client	60
5.1.2.Server	60
5.2.ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	61
5.2.1.Η εφαρμογή LibrApp στην κινητή συσκευή	61
5.2.2.Η εφαρμογή των διαχειριστών	72
6.ΑΞΙΟΛΟΓΗΣΗ ΕΦΑΡΜΟΓΗΣ LIBRAPP	74
6.1.ΠΕΡΙΓΡΑΦΗ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ ΑΞΙΟΛΟΓΗΣΗΣ	74
6.1.1.Ανάλυση αποτελεσμάτων ερωτηματολογίου	74
6.2.ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΜΟΝΤΕΛΟΥ	82
6.2.1.Αποτελέσματα του μοντέλου	83
6.3 ΣΥΜΠΕΡΑΣΜΑΤΑ	86
7.ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΤΑΣΕΙΣ	87
ΒΙΒΛΙΟΓΡΑΦΙΑ	88
ΠΑΡΑΡΤΗΜΑ 1	92
A.ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ	92
A.1.Κώδικας εφαρμογής διαχειριστών	92
A.2.Κώδικας ενδιάμεσου λογισμικού	94
A.3.Κώδικας εφαρμογής κινητής συσκευής	97
B.ΣΧΕΔΙΑΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	105
Γ. ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ	107
Γ.1.Εφαρμογή διαχειριστών και ενδιάμεσο λογισμικό	107
Γ.2.Εφαρμογή κινητού τηλεφώνου	107
ΠΑΡΑΡΤΗΜΑ 2	109

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

Σχήμα 1 Η δομή μιας ψηφιακής βιβλιοθήκης (http://www.dlib.org)	14
Σχήμα 2 Δομή μιας αποθήκης δεδομένων	16
Σχήμα 3 Σχέσεις αποθήκης, εφαρμογής και συστήματος τιλοποίησης (www.dlib.org).....	17
Σχήμα 5 Το επίπεδο διαδραστικότητας των μοντέλων σε σχέση με τον αριθμό των χρηστών ..	26
Σχήμα 7 Δομή ενός smartphone (www.eetimes.com)	30
Σχήμα 8 Δραστηριότητες των χρηστών smartphones (www.adage.com).....	31
Σχήμα 9 Η αρχιτεκτονική του Android	33
Σχήμα 11 Παράδειγμα Android Manifest	38
Σχήμα 12 Ο κύκλος ζωής των activities (http://developer.android.com)	39
Σχήμα 13 Καταστάσεις των δραστηριοτήτων (docs.xamarin.com)	40
Σχήμα 14 Δομή αντικειμένου JSON	45
Σχήμα 15 Κατανομή δείγματος με βάση την ηλικία	75
Σχήμα 16 Κατανομή γένους ανά επίπεδο εκπαίδευσης.....	75
Σχήμα 17 Μορφωτικό επίπεδο δείγματος.....	76
Σχήμα 18 Εργασιακή απασχόληση των ερωτηθέντων	76
Σχήμα 19 Βαθμός εξοικείωσης με το Android.....	77
Σχήμα 20 Αναζήτηση περιεχομένου χωρίς σύνδεση	78
Σχήμα 21 Αναζήτηση περιεχομένου μόνο για συνδεδεμένους χρήστες.....	78
Σχήμα 22 Χρησιμότητα λειτουργίας εμφάνισης αντικειμένων	79
Σχήμα 23 Χρησιμότητας της σύνδεσης με φίλους	79
Σχήμα 24 Αξιολόγηση συνεργατικών λειτουργιών	80
Σχήμα 25 Αξιολόγηση λειτουργιών task management	81
Σχήμα 26 Αξιολόγηση ανάγκης για περαιτέρω ανάπτυξη.....	81
Σχήμα 27 Αξιολόγηση εισόδου με social account.....	82
Σχήμα 28 Αποτελέσματα μοντέλου ανά κατηγορία	85
Σχήμα 29 Συνολικά αποτελέσματα μοντέλου	86
Σχήμα 30 Διάγραμμα κλάσης των λειτουργιών με τον server	100
Σχήμα 31 Διάγραμμα κλάσης για της λειτουργίες login και register	103

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1 Παράδειγμα JSon	45
Εικόνα 2 Παράδειγμα κώδικα PHP	46
Εικόνα 3 Η δομή του JQuery	47
Εικόνα 4 Client-Server Αρχιτεκτονική	48
Εικόνα 5 MVC Αρχιτεκτονική	50
Εικόνα 6 MVC αρχιτεκτονική στο σύστημα Android	50
Εικόνα 7 Η αρχιτεκτονική του συστήματος	55
Εικόνα 8 Αρχική οθόνη	61
Εικόνα 9 Οθόνη έναρξης εφαρμογής	61
Εικόνα 10 Μήνυμα λάθους e-mail ή κωδικού	62
Εικόνα 11 Εισαγωγή στοιχείων εγγεγραμμένου χρήστη	62
Εικόνα 12 Ζήτηση ηλεκτρονικής διεύθυνσης για ανάκτηση κωδικού	62
Εικόνα 13 Ενημέρωση αποστολή e-mail	63
Εικόνα 14 Οθόνη εγγραφής νέου χρήστη	63
Εικόνα 15 Συμπληρωμένη φόρμα νέου χρήστη	63
Εικόνα 16 Σύνδεση με server	64
Εικόνα 17 Ήδη χρησιμοποιούμενο όνομα χρήστη	64
Εικόνα 18 Μη αποδεκτό e-mail	64
Εικόνα 19 Μη αποδεκτό όνομα χρήστη	64
Εικόνα 20 Κεντρικό μενού	65
Εικόνα 21 Οθόνη αναζήτησης	65
Εικόνα 22 Αποτελέσματα αναζήτησης	65
Εικόνα 23 Βαθμολόγηση υλικού	66
Εικόνα 24 Αναλυτική εμφάνιση αποτελέσματος	66
Εικόνα 25 Πρόσδος download	66
Εικόνα 26 Download στο background της εφαρμογής	66
Εικόνα 27 Share 2	67
Εικόνα 28 Share 1	67
Εικόνα 29 Προσωπικό "ράφι"	67
Εικόνα 30 Διαχείριση έργου	68
Εικόνα 31 Οθόνη διαχείρισης έργου	68
Εικόνα 32 Οθόνη προσθήκης νέου έργου	68
Εικόνα 33 Ειδοποίηση για συμμετοχή σε νέο έργο 2	69
Εικόνα 34 Ειδοποίηση για συμμετοχή σε νέο έργο 1	69
Εικόνα 35 Αποδοχή πρόσκλησης για συμμετοχή σε νέο έργο	69
Εικόνα 36 Οθόνη διαχείρισης λίστας καθηκόντων	70
Εικόνα 37 Λίστα καθηκόντων	70
Εικόνα 38 Προσθήκη νέας λίστας καθηκόντων	70
Εικόνα 39 Οθόνη συνομιλίας με φίλο	71
Εικόνα 40 Οθόνη διαχείρισης φίλων	71
Εικόνα 41 Διαχείριση προφίλ	71
Εικόνα 42 Οθόνη διαχείρισης	72
Εικόνα 43 Οθόνη αυθεντικοποίησης διαχειριστή	72
Εικόνα 44 Προσθήκη χρήστη	73
Εικόνα 45 Προσθήκη αντικειμένου	73
Εικόνα 46 Android SDK Manager	107
Εικόνα 47 Android Virtual Device Manager	108

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

Πίνακας 1 Λειτουργίες του RAP.....	17
Πίνακας 2 Metadata.....	22
Πίνακας 3 Μοντέρνες Ψηφιακές Βιβλιοθήκες	25
Πίνακας 4 Μερίδιο αγοράς λειτουργικών συστημάτων, Q2 2013 (σε εκατομμύρια \$).....	31
Πίνακας 5 Διάγραμμα ακολουθίας απλών ενεργειών χρήστη	56
Πίνακας 6 Διάγραμμα ακολουθίας συνεργατικών ενεργειών	57
Πίνακας 7 Διάγραμμα ακολουθίας ενεργειών διαχείρισης καθηκόντων	58
Πίνακας 8 Διάγραμμα ακολουθίας ενεργειών διαχειριστή.....	59
Πίνακας 9 Αποτίμηση απαντήσεων ερωτηματολογίου	82

1. Εισαγωγή

Ένα από τα χαρακτηριστικά της σημερινής εποχής είναι η αναζήτηση και η εύρεση χρήσιμων και έγκυρων πληροφοριών. Ο καθένας από εμάς, καθημερινά, αναζητά να ολοκληρώσει τις εργασίες του (επαγγελματικές, ακαδημαϊκές ή απλά ερασιτεχνικές), χρησιμοποιώντας μία πληθώρα εργαλείων αναζήτησης πληροφορίας. Η πρόκληση είναι να μπορέσει ο χρήστης να απομονώσει τις χρήσιμες σε αυτόν πληροφορίες, ώστε να τις χρησιμοποιήσει με επάρκεια, αλλά και να τις μοιραστεί με τους συναδέλφους/συνεργάτες του.

Υπάρχουν πολλοί τρόποι για να ανακτήσει κάποιος πληροφορίες και στοιχεία. Ειδικά με την άνθιση της τεχνολογίας και του παγκόσμιου ιστού τα τελευταία χρόνια, οι πηγές από τις οποίες μπορεί να γίνει εξόρυξη γνώσης έχουν αυξηθεί σημαντικά. Οι πηγές αυτές ποικίλουν, από τις κλασσικές, όπως για παράδειγμα φυσικές βιβλιοθήκες και αρχεία, μέχρι τις πιο τεχνολογικά ανεπτυγμένες. Η άνοδος του διαδικτύου μας εξόπλισε με εργαλεία όπως μηχανές αναζήτησης, αχανείς βάσεις δεδομένων, ιστότοπους, αλλά και με ακόμα πιο περίπλοκα, όπως ψηφιακές βιβλιοθήκες και χώρους αποθήκευσης πληροφορίας σε υπηρεσίες υπολογιστικού νέφους (cloud computing).

Μία ψηφιακή βιβλιοθήκη είναι μία συλλογή ψηφιακών αρχείων, οποιουδήποτε τύπου, τα οποία βρίσκονται αποθηκευμένα σε ένα αποθετήριο ή σε μία βάση δεδομένων. Αποτελεί ουσιαστικά την προσπάθεια μεταφοράς των παραδοσιακών φυσικών βιβλιοθηκών, στον ψηφιακό κόσμο, μία προσπάθεια που ξεκίνησε στις αρχές της δεκαετίας του 1990. Οι ψηφιακές βιβλιοθήκες μπορεί να είναι αυτόνομες ή να αποτελούν προσθήκη σε μια παραδοσιακή βιβλιοθήκη, η οποία με την σειρά της μπορεί να ανήκει σε κάποιο κρατικό ή ακαδημαϊκό οργανισμό, ή σε κάποια ιδιωτική επιχείρηση. Τα τελευταία χρόνια οι περισσότεροι οργανισμοί και επιχειρήσεις έχουν επενδύσει στην απόκτηση κάποιου ηλεκτρονικού αποθετηρίου, μιας και η άμεση πρόσβαση στην πληροφορία συμβάλλει σε μεγάλο βαθμό στην αποδοτικότητα των χρηστών.

Μεγάλο ρόλο στην αποδοτικότητα των χρηστών, εκτός από την προαναφερθείσα πρόσβαση σε επαρκή πληροφόρηση, έχει παίξει και η εξελισσόμενη ανάπτυξη των συνεργατικών εργαλείων και των επικοινωνιών. Είναι πλέον γεγονός ότι στις μέρες μας, μπορεί κανείς μέσω ενός κινητού τηλεφώνου και από οποιοδήποτε σημείο και να βρίσκεται, να έχει πρόσβαση στο Διαδίκτυο και να επικοινωνήσει/συνεργαστεί με τον οποιονδήποτε. Επίσης με την χρήση συνεργατικών υπηρεσιών δίνεται η δυνατότητα στον χρήστη να δημιουργήσει, να επεξεργαστεί και να μοιραστεί την δουλειά του.

Συνεπώς είναι φανερό, ότι ένας συνδυασμός του τομέα της διαδικτυακής συνεργατικότητας, με τον κλάδο των κινητών επικοινωνιών, θα μπορούσε να συνδράμει σημαντικά στην ανάπτυξη και την εξέλιξη των ψηφιακών βιβλιοθηκών. Μία ψηφιακή βιβλιοθήκη διαθέσιμη στους χρήστες της μέσω κινητού τηλεφώνου αποτελεί μία ελκυστική ιδέα που ταυτόχρονα θα προκαλέσει την αύξηση της αποδοτικότητας των χρηστών.

Στην παρούσα μεταπτυχιακή διατριβή, αναπτύχθηκε ένα πληροφοριακό σύστημα με την μορφή ψηφιακής βιβλιοθήκης, το οποίο εκμεταλλεύεται τις εφαρμογές κινητών τηλεφώνων για να προσφέρει υπηρεσίες αναζήτησης και εύρεσης πληροφορίας, αλλά και συνεργατικότητας και διαχείρισης έργου, ανά πάσα στιγμή και σε οποιοδήποτε μέρος. Δίνει δηλαδή στους χρήστες την δυνατότητα να παράγουν έργο μέσα από το κινητό τους, και τους εντάσσει ταυτόχρονα σε ένα συνεργατικό πλαίσιο, το οποίο τους παρέχει όλα τα απαραίτητα εργαλεία για να αυξήσουν την παραγωγικότητά τους.

1.1. Σκοπός της μεταπτυχιακής διατριβής

Αντικείμενο της παρούσας μεταπτυχιακής διατριβής αποτελεί η μελέτη και η ανάπτυξη μίας κινητής εφαρμογής, της LibrApp, η οποία παρέχει υπηρεσίες συνεργατικής ψηφιακής βιβλιοθήκης στους χρήστες της. Η εφαρμογή αναπτύχθηκε με σκοπό να δώσει σε χρήστες που επιθυμούν να αναπτύξουν έργο σε επαγγελματικό, ακαδημαϊκό ή και ερασιτεχνικό επίπεδο, την δυνατότητα να μπορούν να ανακτήσουν και να μοιραστούν υλικό που βοηθάει στην εξέλιξη της εργασίας τους, καθώς και να αυξήσουν την συνεργασία τους με άλλα άτομα που εκτελούν το

ίδιο ή παρόμοιο έργο. Μιας και παρόμοια εργαλεία είναι σπάνια, η διατριβή στοχεύει στην εδραίωση των ψηφιακών βιβλιοθηκών μέσω κινητών εφαρμογών.

Η βασική ιδέα της εφαρμογής είναι, να δοθεί η δυνατότητα σε ένα ίδρυμα (ιδιωτικό, κρατικό ή ακαδημαϊκό) το οποίο διαθέτει έναν κεντρικό εξυπηρετητή (server) με το σύνολο των πληροφοριών του αποθηκευμένο σε μία βάση δεδομένων, να επιτρέπει στους χρήστες του να συνδεθούν και να χρησιμοποιούν της παρεχόμενες υπηρεσίες μέσω του κινητού τους τηλεφώνου το οποίο τρέχει το λειτουργικό σύστημα Android. Οι χρήστες θα μπορούν να συνδεθούν στον κεντρικό server, για να αποκτήσουν πρόσβαση στο περιεχόμενο της βιβλιοθήκης (βιβλία, εργασίες, συγγράμματα), να συνδεθούν με φίλους ή συνεργάτες και να διαχειριστούν τα έργα τους.

Στόχο επίσης αποτελεί η αξιολόγηση του εν λόγω συστήματος και η μέτρηση της αύξησης της αποδοτικότητας των χρηστών την οποία καλείται να επιτύχει.

1.2. Δομή της διατριβής και μεθοδολογία που ακολουθήθηκε

Στην παρούσα μεταπτυχιακή διατριβή, στην αρχή κάνουμε μία αναφορά στις ψηφιακές βιβλιοθήκες. Γίνεται μία ιστορική αναδρομή και ανάλυση της δομής τους καθώς και των στοιχείων που τις αποτελούν. Έπειτα γίνεται μία παρουσίαση κάποιων από τις πιο γνωστές και σημαντικές ψηφιακές βιβλιοθήκες που υπάρχουν σήμερα. Στη συνέχεια, αναφερόμαστε στην συνεργατικότητα μέσα από μία ψηφιακή βιβλιοθήκη και στα μοντέλα τα οποία χρησιμοποιούνται για την υλοποίησή της. Ακόμα, παρουσιάζονται οι τελευταίες τάσεις και εξελίξεις, οι οποίες αφορούν τον τομέα.

Το επόμενο κεφάλαιο αναφέρεται στο λειτουργικό σύστημα Android και στις εφαρμογές κινητών τηλεφώνων, οι οποίες είναι γραμμένες ώστε να λειτουργούν σε συσκευές που το χρησιμοποιούν. Γίνεται μία αναλυτική παρουσίαση του τρόπου με τον οποίο υλοποιούνται οι εφαρμογές καθώς και μία ανάλυση των κύριων συστατικών τους και του κύκλου ζωής τους. Δίνεται επίσης σημασία στις καταστάσεις στις οποίες μπορεί να βρεθεί μία εφαρμογή κατά την διάρκεια του κύκλου ζωής της, και στο πώς πρέπει να τις διαχειριστεί το λειτουργικό σύστημα.

Σειρά στο κείμενο κατέχει η σχεδίαση της εφαρμογής LibrApp, και η παρουσίαση και ανάλυση των τεχνολογιών και των τεχνικών που χρησιμοποιήθηκαν για την υλοποίησή της. Παρουσιάζεται, η ανάλυση των απαιτήσεων, ο καθορισμός των λειτουργιών, η περιγραφή του σχεδιασμού της εφαρμογής και ο τρόπος υλοποίησης. Επίσης γίνεται ανάλυση των τεχνολογιών PHP, JSON, JQuery και MySQL που χρησιμοποιήθηκαν για την ανάπτυξη, καθώς και του μοντέλου πελάτη-εξυπηρετητή στο οποίο βασίστηκε η υλοποίηση.

Έπειτα, γίνεται η περιγραφή της λειτουργίας της εφαρμογής. Παρατίθεται μεγάλος αριθμός screenshots, τα οποία περιγράφουν επαρκώς τη λειτουργία και τη χρήση της LibrApp, μαζί με την απαραίτητη ανάλυση. Το κεφάλαιο αυτό μπορεί να λειτουργήσει και σαν εγχειρίδιο χρήστη.

Μετά την παρουσίαση της εφαρμογής ακολουθεί η αξιολόγησή της. Για την αξιολόγηση, αναπτύχθηκε ένα ερωτηματολόγιο με γνώμονα την αύξηση της αποδοτικότητας στην εργασία την οποία παρέχει η εφαρμογή στους χρήστες. Καθένας από τους ερωτώμενους κλήθηκε να κρίνει το κατά πόσο κάθε λειτουργία της εφαρμογής βοηθάει στην αύξηση της παραγωγικότητάς του. Πέραν από την κλασική ανάλυση των αποτελεσμάτων, γίνεται χρήση ενός μαθηματικού μοντέλου το οποίο υπολογίζει το ποσοστό κατά το οποίο η εφαρμογή φαίνεται να επηρεάζει την αποδοτικότητα.

Στο τέλος της μεταπτυχιακής διατριβής, γίνεται ανάλυση των συμπερασμάτων στα οποία καταλήγουμε μετά από την αξιολόγηση της εφαρμογής. Επίσης εξετάζονται τυχόν βελτιώσεις τις οποίες θα μπορούσαμε να λάβουμε υπόψη μας σε μελλοντικές εργασίες.

Ακόμα, στο τέλος της διατριβής παρατίθενται δύο παραρτήματα. Στο πρώτο παράρτημα παρουσιάζονται τα σημαντικότερα κομμάτια του κώδικα της εφαρμογής, καθώς και το σχήμα της βάσης δεδομένων που χρησιμοποιήθηκε. Στο δεύτερο παράρτημα παρατίθεται το ερωτηματολόγιο το οποίο χρησιμοποιήθηκε για την αξιολόγηση του συστήματος.

2. Ψηφιακές Βιβλιοθήκες

2.1. Ορισμός

Μια ηλεκτρονική βιβλιοθήκη ή αλλιώς ψηφιακή βιβλιοθήκη είναι μια εστιασμένη συλλογή ψηφιακών αντικειμένων που μπορεί να περιλαμβάνει κείμενο, οπτικό υλικό, ηχητικό υλικό και υλικό βίντεο. Τα παραπάνω αποθηκεύονται στην βιβλιοθήκη σε ηλεκτρονική μορφή σε αντίθεση με τις μορφές μιας κλασσικής βιβλιοθήκης όπως εκτυπωμένο αρχείο, μικροφίλμ ή άλλα μέσα. Ένα τέτοιο σύστημα παρέχει στους χρήστες πέραν από την αποθήκευση των αρχείων, υπηρεσίες οργάνωσης και ανάκτησης αρχείων και μέσων που περιέχονται στη συλλογή της βιβλιοθήκης. Οι ψηφιακές βιβλιοθήκες μπορεί να διαφέρουν πάρα πολύ μεταξύ τους σε μέγεθος ή/και σε πεδίο εφαρμογής, και μπορεί να διατηρούνται από ιδιώτες, οργανισμούς, ή να είναι συσχετισμένες με παραδοσιακές βιβλιοθήκες ή βιβλιοθήκες ιδρυμάτων και πανεπιστημίων [1]. Το ηλεκτρονικό περιεχόμενο μπορεί να αποθηκευτεί τοπικά, να βρίσκεται σε κάποιον απομακρυσμένο εξυπηρετητή ή να διαμοιράζεται μέσω του υπολογιστικού νέφους. Μια ηλεκτρονική βιβλιοθήκη είναι ουσιαστικά ένα είδος συστήματος ανάκτησης πληροφορίας.

Υπάρχει διάκριση μεταξύ του περιεχομένου το οποίο δημιουργήθηκε σε ψηφιακή μορφή, γνωστό ως γεννημένο ψηφιακά (digitally born), και σε αυτό που έχει μετατραπεί από ένα φυσικό μέσο, πχ. χαρτί, με ψηφιοποίηση. Επίσης θα πρέπει να σημειωθεί ότι δεν είναι όλα τα περιεχόμενα μιας ψηφιακής βιβλιοθήκης σε ψηφιακή μορφή. Ο όρος υβριδική βιβλιοθήκη χρησιμοποιείται για τις βιβλιοθήκες που έχουν τόσο φυσικές όσο και ηλεκτρονικές συλλογές. Για παράδειγμα, η American Memory είναι μια ψηφιακή βιβλιοθήκη μέσα στη Βιβλιοθήκη του Κογκρέσου.

Ορισμένες σημαντικές ψηφιακές βιβλιοθήκες χρησιμεύουν ως αρχεία, όπως το arXiv και το Internet Archive. Άλλες επιδιώκουν να κάνουν την ψηφιακή πληροφορία ευρέως προσβάσιμη μέσω των δημοσίων βιβλιοθηκών [5].

Αρχικά ο όρος ψηφιακή βιβλιοθήκη χρησιμοποιήθηκε μαζί με τον όρο εικονική βιβλιοθήκη, ο οποίος τώρα χρησιμοποιείται κυρίως για τις βιβλιοθήκες που είναι εικονικές σε άλλο επίπεδο (όπως οι βιβλιοθήκες οι οποίες συγκεντρώνουν καταμετρημένο περιεχόμενο). Διάφοροι λόγοι οδήγησαν στην καθιέρωση του όρου ψηφιακή βιβλιοθήκη. Η αυξανόμενη χρήση όρων όπως ψηφιακά δίκτυα, ψηφιακός ήχος και ψηφιακό βίντεο οδήγησαν σταδιακά στον παραγκωνισμό του όρου ηλεκτρονική βιβλιοθήκη και βοήθησαν στην επικράτηση του ονόματος με το οποίο είναι γνωστές σήμερα. Αυτή η μετατόπιση παρατηρείται την περίοδο 1991-1993, και το 1994 έχουμε πλέον τη διάδοση του όρου ψηφιακή βιβλιοθήκη, όπου το NSF, η DARPA και η NASA δημιουργούν μια πρωτοβουλία για τις ψηφιακές βιβλιοθήκες την DLI (Digital Library Initiative) [2].

Το πρώτο μεγάλο ερευνητικό πρόγραμμα ήταν το Computer Science Technical Report, μία συνδυασμένη προσπάθεια από πέντε μεγάλα αμερικανικά πανεπιστήμια και την Βιβλιοθήκη του Κογκρέσου. Το πρόγραμμα αυτό χρηματοδοτήθηκε από το ARPA από το 1992-1995.

Στο πλαίσιο του Delos , ενός Δικτύου Αριστείας σε Ψηφιακές Βιβλιοθήκες και του DL.org, μιας δράσης συντονισμού για τις ψηφιακές βιβλιοθήκες, ένα σύνολο ερευνητών παράγγαγε το Μοντέλο Αναφοράς σε Ψηφιακές Βιβλιοθήκες (Digital Library Reference Model) [3],[4], το οποίο ορίζει μία ψηφιακή βιβλιοθήκη ως: «Ένας δυναμικός εικονικός οργανισμός, που συγκεντρώνει συνολικά, διαχειρίζεται και διατηρεί για μεγάλο βάθος χρόνου, πλούσιο ψηφιακό περιεχόμενο, και προσφέρει σε διάφορες κοινότητες χρηστών εξειδικευμένες λειτουργίες σε αυτό το περιεχόμενο, καθορισμένης ποιότητας και σύμφωνα με συγκεκριμένες πολιτικές» [3].

Σύμφωνα με τον Nórnberrg οι ψηφιακές βιβλιοθήκες αποτελούν ένα αναδυόμενο πεδίο που συγκεντρώνει πολλούς ερευνητές από ήδη υπάρχουσες επιστημονικές περιοχές. Αυτή η ιδιαιτερότητά τους, τις στερεί από μία ξεκάθαρη διάταξη και κάνει το πεδίο εντελώς αυτόνομο από όλες τις άλλες περιοχές [6]. Επίσης, στο έργο του Nórnberrg παρατίθενται κάποιες ενδιαφέρουσες προσεγγίσεις σχετικά με τις ψηφιακές βιβλιοθήκες, από την οπτική γωνία διαφορετικών ερευνητών. Συγκεκριμένα κάποιες από αυτές συνοψίζονται στα εξής:

- Οι βιβλιοθηκονόμοι και γενικά οι απασχολούμενοι σε βιβλιοθήκες βλέπουν τις ψηφιακές βιβλιοθήκες ως μία τάση προς την συνεχή αυτοματοποίηση και εξέλιξη των φυσικών βιβλιοθηκών. Ως εκ τούτου, επικεντρώνονται στην ανακάλυψη νέων πηγών

- πληροφόρησης, νέων τρόπων απόκτησης γνώσης, νέων μεθόδων αποθήκευσης κ.ο.κ.
- Όσοι έχουν ως τομέα τις βάσεις δεδομένων ή την ανάκτηση γνώσης αντιλαμβάνονται τις ψηφιακές βιβλιοθήκες ως μία συνέχεια των βάσεων δεδομένων και για την ακρίβεια ως ένα σύνολο ενοποιημένων βάσεων.
- Τέλος, όσοι ασχολούνται με τον Παγκόσμιο Ιστό, θεωρούν τις ψηφιακές βιβλιοθήκες ένα υποπεδίο του Ιστού και εστιάζουν στην ανάπτυξη νέων διαδικτυακών εφαρμογών και υπηρεσιών για την καλύτερη πρόσβαση και διαχείριση.

Θα μπορούσαμε λοιπόν να ορίσουμε τις ψηφιακές βιβλιοθήκες ως ένα σύνολο υπηρεσιών, οι οποίες αναφέρονται στην διαχείριση της ψηφιακής πληροφορίας, οποιαδήποτε και να είναι η μορφή της. Δεδομένης της πολυμορφικής προσέγγισης που διαφαίνεται ανωτέρω θα ήταν σημαντικό αρχικά να παρατηρήσουμε τον τρόπο με τον οποίο μία ψηφιακή βιβλιοθήκη θα μπορούσε να ικανοποιήσει ταυτόχρονα τους διαφορετικούς χρήστες, και εν συνεχεία να μελετήσουμε μία εφαρμογή ψηφιακής βιβλιοθήκης, η οποία θα αυξήσει την ικανοποίηση και την παραγωγικότητα των χρηστών της.

2.2.Διάκριση από τις κοινές βιβλιοθήκες

Το πως σχετίζονται οι ψηφιακές βιβλιοθήκες με τις φυσικές, αποτελεί ένα αρκετά ενδιαφέρον ζήτημα. Ειδικότερα, αξίζει να ερευνηθούν οι διαφορές τους, αν αυτές υπάρχουν. Πλέον στις μέρες μας, οι λέξεις «βιβλιοθήκη» και «ψηφιακή» είναι άρρηκτα συνδεδεμένες, γι' αυτό και είναι σημαντικό να διαχωριστούν οι ομοιότητες και οι διαφορές τους πριν δοθεί βάρος στο πως πλέον αυτές συνδράμουν στην εξυπηρέτηση των χρηστών. Η λειτουργία μιας παραδοσιακής βιβλιοθήκης μπορεί να συνοψιστεί στα παρακάτω:

1. Συλλογή πληροφορίας
2. Οργάνωση του περιεχομένου
3. Ανάκτηση πληροφορίας
4. Διανομή πληροφορίας

Στο πρώτο στάδιο ουσιαστικά περιλαμβάνονται οι διάφορες τεχνικές εύρεσης και κατανόησης των πληροφοριών. Είναι σημαντικό να γίνει ακριβής ορισμός της έννοιας η οποία αναζητείται, ώστε η πληροφορία η οποία θα ανακαλυφθεί και αποθηκευθεί να είναι ακριβής. Το δεύτερο στάδιο ασχολείται με την οργάνωση και την ταξινόμηση του περιεχομένου το οποίο αποκτήθηκε από το πρώτο στάδιο, με τέτοιο τρόπο ώστε να είναι κατανοητό και εύκολα προσβάσιμο από το χρήστη. Το τρίτο στάδιο αναζητά τεχνικές και τρόπους ώστε να γίνεται εφικτή η άμεση πρόσβαση στο υλικό και να καθίσταται απλή η αναζήτηση και η ανάκτηση της υπάρχουσας πληροφορίας. Τέλος, το τέταρτο στάδιο αφορά τις λειτουργίες επερωτήσεων, παραγωγής περιλήψεων και γενικά διανομής της πληροφορίας στους χρήστες σε οποιαδήποτε μορφή αυτή ζητηθεί.

Πάνω σε αυτά τα στάδια βασίζεται και η ιδέα των ψηφιακών βιβλιοθηκών έχοντας όμως ως βασικό στόχο να αυτοματοποιήσει όσο το δυνατόν περισσότερο καθεμία από αυτές τις λειτουργίες. Γενικά θα ήταν δύσκολο να παραλείψει κάποιος οποιοδήποτε από αυτά τα στάδια, όμως γίνονται προσπάθειες ενοποίησης και εκμοντέρνισης της δομής τους [7]. Πάντως όσο και να αυξάνονται οι τεχνολογικές υποδομές, ο σχεδιασμός μιας βιβλιοθήκης πρέπει πάντοτε να πατάει πάνω σε κάποια κοινά πρότυπα.

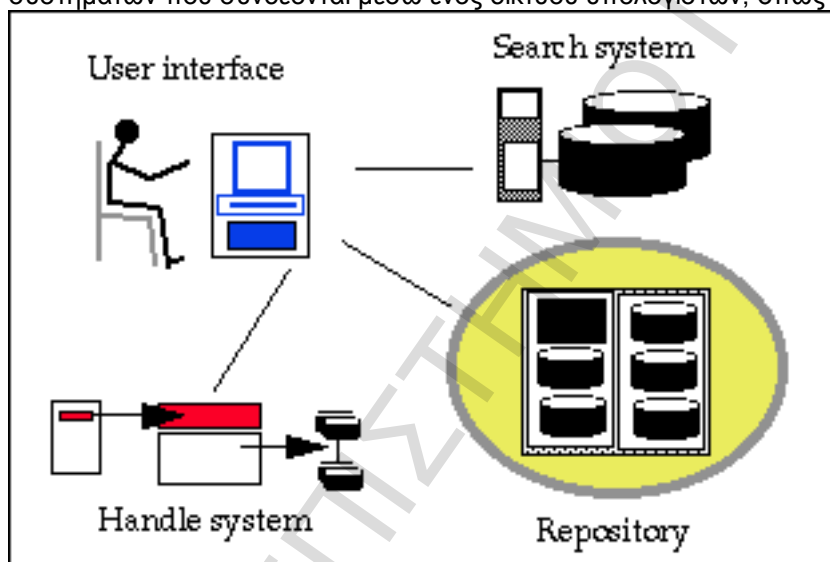
Η έρευνα γύρω από τις ψηφιακές βιβλιοθήκες επικεντρώθηκε στην αυτοματοποίηση των βιβλιοθηκονομικών λειτουργιών. Προσπάθησε δηλαδή να δημιουργήσει έτοιμες διαδικασίες για την ταξινόμηση, αποθήκευση και ανάκτηση της πληροφορίας και να τις εντάξει σε έμπειρα συστήματα. Η παραδοσιακή δημιουργία καταλόγων προϋποθέτει ότι οι χρήστες της μπορούν να θέσουν λίγες μόνο λέξεις-κλειδιά, πολλές φορές προκαθορισμένες. Πάνω σε αυτό το σημείο βλέπουμε τις προσθήκες που έκανε η δημιουργία των ψηφιακών βιβλιοθηκών στις βασικές αρχές υλοποίησης μιας βιβλιοθήκης, όπως για παράδειγμα την ύπαρξη πολλών λέξεων-κλειδιών σε ένα ψηφιακό κατάλογο σε αντίθεση με έναν παραδοσιακό. Έτσι φτάνουμε στο συμπέρασμα ότι μία ψηφιακή βιβλιοθήκη έχει πολλά κοινά με την παραδοσιακή αντίστοιχη της, και ουσιαστικά η διαφορά εντοπίζεται στο ότι επαναπροσδιορίζει της λειτουργίες της για να πετύχει μεγαλύτερη αποδοτικότητα.

2.3 Δομή μιας ψηφιακής βιβλιοθήκης

Μια ψηφιακή βιβλιοθήκη αποτελείται από απλά αντικείμενα, τα ψηφιακά αντικείμενα (digital objects). Ένα ψηφιακό αντικείμενο είναι ένας τρόπος δόμησης της πληροφορίας σε ψηφιακή μορφή. Η μορφή αυτή ποικίλει από δεδομένα πληροφορίας, μέχρι μεταδεδομένα (metadata), δηλαδή δεδομένα τα οποία ουσιαστικά περιγράφουν τα δεδομένα πληροφορίας, για παράδειγμα, κάποιο αναγνωριστικό ή τίτλος. Εντούτοις, οι πληροφορίες σε μια ψηφιακή βιβλιοθήκη είναι αρκετά πολύπλοκες.

Η βασική της δομή περιλαμβάνει τον απαραίτητο ηλεκτρονικό εξοπλισμό για τη δημιουργία, αποθήκευση και προσπέλαση του ψηφιακού υλικού μαζί με το αναγκαίο σε κάθε περίπτωση λογισμικό για τη διαχείριση και οργάνωση του υλικού, τη δημιουργία ευρετηρίων και καταλόγων και την αναζήτηση πληροφοριών σε αυτό. Ας σημειωθεί πως αυτές οι λειτουργίες αφορούν τη γενική περίπτωση μιας ψηφιακής βιβλιοθήκης ενώ από εκεί και πέρα υπάρχουν διαφοροποιήσεις ανάλογα με το εάν η βιβλιοθήκη είναι προσωπική, δημόσια, εμπορική ή εκπαιδευτική. Ωστόσο σε όλες τις περιπτώσεις η λειτουργία της στηρίζεται σε τεχνικές και μεθόδους που συναντώνται στο χώρο των βάσεων δεδομένων με την έμφαση να δίδεται στην ακεραιότητα και την προστασία του αποθηκευμένου ψηφιακού υλικού.

Η οργάνωση μιας τυπικής ψηφιακής βιβλιοθήκης επιτρέπει σε πολλά συστήματα ηλεκτρονικών υπολογιστών να συνυπάρχουν. Τα βασικά συστατικά μέρη της δομής μιας τυπικής ψηφιακής βιβλιοθήκης παρουσιάζονται στην Εικόνα 1. Δίνεται μία σύντομη περιγραφή τους παρακάτω. Αυτά μπορούν να χρησιμοποιηθούν από μια ευρύτατη ποικιλία υπολογιστικών συστημάτων που συνδέονται μέσω ενός δικτύου υπολογιστών, όπως είναι το Διαδίκτυο.



Σχήμα 1 Η δομή μιας ψηφιακής βιβλιοθήκης (<http://www.dlib.org>)

Τα συστατικά που αποτελούν μία ψηφιακή βιβλιοθήκη μπορεί να συνοψιστούν στα παρακάτω:

- Διεπαφές με τον χρήστη (User Interfaces): Το μέσο δηλαδή με το οποίο οι χρήστες επικοινωνούν με την βιβλιοθήκη και εκτελούν τις προσφερόμενες λειτουργίες
- Αποθήκη δεδομένων (Repository): Οι αποθήκες δεδομένων αποθηκεύουν και διαχειρίζονται ψηφιακά αντικείμενα και άλλες πληροφορίες. Μια μεγάλη ψηφιακή βιβλιοθήκη πρέπει να διαθέτει αποθήκες δεδομένων διάφορων τύπων, δηλαδή σύγχρονες βάσεις δεδομένων και ισχυρούς Web servers. Έχουν αναπτυχθεί διάφορες εφαρμογές και πρωτόκολλα για τις αποθήκες δεδομένων αλλά το πιο διαδεδομένο είναι το πρωτόκολλο πρόσβασης αποθήκης δεδομένων (repository access protocol - RAP).
- Σύστημα τιλοποίησης (Handle System): Οι τίτλοι είναι προσδιοριστές γενικού σκοπού, που χρησιμεύουν στην αναγνώριση ψηφιακών πηγών. Το CNRI είναι ένα από τα πιο γνωστά συστήματα τιλοποίησης [8].
- Σύστημα αναζήτησης (Search System): Η δημιουργία μιας ψηφιακής βιβλιοθήκης θέτει ως δεδομένο ότι θα υπάρχουν πολλά ευρετήρια (indexes) και κατάλογοι, των οποίων η

εξερεύνηση θα οδηγήσει στην εύρεση της θέσης των πληροφοριών, πριν γίνει η ανάκτησή τους από την αποθήκη δεδομένων. Αυτά τα ευρετήρια μπορούν ελεύθερα να διαχειριστούν και να υποστηρίξουν ένα ευρύ πεδίο πρωτοκόλλων.

- **Αποθήκευση (Storage):** Το σύστημα αποθήκευσης μιας ψηφιακής βιβλιοθήκης πρέπει να έχει τη δυνατότητα να αποθηκεύει μια μεγάλη ποσότητα δεδομένων με ποικιλία μορφοποιήσεων (formats) και να επιτρέπει την πρόσβαση αυτών στο ταχύτερο δυνατό χρόνο. Τα έγγραφα τύπου κειμένου αποθηκεύονται σε μορφές ASCII, LaTeX, HTML, SGML και PostScript και είναι με διαφορά τα πιο εύκολα αποθηκεύσιμα. Τα αρχεία ψηφιακού ήχου και βίντεο αποθηκεύονται πολύ δυσκολότερα γιατί απαιτούν σαφώς μεγαλύτερο χώρο στην αποθήκη δεδομένων και η διανομή τους γίνεται σε περισσότερο χρόνο.
- **Καταλογογράφηση και Ευρετηρίαση (Classification and Indexing):** Τα συστήματα καταλογογράφησης και ευρετηρίασης χρησιμοποιούνται για την συλλογή των σχετικών δεδομένων σε ομάδες, έτσι ώστε να διευκολύνεται η χρήση τους από τους χρήστες. Ένας πολύπλοκος παράγοντας στην καταλογογράφηση και ευρετηρίαση είναι η τεράστια ποσότητα των ενδεχόμενων δεδομένων που πρέπει να ευρετηριασθεί.
- **Ανάκτηση πληροφοριών (Information retrieval):** Στο πεδίο των ψηφιακών βιβλιοθηκών, υπάρχει μια ποικιλία τεχνικών ανάκτησης πληροφοριών, όπως η αναζήτηση μεταδεδομένων, εγγράφων ελεύθερου κειμένου και άλλων τύπων δεδομένων.
- **Διανομή περιεχομένου (Content Delivery):** Τα σύγχρονα πρότυπα ανοικτών δικτύων, όπως το TCP/IP και η διαρκής άνθηση του Internet έχουν ξεκαθαρίσει ότι οι επιτυχημένες ψηφιακές βιβλιοθήκες πρέπει να χτίζονται πάνω σε ανοικτά δίκτυα. Οι σύγχρονες ψηφιακές βιβλιοθήκες μπορεί να δημιουργούνται σε ένα μόνο υπολογιστή, σε αρκετούς υπολογιστές που αποτελούν ένα τοπικό δίκτυο, ή σε ένα μεγάλο αριθμό υπολογιστών που είναι διασπαρμένοι σε ένα δίκτυο ευρείας περιοχής.
- **Παρουσίαση (Presentation):** Το σύστημα παρουσίασης μιας ψηφιακής βιβλιοθήκης πρέπει να είναι ευέλικτο και εύχρηστο για τον χρήστη. Επίσης πρέπει να λαμβάνει υπόψη του, τις δυνατότητες αλλά και τους περιορισμούς του υλικού του πελάτη και αυτόματα να προσαρμόζεται ώστε να διαθέτει την καλύτερη δυνατή ποιότητα παρουσίασης κάθε φορά.
- **Διαχείριση (Administration):** Οι ψηφιακές βιβλιοθήκες αποθηκεύουν αρκετές εκδόσεις ενός εγγράφου κατά έναν τρόπο που καθιστά ικανές τις πολλαπλές αναθεωρήσεις του από διάφορους συγγραφείς. Επιπροσθέτως, το περιεχόμενο ενός αντικείμενου μπορεί να έχει πολλούς κατόχους και αυτό πρέπει να σημειώνεται. Ένα σύστημα διαχείρισης διασφαλίζει ότι τα αντικείμενα που προορίζονται για το ευρύ κοινό μπορούν να ανακτηθούν από τον καθένα. Ακόμα, είναι σημαντικό να μπορεί ο διαχειριστής να επεξεργαστεί και να τροποποιήσει τις συνιστώσες του συστήματος εύκολα και με φιλικό προς τον χρήστη τρόπο.

Έχοντας αναφέρει τα βασικά κομμάτια μιας ψηφιακής βιβλιοθήκης, παρακάτω θα αναλύσουμε κάποια από τα πιο σημαντικά, τονίζοντας ωστόσο ότι μία βιβλιοθήκη βασίζεται εξίσου σε όλα. Περισσότερο βάθος δίνεται στον τρόπο λειτουργίας της αποθήκης δεδομένων, αλλά και στο πως πρέπει να σχεδιάζεται και να λειτουργεί μία διεπαφή χρήστη, το κομμάτι δηλαδή με το οποίο έρχονται σε επαφή οι περισσότεροι. Επίσης θα γίνει αναφορά στη δομή της πληροφορίας, στο σύστημα OPAC, καθώς και στις διάφορες μορφές με τις οποίες μπορεί να απεικονίζεται η πληροφορία μέσα στην ψηφιακή βιβλιοθήκη (MARC, XML, Metadata).

2.3.1 Δομή της αποθήκης δεδομένων

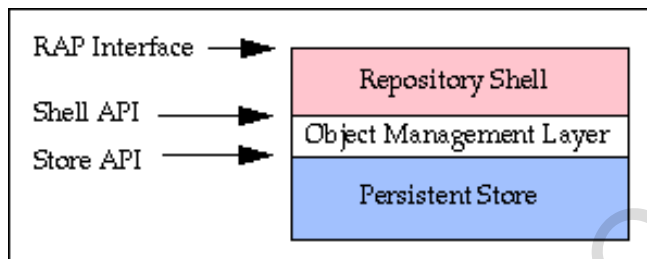
Μία αποθήκη δεδομένων είναι ένα δικτυακά βασισμένο σύστημα για την αποθήκευση και την πρόσβαση σε ψηφιακά αντικείμενα. Όλες οι αλληλεπιδράσεις με την αποθήκη χρησιμοποιούν ένα απλό πρωτόκολλο, που είναι γνωστό ως Πρωτόκολλο Πρόσβασης Αποθήκης (Repository Access Protocol ή RAP). Το RAP έχει ένα μικρό αριθμό από βασικές λειτουργίες, όπως η «κατάθεση», η οποία αποθηκεύει ένα ψηφιακό αντικείμενο στην αποθήκη, και η «πρόσβαση», η οποία παρέχει πρόσβαση σε ένα ψηφιακό αντικείμενο. Έτσι, το RAP παρέχει μια σαφώς

καθορισμένη, ανοικτή διεπαφή στην αποθήκη πληροφορίας, που επιτρέπει σε οποιονδήποτε προγραμματιστή να υλοποιήσει τις δικές του εφαρμογές-πελάτες ή διεπαφές.

Η αποθήκη δεδομένων δεν κάνει υποθέσεις σχετικά με το περιεχόμενο των ψηφιακών αντικειμένων. Μπορεί να αποθηκεύσει ψηφιοποιημένο κείμενο, εικόνες, ήχο, δυαδικά δεδομένα, κλπ. Το repository αποθηκεύει ένα τύπο δεδομένων για κάθε στοιχείο του κάθε ψηφιακού αντικειμένου, δεν μπορεί όμως να κατανοήσει την εσωτερική δομή για κάθε συγκεκριμένη μορφή, όπως για παράδειγμα .Jpeg ή SGML. Όλες οι απαιτήσεις για τη μετατροπή μεταξύ μορφοποιήσεων ή παρουσιάσεων για τους χρήστες είναι εξωτερική προς την αποθήκη. Σε γενικές γραμμές, η αποθηκευμένη μορφή ενός ψηφιακού αντικειμένου μπορεί να είναι διαφορετική από τη μορφή στην οποία έχει διαδοθεί ή στην οποία παρουσιάζεται σε ένα χρήστη.

Η ασφάλεια είναι ένα αναπόσπαστο μέρος του χώρου αποθήκευσης του σχεδιασμού και του RAP. Η αποθήκευση ενός ψηφιακού αντικειμένου σε μία αποθήκη πληροφορίας συσχετίζεται με τα δικαιώματα αυτού του αντικείμενου, γι' αυτό η υλοποίηση κάθε εντολής RAP περιλαμβάνει μια ρητή επικύρωση των όρων και των προϋποθέσεων πρόσβασης.

Στο Σχήμα 2 φαίνονται τα τρία στρώματα της υλοποίησης μίας αποθήκης δεδομένων.



Σχήμα 2 Δομή μιας αποθήκης δεδομένων

Το κέλυφος της αποθήκης (repository shell) είναι το τμήμα της αποθήκης, το οποίο αποτελεί τη διασύνδεση με τον έξω κόσμο. Υλοποιεί το πρωτόκολλο RAP, το οποίο όπως αναφέραμε παραπάνω μετατρέπει μεταξύ της εσωτερικής και της εξωτερικής μορφής των ψηφιακών αντικειμένων, και διαχειρίζεται τις άδειες και τα δικαιώματα χρήσης.

Το persistence store περιέχει τις πληροφορίες οι οποίες βρίσκονται αποθηκευμένες στην αποθήκη. Η λειτουργία του είναι εντελώς κρυμμένη από τον χρήστη και από το κέλυφος της αποθήκης το οποίο είναι σχεδιασμένο έτσι ώστε να λειτουργεί με ένα ευρύ φάσμα από persistence stores. Θα μπορούσαμε για παράδειγμα να χρησιμοποιήσουμε μία σχεσιακή βάση δεδομένων ή το σύστημα διαχείρισης αρχείων ενός λειτουργικού συστήματος χωρίς το RAP να καταλάβει την διαφορά.

Το στρώμα διαχείρισης αντικειμένων παρέχει τη διεπαφή η οποία παρέχει αντικειμενοστρεφείς υπηρεσίες μεταξύ του persistence store και του κελύφους της αποθήκης. Αυτό το στρώμα παρέχει την αντιστοίχιση μεταξύ των ψηφιακών αντικειμένων, τη θέση στην οποία αυτά βρίσκονται αποθηκευμένα, και τις υπηρεσίες του συστήματος που απαιτούνται για την αξιόπιστη λειτουργία του συστήματος.

Όπως φαίνεται και από το προηγούμενο σχήμα, οι διασυνδέσεις μεταξύ των τριών στρωμάτων είναι σαφώς καθορισμένες διεπαφές προγραμματισμού εφαρμογών (APIs). Αυτό επιτρέπει σε όλα τα επιμέρους στρώματα να τροποποιηθούν ανεξάρτητα από τα άλλα.

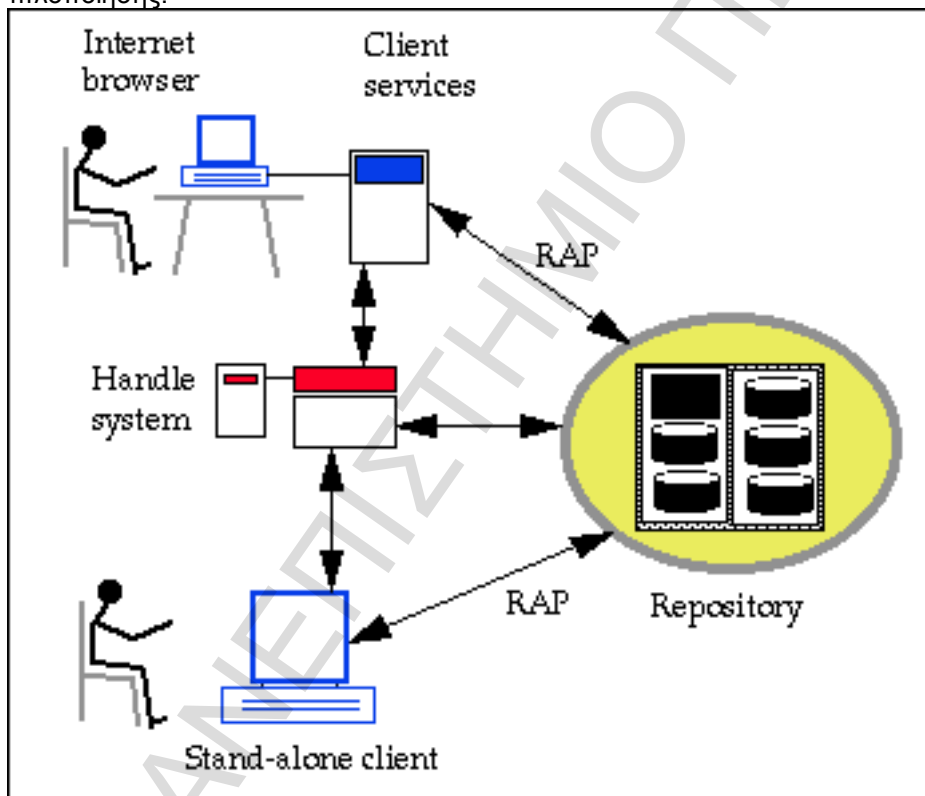
Όλες οι αλληλεπιδράσεις με τον αποθηκευτικό χώρο χρησιμοποιούν το πρωτόκολλο RAP. Μερικές από τις λειτουργίες του RAP αναλύονται στον παρακάτω πίνακα.

Λειτουργία	Περιγραφή
VerifyHandle:	Επιβεβαιώνει ότι ο τίτλος έχει καταχωρηθεί στο σύστημα τιτλοποίησης.
AccessRepoMeta:	Δίνει πρόσβαση στην αποθήκη μεταδεδομένων.
Verify_DO:	Επιβεβαιώνει ότι η αποθήκη δεδομένων αποθηκεύει ένα ψηφιακό αντικείμενο με καθορισμένο τίτλο.
AccessMeta:	Δίνει πρόσβαση στα μεταδεδομένα για ένα συγκεκριμένο ψηφιακό αντικείμενο.
Access_DO:	Δίνει πρόσβαση ψηφιακό αντικείμενο.
Deposit_DO:	Καταχωρεί ένα ψηφιακό αντικείμενο σε μία αποθήκη.
Delete_DO:	Διαγράφει ένα ψηφιακό αντικείμενο από μία αποθήκη.
MutateMeta:	Επεξεργάζεται τα μεταδεδομένα για ένα ψηφιακό αντικείμενο.
Mutate_DO:	Επεξεργάζεται ένα ψηφιακό αντικείμενο.

Πίνακας 1 Λειτουργίες του RAP

2.3.2 Διεπαφές χρήστη

Οι εφαρμογές-πελάτες μίας αποθήκης δεδομένων χρησιμοποιούνται για τον εντοπισμό και τον χειρισμό των ψηφιακών αντικειμένων που περιέχονται μέσα στην αποθήκη. Στο Σχήμα 3 περιγράφονται οι σχέσεις μεταξύ της αποθήκης, της εφαρμογής-πελάτη, και του συστήματος τιτλοποίησης.



Σχήμα 3 Σχέσεις αποθήκης, εφαρμογής και συστήματος τιτλοποίησης (www.dlib.org)

Μια διεπαφή χρήστη για τις ψηφιακές βιβλιοθήκες πρέπει να εμφανίζει μεγάλους όγκους δεδομένων αποτελεσματικά [10]. Τυπικά ο χρήστης παρουσιάζεται με ένα ή περισσότερα επικαλυπτόμενα παράθυρα στα οποία μπορεί να αλλάξει το μέγεθος και την διάταξη. Στις ψηφιακές βιβλιοθήκες, ένα μεγάλο μέρος των δεδομένων τα οποία εκτείνονται μέσα σε μία σειρά πόρων, απαιτούν διαισθητικές διεπαφές για τους χρήστες, ώστε να αναζητούν και να ανακτούν πληροφορίες. Σημαντική πρόκληση για τους ερευνητές διεπαφών χρήστη για ψηφιακές βιβλιοθήκες, αποτελεί η διερεύνηση της ικανότητας να αλλάξει ομαλά η σκοπιά του χρήστη από υψηλό επίπεδο αφαίρεσης σε επίπεδο παραγράφου.

Τα τελευταία έτη έχουν προκύψει μεγάλες αλλαγές στον σχεδιασμό και την νοοτροπία που αφορά την ανάπτυξη διεπαφών χρήστη για τις ψηφιακές βιβλιοθήκες. Σε αυτό οφείλεται η ανάπτυξη σύγχρονων τεχνολογιών, όπως βάσεις δεδομένων τύπου XML, τεχνολογίες διαδικτύου και το Web 2.0, οι οποίες παρέχουν ένα άριστο περιβάλλον για τη διάδοση των πληροφοριών στον απλό χρήστη. Αυτές οι σύγχρονες τεχνολογίες κατέστησαν εφικτή την παγκόσμια συνδεσιμότητα των υπολογιστών και την ανάπτυξη διαφόρων εργαλείων και τεχνολογιών για την πρόσβαση και παροχή διαδικτυακής πληροφορίας.

Οι ψηφιακές βιβλιοθήκες παρέχουν την υποδομή για τη δημιουργία, τη διάρθρωση, την αποθήκευση, την οργάνωση, την επεξεργασία, την ανάκτηση και τη διανομή της ψηφιακής πληροφορίας [11], μέσω του διαδικτύου ή κάποιου εσωτερικού δικτύου. Από όλα τα συστατικά που την αποτελούν θα λέγαμε ότι από τα πιο σημαντικά είναι η διεπαφή χρήστη.

Το σύστημα διεπαφής χρήστη έχει προκαλέσει συζητήσεις λόγω του γεγονότος ότι επηρεάζει τη χρηστικότητα, η οποία αποτελεί βασικό παράγοντα για την επιτυχία ενός προϊόντος. Η εμπορική και τεχνική σημαντικότητα της διεπαφής χρήστη έχει πλέον εδραιωθεί στον ερευνητική κοινότητα της επιστήμης των υπολογιστών, καθώς πολλά συνέδρια και δημοσιεύσεις σε όλο τον κόσμο ασχολούνται αποκλειστικά με τον ανθρώπινο παράγοντα και τα θέματα διεπαφής χρήστη. Εξακολουθούν να υπάρχουν ανοιχτά προβλήματα σε αυτόν τον τομέα, τα οποία συνεχώς αυξάνονται κατά την ανάπτυξη των ψηφιακών βιβλιοθηκών.

Μια ψηφιακή βιβλιοθήκη μπορεί να παρέχει μέσω του Διαδικτύου ένα κοινό σημείο πρόσβασης σε μια τεράστια ποσότητα ψηφιακών πληροφοριών, οι οποίες είναι διαθέσιμες σε χρήστες με διαφορετικό ψυχολογικό, εκπαιδευτικό και κοινωνικό υπόβαθρο και ανάγκες πληροφόρησης. Τεράστιο ποσό χρημάτων έχουν δαπανηθεί για την κατασκευή αξιοποιήσιμων ψηφιακών βιβλιοθηκών. Ωστόσο, σχετική έρευνα έχει δείξει ότι οι ψηφιακές βιβλιοθήκες δεν χρησιμοποιούνται τόσο όσο θα έπρεπε [12].

Οι ψηφιακές βιβλιοθήκες είναι εγγενώς διαδραστικά συστήματα με σταθερή αύξηση του αριθμού των τελικών χρηστών. Θα πρέπει όχι μόνο να βασίζονται σε αποτελεσματικούς και εξελιγμένους μηχανισμούς ανάκτησης αλλά και να παρέχουν αποτελεσματική αλληλεπίδραση με τους τελικούς χρήστες [13].

Τις τελευταίες δύο δεκαετίες, διάφοροι επιστήμονες της πληροφορικής έχουν δώσει προσοχή στην αλληλεπίδραση ανθρώπου-υπολογιστή (HCI) και στον σχεδιασμό διεπαφών χρήστη για συστήματα ανάκτησης πληροφοριών, καθώς και για ψηφιακές βιβλιοθήκες [14],[15]. Προκειμένου να ενισχυθεί η χρηστικότητα και η προσβασιμότητα των ψηφιακών βιβλιοθηκών, είναι απαραίτητο να δούμε πώς οι θεωρίες και τα μοντέλα από τον τομέα της HCI μπορούν να εφαρμοστούν στον σχεδιασμό διεπαφών χρηστών για ψηφιακές βιβλιοθήκες.

Ο σχεδιασμός μιας διεπαφής για ψηφιακές βιβλιοθήκες είναι μια πολύπλοκη διαδικασία που είναι προσανατολισμένη στην επίδειξη των δυνατοτήτων του συστήματος [16]. Η προσεκτική εξέταση των χαρακτηριστικών του συστήματος και των χρηστών αποτελεί το κλειδί της επιτυχίας για τον αποτελεσματικό σχεδιασμό. Ένας σχεδιασμός θεωρείται καθολικά αξιοποιήσιμος όταν αντιμετωπίζει τα τυπικά χαρακτηριστικά που αφορούν τους χρήστες, όπως η χρήση ηλεκτρονικών υπολογιστών, η σωματική αναπηρία, οι ικανότητες σε μια ξένη γλώσσα, η ηλικιακή ομάδα [17], ή η ικανότητα στον τομέα του περιεχομένου ψηφιακών βιβλιοθηκών.

Έρευνα πάνω στις διαφορετικές απαιτήσεις των χρηστών και των τεχνικών σκοπιμοτήτων, προτείνει το ακόλουθο σύνολο κανόνων σχεδιασμού διεπαφών χρήστη για ψηφιακές βιβλιοθήκες [18]:

- Η διεπαφή χρήστη θα πρέπει να είναι απλή και άμεση. Οι βασικές λειτουργίες θα πρέπει να είναι άμεσα εμφανείς, ενώ οι προηγμένες λειτουργίες μπορεί να είναι λιγότερο προφανείς για τους νέους χρήστες. Ένα καλά οργανωμένο περιβάλλον που υποστηρίζει εργασίες χρήστη (user tasks), θα πρέπει να δουλεύει στο παρασκήνιο και να επιτρέπει στους χρήστες να εργάζονται αποτελεσματικά. Εργασίες όπως η αντιγραφή, η εκτύπωση, και η αποθήκευση θα πρέπει να περιλαμβάνονται μόνο εάν η ανάλυση του συγκεκριμένου task το κρίνει απαραίτητο. Το καλύτερο περιβάλλον είναι ένα που δεν γίνεται αντιληπτό από τον χρήστη, και που επιτρέπει στο χρήστη να εστιάσει στις πληροφορίες αντί στον μηχανισμό που χρησιμοποιείται για την παρουσίασή τους.
- Η διεπαφή χρήστη θα πρέπει να παρέχει στους χρήστες έλεγχο επάνω στην ψηφιακή βιβλιοθήκη. Πρέπει να επιτρέπει στους χρήστες να εκτελούν τις εργασίες τους χρησιμοποιώντας οποιαδήποτε ακολουθία βημάτων, την οποία θα χρησιμοποιούσαν

φυσικά. Η διεπαφή δεν θα πρέπει να τις περιορίζει τις επιλογές τους στο τι θεωρείται σωστό από το σύστημα. Θα πρέπει να λειτουργεί περισσότερο με γνώμονα τα γεγονότα παρά το πώς είναι οργανωμένο το μενού της διεπαφής. Η διεπαφή χρήστη του συστήματος θα πρέπει να υποστηρίζει διαφορετικές εργασίες. Οι περισσότεροι χρήστες εκτελούν μια ποικιλία εργασιών και μπορεί να είναι ειδικοί σε μερικές και αρχάριοι σε άλλες. Εκτός από την παροχή βοήθειας όταν ζητείται, το σύστημα θα πρέπει να αναγνωρίζει και να προβλέπει τους στόχους των χρηστών και να προσφέρει βοήθεια για να κάνει την εκτέλεση της εργασίας πιο εύκολη.

- Η διεπαφή της ψηφιακής βιβλιοθήκης πρέπει να είναι οικεία στους χρήστες της, επιτρέποντάς τους να βασίζονται σε προηγούμενες γνώσεις, ιδιαίτερα σε αυτές που έχουν αποκτηθεί από την εμπειρία τους στον πραγματικό κόσμο. Οι χρήστες δεν θα πρέπει να μάθουν νέα πράγματα για την εκτέλεση των καθηκόντων με τα οποία είναι εξοικειωμένοι.
- Η διεπαφή πρέπει να παρέχει ενημερωτικές ανατροφοδοτήσεις για τους χρήστες της. Κατά γενικό κανόνα, πρέπει να υπάρχουν τα σχόλια των χρηστών για κάθε λειτουργία της διεπαφής. Ως προς την έκταση και τη μορφή της ανατροφοδότησης, αυτές εξαρτώνται από την εργασία που εκτελείται. Για μικρής σημασίας και συχνές δράσεις, η ανατροφοδότηση μπορεί να είναι διακριτική, για παράδειγμα, η μεταβολή του περιεχομένου της σελίδας ή ένας απαλός ήχος θα αρκούσε. Για σημαντικές δράσεις, όπως για παράδειγμα, η αλλαγή των κωδικών πρόσβασης ή μηνύματα λάθους, η ανατροφοδότηση πρέπει να είναι πιο εμφανής, ίσως με την μορφή pop-ups και / ή θορύβους για να προειδοποιήσει τον χρήστη για την κατάσταση.
- Η ενημερωτική ανατροφοδότηση κατά την ολοκλήρωση μιας σειράς ενεργειών θα δώσει στους χρήστες την ικανοποίηση της ολοκλήρωσης και θα τους προετοιμάσει για την επόμενη σειρά ενεργειών.
- Το σύστημα θα πρέπει να σχεδιαστεί με τέτοιο τρόπο ώστε ο χρήστης να μην μπορεί να κάνει σοβαρά σφάλματα. Κάθε δυνατότητα για λάθος θα πρέπει να ελαχιστοποιείται. Η διεπαφή θα πρέπει να παρέχει ένα μηχανισμό για την ανίχνευση των λαθών των χρηστών.
- Οι πληροφορίες σήμερα δεν βρίσκονται μόνο σε μορφή κειμένου, αλλά σε μία πληθώρα ψηφιακών μορφών, όπως χάρτες, φωτογραφίες, ήχος και βίντεο. Οι ψηφιακές βιβλιοθήκες πρέπει να παρέχουν αυτού του είδους τις πολυμεσικές πληροφορίες. Γι' αυτό το λόγο η διεπαφή οφείλει να τις υποστηρίζει.
- Οι ψηφιακές βιβλιοθήκες διατηρούν τα προφίλ των χρηστών προκειμένου να παρέχουν αποτελεσματικές υπηρεσίες. Η διεπαφή πρέπει να ενεργεί σύμφωνα με τις ρυθμίσεις που προδιαγράφονται στο προφίλ του κάθε χρήστη.
- Οι περισσότερες από τις ψηφιακές βιβλιοθήκες είναι προσβάσιμες στους χρήστες μέσω του διαδικτύου. Λόγω της μη σταθερής ταχύτητας του κάθε χρήστη, η διεπαφή χρήστη της ψηφιακής βιβλιοθήκης πρέπει να είναι ευκίνητη και απλή, χωρίς βαρέα και περιπτά γραφικά που προκαλούν επιβάρυνση στο δίκτυο και άσκοπες καθυστερήσεις.
- Η κακή εμφάνιση των πληροφοριών, τα ορθογραφικά και γραμματικά λάθη επηρεάζουν την αξιοπιστία της διεπαφής. Ως εκ τούτου, η διεπαφή θα πρέπει να παρέχει όσο το δυνατόν περισσότερο σαφείς και καλά οργανωμένες πληροφορίες για τον χρήστη.
- Οι περισσότεροι από τους χρήστες επισκέπτονται τις ψηφιακές βιβλιοθήκες για να πάρουν τις απαιτούμενες πληροφορίες σύμφωνα με τις ανάγκες τους. Η ψηφιακή βιβλιοθήκη και κατά συνέπεια η διεπαφή, πρέπει να περιέχει ένα επαρκές σύστημα αναζήτησης. Ο χρήστης αναζητά κυρίως χρησιμοποιώντας φυσική γλώσσα. Γι' αυτό είναι σημαντικό η τεχνολογία αλλά και η απεικόνιση της αναζήτησης πληροφορίας να είναι χρηστική και φιλική.
- Οι τεχνολογίες RDF και η σημασιολογική προσέγγιση παρέχουν στον χρήστη έναν εύκολο και αποδοτικό τρόπο αναζήτησης. Με την ενεργοποίηση αυτών των τεχνολογιών στο σχεδιασμό της διεπαφής θα δοθούν περισσότερα προνόμια στους χρήστες.
- Το περιβάλλον πρέπει να υποστηρίζει την ανταλλαγή και την επαναχρησιμοποίηση της πληροφορίας. Οι αναζητήσεις των χρηστών πρέπει να μπορούν εύκολα να καταγραφούν και να επαναχρησιμοποιηθούν. Οι μηχανισμοί που χρησιμοποιούνται για

την βελτίωση των αναζητήσεων είναι: α) Συνεργατικό Φιλτράρισμα και β) Κοινωνικό Φιλτράρισμα.

- Η διαθεσιμότητα της γνώσης και της πληροφορίας δεν είναι μόνο σε μία γλώσσα, ως εκ τούτου, η διεπαφή χρήστη της ψηφιακής βιβλιοθήκης πρέπει να διαθέτει πολυγλωσσική υποστήριξη.
- Δεδομένου ότι οι ψηφιακές βιβλιοθήκες τρέχουν κυρίως στο διαδίκτυο, το περιβάλλον του κάθε χρήστη μπορεί να μην είναι το ίδιο. Συνεπώς, η διεπαφή χρήστη πρέπει να είναι μία ανεξάρτητη πλατφόρμα, εφαρμόσιμη σε όλους τους τύπους των λειτουργικών συστημάτων.
- Η ανάπτυξη στον τομέα των τεχνολογιών είναι συνεχώς αυξανόμενη, παρέχοντας έτσι νέες μορφές πρόσβασης σε πληροφόρηση. Για την μελλοντική υποστήριξη των αναγκών πληροφόρησης η διεπαφή πρέπει να είναι ικανή να ενσωματώσει τις μελλοντικές εξελίξεις.

2.3.3 Δομή της πληροφορίας

Για να αναπαρασταθεί η πολύπλοκη φύση της πληροφορίας σε μία ψηφιακή βιβλιοθήκη, αρκετά ψηφιακά αντικείμενα μπορεί να ομαδοποιηθούν. Αυτή η ομαδοποίηση ονομάζεται σύνολο ψηφιακών αντικειμένων. Όλα τα ψηφιακά αντικείμενα έχουν την ίδια βασική μορφή, αλλά η δομή ενός συνόλου ψηφιακών αντικειμένων εξαρτάται από το περιεχόμενο των πληροφοριών που περιέχει.

Οι διάφοροι τύποι υλικών σε μία ψηφιακή βιβλιοθήκη μπορεί να χωριστούν σε διάφορες κατηγορίες, λόγω χάρη κείμενο σε γλώσσα υπολογιστών, αντικείμενα World Wide Web, προγράμματα υπολογιστών, ή ψηφιοποιημένα ραδιοπρογράμματα. Στο πλαίσιο κάθε κατηγορίας περιέχονται κανόνες και μετατροπές που περιγράφουν πώς θα οργανωθούν οι πληροφορίες σε σύνολα ψηφιακών αντικειμένων. Για παράδειγμα, ειδικοί κανόνες περιγράφουν πώς αναπαρίσταται ένα ψηφιοποιημένο μουσικό κομμάτι. Για κάθε κατηγορία, οι κανόνες που περιγράφουν τα ψηφιακά αντικείμενα αφορούν τον τρόπο με τον οποίο αυτά αναπαριστούν το υλικό μιας ψηφιακής βιβλιοθήκης, πώς ομαδοποιούνται σε σύνολα ψηφιακών αντικειμένων, την εσωτερική δομή κάθε ψηφιακού αντικειμένου και τα σχετικά μεταδεδομένα τους.

Λόγω της διαφορετικότητας των αποθηκευμένων δεδομένων, απαιτείται οι πληροφορίες να είναι οργανωμένες αποτελεσματικά. Μέσα στη βιβλιοθήκη, οι πληροφορίες αποθηκεύονται ως βασικές μονάδες των ψηφιακών πληροφοριών, π.χ. , ένας ψηφιοποιημένος χάρτης, ένα τμήμα του κειμένου, μια ιστοσελίδα, μία σαρωμένη φωτογραφία, κλπ. Σε ψηφιακή μορφή, κάθε βασική μονάδα είναι μια ακολουθία από bits, αλλά οι χρήστες συχνά χρειάζονται το υλικό να βρίσκεται σε ένα μεγαλύτερο επίπεδο αφαίρεσης. Κοινοί όροι, όπως η “έκθεση”, ένα “πρόγραμμα ηλεκτρονικού υπολογιστή”, ή ένας τίτλος βιβλίου μπορεί να αναφέρονται σε πολλά στοιχεία που αποτελούν παραλλαγές του ίδιου. Μπορεί να έχουν διαφορετική μορφή, μικρές διαφορές στο περιεχόμενο, διαφορετικούς περιορισμούς χρήσης, αλλά για κάποιους λόγους οι χρήστες είναι πρόθυμοι να τα θεωρήσουν ισοδύναμα .

Τα θέματα που πρέπει να αντιμετωπιστούν στη δόμηση της πληροφορίας περιλαμβάνουν τα ακόλουθα. Τα ψηφιακά υλικά συχνά σχετίζονται με άλλα υλικά με σχέσεις όπως μέρος / σύνολο, ακολουθία, κλπ. Για παράδειγμα, ένα ψηφιοποιημένο κείμενο μπορεί να αποτελείται από σελίδες, κεφάλαια, ένα ευρετήριο, απεικονίσεις, και ούτω καθεξής. Στον παγκόσμιο ιστό, ένα τυπικό στοιχείο μπορεί να περιλαμβάνει πολλές σελίδες του κειμένου, με ενσωματωμένες εικόνες, και συνδέσεις με άλλες πληροφορίες. Ένα ενιαίο πρόγραμμα υπολογιστή συναρμολογείται από πολλά αρχεία, τόσο δυαδικού όσο και πηγαίου κώδικα.

Το ίδιο αντικείμενο μπορεί να αποθηκευθεί σε διάφορες ψηφιακές μορφές. Μερικές φορές, αυτές οι μορφές είναι ακριβώς ισοδύναμες και είναι δυνατή η μετατροπή τους από τη μία στην άλλη (π.χ., μια μη συμπίεσμένη εικόνα και η αντίστοιχη συμπίεσμένη μορφή της). Σε άλλες περιπτώσεις, οι διάφορες μορφές περιέχουν διαφορετικές πληροφορίες (π.χ., διαφορετικές αναπαραστάσεις μιας σελίδας των κειμένων σε μορφή SGML και σε μορφή PostScript) .

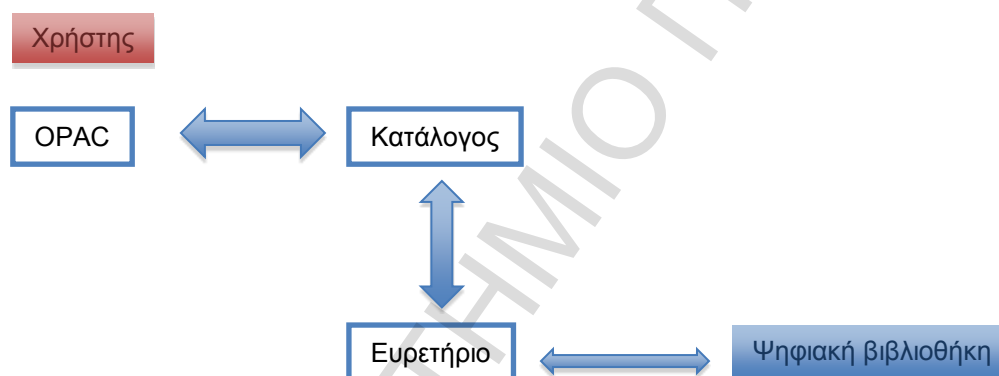
Επειδή τα ψηφιακά αντικείμενα είναι εύκολο να αλλάξουν, δημιουργούνται συνεχώς διαφορετικές εκδόσεις (ορισμένοι οργανισμοί αλλάζουν το περιεχόμενο της ιστοσελίδας τους

αρκετές φορές το μήνα). Οι εκδόσεις μπορεί να διαφέρουν σε ένα μόνο κομμάτι ή μπορεί να είναι πολύ διαφορετικές μεταξύ τους. Όταν το υπάρχον υλικό μετατρέπεται σε ψηφιακή μορφή, το ίδιο φυσικό στοιχείο μπορεί να μετατραπεί αρκετές φορές. Έτσι, μία σαρωμένη φωτογραφία μπορεί να έχει μια έκδοση υψηλής ανάλυσης, μια έκδοση μέσης ποιότητας, και μια μικρογραφία.

Κάθε έκδοση της ψηφιακής πληροφορίας μπορεί να έχει διαφορετικά δικαιώματα και άδειες χρήσης. Ο τρόπος με τον οποίο ο χρήστης επιθυμεί να έχει πρόσβαση στο υλικό μπορεί να εξαρτάται από τα χαρακτηριστικά του δικτύου, ή/και από το μέγεθος του υλικού. Η αρχιτεκτονική πληροφοριών παρέχει μια γενική προσέγγιση για την οργάνωση του υλικού μέσα στην ψηφιακή βιβλιοθήκη με τέτοιο τρόπο ώστε τα προγράμματα ηλεκτρονικών υπολογιστών να μπορούν να κατανοήσουν τη δομή του υλικού, ούτως ώστε να διεξάγουν τις αλληλεπιδράσεις που ο χρήστης επιθυμεί.

2.3.4 Ψηφιακά ράφια και OPAC

Τα ψηφιακά τεκμήρια φιλοξενούνται πλέον όχι σε συμβατικά ράφια βιβλιοθήκης αλλά σε εικονικά δηλαδή τη Περιφερειακή Μνήμη (Π.Μ.) των υπολογιστικών συστημάτων. Είναι αλήθεια ότι τα Υπολογιστικά Συστήματα που φιλοξενούν μία Ψηφιακή Βιβλιοθήκη θα πρέπει να έχουν τεράστιες δυνατότητες επέκτασης της Π.Μ. για να μπορούν να ανταποκρίνονται στον αυξανόμενο όγκο της εισαγόμενης πληροφορίας.



Σχήμα 4 OPAC Σχήμα αναζήτησης

Μέχρι τώρα έχουμε προσδιορίσει το σκέλος εκείνο που αφορά την αναζήτηση και τον εντοπισμό του υλικού. Η σχέση αυτή έχει ως εξής: ο χρήστης έχει συνήθως στη διάθεση του ένα WEB client (OPAC ή Online Public Access Catalog) μέσω μιας επιφάνειας διεπαφής (ένα interface), που του παρέχει τη δυνατότητα με ένα δομημένο τρόπο να αναζητήσει υλικό είτε στη τοπική Ψηφιακή Βάση Δεδομένων είτε αλλού. Αυτό το οποίο συμβαίνει σε αυτή την περίπτωση είναι ότι ο πελάτης Ιστού αποστέλλει την ερώτηση στο κατάλογο με τα μεταδεδομένα. Εφόσον η απάντηση από τον κατάλογο είναι θετική, θα γίνει η αντίστοιχη ευρετηρίαση με τα σχετικά αποτελέσματα και θα προβληθεί στο χρήστη. Κάθε όρος του ευρετηρίου εσωτερικά έχει ενσωματώσει και τις σχετικές παραπομπές για την ανεύρεση και προβολή του ψηφιακού τεκμηρίου.

2.3.5 XML, Marc και Μεταδεδομένα

Η XML είναι ένα σύνολο κανόνων (ή διαφορετικά ένα πακέτο κατευθυντήριων γραμμών ή συμβάσεων) για το σχεδιασμό μορφών κειμένου οι οποίες διευκολύνουν τη δόμηση των δεδομένων. Η XML δεν είναι γλώσσα προγραμματισμού. Η XML διευκολύνει τον υπολογιστή να παράγει δεδομένα, να διαβάζει δεδομένα και να εξασφαλίζει τη σαφήνεια της δομής των δεδομένων. Η XML αποφεύγει τις συνήθεις παγίδες του σχεδιασμού γλωσσών: είναι επεκτάσιμη, ανεξάρτητη συστήματος υλικού και μπορεί να υποστηρίξει διεθνείς και τοπικές προσαρμογές. Η XML είναι πλήρως συμβατή με Unicode.

Προκειμένου να κατανοήσουμε την λειτουργία της Ψηφιακής Βιβλιοθήκης θα πρέπει να κατανοήσουμε την σημασία και τον ρόλο των μεταδεδομένων. Σε μια συμβατική βιβλιοθήκη το περιεχόμενο ενός τεκμηρίου ορίζεται από μια εγγραφή MARC. Σε μια Ψηφιακή Βιβλιοθήκη το περιεχόμενο ενός ψηφιακού τεκμηρίου ορίζεται από τα Metadata. Τα Metadata είναι η MARC εγγραφή του ψηφιακού τεκμηρίου. Είναι τα κύρια χαρακτηριστικά του τεκμηρίου (συνήθως στη διεθνή βιβλιογραφία περιγράφονται επίσης ως data about data).

Περιεχόμενο	Πνευματικά Δικαιώματα	Instantiation
Title	Creator	Date
Subject	Publisher	Type
Description	Contributor	Format
Source	Rights	Identifier
Language		
Relation		
Coverage		

Πίνακας 2 Μεταδεδομένα

Όπως σε μια συμβατική βιβλιοθήκη ένας κατάλογος οργανώνει και ταξινομεί το υλικό, κατά παρόμοιο τρόπο σε μια Ψηφιακή Βιβλιοθήκη ο κατάλογος με τα Metadata είναι αυτός που επιτυγχάνει την οργάνωση και την ταξινόμηση του υλικού της Ψηφιακής Βιβλιοθήκης. Αυτό γίνεται ως εξής. Ένας χρήστης έχει πρόσβαση στη Ψηφιακή Βιβλιοθήκη μέσω ενός OPAC. Ο OPAC έχει ως σημείο αναφοράς τον κατάλογο των metadata. Ο κατάλογος των metadata έχει δυνατότητες ταξινόμησης ανά συγγραφέα, τίτλο, θέμα κλπ άρα η πληροφορία ευρετηριάζεται σύμφωνα με αυτά τα δεδομένα. Ταυτόχρονα δε, ενσωματώνει τις αντίστοιχες παραπομπές στα ψηφιακά τεκμήρια όπως ένας τοπογραφικός κατάλογος.

2.4 Οι ψηφιακές βιβλιοθήκες τώρα

Κατά τη διάρκεια των τελευταίων ετών, παρατηρούμε ότι οι ψηφιακές βιβλιοθήκες άρχισαν να εκσυγχρονίζονται και να δείχνουν μια πιο κοινωνική εικόνα [19]. Αυτό οφείλεται στις αλλαγές που έχουν λάβει χώρα στο World Wide Web (π.χ. Web 2.0) [20], [21], φέρνοντας στο φως νέες τεχνολογίες και νέα μέσα συνεργατικότητας και επικοινωνίας. Σύμφωνα με τον Maness [19], οι τεχνολογίες του Web 2.0, όπως για παράδειγμα η σύγχρονη συζήτηση (instant messaging), τα blogs, τα wikis, τα RSS feeds και τα κοινωνικά δίκτυα, έχουν επηρεάσει τις ψηφιακές κοινότητες βιβλιοθηκών και τις αλλάζουν σταδιακά από την παραδοσιακή Βιβλιοθήκη στη Βιβλιοθήκη 2.0.

Η Βιβλιοθήκη 2.0 έχει μελετηθεί από πολλούς ερευνητές τα τελευταία χρόνια. Τα έργα αυτά, έχουν προσδιορίσει τον ορισμό και τις προδιαγραφές που την αφορούν [22], ενώ την ίδια στιγμή περιγράφονται οι αρχές που την διέπουν. Ειδικότερα, ο Habib παρουσιάζει στο έργο του ένα θεωρητικό μοντέλο της Ακαδημαϊκής Βιβλιοθήκης 2.0. Περιγράφει ποιες λειτουργίες θα πρέπει να διαθέτει ένα τέτοιο σύστημα. Η έρευνά του παρουσιάζει ένα πιθανό λειτουργικό σενάριο ενός τέτοιου συστήματος, όπου μεταξύ άλλων αναφέρεται ότι φοιτητές και διδακτικό και ερευνητικό προσωπικό θα μπορούσαν να δημιουργήσουν ιδιωτικές ομάδες εργασίας στις οποίες θα μπορούσαν να μοιραστούν και να σχολιάσουν θέματα με τρόπο σύγχρονο αλλά και ασύγχρονο [23].

Εκτός από τον ορισμό του θεωρητικού μοντέλου, έχουν επίσης συζητηθεί θέματα εφαρμογής. Σύμφωνα με την έρευνα του Chalou, εκτός από τη χρήση ενός ολοκληρωμένου συστήματος βιβλιοθηκών, που ενδεχομένως δεν παρέχει όλες τις λειτουργίες 2.0, μπορούμε να χρησιμοποιήσουμε ένα ειδικά προσαρμοσμένο σύστημα διαχείρισης περιεχομένου, όπως το Drupal, το SPIP ή το WordPress. Στην ψηφιακή βιβλιοθήκη του βελγικού Κέντρου Γνώσης Υγείας (KCE), η ερευνητική ομάδα του Chalou, εφάρμοσε τις λειτουργίες Web 2.0 και επέτρεψε στους χρήστες του να συμμετάσχουν στον εμπλουτισμό τους [24].

Στην επόμενη ενότητα θα αναφέρουμε μερικές από τις πιο σύγχρονες ψηφιακές βιβλιοθήκες του σήμερα και θα αναλύσουμε τα χαρακτηριστικά τους.

2.4.1. Γνωστές ψηφιακές βιβλιοθήκες

Axiell Arena: Το Axiell Arena (http://www.axiell.com/axiell_arena) είναι μία εμπορική ψηφιακή βιβλιοθήκη, υλοποιημένη από την Axiell, που αναπτύχθηκε με τη χρήση λογισμικού ανοιχτού κώδικα. Στο Axiell Arena, μία βιβλιοθήκη υλοποιείται σύμφωνα με τις αρχές της Βιβλιοθήκης 2.0, προσφέροντας μια ιστοσελίδα που είναι πάντα διαθέσιμη. Οι επισκέπτες της βιβλιοθήκης μπορούν να επικοινωνήσουν στο φόρουμ, ενώ μπορούν επίσης να δημιουργήσουν το δικό τους ψηφιακό ράφι, το οποίο περιλαμβάνει πληροφορίες και μια ποικιλία από διαφορετικούς τίτλους. Επίσης, ενθαρρύνονται να ξεκινήσουν διάλογο με τη βιβλιοθήκη και να παρέχουν ποιοτικά δεδομένα για την ανάπτυξη των καταλόγων της, π.χ. μέσω tagging, reviews, αξιολόγησης ή την έναρξη μιας συζήτησης στο φόρουμ σε σχέση με ένα συγκεκριμένο τίτλο.

OPACIAL: Το OPACIAL (<http://library.panteion.gr/opacial/>), είναι ένα έργο ανοιχτού πηγαίου κώδικα που μπορεί να ενισχύσει τις υπηρεσίες που παρέχονται από τη βιβλιοθήκη ενός ιδρύματος με τη λειτουργικότητα του Web 2.0. Αποτελεί ουσιαστικά παράδειγμα ενός πρώιμου συστήματος Βιβλιοθήκης 2.0 [25]. Το OPACIAL προσφέρει στους χρήστες του τη δυνατότητα να αναζητήσουν και να περιηγηθούν, καθώς και να ανακτήσουν αντικείμενα (έγγραφα), με βάση τις ετικέτες μέσω ενός σύννεφου ετικετών (tag cloud), ενώ είναι σε θέση να προτείνει σχετικές ετικέτες που ταιριάζουν σε ένα αντικείμενο. Επίσης, δίνει τη δυνατότητα για μια ενοποιημένη αναζήτηση των σχετικών εγγράφων από την ψηφιακή βιβλιοθήκη του ιδρύματος, με βάση κοινές θεματικές ετικέτες. Ένα πολύ σημαντικό θέμα της προαναφερθείσας εργασίας είναι ότι, προκειμένου να διασφαλιστεί η επιτυχία του εκσυγχρονισμού των ψηφιακών βιβλιοθηκών, θα πρέπει πρώτα να ερευνηθεί ποιες υπηρεσίες θα πρέπει να αναβαθμιστούν σε έκδοση 2.0 [26].

Arnetminer: Το σύστημα Arnetminer (<http://arnetminer.org>) έχει σχεδιαστεί για την αναζήτηση και την εκτέλεση εργασιών εξόρυξης δεδομένων από ακαδημαϊκές δημοσιεύσεις στο διαδίκτυο, με τη χρήση της ανάλυσης κοινωνικών δικτύων για τον εντοπισμό των συνδέσεων μεταξύ ερευνητών, συνεδρίων και εκδόσεων [59]. Αυτό επιτρέπει την παροχή υπηρεσιών, όπως η γεωγραφική αναζήτηση, ένωση αναζητήσεων, η αναζήτηση προγράμματος, η ακαδημαϊκή αξιολόγηση επιδόσεων και τα θέματα μοντελοποίησης.

Δημιουργήθηκε από τον καθηγητή Jie Tang από το Πανεπιστήμιο Tsinghua της Κίνας και ξεκίνησε τον Μάρτιο του 2006 ως ένα ερευνητικό πρόγραμμα για την ανάλυση της κοινωνικής επιρροής, και της κατάταξης και εξόρυξης των κοινωνικών δικτύων. Έχει τεθεί σε λειτουργία για περισσότερα από τρία χρόνια, εξυπηρετεί πάνω από 1.300.000 ερευνητές και διαθέτει πάνω από τρία εκατομμύρια δημοσιεύσεις. Χρησιμοποιείται συνήθως σε πανεπιστήμια για τον εντοπισμό σχέσεων και εξαγωγή στατιστικών συσχετίσεων σχετικά με την έρευνα και τους ερευνητές. Το προϊόν έχει χρησιμοποιηθεί σε πλατφόρμα SciVerse του οίκου Elsevier, καθώς και σε επιστημονικά συνέδρια, όπως τα SIGKDD, ICDM, PKDD, WSDM.

Το σύστημα Arnetminer εξάγει αυτόματα το προφίλ των ερευνητών από το διαδίκτυο. Συλλέγει και προσδιορίζει τις σχετικές σελίδες και στη συνέχεια χρησιμοποιεί μια ενοποιημένη προσέγγιση για την εξαγωγή δεδομένων από τα αναγνωρισμένα έγγραφα. Εξάγει επίσης εκδόσεις από online ψηφιακές βιβλιοθήκες χρησιμοποιώντας ευρετικούς κανόνες. Ως αναγνωριστικό χρησιμοποιείται το όνομα του ερευνητή.

arXiv: Το arXiv (<http://arxiv.org>) αναπτύχθηκε αρχικά από τον Paul Ginsparg [27], εν μέρει, για να αντικαταστήσει μια λίστα διανομής ηλεκτρονικού ταχυδρομείου για προδημοσιεύσεις που λειτουργούσε χειροκίνητα από τον Joanne Cohn για περίπου δύο χρόνια. Ξεκίνησε τον Αύγουστο του 1991 ως αποθήκη για τις προδημοσιεύσεις στη φυσική και αργότερα επεκτάθηκε για να συμπεριλάβει την αστρονομία, τα μαθηματικά, την επιστήμη των υπολογιστών, την ποσοτική βιολογία και πιο πρόσφατα τα στατιστικά στοιχεία. Σύντομα έγινε φανερό ότι υπήρχε ζήτηση για μακροχρόνια - μακροπρόθεσμη διατήρηση των προδημοσιεύσεων. Ο όρος e-print, εγκρίθηκε για να περιγράψει τα άρθρα τα οποία περιέχει. Ο Ginsparg έγινε αποδέκτης του βραβείου MacArthur Fellowship το 2002 για τη δημιουργία του arXiv.

Το 2004, εισήχθη ένα σύστημα θεώρησης. Το νέο σύστημα έχει προσελκύσει το δικό του μερίδιο της κριτικής για δήθεν περιορισμό της έρευνας. Σύμφωνα με το σύστημα αυτό, ένας συγγραφέας πρέπει πρώτα να εγκριθεί. Η θεώρηση γίνεται είτε από άλλο συγγραφέα του arXiv, ή είναι αυτόματη, και εκτελείται με βάση διάφορα κριτήρια τα οποία δεν ανακοινώνονται. Η

πλειοψηφία των e-prints υποβάλλονται επίσης σε περιοδικά για δημοσίευση, αλλά κάποιες εργασίες παραμένουν μόνο ως e-prints και ποτέ δεν δημοσιεύονται σε ένα περιοδικό με κριτές.

Τα έγγραφα μπορούν να υποβληθούν σε μία πληθώρα μορφών, συμπεριλαμβανομένων των LaTeX και PDF.

Invenio: Το Invenio (<https://invenio-software.org>) είναι ένα πακέτο λογισμικού ανοιχτού κώδικα που παρέχει τα εργαλεία για τη διαχείριση των ψηφιακών στοιχείων σε μία αποθήκη πληροφοριών ενός ιδρύματος. Το λογισμικό χρησιμοποιείται συνήθως για τη δημιουργία αποθετηρίων ανοικτής πρόσβασης και ως ψηφιακή βιβλιοθήκη. Το Invenio έχει αναπτυχθεί από το Ινστιτούτο CERN, και είναι ελεύθερα διαθέσιμο στους χρήστες. Πριν από την 1 Ιουλίου 2006, το πακέτο ονομαζόταν CDSware. Από την έκδοση 0.90.0 του 2006, το λογισμικό έχει μετονομαστεί σε CDS Invenio. Ο πάροχος υπηρεσιών Tind Technologies ιδρύθηκε το 2013 για να φιλοξενήσει την αυξανόμενη ζήτηση για την υποστήριξη του Invenio [28].

CiteSeer: Το CiteSeer (<https://citeseer.ist.psu.edu/>) ήταν μια δημόσια μηχανή αναζήτησης και ψηφιακή βιβλιοθήκη, η οποία αφορούσε επιστημονικές εργασίες, κυρίως στους τομείς των ηλεκτρονικών υπολογιστών και της επιστήμης της πληροφορίας, το οποίο έχει πλέον αντικατασταθεί από το CiteSeerX. Πολλοί θεωρούν ότι αποτελεί την πρώτη προσέγγιση σχετικά με επιστημονικές μηχανές αναζήτησης [29]. Δώθηκε στο κοινό το 1998 και είχε πολλά νέα χαρακτηριστικά διαθέσιμα σε σχέση με τις ακαδημαϊκές βιβλιοθήκες που υπήρχαν μέχρι εκείνη τη στιγμή. Αυτές περιελάμβαναν:

- Αυτόνομη Ευρετηρίαση Παραπομπών: Δημιουργείται αυτόματα ένα ευρετήριο παραπομπών που μπορεί να χρησιμοποιηθεί για την αναζήτηση βιβλιογραφίας και την αξιολόγηση.
- Στατιστικά δεδομένα για τις παραπομπές αλλά και τα συναφή έγγραφα υπολογίζονται για όλα τα άρθρα που αναφέρονται στη βάση δεδομένων, και όχι μόνο για τα άρθρα που βρίσκονται στο ευρετήριο.
- Σύνδεση αναφορών, η οποία επιτρέπει την περιήγηση στην βάση δεδομένων χρησιμοποιώντας τις παραπομπές.
- Το πλαίσιο των παραπομπών σε κάθε συγκεκριμένο άρθρο, επιτρέπει στους ερευνητές να δουν γρήγορα και εύκολα τι έχουν πει άλλοι ερευνητές σχετικά με το άρθρο.
- Σχετικά έγγραφα με το κάθε άρθρο εμφανίζονται χρησιμοποιώντας παραπομπές.

Συχνά θεωρείται ότι είναι το πρώτο αυτοματοποιημένο σύστημα ευρετηρίασης παραπομπών και θεωρήθηκε πρόδρομος των γνωστών ακαδημαϊκών εργαλείων αναζήτησης, όπως το Google Scholar και το Microsoft Academic Search. Μηχανές αναζήτησης όπως το CiteSeer, συνήθως συλλέγουν έγγραφα από κοινούς δικτυακούς τόπους και δεν κάνουν διάσχιση των ιστοσελίδων των εκδοτών. Ως εκ τούτου, οι συγγραφείς των οποίων τα έγγραφα είναι ελεύθερα διαθέσιμα είναι πιο πιθανό να εκπροσωπούνται στη μηχανή αναζήτησης.

Στόχος του CiteSeer είναι να βελτιώσει τη διάδοση και την πρόσβαση στην ακαδημαϊκή και επιστημονική βιβλιογραφία. Ως υπηρεσία μη κερδοσκοπικού χαρακτήρα, που μπορεί να χρησιμοποιείται ελεύθερα από όλους, έχει θεωρηθεί ως μέρος του κινήματος της ανοικτής πρόσβασης, που έχει ως στόχο να επιτρέψει μεγαλύτερη και ευκολότερη πρόσβαση στην επιστημονική βιβλιογραφία.

CiteSeerX: Το CiteSeerX (<http://citeseerx.ist.psu.edu/>) αντικατέστησε το CiteSeer και όλα τα επερωτήματα του διοχετεύθηκαν στο CiteSeerX. Το CiteSeerX είναι μια δημόσια μηχανή αναζήτησης, ψηφιακή βιβλιοθήκη και αποθετήριο επιστημονικών εργασιών με επίκεντρο την πληροφορική και την επιστήμη της πληροφορίας [30]. Κυκλοφόρησε το 2008 και συνεχίζει να υποστηρίζει τους στόχους που περιγράφονται από CiteSeer.

Το CiteSeerX εξακολουθεί να βαθμολογείται ως ένα από τα κορυφαία αποθετήρια του κόσμου. Διαθέτει σήμερα πάνω από 2 εκατομμύρια έγγραφα με σχεδόν 2 εκατομμύρια μοναδικούς συγγραφείς και 40 εκατομμύρια αναφορές και παραπομπές.

Civil Rights Litigation Clearinghouse: Η Civil Rights Litigation Clearinghouse (<http://www.clearinghouse.net>) είναι μια ιστοσελίδα που δημοσιεύει πληροφορίες και έγγραφα που σχετίζονται με την άσκηση των πολιτικών δικαιωμάτων. Η συγκεκριμένη ιστοσελίδα ιδρύθηκε από τον καθηγητή της νομικής Margo Schlanger το 2005, στο Πανεπιστήμιο Ουάσιγκτον στο Σεντ Λούι, και μετακόμισε το 2009 στο Πανεπιστήμιο του Michigan [31].

Το Clearinghouse διαθέτει τις πληροφορίες και τα έγγραφα του χωρίς κόστος σε νομοθέτες, ερευνητές, δικηγόρους, καθηγητές, φοιτητές και το ευρύ κοινό. Αποτελεί την κύρια πηγή στο διαδίκτυο για το είδος των περιπτώσεων που καλύπτει, επιτρέποντας στο κοινό πρόσβαση σε έγγραφα, συμπεριλαμβανομένων ενστάσεων κ.α.

Το World Wide Web Virtual Library ήταν ο πρώτος δείκτης περιεχομένου στον Παγκόσμιο Ιστό και εξακολουθεί να λειτουργεί ως ένας κατάλογος των ηλεκτρονικών κειμένων και πηγών πληροφόρησης στο διαδίκτυο. Ξεκίνησε από τον Tim Berners-Lee, τον δημιουργό της HTML και του World Wide Web, το 1991 στο CERN στη Γενεύη [32]. Σε αντίθεση με τις εμπορικές ιστοσελίδες-ερευτήρια, διευθύνεται από μία συνομοσπονδία εθελοντών. Οι επιμέρους δείκτες, ή εικονικές βιβλιοθήκες, βρίσκονται σε εκατοντάδες διαφορετικούς εξυπηρετητές σε όλο τον κόσμο.

Το World Wide Web Virtual Library έχει αυξηθεί τα τελευταία χρόνια, έτσι ώστε να υπάρχουν σήμερα περίπου 300 υπο-βιβλιοθήκες στο πλαίσιο της κεντρικής βιβλιοθήκης.

Μία συνοπτική περιγραφή των συστημάτων που αναλύθηκαν παραπάνω, δίνεται στον Πίνακα 3.

Όνομα	Θέμα	Περιγραφή	Πάροχος
Axiell Arena	Εμπόριο	Εμπορική Βιβλιοθήκη	Ψηφιακή Axiell
OPACIAL	Library 2.0	Library 2.0	OPACIAL
Arnetminer	Επιστήμη Υπολογιστών	Δωρεάν online υπηρεσία για ακαδημαϊκές αναζητήσεις.	
ArXiv	Επιστήμη	Έρευνες στα μαθηματικά, φυσική, επιστήμη των υπολογιστών, ποσοτική βιολογία και τις στατιστικές.	Cornell University
Invenio	Επιστήμη	Ηλεκτρονική βιβλιοθήκη για τα σχετικά έγγραφα του CERN	CERN
CiteSeerX	Επιστήμη Υπολογιστών	Ψηφιακή Βιβλιοθήκη και μηχανή αναζήτησης για επιστημονική λογοτεχνία	Pennsylvania State University
Civil Rights Litigation Clearinghouse	Πολιτικά Δικαιώματα	Ψηφιακή Βιβλιοθήκη με έγγραφα νομικού και πολιτικού περιεχομένου	University of Michigan
World Wide Web Virtual Library	Γενικό	Κατάλογος ηλεκτρονικών κειμένων	European University Institute

Πίνακας 3 Μοντέρνες Ψηφιακές Βιβλιοθήκες

2.4.2 Συνεργατικότητα σε ψηφιακές βιβλιοθήκες

Το έργο των βιβλιοθηκονόμων, καθώς και η χρήση των πόρων της βιβλιοθήκης θεωρείται μία μοναχική διαδικασία. Παρόλα αυτά, είναι προφανές ότι όλα τα ενδιαφερόμενα μέλη πρέπει να μπορούν να έχουν μία επικοινωνιακή συνεργασία.

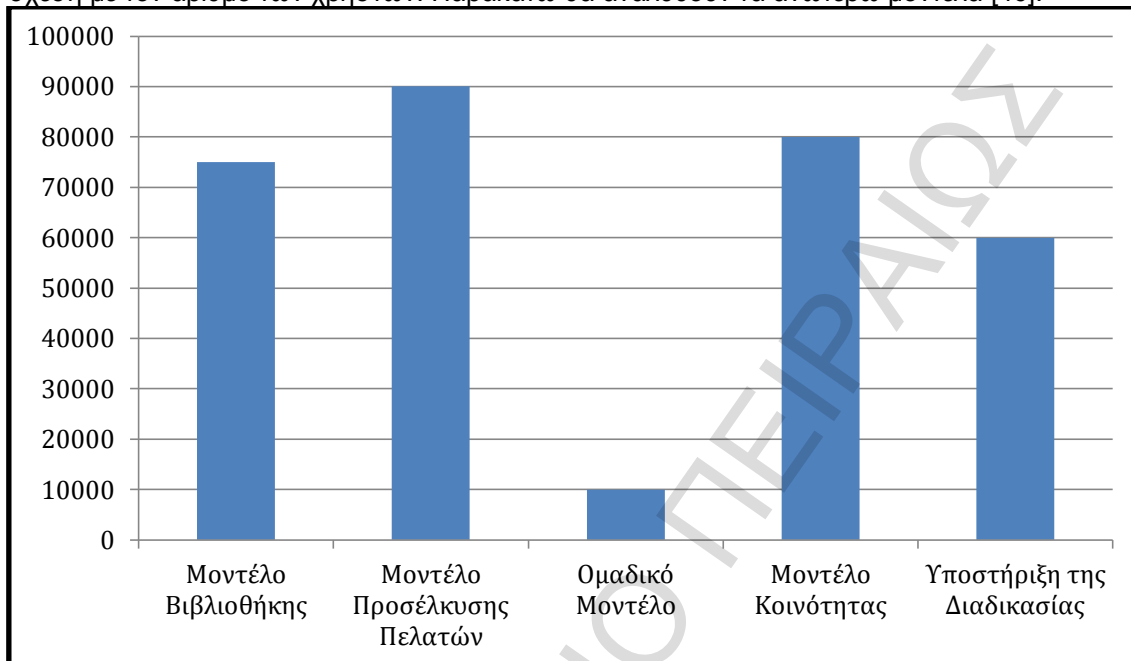
Η συνεργατικότητα είναι ένα θέμα που αφορά ένα μεγάλο αριθμό ιδρυμάτων και οργανώσεων. Υπάρχουν ορισμένες πτυχές της συνεργατικότητας που είναι κοινές για όλους. Στο έργο των Butler και Coleman εξετάζονται πέντε γενικά μοντέλα τα οποία ποικίλουν σε σχέση με το επίπεδο της διαδραστικότητας [44]. Σύμφωνα με αυτούς, οι βιβλιοθήκες είναι ο τύπος του οργανισμού που παρουσιάζει περισσότερο συλλογική λειτουργικότητα. Η πλειοψηφία των συνεργατικών περιβαλλόντων μπορεί να ταξινομηθεί σε ένα από αυτά τα πέντε μοντέλα.

Αυτά τα γενικά πρότυπα συνεργατικότητας είναι:

- Μοντέλο Βιβλιοθήκης
- Μοντέλο Προσέλκυσης Πελατών

- Ομαδικό Μοντέλο
- Μοντέλο Κοινότητας
- Υποστήριξη της διαδικασίας

Στο Σχήμα 9 μπορούμε να δούμε το επίπεδο διαδραστικότητας του κάθε μοντέλου σε σχέση με τον αριθμό των χρηστών. Παρακάτω θα αναλυθούν τα ανωτέρω μοντέλα [46].



Σχήμα 5 Το επίπεδο διαδραστικότητας των μοντέλων σε σχέση με τον αριθμό των χρηστών

Μοντέλο Βιβλιοθήκης

Αυτό είναι το απλούστερο και ίσως το πιο κοινό μοντέλο. Το μοντέλο προσφέρει κοινόχρηστα στοιχεία που μπορούν να προσπελαστούν από ένα μεγάλο αριθμό χρηστών. Το μοντέλο αυτό δεν είναι τόσο κατάλληλο για ενεργή αλληλεπίδραση με άλλα μοντέλα. Παρέχει αμοιβαία πρόσβαση στην κοινή συνήθως μακροχρόνια χρήση του περιεχομένου. Εδώ υπάρχουν πολλοί περισσότεροι χρήστες του περιεχομένου από ότι υπάρχουν πάροχοι.

Το περιεχόμενο το διαχειρίζεται συνήθως μια μικρή ομάδα ανθρώπων, και υπάρχει συνήθως λίγο ή και καθόλου ανατροφοδότηση ή αλληλεπίδραση με τους δημιουργούς του περιεχομένου. Το περιεχόμενο χρησιμοποιείται συχνά με ασύγχρονες μορφές συνεργατικότητας αν και είναι δυνατόν να χρησιμοποιηθεί και συνεργατικότητα σε πραγματικό χρόνο.

Υπάρχει μια περίπλοκη ευρετηρίαση και ο μηχανισμός ανάκτησης λειτουργεί με βάση τις λέξεις-κλειδιά. Μια πιθανή χρήση ενός μοντέλου βιβλιοθήκης σε επίπεδο συνεργατικότητας, είναι να προσφέρει το περιεχόμενο όλων των συμμετεχόντων για κοινή χρήση.

Μία τέτοια βιβλιοθήκη θα διαθέτει μία κοινή βάση δεδομένων ώστε να παρέχει τους καταλόγους και τις λειτουργίες αναζήτησης στους χρήστες για να βρουν τις πληροφορίες που χρειάζονται. Η Βάση δεδομένων θα πρέπει να υπάρχει για ένα μεγάλο χρονικό διάστημα, να συντηρείται συνεχώς και να αναθεωρείται από μια μικρή ομάδα προσωπικού.

Μοντέλο Προσέλκυσης Πελατών

Αυτό είναι το επόμενο μοντέλο που παρουσιάζεται από τους Butler και Coleman [44]. Αφορά αιτήσεις από ένα μικρό αριθμό αιτούντων και πολλές απαντήσεις από τους ερωτηθέντες. Το μοντέλο χαρακτηρίζεται από ένα συνήθως μεγαλύτερο αριθμό ερωτηθέντων από ότι αιτούντων. Η συνεργατική αλληλεπίδραση είναι σχεδόν πάντα ασύγχρονη και περνά μέσα από

e-mail ή ιστοσελίδες. Υπάρχει συνήθως αυτόματη ειδοποίηση προς τους συμμετέχοντες των νέων αιτήσεων και απαντήσεων.

Αυτό το μοντέλο είναι λιγότερο κατάλληλο για την υλοποίηση της συνεργατικότητας σε μία ψηφιακή βιβλιοθήκη, αλλά θα μπορούσε να λειτουργήσει και γι' αυτό χρησιμοποιείται από οργανισμούς, κυρίως για την κριτική σχολίων σε κείμενα.

Ομαδικό Μοντέλο

Το τρίτο μοντέλο είναι το μοντέλο της ομάδας. Το μοντέλο αυτό συνήθως χρησιμοποιείται για να ενεργοποιήσει τις δραστηριότητες της ομάδας. Χαρακτηρίζεται από το γεγονός ότι τα μέλη της μοιράζονται μερικούς κοινούς στόχους και έχουν ένα κοινό μερίδιο στην επιτυχία τους.

Οι συμμετέχοντες συχνά δεσμεύονται από τις παραμέτρους του έργου. Το μέγεθος της ομάδας συνήθως περιορίζεται σε ένα μικρό αριθμό, από δύο μέχρι είκοσι μέλη. Υπάρχει ένα πολύ υψηλό επίπεδο διαδραστικότητας και τα περισσότερα μέλη μπορούν να διαβάσουν αλλά και να γράψουν περιεχόμενο. Συχνές εκδηλώσεις διαδραστικότητας είναι η ταυτόχρονη επεξεργασία κειμένων, οι ερωταπαντήσεις και οι ενημερωτικές συναντήσεις.

Η ηλεκτρονική υλοποίηση του μοντέλου πρέπει να παρέχει αρκετές δυνατότητες ώστε τα μέλη να μπορούν να μοιραστούν ιδέες, να κάνουν ερωτήσεις, και να αποθηκεύουν τα σημαντικά έγγραφα της ομάδα. Τα μέλη της ομάδας πρέπει να διαθέτουν τα μέσα για την πραγματοποίηση των ηλεκτρονικών συζητήσεων προκειμένου να εργαστούν πάνω στα ζητήματά τους. Οι ομάδες εργάζονται τόσο σε πραγματικό χρόνο όσο και ασύγχρονα, και έτσι το εργαλείο το οποίο υλοποιεί το μοντέλο πρέπει να υποστηρίζει και τους δύο τύπους.

Στο πεδίο της συνεργατικότητας των ψηφιακών βιβλιοθηκών, το μοντέλο αυτό δεν θα βρει συχνή εφαρμογή, αλλά εξακολουθεί να μπορεί να χρησιμοποιηθεί περιστασιακά. Για παράδειγμα, θα μπορούσαμε να φανταστούμε ότι οι ψηφιακές βιβλιοθήκες ξεκινούν να εργάζονται επάνω σε κάποιο κοινό ερευνητικό έργο και επιλέγουν αρκετά μέλη του προσωπικού για αυτό το σκοπό. Αυτό σίγουρα θα οδηγήσει σε μια ενεργή, και όχι παθητική συνεργασία μεταξύ των ψηφιακών βιβλιοθηκών, και θα δημιουργήσει την ανάγκη για ενεργή αλληλεπίδραση. Σε αυτή την περίπτωση, το ομαδικό μοντέλο είναι αρκετά αποτελεσματικό.

Μοντέλο Κοινότητας

Αυτό είναι ένα λιγότερο κοινό, αλλά πολύ ισχυρό μοντέλο. Χρησιμοποιείται για να διευκολύνει τις δραστηριότητες μιας κοινότητας. Τα μέλη της κοινότητας επιδιώκουν να επεκτείνουν τις γνώσεις τους και την κατανόησή τους πάνω στην περιοχή του ενδιαφέροντός τους. Οι κοινότητες μπορούν να διαχειριστούν, ή να επεξεργαστούν, και όλα τα μέλη είναι ευπρόσδεκτα να διαβάσουν και να γράψουν το περιεχόμενο, αν και τα περισσότερα μέλη ενδιαφέρονται κυρίως για την ανάγνωση. Ο αριθμός των ενεργών συνεισφερόντων είναι συνήθως περίπου το 10% του πληθυσμού της κοινότητας. Οι περισσότερες από τις αλληλεπιδράσεις είναι ασύγχρονες, αλλά με την πάροδο του χρόνου και την άνοδο της τεχνολογίας έχουν προκύψει αρκετές κοινότητες που βασίζονται σε εργαλεία τύπου chat. Ένα μοντέλο κοινότητας συνήθως οργανώνεται γύρω από τους ανθρώπους που μοιράζονται ένα κοινό επαγγελματικό πεδίο.

Παλιότερα, διάφορες διαδικτυακές κοινότητες διέθεταν συνήθως λίστες διανομής e-mail. Οι πρόσφατες εφαρμογές επιτρέπουν προηγμένες δυνατότητες για συζήτηση. Εκτός αυτού, μπορούν να χρησιμοποιηθούν και άλλες λειτουργίες, όπως αναζήτηση, χάρτες, κυρίως από μεγάλες σε μέγεθος κοινότητες. Η λειτουργία αναζήτησης επιτρέπει την ανάκτηση πληροφοριών που μπορεί να είναι παλιές και θα χανόταν αλλιώς. Εάν υπάρχουν συντονιστές, μπορούν να παρακολουθούν την ποιότητα του συστήματος, να αφαιρούν περιττές καταχωρήσεις και να διευκολύνουν τις απαντήσεις από αναγνωρισμένους εμπειρογνώμονες. Η ειδοποίηση με e-mail έχει αποδειχθεί ότι είναι ένα ισχυρό εργαλείο για την προσέλκυση μελών στην κοινότητα.

Για τη συνεργατικότητα στις ψηφιακές βιβλιοθήκες αυτό το μοντέλο μπορεί να χρησιμοποιηθεί εξίσου με τα προηγούμενα μοντέλα. Για παράδειγμα, οι βιβλιοθήκες θα μπορούσαν να συμφωνήσουν σε συγκεκριμένες τεχνολογίες, πρωτόκολλα και πρότυπα που θα χρησιμοποιηθούν για την οικοδόμηση και τη διατήρηση της πληροφορίας. Στη συνέχεια, θα

μπορούσαν να ενδιαφερθούν να μάθουν για την ανάπτυξη της τεχνικής υποστήριξης που έχουν επιλέξει, σχετικά με τις νέες δυνατότητες και λειτουργίες. Ή μπορεί να έχουν ερωτήσεις σχετικά με τη βελτίωση της υλοποίησης.

Υποστήριξη της διαδικασίας

Το τελευταίο μοντέλο που περιγράφεται από τους Butler και Coleman εξυπηρετεί τη χρήση των τεχνολογιών συνεργατικότητας σε μια ροή εργασίας (work flow). Μερικά από τα χαρακτηριστικά του αποτελούν πολύπλοκες ή έκτακτες διαδικασίες, οι οποίες εκτελούνται συχνά. Το μοντέλο χρησιμοποιείται συχνά σε συνδυασμό με άλλα μοντέλα. Οι κρίσιμες οργανωτικές συνεργατικές διαδικασίες που το μοντέλο θα μπορούσε να χρησιμοποιεί περιλαμβάνουν την ανάπτυξη νέων προϊόντων, πωλήσεων/μάρκετινγκ, εξυπηρέτηση και υποστήριξη πελατών, κατάρτιση κλπ. Το μοντέλο προσφέρει τη δυνατότητα να δημιουργούνται προσαρμοσμένες μορφές που υποστηρίζουν αυτές τις διεργασίες. Το μοντέλο αυτό προσφέρει συχνά μια επισκόπηση της προόδου των διαδικασιών στον διαχειριστή του συστήματος.

Αυτό το μοντέλο ταιριάζει στα συνεργατικά πρότυπα, επειδή υπάρχει συνήθως μια ισχυρή ανάγκη να αλληλεπιδρούν γύρω από μια διαδικασία για την επίλυση προβλημάτων και επιπλοκών. Σε αυτό το πλαίσιο, είναι άξιος αναφοράς ο νόμος του Pareto (ο κανόνας 20/80): Το 20% των πόρων χρησιμοποιούνται για να υποστηρίξουν το 80% των τυποποιημένων αλληλεπιδράσεων, ενώ το υπόλοιπο 80% των πόρων πηγαίνουν για την υποστήριξη του 20% των μη τυποποιημένων αλληλεπιδράσεων.

Είναι δύσκολο να βρεθεί ένα παράδειγμα της εφαρμογής αυτού του μοντέλου για την ανάπτυξη της συνεργατικότητας σε ψηφιακές βιβλιοθήκες. Ένα πιθανό σενάριο θα μπορούσε να είναι το εξής: μία από τις πληροφορίες, που προσφέρει η συνεργατική ψηφιακή βιβλιοθήκη είναι ένα προστατευμένο έγγραφο, το οποίο προσφέρεται προς πώληση. Ένας από τους χρήστες αποφασίζει να αγοράσει αυτό το προϊόν και το ζητάει σε έναν από τους ιστότοπους της βιβλιοθήκης, και κάνει μια πληρωμή μέσω του μηχανισμού που προσφέρεται σε αυτό το site. Η παραγγελία μεταφέρεται αυτόματα στη βάση δεδομένων μίας άλλης ψηφιακής βιβλιοθήκης, που βρίσκεται σε συνεργασία με την πρώτη. Σε λίγες ημέρες ο πελάτης καταγγέλλει ότι έλαβε λάθος προϊόν. Εδώ, προκύπτει μία κατάσταση σφάλματος, όπου ο αντιπρόσωπος πρέπει να λάβει πολλά μέτρα για να μάθει πώς αυτό θα μπορούσε να συμβεί, όπως η παρακολούθηση της παραγγελίας, η επικοινωνία με την άλλη βιβλιοθήκη, η επικοινωνία με την τεχνική υποστήριξη κλπ. Αυτό είναι ένα παράδειγμα μιας κατάστασης, όπου το μοντέλο υποστήριξης διαδικασίας θα μπορούσε να είναι χρήσιμο.

Αυτά τα πέντε μοντέλα συνεργασίας καθορίζουν μερικά από τα βασικά δομικά στοιχεία της συνεργατικότητας. Πολλά από αυτά τα μοντέλα μπορούν να αθροιστούν, ή να χρησιμοποιηθούν ταυτόχρονα, χωριστά ή να συνδυαστούν σε υβριδικά μοντέλα. Ένα παράδειγμα αποτελεί ο συνδυασμός των μοντέλων Βιβλιοθήκης και Προσέλκυσης Πελατών. Αυτό θα μπορούσε να συμβάλει στη δημιουργία μίας βιβλιοθήκης βέλτιστης πρακτικής, συγκεντρώνοντας ανατροφοδότηση για κάθε πρακτική.

Για να δημιουργήσουμε ένα πλαίσιο για την ανάλυση και το σχεδιασμό της συνεργατικότητας είναι αναγκαίο να κατανοήσουμε κάθε μοντέλο ξεχωριστά και στη συνέχεια να αναλύσουμε τους πιθανούς συνδυασμούς τους. Για να βρούμε την καλύτερη λύση, είναι σημαντικό να μελετήσουμε την κατάσταση και να κάνουμε μια λίστα με τις απαιτήσεις. Τότε θα μπορούσε κανείς να παράγει ένα σχέδιο των διαθέσιμων μοντέλων, αυτούσιων ή ενωμένων. Πολλά συστατικά μπορούν να τεθούν μαζί σε ένα ολοκληρωμένο συνεργατικό σύστημα. Μια διεξοδική κατανόηση των απαιτήσεων είναι αναπόφευκτη για να βρεθεί η βέλτιστη λύση.

2.4.3 Μελλοντική Έρευνα πάνω στον τομέα των ψηφιακών βιβλιοθηκών

Τα πιο πρόσφατα έργα στον τομέα αυτό επικεντρώνεται σε δύο τομείς. Ο πρώτος τομέας είναι η ενσωμάτωση των μέσων μαζικής ενημέρωσης, όπως το βίντεο σε ψηφιακές βιβλιοθήκες. Δύο τέτοια συστήματα παρουσιάζονται στην εργασία των Lanagan και Smeaton [33]. Τα συστήματα αυτά υποστηρίζουν σχολιασμούς του χρήστη και μπορούν να αλληλεπιδράσουν με διαφορετικούς τύπους δεδομένων. Ο δεύτερος τομέας είναι το semantic web και οι τεχνολογίες Web 3.0. Μια σύγκριση μεταξύ του World Wide Web και του semantic web, με τη μορφή ενός μοντέλου δεδομένων που χρησιμοποιείται για την αναγνώριση, την περιγραφή και την

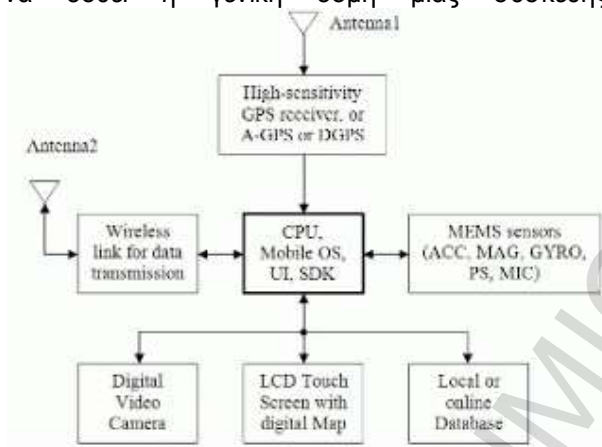
ανακάλυψη των στοιχείων, δείχνει ότι το semantic web είναι πιο κατάλληλο ως ψηφιακή βιβλιοθήκη απ' ότι το διαδίκτυο [3],[34]. Αυτά τα συστήματα λειτουργούν στην προσαρμογή της Ψηφιακής Βιβλιοθήκης με σημασιολογική παρουσίαση των δεδομένων, καθιστώντας την ταξινόμηση και την πρόσβαση αποδοτικότερες. Από το 2009, υπάρχει μια γενικότερη τάση προς την υιοθέτηση σημασιολογικών γλωσσών διαδικτύου, για τη διατήρηση και τη διαχείριση των εξαρτήσεων σε ψηφιακά αρχεία βιβλιοθήκης [35], καθώς και στην υλοποίηση των μεταδεδομένων, όπως ένα τέτοιο σύστημα χρησιμοποιείται στην Βιβλιοθήκη του Πανεπιστημίου Πατρών [36]. Τέτοιες προσπάθειες συνεχίστηκαν στο έργο του Liangxian [37]. Επίσης, πολύ ενθαρρυντικά αποτελέσματα προέρχονται από το σύστημα CallimachusDL των Garcia - Crespo [38]. Σε αυτό το έργο, γίνεται εμφανές ότι τα χαρακτηριστικά του σημασιολογικού ιστού, όπως η φορητότητα των δεδομένων και η συνεχής παρουσία (ubiquity), μπορεί να έχει θετικό αντίκτυπο στις Ψηφιακές Βιβλιοθήκες. Επίσης, έρευνες επάνω στο cloud computing σε σχέση με τις ψηφιακές βιβλιοθήκες έχει συζητηθεί στο έργο του Romero με πολύ ενδιαφέροντα αποτελέσματα [40].

3. Κινητές Εφαρμογές και το Σύστημα Android

Στην σημερινή εποχή, έχει επικρατήσει η ανάγκη για κινητή υπολογιστικότητα. Αυτή η ανάγκη έχει γίνει ακόμα πιο επιτακτική τα τελευταία έτη, με την εμφάνιση έξυπνων συσκευών κινητών τηλεφώνων, τα γνωστά και ως smartphones, τα οποία ουσιαστικά είναι τηλέφωνα με υψηλή υπολογιστική ισχύ. Τα πρώτα smartphones, συνήθως συνδύαζαν τα χαρακτηριστικά ενός συμβατικού κινητού τηλεφώνου με αυτά κάποιας άλλης συσκευής, όπως για παράδειγμα ένας προσωπικός ψηφιακός βοηθός (PDA), ένα media player, μία ψηφιακή φωτογραφική μηχανή, ή μια μονάδα πλοήγησης GPS. Τα σύγχρονα smartphones περιλαμβάνουν όλα αυτά τα χαρακτηριστικά καθώς και τα χαρακτηριστικά ενός υπολογιστή αφής, συμπεριλαμβανομένων web browsing, Wi-Fi, και εφαρμογών από τρίτους. Μία αφαιρετική προσπάθεια για να δοθεί η γενική δομή μίας συσκευής smartphone δίνεται στο Σχήμα 6.



Σχήμα 6 Ένα smartphone



Σχήμα 7 Δομή ενός smartphone (http://www.eetimes.com/document.asp?doc_id=1278857)

Σύμφωνα με το σχήμα 7, στην καρδιά της δομής βρίσκεται μία κεντρική μονάδα επεξεργασίας (CPU), καθώς και το λειτουργικό σύστημα. Από την CPU υπάρχουν συνδέσεις με όλα τα περιφερειακά στοιχεία. Αναλυτικά υπάρχουν: οθόνη LCD με δυνατότητα αφής για την είσοδο των δεδομένων από το χρήστη, κάποιο αποθηκευτικό σύστημα (εξωτερικό ή εσωτερικό), διάφοροι αισθητήρες (επιταχυνσιόμετρο, βαρόμετρο κ.α.), μία ή περισσότερες ψηφιακές φωτογραφικές μηχανές, καθώς και δύο κεραίες, μία για το σήμα κινητής τηλεφωνίας και το ασύρματο internet (wifi), και μία για την λήψη σημάτων GPS.

Πέραν από το υλικό, βασικό ρόλο στην κατασκευή και στην χρησιμότητα μίας τέτοιας συσκευής παίζει το λειτουργικό σύστημα και οι λειτουργίες που προσφέρει στον χρήστη. Αυτή τη στιγμή, κυκλοφορούν αρκετά λειτουργικά συστήματα στην αγορά με τα σημαντικότερα από αυτά να είναι τα Google Android, Apple iOS, Microsoft Windows Phone 7, BlackBerry OS αλλά και άλλα τα οποία σιγά σιγά κερδίζουν μερίδιο της αγοράς όπως τα Firefox OS της Mozilla και το Ubuntu Touch Os της Canonical. Από αυτά το λειτουργικό σύστημα της Google και αυτό της Apple ανταγωνίζονται για το μεγαλύτερο μερίδιο της αγοράς με το πρώτο να έχει κερδίσει έδαφος μέσα στο 2013, όπως φαίνεται και από σχετική έρευνα¹, η οποία φαίνεται στον πίνακα 4.

1 <https://www.idc.com/getdoc.jsp?containerId=prUS24257413>

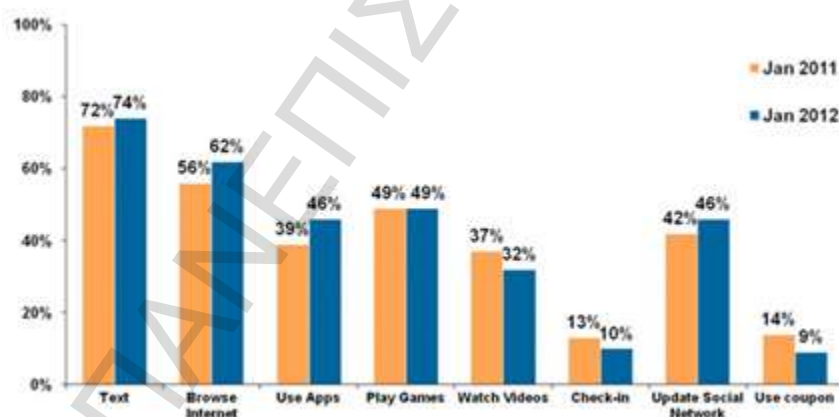
Λειτουργικό Σύστημα	2Q13 Πωλήσεις	2Q13 Μεριδίο Αγοράς	2Q12 Πωλήσεις	2Q12 Μεριδίο Αγοράς	Διαφορά πωλήσεων μεταξύ ετών
Android	187.4	79.3%	108	69.1%	73.5%
iOS	31.2	13.2%	26	16.6%	20.0%
Windows Phone	8.7	3.7%	4.9	3.1%	77.6%
BlackBerry OS	6.8	2.9%	7.7	4.9%	-11.7%
Linux	1.8	0.8%	2.8	1.8%	-35.7%
Symbian	0.5	0.2%	6.5	4.2%	-92.3%
Others	N/A	0.0%	0.3	0.2%	-100.0%
Total	236.4	100.0%	156.2	100.0%	51.3%

Πίνακας 4 Μεριδίο αγοράς λειτουργικών συστημάτων, Q2 2013 (σε εκατομμύρια \$)

Δεδομένης της αύξησης στην χρήση του λειτουργικού συστήματος Android, αλλά και των χαρακτηριστικών του, τα οποία θα αναλυθούν παρακάτω, επιλέξαμε αυτό, ως το σύστημα πάνω στο οποίο θα υλοποιήσουμε την ψηφιακή βιβλιοθήκη στην οποία αναφέρεται η παρούσα διπλωματική εργασία.

Ας δούμε όμως ποιοι λόγοι μας οδηγούν στην ενασχόληση με την δημιουργία εφαρμογών για κινητά τηλέφωνα. Το μεγαλύτερο ποσοστό των χρηστών χρησιμοποιεί μία πληθώρα εφαρμογών οι οποίες είναι φτιαγμένες από τρίτους, για να αυτοματοποιήσει τις εργασίες του, να διασκεδάσει ή να εκτελέσει μία εκ των πολλών λειτουργιών που του δίνονται. Δεδομένου ότι οι χρήστες θέλουν να έχουν την πληροφορία συνεχώς διαθέσιμη, και ότι ένα smartphone μπορεί να τον βοηθήσει στο να την αποκτήσει, οδηγούμαστε στο συμπέρασμα, ότι μία ψηφιακή βιβλιοθήκη στην μορφή μίας κινητής εφαρμογής θα είχε μεγάλο αντίκτυπο στους χρήστες και θα μπορούσε να οδηγήσει στην αύξηση της αποδοτικότητάς τους. Στο Σχήμα 7 μπορούμε να δούμε πως οι χρήστες smartphones κατανέμουν τον χρόνο τους. Το αποτέλεσμα αυτής της έρευνας², ενισχύει το συμπέρασμα μας ότι θα ήταν χρήσιμο να υλοποιήσουμε την ψηφιακή βιβλιοθήκη σαν κινητή εφαρμογή, μιας και το 46% των χρηστών χρησιμοποιεί κατά κόρον τέτοιες εφαρμογές.

Στην επόμενη ενότητα, θα αναλύσουμε με λεπτομέρειες το λειτουργικό σύστημα Android, και τα συστατικά που το απαρτίζουν.



Σχήμα 8 Δραστηριότητες των χρηστών smartphones (www.adage.com)

² <http://adage.com/article/digitalnext/smartphone-users-people-special-anymore/232170/>

3.1 Το λειτουργικό σύστημα Android

Το Android είναι ένα λειτουργικό σύστημα βασισμένο στον πυρήνα του Linux [40], και έχει σχεδιαστεί κυρίως για φορητές συσκευές με οθόνη αφής, όπως smartphones και tablets. Αρχικά αναπτύχθηκε από την Android Inc, η οποία ιδρύθηκε στο Πάλο Άλτο της Καλιφόρνια τον Οκτώβριο του 2003 από τους Andy Rubin, Rich Miner, Nick Sears, και Chris White με σκοπό την ανάπτυξη έξυπνων κινητών συσκευών που είναι ενημερωμένες σχετικά με την τοποθεσία και τις προτιμήσεις του ιδιοκτήτη τους. Η Android Inc αργότερα εξαγοράστηκε από την Google. Το σύστημα Android παρουσιάστηκε το 2007 μαζί με την ίδρυση του Open Handset Alliance [41], μια κοινοπραξία εταιρειών υλικού, λογισμικού και τηλεπικοινωνιών αφοσιωμένων στην προώθηση των ανοικτών προτύπων για τις κινητές συσκευές (OHA, 2012). Το πρώτο smartphone που έγινε διαθέσιμο με λειτουργικό σύστημα Android, ήταν το HTC Dream, το οποίο κυκλοφόρησε στις 22 Οκτωβρίου του 2008.

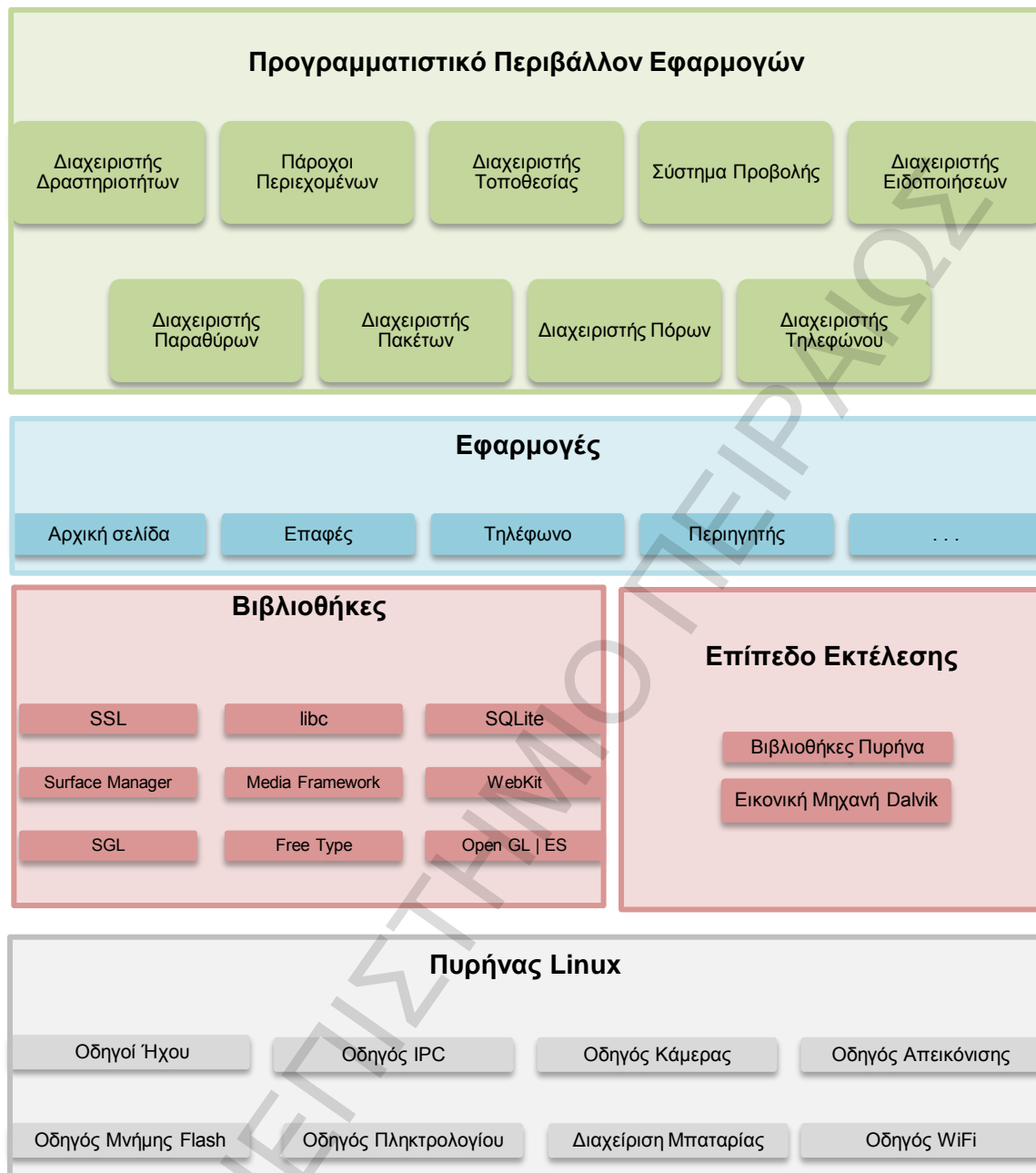
Η διεπαφή χρήστη του Android βασίζεται σε άμεσο χειρισμό, χρησιμοποιώντας τις κινήσεις αφής, όπως το σύρσιμο (swiping), το χτύπημα (tapping) και το τσίμπημα (pinching), για να χειραγωγήσει τα αντικείμενα που βρίσκονται στην οθόνη. Εσωτερικό υλικό, όπως επιταχυνσιόμετρα, γυροσκόπια και αισθητήρες εγγύτητας χρησιμοποιούνται από ορισμένες εφαρμογές για να ανταποκριθούν σε πρόσθετες ενέργειες του χρήστη, όπως για παράδειγμα, για την προσαρμογή της οθόνης από κατακόρυφο σε οριζόντιο προσανατολισμό ανάλογα με το πώς η συσκευή είναι προσανατολισμένη.

Ο πηγαίος κώδικας του Android διατίθεται από την Google υπό την άδεια Apache License, η οποία επιτρέπει στο λογισμικό να τροποποιηθεί ελεύθερα και να διανεμηθεί από τους κατασκευαστές συσκευών, καθώς και από ανεξάρτητους προγραμματιστές. Από τον Ιούλιο του 2013, το Android διαθέτει πάνω από 1 εκατομμύριο εφαρμογές διαθέσιμες για μεταφόρτωση, και μετρά πάνω από 50 δισεκατομμύρια downloads. Μια έρευνα που πραγματοποιήθηκε τον Απρίλιο – Μάιο του 2013 διαπίστωσε ότι το Android είναι η πιο διαδεδομένη πλατφόρμα μεταξύ των προγραμματιστών και χρησιμοποιείται από το 71% του πληθυσμού των προγραμματιστών[42].

Το Android είναι δημοφιλές στις εταιρείες τεχνολογίας που χρειάζονται ένα έτοιμο, χαμηλού κόστους και προσαρμόσιμο λειτουργικό σύστημα για συσκευές υψηλής τεχνολογίας. Παρά το γεγονός ότι σχεδιάστηκε αρχικά για κινητά τηλέφωνα και tablets, έχει επίσης χρησιμοποιηθεί σε τηλεοράσεις, κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές και άλλες ηλεκτρονικές συσκευές.

3.1.1 Η αρχιτεκτονική του Android

Η αρχιτεκτονική του Android αποτελείται από 4 επίπεδα. Καθένα από αυτά τα επίπεδα μπορεί να αποτελείται από άλλα υποστοιχεία, και χρησιμοποιεί τις υπηρεσίες των επιπέδων τα οποία βρίσκονται από κάτω του. Μία συνολική εικόνα της αρχιτεκτονικής του Android δίνεται στο Σχήμα 9 [43].



Σχήμα 9 Η αρχιτεκτονική του Android

Linux Kernel

Η βασική στιβάδα είναι ο πυρήνας του Linux. Το σύνολο του Android OS είναι χτισμένο πάνω στον Linux Kernel 2.6 με εξαίρεση κάποιες περαιτέρω αρχιτεκτονικές αλλαγές που έγιναν από την Google. Είναι το πιο χαμηλό επίπεδο της αρχιτεκτονικής, το οποίο αλληλεπιδρά με το υλικό και περιέχει όλα τα απαραίτητα προγράμματα οδήγησης. Τα προγράμματα οδήγησης είναι προγράμματα που ελέγχουν και επικοινωνούν με το υλικό. Ας πάρουμε για παράδειγμα την λειτουργία Bluetooth. Όλες οι συσκευές διαθέτουν υλικό Bluetooth. Ως εκ τούτου, ο πυρήνας πρέπει να περιλαμβάνει ένα πρόγραμμα οδήγησης για το Bluetooth ούτως ώστε να επικοινωνεί με το υλικό. Ο πυρήνας του Linux λειτουργεί επίσης ως ένα επίπεδο αφαίρεσης μεταξύ του υλικού και των άλλων επιπέδων λογισμικού. Το Android χρησιμοποιεί τον πυρήνα Linux για

όλες τις βασικές λειτουργίες του, όπως τη διαχείριση μνήμης, τη διαχείριση των διαδικασιών, τη δικτύωση, τις ρυθμίσεις ασφαλείας κλπ. Επειδή το Android είναι χτισμένο επάνω σε ένα πιο δημοφιλές και αποδεδειγμένο θεμέλιο, έκανε την προσαρμογή του Android στην ποικιλία των συσκευών ένα σχετικά ανώδυνο έργο.

Βιβλιοθήκες

Το επόμενο στρώμα, το οποίο βρίσκεται επάνω από το επίπεδο του πυρήνα, είναι αυτό που περιλαμβάνει τις ενσωματωμένες βιβλιοθήκες του Android, καθώς και το κομμάτι της αρχιτεκτονικής μέσα στο οποίο γίνεται η εκτέλεση όλων των εντολών. Είναι αυτό το επίπεδο το οποίο επιτρέπει στη συσκευή να χειρίζεται διαφορετικούς τύπους δεδομένων. Αυτές οι βιβλιοθήκες είναι γραμμένες σε γλώσσα C ή C++ και παρέχονται στους προγραμματιστές μέσω ενός Java API. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν το πλαίσιο εφαρμογών της Java για να αποκτήσουν πρόσβαση σε αυτές τις βιβλιοθήκες. Μερικές από τις πιο σημαντικές βιβλιοθήκες είναι οι ακόλουθες:

- *Διαχειριστής Οθόνης (Surfave Manager)*: Χρησιμοποιείται για τη σύνθετη διαχείριση παραθύρων με buffering εκτός οθόνης. Το εκτός οθόνης buffering σημαίνει ότι δεν μπορούμε να σχεδιάσουμε απευθείας στην οθόνη, αλλά τα σχέδια μας, πηγαίνουν στον buffer, ο οποίος βρίσκεται εκτός οθόνης. Εκεί συνδυάζεται με άλλα σχέδια με τα οποία συνθέτει την τελική οθόνη, η οποία θα γίνει εμφανής στον χρήστη.
- *Media Framework*: Το media framework παρέχει διάφορους αποκωδικοποιητές πολυμέσων που επιτρέπουν την καταγραφή και την αναπαραγωγή των διαφόρων μορφών των media που γίνονται διαθέσιμα στην συσκευή.
- *SQLite*: Η SQLite είναι η μηχανή της βάσης δεδομένων που χρησιμοποιείται στο Android για σκοπούς αποθήκευσης δεδομένων. Αποτελεί ουσιαστικά μία πιο ελαφριά έκδοση της SQL.
- *WebKit*: Είναι η μηχανή περιήγησης που χρησιμοποιείται για την εμφάνιση του περιεχόμενου HTML.
- *OpenGL*: Χρησιμοποιείται για να εμφανίσει γραφικά 2D ή 3D στην οθόνη.

Το επίπεδο εκτέλεσης μοιράζεται τον ίδιο χώρο με το επίπεδο των βιβλιοθηκών και περιλαμβάνει την εικονική μηχανή Dalvik (Dalvik Virtual Machine) και τις βιβλιοθήκες πυρήνα, οι οποίες είναι γραμμένες σε Java.

Η Dalvik Virtual Machine είναι ένα είδος της Java Virtual Machine (JVM), η οποία χρησιμοποιείται σε συσκευές Android για να εκτελέσει εφαρμογές και είναι βελτιστοποιημένη ώστε να λειτουργεί με χαμηλή επεξεργαστική ισχύ και σε περιβάλλοντα χαμηλής μνήμης. Σε αντίθεση με την JVM, η Dalvik Virtual Machine δεν εκτελεί .class αρχεία, αλλά .dex αρχεία. Τα .dex αρχεία κατασκευάζονται από .class αρχεία κατά το χρόνο της μεταγλώττισης (compilation) και παρέχουν υψηλότερη απόδοση σε συνθήκες χαμηλών πόρων. Η Dalvik VM επιτρέπει την ταυτόχρονη δημιουργία πολλαπλών στιγμιότυπων ψηφιακών μηχανών παρέχοντας ασφάλεια, απομόνωση, διαχείριση μνήμης και υποστήριξη πολλαπλών νημάτων.

Οι βιβλιοθήκες πυρήνα Java είναι διαφορετικές από τις βιβλιοθήκες Java SE και Java ME. Ωστόσο, αυτές οι βιβλιοθήκες παρέχουν τις περισσότερες από τις λειτουργίες που καθορίζουν τις βιβλιοθήκες της Java SE.

Πλαίσιο Εφαρμογών

Αυτά είναι τα στοιχεία με τα οποία αλληλεπιδρούν οι εφαρμογές μας. Το επίπεδο αυτό περιέχει όλα τα εργαλεία τα οποία θα χρησιμοποιήσει ένας προγραμματιστής για να αναπτύξει εφαρμογές σε Android. Τα προγράμματα αυτά διαχειρίζονται τις βασικές λειτουργίες του τηλεφώνου, όπως τη διαχείριση των πόρων, τη διαχείριση φωνητικής κλήσης, κλπ. Σημαντικά τμήματα του πλαισίου εφαρμογών είναι :

- *Διαχειριστής Δραστηριοτήτων (Activity Manager)*: Διαχειρίζεται τον κύκλο ζωής των εφαρμογών.

- *Πάροχοι Περιεχομένου (Content Providers)*: Διαχειρίζονται την ανταλλαγή δεδομένων μεταξύ των εφαρμογών.
- *Διαχειριστής Τηλεφωνίας (Telephony Manager)*: Διαχειρίζεται όλες τις φωνητικές κλήσεις. Τον χρησιμοποιούμε αν θέλουμε η εφαρμογή μας να έχει πρόσβαση στις φωνητικές κλήσεις.
- *Διαχειριστής Τοποθεσίας (Location Manager)*: Επιτρέπει στις εφαρμογές να γνωρίζουν την ακριβή τοποθεσία στην οποία βρίσκεται η συσκευή, συνήθως με χρήση GPS.
- *Διαχειριστής Πόρων (Resource Manager)*: Διαχειρίζεται τα διάφορα είδη των πόρων που χρησιμοποιούμε στην εφαρμογή μας.
- *Διαχειριστής Ειδοποιήσεων (Notification Manager)*: Επιτρέπει στην εφαρμογή να ειδοποιήσει τον χρήστη για κάποιο σημαντικό γεγονός, χωρίς να διακόπτει την τρέχουσα εφαρμογή. Χρησιμοποιεί την μπάρα ειδοποιήσεων (notification bar) για να το επιτύχει.

Επίπεδο Εφαρμογών

Οι εφαρμογές είναι το ανώτερο επίπεδο της αρχιτεκτονικής του Android και αυτό στον οποίο εμφανίζονται οι εφαρμογές. Πολλές εφαρμογές έρχονται προεγκατεστημένες με κάθε συσκευή. Κάποιες από αυτές είναι:

- Εφαρμογή SMS
- Εφαρμογή κλήσεων (dialer)
- Πρόγραμμα περιήγησης στο Διαδίκτυο (Browser)
- Διαχειριστής επαφών (Contact Manager)

Ένας προγραμματιστής είναι σε θέση να γράψει μια εφαρμογή που αντικαθιστά κάποια από τις υφιστάμενες εφαρμογές. Δεν περιορίζεται δηλαδή στην πρόσβαση που δίνει η συσκευή σε οποιοδήποτε ιδιαίτερο χαρακτηριστικό.

3.1.2. Εφαρμογές Android

Οι εφαρμογές σε Android είναι γραμμένες στην γλώσσα προγραμματισμού Java. Τα εργαλεία που περιλαμβάνονται στο Android SDK μεταγλωττίζουν τον κώδικα, μαζί με όλα τα δεδομένα και τους πόρους (resource files), δημιουργώντας ένα APK: ένα πακέτο Android, το οποίο ουσιαστικά είναι ένα συμπιεσμένο αρχείο με κατάληξη .apk. Ένα αρχείο APK περιέχει όλα τα περιεχόμενα μίας εφαρμογής Android και αποτελεί το αρχείο το οποίο οι συσκευές που τρέχουν Android χρησιμοποιούν για να εγκαταστήσουν την εφαρμογή.

Μόλις εγκατασταθεί σε μια συσκευή, κάθε εφαρμογή Android “ζει” στο δικό της ασφαλές περιβάλλον (sandbox):

- Το λειτουργικό σύστημα Android είναι ένα λειτουργικό σύστημα Linux με πολλούς χρήστες, στο οποίο κάθε εφαρμογή αποτελεί και έναν διαφορετικό χρήστη.
- Από προεπιλογή, το σύστημα εκχωρεί σε κάθε εφαρμογή ένα μοναδικό αναγνωριστικό χρήστη (το αναγνωριστικό χρησιμοποιείται μόνο από το σύστημα και είναι άγνωστο προς στην εφαρμογή). Το σύστημα ορίζει δικαιώματα για όλα τα αρχεία της εφαρμογής έτσι ώστε μόνο ο κωδικός ταυτότητας χρήστη που έχει εκχωρηθεί στην συγκεκριμένη εφαρμογή να μπορεί να έχει πρόσβαση σε αυτά.
- Κάθε διαδικασία έχει τη δική της εικονική μηχανή (VM), έτσι ώστε ο κώδικας της εφαρμογής να τρέχει σε απομόνωση από τις άλλες εφαρμογές.
- Από προεπιλογή, κάθε εφαρμογή εκτελείται στη δική της διαδικασία Linux. Το Android ξεκινά τη διαδικασία όταν οποιοδήποτε από τα συστατικά της εφαρμογής θα πρέπει να εκτελεστούν, έπειτα τερματίζει την διαδικασία όταν δεν χρειάζονται πλέον ή όταν το σύστημα πρέπει να ανακτήσει τη μνήμη για άλλες εφαρμογές.

Με τον τρόπο αυτό, το σύστημα Android εφαρμόζει την αρχή του ελάχιστου προνομίου (Least Privilege Principle) [47]. Δηλαδή, κάθε εφαρμογή, από προεπιλογή, έχει πρόσβαση μόνο

στα στοιχεία τα οποία χρειάζεται για να πραγματοποιήσει την λειτουργία της και σε τίποτα παραπάνω. Αυτό δημιουργεί ένα πολύ ασφαλές περιβάλλον στο οποίο μια εφαρμογή δεν μπορεί να έχει πρόσβαση σε μέρη του συστήματος, για τα οποία δεν της έχει δοθεί η άδεια.

Ωστόσο, υπάρχουν τρόποι για μια εφαρμογή να πραγματοποιήσει ανταλλαγή δεδομένων με άλλες εφαρμογές και να έχει πρόσβαση στις υπηρεσίες του συστήματος. Αυτοί είναι οι εξής:

- Είναι δυνατόν δύο εφαρμογές να μοιράζονται το ίδιο αναγνωριστικό χρήστη Linux, οπότε είναι σε θέση να έχουν πρόσβαση η μία στα αρχεία της άλλης. Για να εξοικονομήσουμε πόρους συστήματος, οι εφαρμογές με το ίδιο αναγνωριστικό χρήστη μπορούν να εκτελούνται στην ίδια διαδικασία Linux και να μοιράζονται το ίδιο VM (Επίσης, οι εφαρμογές πρέπει να υπογράφονται με το ίδιο πιστοποιητικό).
- Μια εφαρμογή μπορεί να ζητήσει άδεια για να αποκτήσει πρόσβαση σε δεδομένα της συσκευής, όπως τις επαφές του χρήστη, τα μηνύματα SMS, τις εξωτερικές συσκευές αποθήκευσης (κάρτα SD), την φωτογραφική μηχανή, το Bluetooth, και πολλά άλλα. Όλα τα δικαιώματα πρέπει να χορηγούνται από το χρήστη κατά τη διάρκεια της εγκατάστασης.

Συστατικά των εφαρμογών



Σχήμα 10 Τα συστατικά του Android

Τα συστατικά των εφαρμογών είναι τα βασικά δομικά στοιχεία κάθε εφαρμογής Android. Κάθε συστατικό είναι ένα διαφορετικό σημείο μέσω του οποίου το σύστημα μπορεί να εισέλθει στην εφαρμογή. Δεν είναι όλα τα συστατικά πραγματικά σημεία εισόδου για το χρήστη και ορισμένα εξαρτώνται από κάποιο άλλο, αλλά καθένα υπάρχει ως αυτόνομη οντότητα και παίζει ένα συγκεκριμένο ρόλο. Το κάθε συστατικό είναι ένα μοναδικό δομικό στοιχείο που βοηθά στον καθορισμό της συνολικής συμπεριφοράς της εφαρμογής.

Υπάρχουν τέσσερις διαφορετικοί τύποι συστατικών. Κάθε τύπος εξυπηρετεί ένα συγκεκριμένο σκοπό και έχει ένα ξεχωριστό κύκλο ζωής που καθορίζει το πώς δημιουργείται και καταστρέφεται.

- **Δραστηριότητες (Activities):** Μια δραστηριότητα αντιπροσωπεύει μια μοναδική οθόνη με ένα περιβάλλον εργασίας χρήστη. Για παράδειγμα, μία εφαρμογή ηλεκτρονικού ταχυδρομείου μπορεί να έχει μια δραστηριότητα που εμφανίζει μια λίστα με τα νέα μηνύματα ηλεκτρονικού ταχυδρομείου, μια άλλη δραστηριότητα για την σύνθεση ενός μηνύματος ηλεκτρονικού ταχυδρομείου, καθώς και άλλη μία δραστηριότητα για την ανάγνωση των μηνυμάτων ηλεκτρονικού ταχυδρομείου. Αν και οι δραστηριότητες λειτουργούν μαζί για να σχηματίσουν την εφαρμογή ηλεκτρονικού ταχυδρομείου, η καθεμία είναι ανεξάρτητη από τις άλλες. Έτσι λοιπόν, μία διαφορετική εφαρμογή μπορεί να ξεκινήσει οποιαδήποτε από αυτές τις δραστηριότητες (εάν η εφαρμογή ηλεκτρονικού ταχυδρομείου το επιτρέπει).
- **Υπηρεσίες (Services):** Μια υπηρεσία είναι ένα στοιχείο που λειτουργεί στο παρασκήνιο για να εκτελέσει μακρόχρονες λειτουργίες ή να εκτελέσει εργασίες για απομακρυσμένες διαδικασίες. Μια υπηρεσία δεν προϋποθέτει την ύπαρξη διεπαφής χρήστη. Για παράδειγμα, μια υπηρεσία μπορεί να παίζει μουσική στο παρασκήνιο, ενώ ο χρήστης βρίσκεται σε μία διαφορετική εφαρμογή, ή θα μπορούσε να μεταφέρει δεδομένα μέσω δικτύου χωρίς να εμποδίζει την αλληλεπίδραση του χρήστη με μια δραστηριότητα. Ένα άλλο συστατικό, όπως για παράδειγμα μια δραστηριότητα, μπορεί να ξεκινήσει την υπηρεσία και να την αφήσει να τρέξει ή να συνδεθεί με αυτή, ούτως ώστε να αλληλεπιδράσει μαζί της.

- **Πάροχοι περιεχομένου (Content Providers):** Ένας πάροχος περιεχομένου διαχειρίζεται ένα κοινόχρηστο κοινό σύνολο δεδομένων της εφαρμογής. Μπορούμε να αποθηκεύσουμε τα δεδομένα στο σύστημα αρχείων, σε μια βάση δεδομένων SQLite, στο διαδίκτυο, ή σε οποιαδήποτε άλλη θέση αποθήκευσης, στην οποία μπορεί να έχει πρόσβαση η εφαρμογή. Μέσα από τον πάροχο του περιεχομένου, άλλες εφαρμογές μπορούν να εκτελέσουν επερωτήσεις ή ακόμη και να τροποποιήσουν τα δεδομένα (αν ο πάροχος περιεχομένου το επιτρέπει). Οι πάροχοι περιεχομένου είναι επίσης χρήσιμοι για την ανάγνωση και εγγραφή δεδομένων τα οποία δεν μοιράζονται με την εφαρμογή μας.
- **Δέκτες μηνυμάτων Broadcast (Broadcast Receivers):** Ένας δέκτης μηνυμάτων broadcast είναι ένα συστατικό, το οποίο απαντά σε μηνύματα διαθέσιμα σε ολόκληρο το σύστημα. Πολλά μηνύματα broadcast προέρχονται από το σύστημα, για παράδειγμα, ένα broadcast που ανακοινώνει ότι η οθόνη είναι πλέον απενεργοποιημένη, ή ότι η μπαταρία είναι χαμηλή, ή ότι μια έγινε λήψη μίας φωτογραφίας. Επίσης, οι εφαρμογές μπορούν να αποστείλουν broadcasts. Για παράδειγμα, για να ειδοποιήσουν τις άλλες εφαρμογές πως ολοκληρώθηκε το κατέβασμα δεδομένων από το διαδίκτυο και τα δεδομένα είναι διαθέσιμα για χρήση. Αν και οι δέκτες broadcast δεν διαθέτουν διεπαφή χρήστη, μπορούν να δημιουργήσουν μια ειδοποίηση στη γραμμή κατάστασης για να ειδοποιήσουν το χρήστη όταν συμβαίνει ένα γεγονός broadcast. Συχνότερα όμως, χρησιμοποιούνται ως μια "πύλη" προς άλλα συστατικά.

Μια μοναδική πτυχή του συστήματος Android είναι ότι κάθε εφαρμογή μπορεί να ξεκινήσει τα συστατικά μίας άλλης. Για παράδειγμα, αν ο χρήστης θέλει να κάνει λήψη μιας φωτογραφίας με την κάμερα της συσκευής, υπάρχει πιθανώς μία εφαρμογή που το κάνει και η εφαρμογή του μπορεί να την χρησιμοποιήσει, αντί να αναπτύξει την δραστηριότητα μόνος του.

Όταν το σύστημα προκαλεί την εκκίνηση ενός συστατικού, ξεκινάει η διαδικασία για την εν λόγω εφαρμογή (αν δεν είναι ήδη σε λειτουργία) και αρχικοποιεί τις κλάσεις που απαιτούνται για το συστατικό. Για παράδειγμα, εάν η εφαρμογή ξεκινά τη δραστηριότητα της εφαρμογής της φωτογραφικής μηχανής, η δραστηριότητα αυτή εκτελείται στο πλαίσιο της διαδικασίας που ανήκει στην εφαρμογή φωτογραφικής μηχανής, και όχι στη διαδικασία της εφαρμογής του χρήστη. Ως εκ τούτου, σε αντίθεση με τα περισσότερα άλλα συστήματα για εφαρμογές, οι εφαρμογές Android δεν έχουν ένα ενιαίο σημείο εισόδου.

Επειδή το σύστημα εκτελεί κάθε εφαρμογή σε μια ξεχωριστή διαδικασία με άδειες που περιορίζουν την πρόσβαση σε άλλες εφαρμογές, η εφαρμογή δεν μπορεί να ενεργοποιήσει άμεσα ένα συστατικό κάποιας άλλης εφαρμογής. Το σύστημα Android, ωστόσο μπορεί. Έτσι, για να ενεργοποιήσουμε ένα συστατικό σε μία άλλη εφαρμογή, θα πρέπει να παραδώσουμε ένα μήνυμα στο σύστημα, το οποίο καθορίζει την πρόθεσή μας να ξεκινήσουμε ένα συγκεκριμένο συστατικό. Το σύστημα ενεργοποιεί στη συνέχεια το στοιχείο για εμάς.

Ενεργοποίηση Συστατικών

Τρεις από τους τέσσερις τύπους συστατικών, οι δραστηριότητες, οι υπηρεσίες, και δέκτες broadcast, ενεργοποιούνται από ένα ασύγχρονο μήνυμα που ονομάζεται Intent. Τα Intents δεσμεύουν τα επιμέρους συστατικά μεταξύ τους κατά το χρόνο εκτέλεσης, ανεξάρτητα αν το κάθε συστατικό ανήκει στην δική μας εφαρμογή ή σε άλλη.

Ένα Intent δημιουργείται χρησιμοποιώντας ένα αντικείμενο τύπου Intent, το οποίο καθορίζει ένα μήνυμα, ώστε να ενεργοποιήσει είτε ένα συστατικό είτε ένα συγκεκριμένο τύπο ενός συστατικού.

Για τις δραστηριότητες και τις υπηρεσίες, ένα Intent καθορίζει την δράση η οποία θα εκτελεστεί (για παράδειγμα, την εμφάνιση ή την αποστολή κάποιου αντικείμενου) και μπορεί να καθορίσει την διεύθυνση των δεδομένων (URI) τα οποία θα χρησιμοποιήσει η εφαρμογή. Για παράδειγμα, ένα Intent μπορεί να μεταφέρει ένα αίτημα σε μια δραστηριότητα, ούτως ώστε να εμφανίσει μια εικόνα ή για να ανοίξει μια ιστοσελίδα. Σε ορισμένες περιπτώσεις, μπορούμε να ξεκινήσουμε μια δραστηριότητα περιμένοντας να λάβουμε κάποιο αποτέλεσμα. Στην περίπτωση αυτή, η δραστηριότητα επιστρέφει επίσης το αποτέλεσμα σε ένα Intent.

Για τους δέκτες broadcast, η πρόθεση απλώς ορίζει την ανακοίνωση η οποία θα μεταδοθεί.

Ο τέταρτος τύπος συστατικών, οι πάροχοι περιεχομένου, δεν ενεργοποιείται από Intents. Αντίθετα, ενεργοποιούνται όταν δέχονται ένα αίτημα από έναν αναλυτή περιεχομένου (ContentResolver). Ο αναλυτής περιεχομένου χειρίζεται όλες τις άμεσες συναλλαγές με τον πάροχο περιεχομένου, έτσι ώστε το συστατικό το οποίο εκτελεί συναλλαγές με τον πάροχο να καλεί μεθόδους στο αντικείμενο του αναλυτή περιεχομένου. Αυτό δημιουργεί ένα στρώμα αφαίρεσης μεταξύ του παρόχου περιεχομένου και του συστατικού που ζητά πληροφορίες, αυξάνοντας της ασφάλεια.

Το αρχείο Manifest

Πριν το σύστημα Android μπορέσει να ξεκινήσει κάποιο συστατικό, πρέπει να γνωρίζει ότι υπάρχει το στοιχείο ψάχνοντας αρχείο AndroidManifest.xml της εφαρμογής. Η εφαρμογή πρέπει να δηλώνει όλα τα στοιχεία της σε αυτό το αρχείο. Το αρχείο manifest εκτελεί μια σειρά από πράγματα, εκτός από το να δηλώνει όλα τα συστατικά της εφαρμογής. Κάποια από αυτά είναι τα εξής:

- Αναγνωρίζει τυχόν δικαιώματα χρήστη που η εφαρμογή απαιτεί, όπως η πρόσβαση στο Internet ή πρόσβαση στις επαφές του χρήστη.
- Δηλώνει το ελάχιστο επίπεδο API που απαιτείται από την εφαρμογή.
- Δηλώνει λειτουργίες υλικού και λογισμικού που χρησιμοποιούνται ή που απαιτούνται από την εφαρμογή, όπως η φωτογραφική μηχανή, οι υπηρεσίες bluetooth, ή μια οθόνη multitouch.
- Βιβλιοθήκες API, τις οποίες χρειάζεται η εφαρμογή, όπως πχ. η βιβλιοθήκη του Google Maps κ.α.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.Libraryapp"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />

    <permission
        android:name="com.example.Libraryapp.permission.C2D_MESSAGE"
        android:protectionLevel="signature" />

    <uses-permission android:name="com.example.Libraryapp.permission.C2D_MESSAGE" />
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
    <uses-permission android:name="android.permission.VIBRATE" />

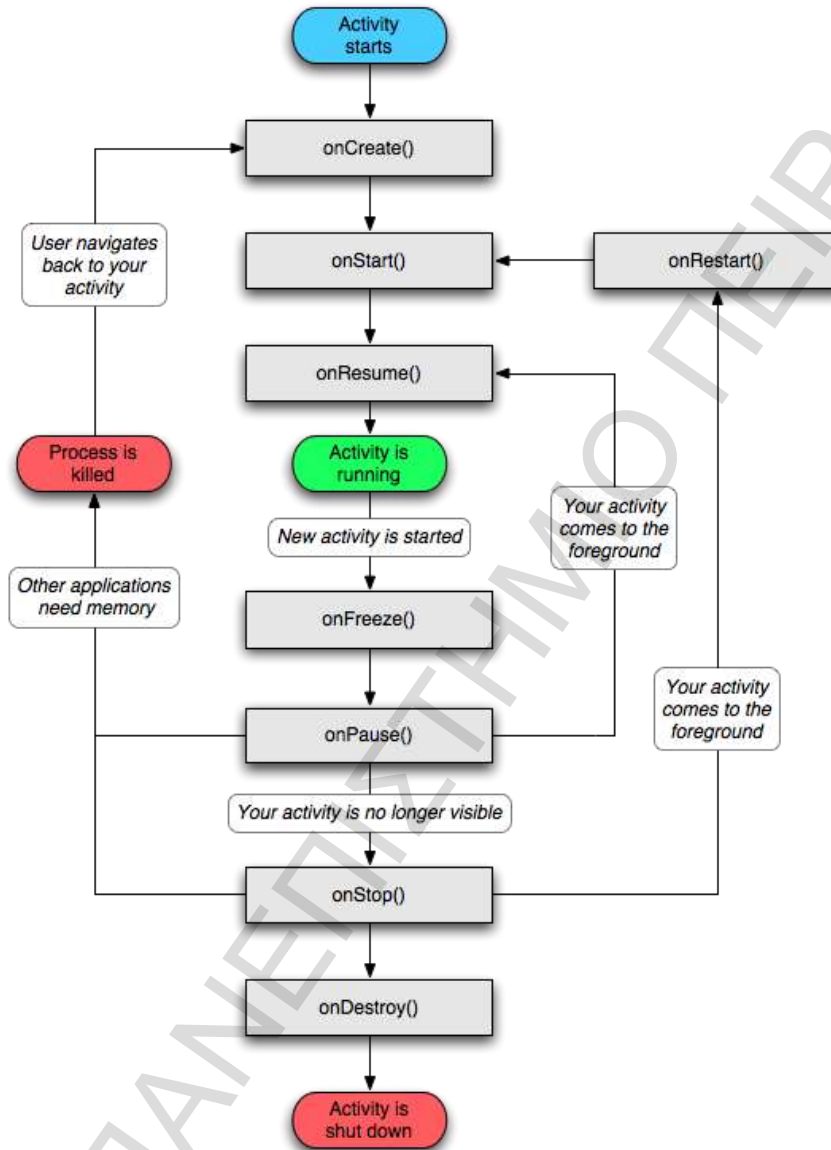
    <application
        android:allowBackup="true"
        android:icon="@drawable/La_Launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.Libraryapp.LibApp"
            android:background="@drawable/background"
            android:label="@string/title_activity_Libr_app"
            />
        />
    />
</manifest>
```

Σχήμα 11 Παράδειγμα Android Manifest από την εφαρμογή LibrApp

Πόροι εφαρμογών

Μια εφαρμογή Android δεν αποτελείται μόνο από κώδικα, αλλά απαιτεί πόρους οι οποίοι είναι ξεχωριστοί από τον πηγαίο κώδικα, όπως εικόνες, αρχεία ήχου, καθώς και οτιδήποτε αφορά την οπτική παρουσίαση της εφαρμογής. Η χρήση των πόρων της εφαρμογής καθιστά εύκολη την ενημέρωση των διαφόρων χαρακτηριστικών της εφαρμογής χωρίς να χρειάζεται αλλαγή στον κώδικα και δίνει τη δυνατότητα βελτιστοποίησης της εφαρμογής.

Κύκλος ζωής των δραστηριοτήτων (Lifecycle of activities)



Σχήμα 12 Ο κύκλος ζωής των activities (<http://developer.android.com>)

Καθώς ο χρήστης πλοηγείται εντός της εφαρμογής, εκτός της εφαρμογής, και επιστρέφει στην εφαρμογή, οι δραστηριότητες της εφαρμογής μεταβαίνουν μεταξύ διαφόρων σταδίων του κύκλου ζωής τους. Για παράδειγμα, όταν μία δραστηριότητα ξεκινά για πρώτη φορά, έρχεται στο προσκήνιο του συστήματος. Κατά τη διάρκεια αυτής της διαδικασίας, το σύστημα Android καλεί μια σειρά από μεθόδους του κύκλου ζωής της δραστηριότητας. Εάν ο χρήστης εκτελεί μια ενέργεια που ξεκινά μια άλλη δραστηριότητα ή τον μεταφέρει σε άλλη

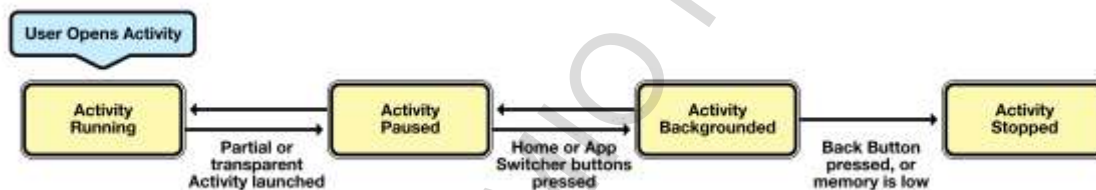
εφαρμογή, το σύστημα καλεί ένα άλλο σύνολο των μεθόδων του κύκλου ζωής της δραστηριότητας, καθώς αυτή κινείται στο παρασκήνιο.

Εντός των μεθόδων του κύκλου ζωής επανάκλησης, μπορούμε να δηλώσουμε πώς θα συμπεριφέρεται η δραστηριότητα όταν ο χρήστης εγκαταλείπει και στην συνέχεια εισέρχεται εκ νέου στη δραστηριότητα. Για παράδειγμα, εάν η εφαρμογή αφορά έναν αναπαραγωγέα βίντεο, θα πρέπει να μπορεί η εφαρμογή να διακόψει την εκτέλεση του βίντεο και να τερματίσει τη σύνδεση με το δίκτυο όταν ο χρήστης μεταβαίνει σε άλλη εφαρμογή. Όταν ο χρήστης επιστρέψει, θα πρέπει να συνδέεται ξανά στο δίκτυο και να επιτρέψει στο χρήστη να συνεχίσει το βίντεο από το ίδιο σημείο.

Ο κύκλος ζωής των δραστηριοτήτων του Android περιλαμβάνει μια συλλογή από μεθόδους που διατίθενται στην κλάση Activity που παρέχουν στον προγραμματιστή ένα πλαίσιο διαχείρισης των πόρων. Το πλαίσιο αυτό επιτρέπει στους προγραμματιστές να ανταποκρίνονται στις ιδιαίτερες απαιτήσεις της διαχείρισης των καταστάσεων, στις οποίες μπορεί να βρεθεί κάθε δραστηριότητα μέσα σε μια εφαρμογή και να χειρίζεται σωστά τη διαχείριση των πόρων.

Καταστάσεις Δραστηριότητας (Activity States)

Το λειτουργικό σύστημα Android διαχειρίζεται τις δραστηριότητες με βάση την κατάστασή τους. Αυτό βοηθά το Android να προσδιορίσει τις δραστηριότητες που δεν είναι πλέον σε χρήση, επιτρέποντας στο λειτουργικό σύστημα να ανακτήσει εκ νέου τη μνήμη και τους πόρους. Το Σχήμα 13 απεικονίζει τις καταστάσεις από τις οποίες μια δραστηριότητα μπορεί να περάσει κατά τη διάρκεια της ζωής της.



Σχήμα 13 Καταστάσεις των δραστηριοτήτων (docs.xamarin.com)

Αυτές οι καταστάσεις μπορούν να χωριστούν σε 4 κύριες ομάδες:

1. *Ενεργή ή Σε εκτέλεση* - Οι δραστηριότητες θεωρούνται ενεργές ή σε εκτέλεση αν είναι στο προσκήνιο, ή αλλιώς στην κορυφή της στοιβάς δραστηριοτήτων. Αυτή θεωρείται η δραστηριότητα με την υψηλότερη προτεραιότητα, και ως τέτοια θα πρέπει να τερματίζεται από το λειτουργικό σύστημα μόνο σε ακραίες καταστάσεις, όπως για παράδειγμα αν η δραστηριότητα προσπαθεί να χρησιμοποιήσει περισσότερη μνήμη από ό, τι είναι διαθέσιμη στη συσκευή, καθώς αυτό θα μπορούσε να προκαλέσει την διεπαφή χρήστη να «κολλήσει».
2. *Σε παύση* - Όταν η συσκευή πηγαίνει σε ανενεργή κατάσταση (sleep mode), ή μια δραστηριότητα εξακολουθεί να είναι ορατή, αλλά είναι εν μέρει κρυμμένη από μια νέα δραστηριότητα, η δραστηριότητα θεωρείται σε παύση. Οι δραστηριότητες σε παύση είναι ακόμα ζωντανές, δηλαδή, διατηρούν όλες τις πληροφορίες και εξακολουθούν να συνδέονται με το διαχειριστή παραθύρων. Αυτή θεωρείται ότι είναι η δεύτερη σε προτεραιότητα κατάσταση δραστηριότητας και ως εκ τούτου, θα πρέπει να τερματίζεται μόνο από το λειτουργικό σύστημα, εάν τερματίζοντας αυτή τη δραστηριότητα θα ικανοποιηθούν οι απαιτήσεις σε πόρων που χρειάζονται για να κρατηθεί σταθερή η ενεργή δραστηριότητα.
3. *Σταματημένη ή στο παρασκήνιο* - Οι δραστηριότητες που είναι εντελώς επισκιασμένες από μια άλλη δραστηριότητα θεωρούνται σταματημένες ή στο παρασκήνιο. Οι σταματημένες δραστηριότητες εξακολουθούν να προσπαθούν να διατηρήσουν τις πληροφορίες τους για όσο το δυνατόν περισσότερο, αλλά θεωρούνται ότι διαθέτουν τη χαμηλότερη προτεραιότητα από τις τρεις καταστάσεις, και ως εκ τούτου, το λειτουργικό

σύστημα θα τερματίσει πρώτα αυτές ώστε να ικανοποιήσει τις απαιτήσεις των πόρων από δραστηριότητες υψηλότερης προτεραιότητας.

4. Σε επανεκκίνηση - Είναι δυνατόν για μια δραστηριότητα που βρίσκεται σε κάποια από τις παραπάνω καταστάσεις να αφαιρεθεί από τη μνήμη του συστήματος. Εάν ο χρήστης περιηγηθεί πίσω στη δραστηριότητα, αυτή πρέπει να επανεκκινηθεί, και να γίνει επαναφορά στην προηγούμενως αποθηκευμένη κατάσταση της.

Επιαναδημιουργία των Δραστηριοτήτων σε απάντηση στις αλλαγές διαμόρφωσης

Για να κάνει τα πράγματα πιο περίπλοκα, το σύστημα Android προσθέτει ένα ακόμα στοιχείο που ονομάζεται αλλαγές διαμόρφωσης. Οι αλλαγές διαμόρφωσης αφορούν την ταχεία καταστροφή/αναδημιουργία της δραστηριότητας που συμβαίνει όταν η διαμόρφωση μίας δραστηριότητας αλλάζει, όπως όταν περιστρέφεται η συσκευή, όταν εμφανίζεται το πληκτρολόγιο ή όταν η συσκευή τοποθετείται σε φόρτιση.

Οι αλλαγές διαμόρφωσης εξακολουθούν να προκαλούν τις ίδιες αλλαγές κατάστασης δραστηριότητας, οι οποίες θα προέκυπταν κατά τη διάρκεια της διακοπής και της επανεκκίνησης της δραστηριότητας.

Μέθοδοι του κύκλου ζωής της Δραστηριότητας

Το Android SDK παρέχει ένα ισχυρό πλαίσιο για τη διαχείριση της κατάστασης των δραστηριοτήτων μέσα σε μια εφαρμογή. Όταν η κατάσταση μιας δραστηριότητας αλλάζει, η δραστηριότητα ειδοποιείται από το λειτουργικό σύστημα, το οποίο καλεί τις ειδικές μεθόδους για την εν λόγω δραστηριότητα. Αυτοί οι μέθοδοι φαίνονται στο Σχήμα 12.

Ο προγραμματιστής, μπορεί να χειριστεί τις αλλαγές της κατάστασης υπερφορτώνοντας (overloading) αυτές τις μεθόδους στον κώδικα μιας δραστηριότητας. Είναι σημαντικό να σημειωθεί, ωστόσο, ότι όλες οι μέθοδοι κύκλου ζωής καλούνται στο νήμα της διεπαφή χρήστης και θα εμποδίσουν το λειτουργικό σύστημα από την εκτέλεση του επόμενου κομματιού της διεπαφής. Ως εκ τούτου, ο κώδικας σε αυτές τις μεθόδους θα πρέπει να είναι όσο το δυνατόν συντομότερος.

Ας εξετάσουμε κάθε μία από αυτές τις μεθόδους του κύκλου ζωής και τη χρήση τους:

onCreate

Αυτή είναι η πρώτη μέθοδος που καλείται όταν δημιουργείται μια δραστηριότητα. Η onCreate πρέπει πάντα να υπερφορτώνεται από τον προγραμματιστή για να εκτελέσει τις αρχικοποιήσεις που μπορεί να απαιτούνται από μια δραστηριότητα, όπως :

- Δημιουργία όψεων (Views)
- Αρχικοποίηση μεταβλητών
- Δέσμευση στατικών δεδομένων σε λίστες

OnCreate παίρνει ως παράμετρο ένα αντικείμενο τύπου Bundle, το οποίο αποτελεί ένα λεξικό για την αποθήκευση και το πέρασμα πληροφοριών κατάστασης μεταξύ των δραστηριοτήτων. Εάν το Bundle δεν είναι μηδενικό, αυτό δείχνει ότι η δραστηριότητα επανεκκινείται, και θα πρέπει να αποκατασταθεί η κατάστασή της από το προηγούμενο στιγμιότυπο. Μόλις onCreate ολοκληρωθεί, εκτελείται η onStart.

onStart

Αυτή η μέθοδος καλείται πάντα από το σύστημα μετά από την onCreate. Οι δραστηριότητες μπορούν να υπερφορτώσουν αυτήν τη μέθοδο, εάν θέλουν να εκτελέσουν συγκεκριμένες εργασίες λίγο πριν γίνει ορατή η δραστηριότητα. Το σύστημα Android θα καλέσει την onResume αμέσως μετά από αυτή τη μέθοδο.

onResume

Το σύστημα καλεί αυτή τη μέθοδο, όταν η δραστηριότητα είναι έτοιμη να ξεκινήσει την αλληλεπίδρασή της με το χρήστη. Οι δραστηριότητες θα πρέπει να υπερφορτώσουν αυτήν τη μέθοδο για να εκτελούν καθήκοντα όπως :

- Αναδιατάξη των ταχυτήτων των καρτέ (frame rate) (μια κοινή τακτική στα παιχνίδια)
- Εκκίνηση animations
- Αναμονή για ενημερώσεις GPS
- Εμφάνιση όλων των σχετικών ειδοποιήσεων ή διαλόγων
- Σύνδεση με εξωτερικούς χειριστές συμβάντων

Η `onResume` είναι σημαντική, διότι κάθε πράξη που εκτελείται στην `OnPause` θα πρέπει να αναιρείται στην `onResume`, δεδομένου ότι είναι η μόνη μέθοδος του κύκλου ζωής που είναι εγγυημένο πως θα εκτελεστεί μετά την `OnPause` όταν η δραστηριότητα επανεκκινείται.

onPause

Αυτή η μέθοδος καλείται όταν το σύστημα είναι έτοιμο να θέσει κάποια δραστηριότητα στο παρασκήνιο ή όταν η δραστηριότητα γίνεται εν μέρει αφανής. Οι δραστηριότητες θα πρέπει να υπερφορτώσουν αυτήν τη μέθοδο, εάν πρέπει να:

- Δεσμεύσει μη αποθηκευμένες αλλαγές στο αποθηκευτικό σύστημα.
- Καταστρέψει ή καθαρίσει άλλα αντικείμενα που καταναλώνουν πόρους.
- Μειώσει το frame rate ή σταματήσει animations.
- Καταργήσει εξωτερικούς χειριστές συμβάντων ή χειριστές κοινοποίησης.
- Αποκρύψει όλες τις σχετικές ειδοποιήσεις ή διαλόγους.

Υπάρχουν δύο πιθανές μέθοδοι κύκλου ζωής που θα κληθούν μετά από την `onPause`:

- Η `onResume` θα κληθεί, εάν η δραστηριότητα πρόκειται να επιστρέψει στο προσκήνιο.
- Η `onStop` θα κληθεί, εάν η δραστηριότητα τίθεται στο παρασκήνιο.

onStop

Αυτή η μέθοδος καλείται όταν η δραστηριότητα δεν είναι πλέον ορατή στο χρήστη. Αυτό συμβαίνει όταν συμβαίνει ένα από τα ακόλουθα:

- Μια νέα δραστηριότητα ξεκίνησε και καλύπτει τη δραστηριότητα αυτή.
- Μια υπάρχουσα δραστηριότητα έρχεται στο προσκήνιο.
- Η δραστηριότητα καταστρέφεται.

Η `onStop` δεν καλείται πάντα σε καταστάσεις χαμηλής μνήμης, όπως όταν το Android χρειάζεται πόρους και δεν μπορεί να τοποθετήσει την δραστηριότητα στο παρασκήνιο. Για το λόγο αυτό, το καλύτερο είναι να μην βασίζομαστε στην `onStop` όταν προετοιμάζουμε μια δραστηριότητα για καταστροφή.

Οι επόμενες μέθοδοι κύκλου ζωής που μπορεί να κληθούν μετά από αυτή είναι η `onDestroy`, εάν η δραστηριότητα πρόκειται να καταστραφεί, ή η `onRestart` εάν η δραστηριότητα έρχεται πίσω για να αλληλεπιδράσει με το χρήστη.

onDestroy

Αυτή είναι η τελική μέθοδος που καλείται σε ένα στιγμιότυπο δραστηριότητας προτού να καταστραφεί και να αφαιρεθεί εντελώς από τη μνήμη. Σε ακραίες καταστάσεις το Android μπορεί να σκοτώσει τη διεργασία εφαρμογής που φιλοξενεί τη δραστηριότητα, η οποία θα οδηγήσει στην μη κλήση της `onDestroy`. Οι περισσότερες δραστηριότητες δεν υπερφορτώνουν αυτή τη μέθοδο, επειδή η περισσότερη δουλειά τερματισμού και καθαρισμού έχει γίνει στην `onPause` και στην `onStop`. Η μέθοδος `onDestroy` συνήθως υπερφορτώνεται για να καθαρίσει πόρους που έχουν μείνει πολύ καιρό ανενεργοί.

Δεν υπάρχουν μέθοδοι του κύκλου ζωής που καλούνται μετά από την καταστροφή της δραστηριότητας.

onRestart

Αυτή η μέθοδος καλείται πριν την επανεκκίνηση μίας σταματημένης δραστηριότητας. Ένα καλό παράδειγμα αυτού είναι όταν ο χρήστης πατήσει το κουμπί αρχικής οθόνης, ενώ βρίσκεται μέσα σε μια δραστηριότητα στην εφαρμογή. Όταν συμβαίνει αυτό, καλούνται οι μέθοδοι `onPause` και `onStop`, και η δραστηριότητα μετακινείται στο παρασκήνιο, αλλά δεν καταστρέφεται. Εάν ο χρήστης στη συνέχεια αποφασίσει να αποκαταστήσει την εφαρμογή, το Android θα καλέσει τη μέθοδο `onRestart` της δραστηριότητας.

Δεν υπάρχουν γενικές κατευθυντήριες γραμμές για το τι είδους λογική πρέπει να υλοποιείται στην `onRestart`. Αυτό συμβαίνει επειδή η `onStart` καλείται πάντα ανεξάρτητα από το αν η δραστηριότητα δημιουργείται ή επανεκκινείται. Έτσι όλοι οι πόροι που απαιτεί η δραστηριότητα θα πρέπει να προετοιμαστούν στην `onStart`, παρά στην `onRestart`.

Η επόμενη μέθοδος του κύκλου ζωής που καλείται μετά την `onRestart` είναι η `onStart`.

4. Ανάλυση των λειτουργιών και των τεχνολογιών της εφαρμογής

Στα παραπάνω κεφάλαια είδαμε κάποια γενικά στοιχεία για τις ψηφιακές βιβλιοθήκες και την συνεργατικότητα σε αυτές. Επίσης αναλύσαμε το προγραμματιστικό περιβάλλον Android, το οποίο χρησιμοποιήθηκε σε αυτήν την μεταπτυχιακή διατριβή για την υλοποίηση της εφαρμογής LibrApp. Στο κεφάλαιο αυτό θα παρουσιάσουμε στην αρχή τις υπόλοιπες τεχνολογίες και τεχνικές που χρησιμοποιήθηκαν στην υλοποίηση, και στη συνέχεια θα παρουσιάσουμε την ανάλυση απαιτήσεων, τον σχεδιασμό της εφαρμογής, καθώς και τον τρόπο υλοποίησής της.

4.1. Περιγραφή λοιπών τεχνολογιών

Στην παρούσα μεταπτυχιακή διατριβή, πέραν από το προγραμματιστικό περιβάλλον Android (Android API), χρησιμοποιήθηκαν και κάποιες άλλες ακόμα τεχνολογίες για την υλοποίηση, επικοινωνία και αλληλεπίδραση με τον κεντρικό εξυπηρετητή και για την αποθήκευση των αντικειμένων που αποτελούν την ψηφιακή βιβλιοθήκη.

Συγκεκριμένα, η ανάπτυξη της εφαρμογής έγινε με την χρήση του Android API 18 (Android 4.3 Jelly Bean³), και για την επικοινωνία της εφαρμογής με τον εξυπηρετητή για την αποστολή και λήψη δεδομένων και πληροφοριών, χρησιμοποιήθηκε η τεχνολογία JSON. Στην πλευρά του εξυπηρετητή χρησιμοποιήθηκε η γλώσσα προγραμματισμού PHP 5, για την υλοποίηση των υπηρεσιών (restful services), οι οποίες επικοινωνούν με την βάση δεδομένων της βιβλιοθήκης και εκτελούν τα αιτήματα του χρήστη. Η βάση δεδομένων είναι μία σχεσιακή βάση MySQL, η οποία όπως και η PHP, εκτελούνται σε περιβάλλον εξυπηρετητή Apache. Τέλος, για την υλοποίηση της διεπαφής Web με την οποία θα μπορεί ο διαχειριστής του συστήματος να επεξεργάζεται στοιχεία της βιβλιοθήκης και των χρηστών, χρησιμοποιήθηκε, πέραν από τις τεχνολογίες HTML και PHP, το JQuery, το οποίο είναι ένα πλαίσιο της γλώσσας διαδικτυακού προγραμματισμού JavaScript.

Στις παρακάτω υποενότητες θα πραγματοποιήσουμε μία συνοπτική παρουσίαση για κάθε μία από αυτές τις τεχνολογίες.

4.1.1. Το πρότυπο JSON

Το JSON⁴ (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων, το οποίο βασίζεται σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript. Το JSON είναι ένα πρότυπο κειμένου, το οποίο είναι εντελώς ανεξάρτητο από την γλώσσα προγραμματισμού που το χρησιμοποιεί, αλλά εκμεταλλεύεται τις συμβάσεις που είναι γνωστές στους προγραμματιστές της οικογένειας γλωσσών που συμπεριλαμβάνει γλώσσες προγραμματισμού όπως οι C, C++, C#, Java, JavaScript, Perl, Python, και πολλές άλλες. Αυτές οι ιδιότητες καθιστούν το JSON μια ιδανική γλώσσα ανταλλαγής δεδομένων. Χρησιμοποιείται κυρίως για τη μετάδοση δεδομένων μεταξύ ενός διακομιστή και μίας web εφαρμογής, ως εναλλακτική λύση στην XML.

Το JSON είναι χτισμένο πάνω σε δύο δομές:

- Μια συλλογή από ζεύγη ονόματος/τιμής. Διαφορετικές γλώσσες αντιλαμβάνονται αυτά τα ζεύγη με διαφορετικούς τρόπους πχ, ως ένα αντικείμενο, εγγραφή, δομή, λεξικό, πίνακα κατακερματισμού, λίστα με κλειδιά, ή πίνακα.
- Μια ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες, αυτό γίνεται αντιληπτό ως μια σειρά, διάνυσμα, λίστα, ή ακολουθία.

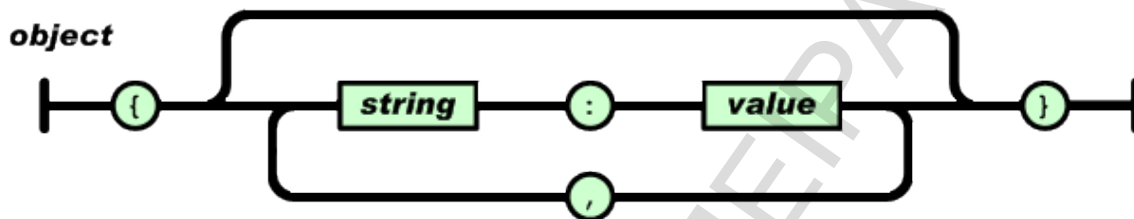
Βασικοί τύποι δεδομένων οι οποίοι περιγράφονται από το JSON είναι :

- Αριθμός (Number) – προσημασμένος δεκαδικός αριθμός που μπορεί να περιέχει ένα κλασματικό μέρος και μπορεί να χρησιμοποιήσει εκθετική σημειογραφία. Το JSON δεν επιτρέπει τη χρήση μη - αριθμών, όπως το NaN, ούτε κάνει καμία διάκριση μεταξύ των ακεραίων και των αριθμών κινητής υποδιαστολής.

3 <http://www.android.com/about/jelly-bean/>

4 <http://www.json.org/>

- Αλφαριθμητικό (String) - μια ακολουθία από μηδέν ή περισσότερους χαρακτήρες Unicode. Τα αλφαριθμητικά οριοθετούνται με διπλά εισαγωγικά και υποστηρίζουν χαρακτήρες διαφυγής.
- Boolean – Μία από τις λέξεις «αληθής» ή «ψευδής» (True or False).
- Πίνακας (Array) - μια ταξινομημένη λίστα με μηδέν ή περισσότερες τιμές, καθεμία από τις οποίες μπορεί να είναι οποιοδήποτε τύπου.
- Αντικείμενο (Object) - Ένας μη διατεταγμένος χάρτης (ζεύγη ονόματος/τιμής). Τα αντικείμενα οριοθετούνται με άγκιστρα και χρησιμοποιούν κόμματα για το διαχωρισμό κάθε ζεύγους. Όλα τα κλειδιά πρέπει να είναι αλφαριθμητικά και πρέπει να είναι μοναδικά σε σχέση με κάθε άλλο μέσα σε αυτό το αντικείμενο. Η δομή ενός αντικειμένου φαίνεται στο Σχήμα 14.
- Κενό (null) - Μια κενή τιμή.



Σχήμα 14 Δομή αντικειμένου JSON

Το JSON αγνοεί γενικά τα κενά γύρω ή ανάμεσα από τα συντακτικά στοιχεία. Ωστόσο το JSON αναγνωρίζει μόνο τέσσερις ειδικές κενούς χαρακτήρες: το space, το tab, την αλλαγή γραμμής, και το carriage return. Παρακάτω φαίνεται ένα απλό παράδειγμα JSON.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "height_cm": 167.64,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    { "type": "home", "number": "212 555-1234" },
    { "type": "fax", "number": "646 555-4567" }
  ]
}
```

Εικόνα 1 Παράδειγμα JSON

Στην εφαρμογή μας, θα χρησιμοποιούμε το πρότυπο JSON για να μεταφέρουμε πληροφορία από και προς τον απομακρυσμένο server. Για τον λόγο αυτό χρησιμοποιούνται διάφοροι parsers που μετατρέπουν την πληροφορία στην απαιτούμενη μορφή JSON. Η παρουσίαση αυτών των parser θα γίνει στο Παράρτημα 2.

4.1.2. Η γλώσσα προγραμματισμού PHP

Η γλώσσα προγραμματισμού PHP⁵ (αναδρομικό ακρωνύμιο του PHP: Hypertext Preprocessor) είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού scripting γλώσσα προγραμματισμού σεναρίων που είναι ιδιαίτερα κατάλληλη για την ανάπτυξη διαδικτυακών εφαρμογών μιας και μπορεί να ενσωματωθεί στην γλώσσα HTML.

Αντί για πολλές εντολές για την παραγωγή κώδικα HTML (όπως συμβαίνει σε γλώσσες όπως η C ή η Perl), οι σελίδες PHP περιέχουν ενσωματωμένο κώδικα HTML. Ο κώδικας PHP είναι εσώκλειστος σε ειδικές οδηγίες έναρξης και λήξης.

Η PHP διαχωρίζεται από τις γλώσσες που εκτελούν τον κώδικά τους στην πλευρά του πελάτη (client-side, όπως η Javascript), από το ότι ο κώδικας εκτελείται στον server, και παράγει HTML η οποία στη συνέχεια αποστέλλεται στον πελάτη.

Η PHP είναι είναι εξαιρετικά απλή, αλλά προσφέρει πολλά προηγμένα χαρακτηριστικά στους προγραμματιστές. Στην εικόνα 2 βλέπουμε ένα παράδειγμα χρήσης της γλώσσας PHP, για την εμφάνιση ενός απλού μηνύματος.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Γεια σου κόσμο";
    ?>

  </body>
</html>
```

Εικόνα 2 Παράδειγμα κώδικα PHP

Στην εφαρμογή που αναπτύχθηκε σε αυτή την μεταπτυχιακή διατριβή, η γλώσσα PHP χρησιμοποιήθηκε για την συγγραφή μεμονωμένων υπηρεσιών, οι οποίες λειτουργούν σαν μεσοσμικό (middleware) με την βάση δεδομένων. Επίσης χρησιμοποιήθηκαν στην δημιουργία της διαδικτυακής διεπαφής του διαχειριστή του συστήματος. Η ανάλυση των κλάσεων PHP που χρησιμοποιήθηκαν γίνεται στο Παράρτημα 2.

4.1.3. Βάση δεδομένων MySQL

Η MySQL αποτελεί το δεύτερο στον κόσμο πιο ευρέως χρησιμοποιημένο σύστημα ανοικτού λογισμικού διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS).

Η MySQL είναι μια δημοφιλής επιλογή βάσης δεδομένων για χρήση σε διαδικτυακές εφαρμογές. Στην εφαρμογή που παρουσιάζουμε εδώ, χρησιμοποιούμε την MySQL, για βάση δεδομένων και την διαχειριζόμαστε μέσω του εργαλείου phpMyAdmin.

Στο παράρτημα 2 θα δοθεί μια αναλυτική περιγραφή των πινάκων και του σχήματος της βάσης.

⁵ <http://www.php.net/>

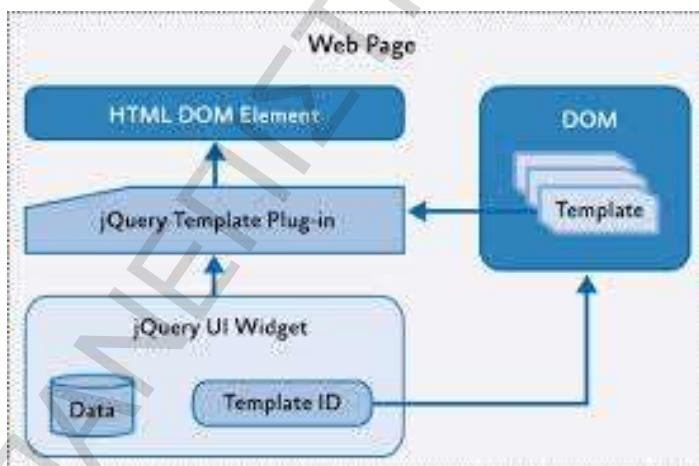
4.1.4.To JQuery Framework

Το jQuery⁶ είναι μια βιβλιοθήκη της γλώσσας προγραμματισμού JavaScript που επιτρέπει στους web developers να προσθέσουν επιπλέον λειτουργικότητα στις ιστοσελίδες τους. Είναι ανοιχτού κώδικα και παρέχεται δωρεάν υπό την άδεια MIT. Τα τελευταία χρόνια, το jQuery έχει γίνει η πιο δημοφιλής βιβλιοθήκη της JavaScript.

Για την χρήση του jQuery, ένας προγραμματιστής πρέπει απλώς να προσθέσει μία αναφορά προς στο αρχείο του jQuery στον κώδικα HTML μιας ιστοσελίδας. Ορισμένες ιστοσελίδες φιλοξενούν το δικό τους τοπικό αντίγραφο του jQuery, ενώ άλλες απλώς χρησιμοποιούν μία αναφορά προς την βιβλιοθήκη η οποία φιλοξενείται από την Google ή το διακομιστή του jQuery.

Μόλις φορτώσει την βιβλιοθήκη του jQuery, μια ιστοσελίδα μπορεί να καλέσει οποιαδήποτε λειτουργία jQuery υποστηρίζεται από τη βιβλιοθήκη. Κοινά παραδείγματα περιλαμβάνουν την τροποποίηση του κειμένου, προσθήκη κινουμένων στοιχείων σε μια σελίδα, και την εκτέλεση animations. Το jQuery μπορεί επίσης να λειτουργήσει με κώδικα Ajax και γλώσσες σεναρίων, όπως η PHP και η ASP για να αποκτήσει πρόσβαση στα δεδομένα μιας βάσης δεδομένων. Το jQuery τρέχει στην πλευρά του πελάτη (αντί για τον web server), και μπορεί να ενημερώσει τις πληροφορίες σε μια ιστοσελίδα σε πραγματικό χρόνο, χωρίς την επαναφόρτωση της σελίδας. Ένα κοινό παράδειγμα είναι η αυτόματη συμπλήρωση, στην οποία μια μορφή αναζήτησης εμφανίζει αυτόματα κοινές αναζητήσεις, καθώς ο χρήστης πληκτρολογεί το ερώτημα του.

Εκτός από την ελεύθερη άδεια χρήσης του, ο άλλος κύριος λόγος που το jQuery έχει κερδίσει τέτοια δημοτικότητα είναι η συμβατότητά του με όλους τους φυλλομετρητές. Δεδομένου ότι κάθε πρόγραμμα περιήγησης ορίζει την HTML, τα CSS, και την JavaScript με διαφορετικό τρόπο, μπορεί να είναι δύσκολο για έναν προγραμματιστή να κάνει μια ιστοσελίδα να εμφανίζεται το ίδιο σε όλους τους φυλλομετρητές. Αντί να χρειάζεται να γράψει προσαρμοσμένες λειτουργίες για κάθε πρόγραμμα περιήγησης, ένας προγραμματιστής μπορεί να χρησιμοποιήσει μια ενιαία λειτουργία jQuery που θα λειτουργήσει το ίδιο σε όλους τους φυλλομετρητές. Αυτή η υποστήριξη multi-browser έχει οδηγήσει πολλούς προγραμματιστές να στραφούν από το πρότυπο jQuery, δεδομένου ότι απλοποιεί σημαντικά τη διαδικασία κωδικοποίησης. Στην Εικόνα 3 μπορούμε να δούμε την δομή του JQuery και πώς αυτό αλληλεπιδρά με το DOM (Document Object Model).



Εικόνα 3 Η δομή του JQuery (msdn.microsoft.com)

Στην παρούσα μεταπτυχιακή διατριβή η βιβλιοθήκη JQuery θα χρησιμοποιηθεί σε συνδυασμό με τις γλώσσες PHP και HTML, για την δημιουργία της διεπαφής του διαχειριστή του συστήματος. Το JQuery επιλέχθηκε για την ευκολία την οποία διαθέτει στην γρήγορη αλλαγή της ιστοσελίδας χωρίς ανανέωση και στην ταχεία εμφάνιση αποτελεσμάτων.

⁶ <http://jquery.com/>

Παραδείγματα χρήσης του JQuery για την δημιουργία της εφαρμογής δίνονται στο Παράρτημα 2.

4.2. Μοντέλο πελάτη/εξυπηρετητή και άλλες τεχνικές

Σε αυτή την υποενότητα θα αναλύσουμε τις τεχνικές τις οποίες χρησιμοποιήσαμε κατά την εκπόνηση αυτής της μεταπτυχιακής διατριβής για την ανάπτυξη της εφαρμογής LibrApp. Το βασικό μοντέλο πάνω στο οποίο αναπτύχθηκε η εφαρμογή είναι το μοντέλο πελάτη/εξυπηρετητή (client/server model).

Επίσης, για την συγγραφή του κώδικα Java που απαρτίζει την εφαρμογή, χρησιμοποιήθηκε μία από τις πιο διαδεδομένες αρχιτεκτονικές κώδικα τον τελευταίο καιρό, η αρχιτεκτονική Μοντέλου-Όψης-Ελεγκτή (Model-View-Controller) ή MVC.

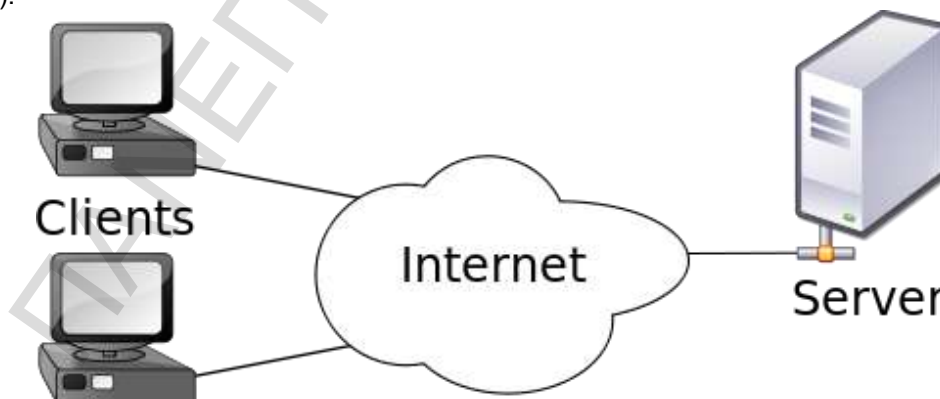
Παρακάτω θα αναλύσουμε αυτές τις δύο τεχνικές:

4.2.1. Το μοντέλο πελάτη/εξυπηρετητή

Το μοντέλο πελάτη-εξυπηρετητή ή αλλιώς client-server είναι μια κατακεντρωμένη δομή εφαρμογής που διαχωρίζει τα καθήκοντα και το φόρτο εργασίας μεταξύ των παρόχων ενός πόρου ή μιας υπηρεσίας, που ονομάζεται εξυπηρετητής (server), και αυτών που αιτούνται την υπηρεσία, και ονομάζονται πελάτες (clients) [48]. Ο server είναι συνήθως ένας υπολογιστής ή ένα σύστημα πολλαπλών υπολογιστών (cluster) με μεγάλη υπολογιστική δυνατότητα ο οποίος προσφέρει στους πελάτες ένα συγκεκριμένο είδος υπηρεσιών. Ένας εξυπηρετητής κατηγοριοποιείται ανάλογα με το είδος της προσφερόμενης υπηρεσίας και μπορεί να είναι database server, email server, application server, web server ή άλλο.

Αν ένας υπολογιστής αποτελεί πελάτη, εξυπηρετητή, ή και τα δύο, καθορίζεται από τη φύση της εφαρμογής που απαιτεί τις λειτουργίες υπηρεσιών. Για παράδειγμα, ένας μόνο υπολογιστής μπορεί να εκτελεί τον διαδικτυακό εξυπηρετητή (web server) και τον εξυπηρετητή αρχείων (file server) ταυτόχρονα, εξυπηρετώντας διαφορετικούς πελάτες οι οποίοι κάνουν διάφορα είδη αιτήσεων. Το λογισμικό του πελάτη μπορεί να επικοινωνήσει με το λογισμικό του εξυπηρετητή εντός του ίδιου υπολογιστή.

Οι πελάτες και οι εξυπηρετητές ανταλλάσσουν μηνύματα χρησιμοποιώντας ένα μοτίβο αιτήματος-απάντησης (request-response pattern). Ο πελάτης αποστέλλει μια αίτηση, και ο εξυπηρετητής επιστρέφει μια απάντηση. Για την επικοινωνία πελάτη-εξυπηρετητή, οι υπολογιστές πρέπει να διαθέτουν μια κοινή γλώσσα, και πρέπει να ακολουθούν κάποιους κανόνες, έτσι ώστε τόσο ο πελάτης όσο και ο εξυπηρετητής να ξέρουν τι να περιμένουν. Η γλώσσα και οι κανόνες επικοινωνίας ορίζονται σε ένα πρωτόκολλο επικοινωνίας. Όλα τα πρωτόκολλα client-server λειτουργούν στο επίπεδο εφαρμογής του μοντέλου OSI (Application layer).



Εικόνα 4 Αρχιτεκτονική Client-Server

Με την αυξανόμενη πρόοδο των τεχνολογιών ασύρματης δικτύωσης[49] και των κινητών συσκευών έχει προκύψει ένας νέος τομέας για την εξυπηρέτηση των χρηστών που χρησιμοποιούν κινητές συσκευές, η κινητή πληροφορική (mobile computing). Το νέο υπολογιστικό περιβάλλον δημιουργεί διαφορετικές ανάγκες αντιμετώπισης και για αυτό το

μοντέλο client/server καθώς και οι εφαρμογές σε ένα κινητό περιβάλλον θα πρέπει να προσαρμοστούν στις ανάγκες που απορρέουν από την κινητικότητα των χρηστών. Θα πρέπει δηλαδή οι εφαρμογές να προσαρμοστούν στην ανάγκη των χρηστών για υπολογιστικότητα on-demand συνεχώς και οπουδήποτε και για χρήση σε συνθήκες χαμηλών πόρων (όπως πχ σε μια κινητή συσκευή με περιορισμένη μπαταρία). Οι περιορισμένοι πόροι μίας κινητής συσκευής, όπως η μπαταρία, η μικρή υπολογιστική ισχύς, ο μικρός αποθηκευτικός χώρος κτλ, δημιουργούν την ανάγκη στις εφαρμογές και στις αρχιτεκτονικές δικτύων να αναπτύξουν μηχανισμούς για την αντιμετώπισή τους. Τέτοιες αρχιτεκτονικές είναι η λεπτή αρχιτεκτονική (thin architecture), η πλήρης αρχιτεκτονική (full architecture), οι proxy server κ.α.

4.2.2.Η αρχιτεκτονική MVC

Το πρότυπο Model-View-Controller (MVC) είναι ένα πρότυπο λογισμικού για την υλοποίηση διεπαφών χρήστη [50]. Διαιρεί μια δεδομένη εφαρμογή λογισμικού σε τρία αλληλένδετα μέρη, έτσι ώστε να διαχωρίζονται οι εσωτερικές αναπαραστάσεις των πληροφοριών από τους τρόπους που οι πληροφορίες παρουσιάζονται στον χρήστη [51] [52]. Η κεντρική συνιστώσα, το μοντέλο (model), αποτελείται από τα δεδομένα της εφαρμογής, τους επιχειρησιακούς κανόνες, τη λογική της εφαρμογής και τις λειτουργίες. Μία όψη (View) μπορεί να είναι οποιαδήποτε μορφή εξόδου των πληροφοριών, όπως ένα γράφημα ή ένα διάγραμμα. Είναι δυνατόν να υπάρχουν πολλαπλές όψεις των ίδιων πληροφοριών, όπως για παράδειγμα ραβδογραφήματα και πίνακες που περιέχουν την ίδια πληροφορία αλλά για διαφορετικούς χρήστες. Η τρίτη συνιστώσα, ο ελεγκτής, δέχεται είσοδο και την μετατρέπει σε εντολές για το μοντέλο ή την όψη [53].

Αλληλεπίδραση των συνιστωσών

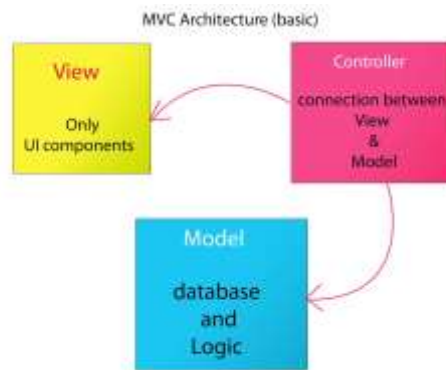
Εκτός από τη διαίρεση της εφαρμογής σε τρία είδη των συστατικών, το μοντέλο MVC καθορίζει επίσης τις αλληλεπιδράσεις μεταξύ τους [54].

- Το μοντέλο ειδοποιεί τις όψεις και τους ελεγκτές με τους οποίους συσχετίζεται, όταν συμβαίνει κάποια αλλαγή στην κατάσταση. Η κοινοποίηση αυτή επιτρέπει στις όψεις να ενημερώσουν το περιεχόμενο που παρουσιάζουν στον χρήστη, και στους ελεγκτές να αλλάξουν το διαθέσιμο σύνολο των εντολών.
- Μια όψη ενημερώνεται από τον ελεγκτή με όλες τις πληροφορίες που χρειάζεται για την παραγωγή μιας της δημιουργίας της εξόδου που θα παρουσιαστεί στον χρήστη. Μπορεί επίσης να παρέχει γενικούς μηχανισμούς για να ενημερώσει τον ελεγκτή σχετικά με τυχόν είσοδο από τον χρήστη.
- Ένας ελεγκτής μπορεί να στέλνει εντολές στο μοντέλο για να ενημερώσει την κατάστασή του. Μπορεί επίσης να στέλνει εντολές και στις συσχετισμένες με αυτόν όψεις ώστε να αλλάξει την παρουσίαση της όψης σχετικά με το μοντέλο.

Χρήση σε εφαρμογές Web

Αν και αρχικά αναπτύχθηκε για την υλοποίηση εφαρμογών desktop, το πρότυπο Model View Controller έχει πλέον υιοθετηθεί ευρέως ως μια αρχιτεκτονική για εφαρμογές Web σε όλες τις μεγάλες γλώσσες προγραμματισμού. Αρκετές εμπορικά και μη εμπορικά πλαίσια εφαρμογών (application frameworks) έχουν δημιουργηθεί που ενισχύουν/χρησιμοποιούν το πρότυπο MVC. Τα πλαίσια αυτά διαφέρουν ως προς τρόπο με τον οποίο χρησιμοποιούν το πρότυπο, κυρίως στην κατανομή των αρμοδιοτήτων μεταξύ του πελάτη και του διακομιστή [55].

Τα πρόωρα πλαίσια MVC web χρησιμοποιούσαν μια λεπτή προσέγγιση πελάτη (thin client approach) που τοποθετούσε σχεδόν ολόκληρο το μοντέλο, την όψη και τη λογική ελέγχου στο διακομιστή. Σε αυτή την προσέγγιση, ο πελάτης αποστέλλει είτε HTTP αιτήματα είτε εισόδους φόρμας στον ελεγκτή και στη συνέχεια λαμβάνει μία πλήρη και ενημερωμένη web σελίδα (ή άλλο έγγραφο) από την όψη, ενώ το μοντέλο υπάρχει εξ ολοκλήρου στο διακομιστή. Πλέον, καθώς οι τεχνολογίες πελάτη έχουν ωριμάσει, τέτοιου είδους πλαίσια έχουν αντικατασταθεί από άλλα τα οποία επιτρέπουν στις συνιστώσες του MVC να εκτελούνται εν μέρει και στην πλευρά του πελάτη.



Εικόνα 5 MVC Αρχιτεκτονική

Η MVC αρχιτεκτονική στο Android

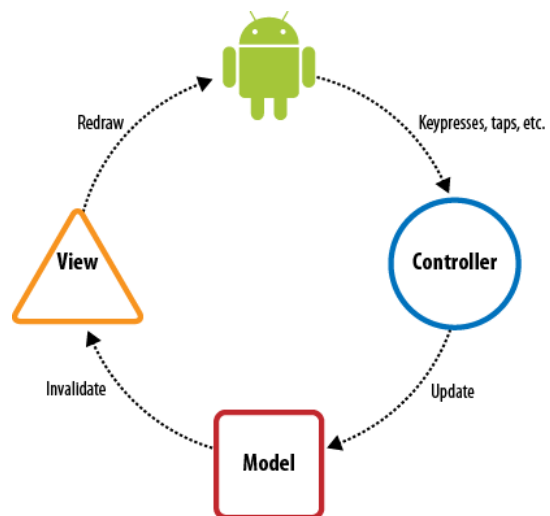
Το προγραμματιστικό περιβάλλον του Android δεν υλοποιεί ακριβώς την αρχιτεκτονική του προτύπου MVC, αλλά ακολουθεί μία παρεμφερή προσέγγιση. Αφήνει στον προγραμματιστή το να μοντελοποιήσει τον κώδικα του ακολουθώντας τους παρακάτω κανόνες σχετικά με τις ουσιαστικές του MVC. Αναλυτικά:

- Μοντέλο: Τι πρέπει να εμφανιστεί στον χρήστη
- Όψη: Πώς να θα εμφανιστεί στον χρήστη
- Ελεγκτής: Είσοδος από τον χρήστη, διαχείριση γεγονότων κλπ.

Ουσιαστικά το Android μας παρέχει τους παρακάτω τρόπους για να λειτουργήσουμε σε MVC αρχιτεκτονική:

- Προσδιορίζουμε την διεπαφή χρήστη σε διάφορα xml αρχεία ορίζοντας τα layouts. Έτσι κατά κάποιον τρόπο δημιουργούμε τις όψεις.
- Μπορούμε επίσης να ορίσουμε τους πόρους που θα χρησιμοποιήσουμε σε διάφορα αρχεία xml, αναπαριστώντας ένα μέρος του μοντέλου..
- Μπορούμε να επεκτείνουμε κλάσεις όπως ListActivity, TabActivity και να κάνουμε χρήση των xml layouts χρησιμοποιώντας inflaters. Αυτό ουσιαστικά ορίζει τις Java κλάσεις που γράφουμε ως τους ελεγκτές των όψεων.

Στην εικόνα 6 μπορούμε να δούμε πως λειτουργεί η MVC αρχιτεκτονική σε χρήση με το προγραμματιστικό περιβάλλον Android. Όπως αναφέρθηκε και παραπάνω, το μοντέλο ουσιαστικά επικυρώνει την όψη (τα xml layouts). Η όψη είναι αυτή που εμφανίζεται στον χρήστη σαν διεπαφή και ο Controller αποτελεί ουσιαστικά τον μηχανισμό ο οποίος λαμβάνει τα user interactions, τα μεταφέρει στο μοντέλο και κατα συνέπεια στις όψεις.



Εικόνα 6 MVC αρχιτεκτονική στο σύστημα Android

4.3.Ανάλυση Απαιτήσεων

Στην ενότητα αυτή γίνεται περιγραφή της ανάλυσης απαιτήσεων του πληροφοριακού συστήματος που αναπτύχθηκε στην παρούσα μεταπτυχιακή διατριβή. Οι αναλύσεις αυτές βασίζονται και επεκτείνουν εκείνες που δώθηκαν σε προηγούμενη μεταπτυχιακή διατριβή [60]. Πιο συγκεκριμένα δίνεται μία αναλυτική περιγραφή των χαρακτηριστικών της Android ψηφιακής βιβλιοθήκης LibrApp:

Οι γενικότερες απαιτήσεις του συστήματος είναι:

- Δημιουργία ηλεκτρονικού αποθετηρίου που θα είναι συνεχώς ενήμερο με την πλειονότητα των αντικειμένων (βιβλίων, περιοδικών, συγγραμμάτων) που διαθέτει το ίδρυμα σε φυσική μορφή (τουλάχιστον του 90% για να θεωρείται έγκυρη η ψηφιακή βιβλιοθήκη), καθώς και άλλου υλικού διαθέσιμου μόνο σε ψηφιακή μορφή.
- Αυτοματοποίηση των δραστηριοτήτων τόσο των χρηστών, όσο των διαχειριστών/βιβλιοθηκονόμων.
- Ασφάλεια του δικτύου και του συστήματος: το ΠΣ καθώς και το δίκτυο στο οποίο θα εγκατασταθεί, θα πρέπει να διαθέτει επαρκή ασφάλεια μιας και είναι σημαντικό να μην συμβούν αλλαγές στις πληροφορίες ή διαρροή αυτών, εφόσον πολλά από τα αντικείμενα θα είναι αγορασμένα από το ίδρυμα και θα διατίθενται στους χρήστες με ειδική άδεια διακίνησης.
- Ευκολία χρήσης
- Παράλληλη λειτουργία από πολλούς χρήστες
- Παροχή πολλών λειτουργιών και εργαλείων τόσο στον χρήστη όσο και στους διαχειριστές/βιβλιοθηκονόμους.
- Αύξηση της παραγωγικότητας των διαχειριστών.
- Αύξηση της αποδοτικότητας των χρηστών.
- Λογικός προϋπολογισμός για την εγκατάσταση και χρήση του ΠΣ. Θα πρέπει το ΠΣ πέραν από το στόχο του να αυξήσει την παραγωγικότητα και την αποδοτικότητα των ανθρώπων που καλείται να εξυπηρετήσει να διαθέτει ένα βιώσιμο κόστος εγκατάστασης και συντήρησης για το ίδρυμα που θα το υιοθετήσει.
- Αποφόρτιση των διαχειριστών, μειώνοντας τα ζητήματα τα οποία καλούνται να λύσουν.

Ειδικότερα οι απαιτήσεις μίας ψηφιακής βιβλιοθήκης είναι:

- Δημιουργία λογαριασμού χρήστη και εύκολη, γρήγορη και φιλική προς τον χρήστη διαχείριση αυτού.
- Δυνατότητα αναζήτησης μέσα στο αποθετήριο για βιβλία ή αντικείμενα χωρίς την ανάγκη λογαριασμού αλλά με περιορισμένες δυνατότητες (μόνο προβολή).
- Ενισχυμένη δυνατότητα αναζήτησης σε χρήστες με ενεργό προφίλ στην βιβλιοθήκη.
- Δυνατότητα επεξεργασίας του προφίλ χρήστη.
- Δυνατότητα προσθήκης των συγγραμμάτων σε προσωπικό ψηφιακό ράφι (personal digital bookshelf).
- Εύρεση εγγεγραμμένων φίλων ή συνεργατών.
- Δυνατότητα σύνδεσης με φίλους και συνεργάτες.
- Κοινοποίηση αγαπημένων ή αποθηκευμένων αντικειμένων στα κοινωνικά μέσα (Facebook, Twitter, LinkedIn).
- Δυνατότητα αποθήκευσης συγκεκριμένων αντικειμένων εφόσον ή άδεια χρήσης του το επιτρέπει.
- Σχολιασμός και αξιολόγηση των αντικειμένων της βιβλιοθήκης.

Επίσης, απαιτήσεις οι οποίες αφορούν ειδικά την συνεργατικότητα μεταξύ των χρηστών είναι:

- Δημιουργία κοινού έργου με τη μορφή ελεύθερου κειμένου (wiki ή παρεμφερές)
- Κοινή χρήση των έργων με συνδεδεμένους συνεργάτες.
- Διατήρηση εκδόσεων των έργων (versioning).
- Ανταλλαγή υλικού μεταξύ των μελών του ίδιου έργου.

- Επικοινωνία με φίλους/συνεργάτες μέσω διαφόρων μορφών (chat, forum, asynchronous messaging etc.).
 - Ενημέρωση για τις δραστηριότητες των φίλων/συνεργατών.
- Οι απαιτήσεις οι οποίες αφορούν την διαχείριση καθηκόντων (task management) είναι:
- Δημιουργία λίστας καθηκόντων για συγκεκριμένο έργο.
 - Διαμοιρασμός αυτής της λίστας σε αυτούς που τους αφορά.
 - Συσχέτιση καθήκοντος με άτομο.
 - Δυνατότητα συμπλήρωσης σχολίων.
 - Δυνατότητα σε αυτούς που έχουν συγκεκριμένα καθήκοντα να ενημερώνουν για την ολοκλήρωσή τους.
- Οι απαιτήσεις από την πλευρά των Διαχειριστών-Βιβλιοθηκονόμων είναι:
- Διαχείριση των χρηστών του συστήματος (πχ σε περίπτωση χαμένου κωδικού ή κάποιου παραπόνου).
 - Διαχείριση του περιεχομένου του συστήματος.
 - Προσθήκη συγγραμμάτων και αντικειμένων.
 - Διαχείριση των δικαιωμάτων τα οποία περιβάλλουν τα αντικείμενα του συστήματος.
 - Ενημέρωση των χρηστών για τυχόν αλλαγές στο σύστημα.

4.4.Σχεδιασμός του συστήματος

Η εφαρμογή LibrApp αναπτύχθηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής με σκοπό να καλύψει τις περισσότερες από τις λειτουργικές απαιτήσεις οι οποίες παρουσιάστηκαν στην προηγούμενη ενότητα. Η LibrApp σχεδιάστηκε για να προσφέρει στους χρήστες και τους διαχειριστές παραδοσιακών βιβλιοθηκών, ψηφιακών βιβλιοθηκών ή παρόμοιων αποθετηρίων την δυνατότητα να έχουν πρόσβαση στην πληροφορία από οποιοδήποτε σημείο και να βρίσκονται, καθώς και να αλληλεπιδρούν και συνεργάζονται με αποτελεσματικότητα με συνεργάτες και συναδέλφους που εκτελούν το ίδιο έργο. Ο απώτερος σκοπός της σχεδίασης του ήταν η βελτιστοποίηση της αποδοτικότητας των χρηστών προσφέροντας τους μία σύγχρονη πλατφόρμα εύρεσης πληροφοριών, δίνοντας τους ταυτόχρονα πρόσβαση σε μία ενημερωμένη παλέτα συνεργατικών εργαλείων.

Για τον σχεδιασμό της κινητής ψηφιακής βιβλιοθήκης LibrApp χρησιμοποιήθηκε το μοντέλο client – server (το οποίο αναλύθηκε σε προηγούμενη υποενότητα αυτού το κεφαλαίου) ειδικά προσαρμοσμένο στις ανάγκες της κινητής υπολογιστικότητας και διαμορφωμένο με τέτοιο τρόπο ώστε να βελτιστοποιείται η χρήση του από την εφαρμογή. Πιο συγκεκριμένα, η LibrApp αποτελείται από έναν διαδικτυακό διακομιστή (web server) ο οποίος είναι συνδεδεμένος με μία βάση δεδομένων (με την χρήση ενός ενδιάμεσου λογισμικού ή middleware το οποίο φροντίζει για την σωστή επικοινωνία) και ένα πλήθος κινητών πελατών οι οποίοι συνδέονται με τον διακομιστή και αλληλεπιδρούν με τη βάση δεδομένων. Η κυρίως εφαρμογή της LibrApp είναι σχεδιασμένη για να εκτελείται από τα κινητά τηλέφωνα των χρηστών (Android OS), ενώ υπάρχει και μία web εφαρμογή η οποία είναι σχεδιασμένη για χρήση από τους διαχειριστές και τους βιβλιοθηκονόμους του ιδρύματος/βιβλιοθήκης. Τα κινητά τηλέφωνα των χρηστών χρησιμοποιούν την εφαρμογή για να συνδεθούν με την απομακρυσμένο web server και να ζητήσουν πρόσβαση στις πληροφορίες της βιβλιοθήκης. Και οι δύο εφαρμογές πελάτες δεν επικοινωνούν άμεσα με τον διακομιστή. Ένα ενδιάμεσο λογισμικό σε μορφή υπηρεσίας ασχολείται με τα αιτήματα των εφαρμογών προς τον διακομιστή, δημιουργώντας μία αρχιτεκτονική τύπου client – middleware – server. Το middleware ουσιαστικά ακούει τα αιτήματα των εφαρμογών, τα εκτελεί αλληλεπιδρώντας με τον server και επιστρέφει τα αποτελέσματα στον client. Με αυτόν τον τρόπο πελάτης και διακομιστής δεν σπαταλούν πόρους και χρόνο συστήματος για την επικοινωνία μεταξύ τους, δίνοντας έτσι (κυρίως στις εφαρμογές client) την δυνατότητα να τρέχουν άλλες εργασίες ταυτόχρονα, ενώ περιμένουν απάντηση από τον server. Αυτή η μορφή ασύγχρονης επικοινωνίας βελτιστοποιεί την υπολογιστική απόδοση του συστήματος. Ο σχεδιασμός της εφαρμογής παρουσιάζεται στην εικόνα 7.

Η εφαρμογή LibrApp μπορεί να ακολουθήσει πολλά σενάρια. Αυτά τα σενάρια μπορεί να κατηγοριοποιηθούν σε 4 υπερκατηγορίες: Απλές ενέργειες χρήστη, συνεργατικές ενέργειες, ενέργειες διαχείρισης καθηκόντων και ενέργειες διαχειριστή. Παρακάτω γίνεται ανάλυση των περισσότερων από αυτά τα σενάρια.

Απλές ενέργειες χρήστη: Σε αυτή την κατηγορία ανήκουν τα σενάρια τα οποία

συσχετίζονται με τις απλές ενέργειες που μπορεί να εκτελέσει ένας χρήστης που χρησιμοποιεί την LibrApp. Ο χρήστης ενεργοποιεί την εφαρμογή μέσω της κινητής συσκευής του. Στην συνέχεια μπορεί να επιλέξει να κάνει αναζήτηση στη βάση δεδομένων της βιβλιοθήκης. Σε αυτή την περίπτωση πληκτρολογεί το αντικείμενο της αναζήτησής του και αποστέλλει ένα αίτημα στο ενδιάμεσο λογισμικό. Αυτό με την σειρά του εκτελεί την αναζήτηση στην βάση δεδομένων και επιστρέφει τα αποτελέσματα στον χρήστη. Επίσης μπορεί να επιλέξει να εγγραφεί στην εφαρμογή. Σε αυτή την περίπτωση συμπληρώνει μία φόρμα η οποία αποστέλλεται στο middleware και αυτό φροντίζει να αποθηκεύσει τις πληροφορίες στην βάση δεδομένων και να επιστρέψει ένα μήνυμα επιτυχίας στην εφαρμογή, η οποία με την σειρά της θα οδηγήσει τον χρήστη στην οθόνη εγγεγραμμένων χρηστών. Αν ο χρήστης διαθέτει λογαριασμό στην εφαρμογή, μπορεί να επιλέξει να κάνει είσοδο με το e-mail και τον κωδικό του. Σε αυτή την περίπτωση, η εφαρμογή θα αποστείλει ένα αίτημα ταυτοποίησης, το middleware θα το εκτελέσει και θα επιστρέψει το αποτέλεσμα στην εφαρμογή. Αν η ταυτοποίηση ήταν επιτυχής, ο χρήστης θα οδηγηθεί στην οθόνη εγγεγραμμένων χρηστών. Από εκεί ο χρήστης μπορεί να πραγματοποιήσει μία αναζήτηση όπως και στην περίπτωση των μη εγγεγραμμένων χρηστών, με την διαφορά ότι μπορεί να πραγματοποιήσει κάποιες έξτρα λειτουργίες. Ο χρήστης μπορεί να αποθηκεύσει το αντικείμενο που βρήκε στο προσωπικό του ράφι. Σε αυτή την περίπτωση αποστέλλεται στο ενδιάμεσο λογισμικό μία αίτηση που περιλαμβάνει τον κωδικό του χρήστη και τα στοιχεία του αντικειμένου. Έπειτα, το ενδιάμεσο λογισμικό θα προσθέσει το αντικείμενο στον πίνακα της βάσης δεδομένων που αντιστοιχεί στο ψηφιακό ράφι του χρήστη. Επίσης ο χρήστης μπορεί να σβήσει αντικείμενα από το ράφι του, λειτουργία η οποία εκτελείται με αντίστοιχο τρόπο. Επίσης ένας χρήστης μπορεί να μοιραστεί ένα αντικείμενο που αναζήτησε σε κάποιο κοινωνικό δίκτυο. Αυτή είναι μια ενέργεια που εκτελείται μόνο από την εφαρμογή χωρίς την χρήση του διακομιστή ή του ενδιάμεσου λογισμικού. Ακόμα ένας συνδεδεμένος χρήστης μπορεί να κατεβάσει στην κινητή του συσκευή κάποιο αντικείμενο, δεδομένου ότι αυτό διαθέτει την απαραίτητη άδεια χρήσης. Σε αυτή την περίπτωση αποστέλλει μία αίτηση στο ενδιάμεσο λογισμικό και αυτό συνδέεται με τον file server στον οποίο βρίσκεται το φυσικό αρχείο, και το αποστέλλει στην συσκευή του χρήστη. Αυτή η ενέργεια εκτελείται στο παρασκήνιο και δεν εμποδίζει τον χρήστη από το να πραγματοποιεί άλλες ενέργειες. Το διάγραμμα ακολουθίας αυτής της κατηγορίας εμφανίζεται στον πίνακα 5.

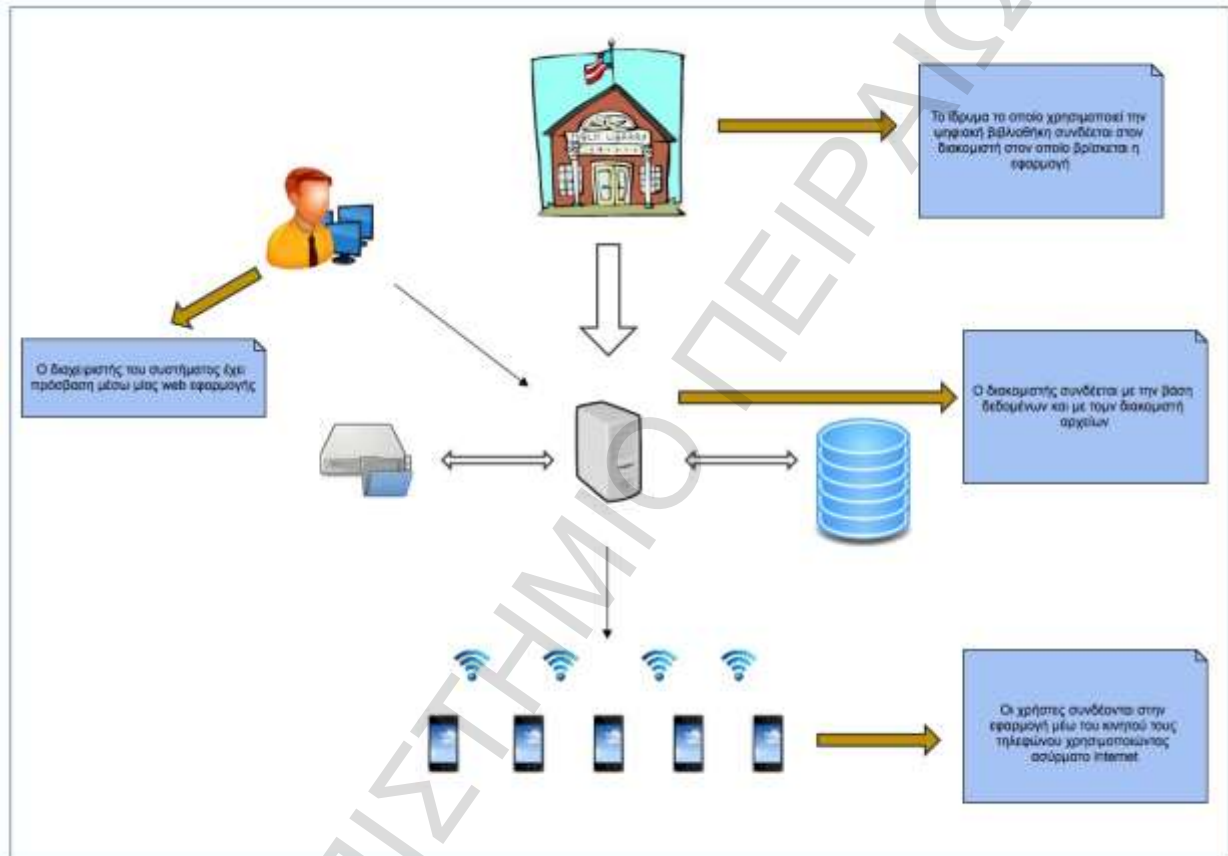
Συνεργατικές ενέργειες: Σε αυτή την κατηγορία ανήκουν τα σενάρια αυτά τα οποία μπορεί να πραγματοποιήσει ο χρήστης για να εκτελέσει τις ενέργειες, οι οποίες αποσκοπούν στην αύξηση της συνεργατικότητάς τους και στην σύνδεσή του με τους άλλους χρήστες. Πιο συγκεκριμένα ο χρήστης μπορεί να εκτελέσει ένα από τα παρακάτω σενάρια. Ο χρήστης μπορεί μέσα από την εφαρμογή να αναζητήσει φίλους ή γνωστούς που χρησιμοποιούν την εφαρμογή, κάνοντας browse μέσα στους χρήστες της εφαρμογής ή πληκτρολογώντας ένα συγκεκριμένο αναγνωριστικό χρήστη ή e-mail. Σε αυτή την περίπτωση οι πληροφορίες σχετικά με τους χρήστες δίνονται από το ενδιάμεσο λογισμικό στην εφαρμογή, αφού προηγηθεί αλληλεπίδραση με την βάση δεδομένων. Αν ο χρήστης επιθυμεί να συνδεθεί με κάποιον αποστέλλει ένα αίτημα φιλίας, το οποίο το επεξεργάζεται το ενδιάμεσο λογισμικό και αν όλα είναι σωστά, το αποστέλλει στην συσκευή του αποδέκτη. Ο αποδέκτης θα λάβει μία ενημέρωση σχετικά με το αίτημα φιλίας και θα πρέπει να το αποδεχτεί ή όχι. Η απάντηση του αποδέκτη, θα διαβιβαστεί από την εφαρμογή στο ενδιάμεσο λογισμικό και από εκεί στην συσκευή του αρχικού χρήστη που ξεκίνησε το αίτημα φιλίας. Αν η απάντηση ήταν θετική, τότε ο χρήστης θα μπορεί να δει τον αποδέκτη στην λίστα φίλων του και να αλληλεπιδράσει μαζί του με διάφορους τρόπους. Ένας από αυτούς αποτελεί το σενάριο επικοινωνίας (chat). Σε αυτό το σενάριο ο χρήστης πληκτρολογεί ένα μήνυμα προς τον αποδέκτη και αυτό μεταβιβάζεται στην συσκευή του αποδέκτη με την βοήθεια του ενδιάμεσου λογισμικού. Στην συνέχεια ο αποδέκτης λαμβάνει μία ειδοποίηση για το μήνυμα και μπορεί να απαντήσει σε αυτό με ένα δικό του, εκτελώντας παρόμοια διαδικασία με του αποστολέα. Ένα ακόμα σενάριο που μπορεί κάποιος να εκτελέσει και ανήκει σε αυτήν την κατηγορία είναι η δημιουργία ενός κοινόχρηστου έργου (project). Ο χρήστης μπορεί να δημιουργήσει ένα κοινόχρηστο project σε μορφή παρόμοια με κάποιο wiki. Δίνει στο project όνομα και περιγράφει και διαλέγει τα άτομα με τα οποία θέλει να το κάνει κοινόχρηστο. Στην συνέχεια, το αίτημα του χρήστη αποστέλλεται στο middleware, το οποίο αποθηκεύει τα στοιχεία του project στην βάση δεδομένων, και προχωράει στην αποστολή αιτήσεων για αποδοχή στις συσκευές των εμπλεκόμενων χρηστών. Οι αποδέκτες των αιτημάτων λαμβάνουν μία ειδοποίηση στην κινητή συσκευή τους και μπορούν είτε να την

αποδεχτούν είτε να την αρνηθούν. Μόλις τελειώσει ο γύρος των αποδοχών και αποθηκευτούν τα στοιχεία των εμπλεκόμενων στην βάση δεδομένων (δουλειά του middleware), τότε το project είναι έτοιμο για χρήση. Οι χρήστες μπορούν να το τροποποιήσουν αποστέλλοντας αιτήσεις στο middleware για αποθήκευση των αλλαγών τους, είτε να το επαναφέρουν σε κάποια προηγούμενη έκδοση. Ακόμα, ο χρήστης μπορεί να εκτελέσει το σενάριο στο οποίο αποστέλλει σε κάποιον από τους συνδεδεμένους φίλους του ένα αντικείμενο αποθηκευμένο στην κινητή του συσκευή. Στην περίπτωση αυτή, ο χρήστης επιλέγει το αντικείμενο με την χρήση ενός browser, διαλέγει παραλήπτη και πατάει το κουμπί αποστολής. Το αίτημα αποστέλλεται στο ενδιάμεσο λογισμικό το οποίο θα αποστείλει στον παραλήπτη ένα αίτημα αποδοχής του αντικειμένου. Αν ο αποδέκτης πατήσει αποδοχή, τότε θα ξεκινήσει η μεταβίβαση του αντικειμένου (ενέργεια που εκτελείται στο παρασκήνιο). Το διάγραμμα ακολουθίας αυτής της κατηγορίας εμφανίζεται στον πίνακα 6.

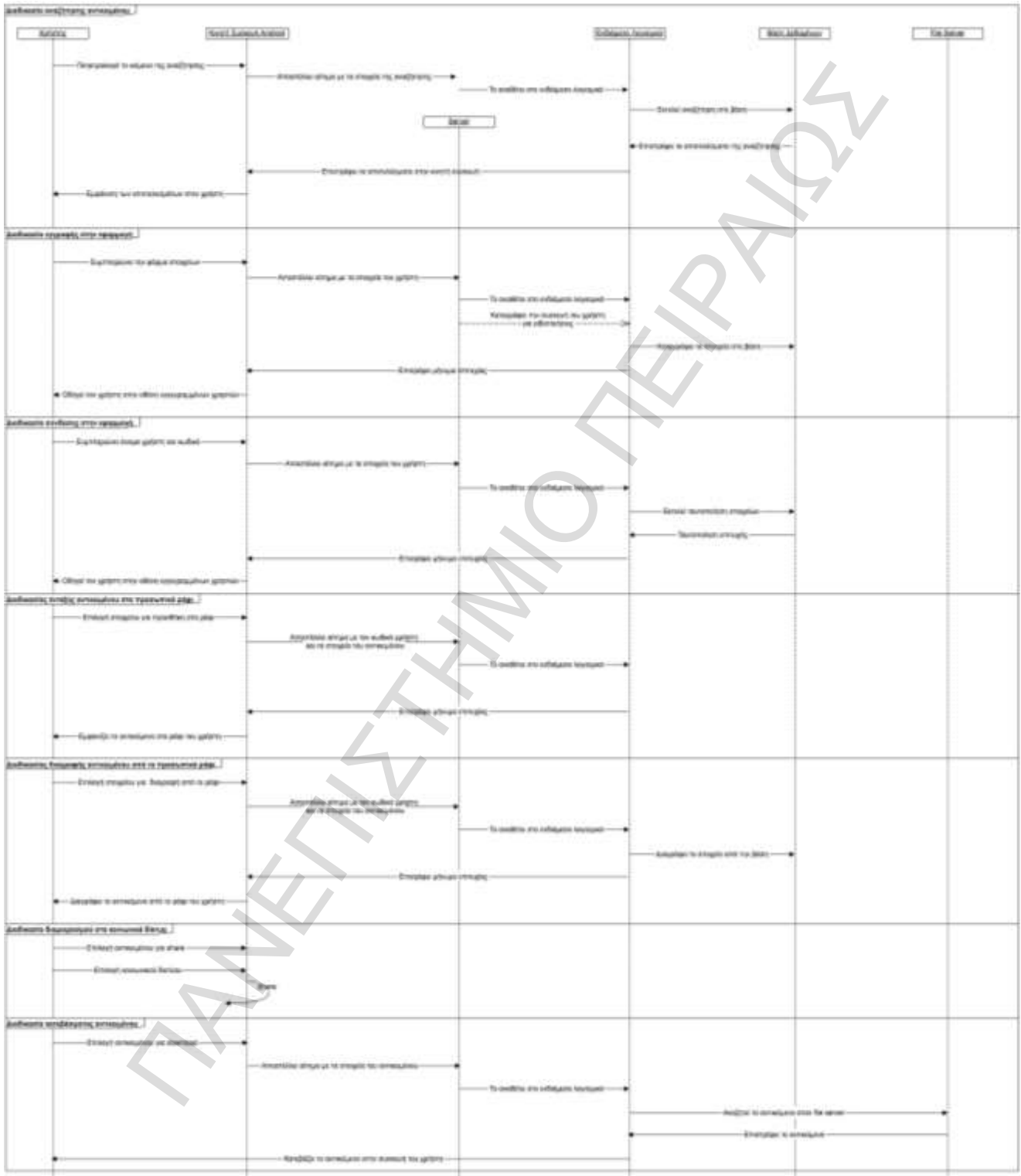
Ενέργειες διαχείρισης καθηκόντων: Σε αυτή την κατηγορία ανήκουν τα σενάρια τα οποία σχετίζονται με την λειτουργία διαχείρισης καθηκόντων που προσφέρει η εφαρμογή LibrApp. Ο χρήστης μπορεί μέσα από το περιβάλλον της εφαρμογής να δημιουργήσει μία νέα λίστα καθηκόντων, να προσθέσει καθήκοντα (ή αντικείμενα τύπου to-do όπως αναφέρονται στην εφαρμογή), και να τα μοιράσει σε συνεργάτες με τους οποίους έχει ήδη συνδεθεί. Ακόμα μπορεί να διαχειριστεί υπάρχουσες λίστες (ενεργές ή μη) και να εκτελέσει διάφορες ενέργειες. Αναλυτικά, ο χρήστης μέσα από την εφαρμογή επιλέγει την δημιουργία της νέας λίστας και της δίνει ένα όνομα, μία περιγραφή και ένα μέγιστο χρόνο υλοποίησης. Στην συνέχεια, δημιουργεί καθήκοντα ή to-do, δίνοντας τους όνομα, χρονοδιάγραμμα και συσχετίζοντας τα με τον εαυτό του ή άλλους χρήστες της ψηφιακής βιβλιοθήκης. Όταν ο χρήστης πατήσει δημιουργία, τότε αποστέλλεται μία αίτηση προς το ενδιάμεσο λογισμικό το οποίο καταγράφει την λίστα καθηκόντων στην βάση δεδομένων και προχωράει στην αποστολή αιτημάτων προς του χρήστες οι οποίοι επιλέχθηκαν να συσχετιστούν με τα καθήκοντα της λίστας. Οι συσχετισμένοι χρήστες λαμβάνουν τις ειδοποιήσεις στην κινητή τους συσκευή και μπορούν είτε να αποδεχτούν την πρόσκληση είτε να την αγνοήσουν. Αν οι χρήστες αποδεχτούν την πρόσκληση τότε το middleware αποστέλλει την λίστα καθηκόντων στην συσκευή του αποδέκτη και η εφαρμογή την αποθηκεύει τοπικά. Ο αποδέκτης βλέπει τα to-dos που τον αφορούν με διαφορετικό χρώμα και έχει την δυνατότητα να τους προσθέσει σχόλια, καθώς και να τα δηλώσει αποπερατωμένα. Ο χρήστης-διαχειριστής της λίστας μπορεί να αποστείλει ενημερώσεις προς τους εμπλεκόμενους, οι οποίες θα διαμοιραστούν σε αυτούς μέσω του ενδιάμεσου λογισμικού. Ακόμα, ο χρήστης-διαχειριστής μπορεί να διαγράψει μία λίστα καθηκόντων. Αυτή η ενέργεια θα αναγκάσει το ενδιάμεσο λογισμικό να αποστείλει αιτήσεις προς τις κινητές συσκευές των εμπλεκόμενων, οι οποίες θα προχωρήσουν στην ενημέρωση των χρηστών ότι η λίστα διαγράφηκε και μετά θα την αφαιρέσουν από την εφαρμογή. Κάθε φορά που κάποιος χρήστης δηλώνει ένα αντικείμενο to-do ως ολοκληρωμένο, αποστέλλεται μία αίτηση στο ενδιάμεσο λογισμικό, το οποίο ενημερώνει αντιστοίχως την βάση δεδομένων και στη συνέχεια αποστέλλει μία ειδοποίηση στον χρήστη-διαχειριστή. Όταν μία λίστα καθηκόντων ολοκληρωθεί, όταν δηλαδή όλα τα to-do αντικείμενα δηλωθούν ως ολοκληρωμένα από τους χρήστες στους οποίους είχαν ανατεθεί, τότε αποστέλλεται μία ενημέρωση από το ενδιάμεσο λογισμικό προς την κινητή συσκευή του χρήστη-διαχειριστή ότι η λίστα ολοκληρώθηκε. Ο χρήστης διαχειριστής μπορεί τότε είτε να το αποδεχτεί, αποστέλλοντας έτσι μηνύματα προς όλους τους εμπλεκόμενους, είτε να προσθέσει άμεσα και άλλα αντικείμενα to-do και να τα συσχετίσει με χρήστες. Το διάγραμμα ακολουθίας αυτής της κατηγορίας εμφανίζεται στον πίνακα 7.

Ενέργειες διαχειριστή: Σε αυτή την κατηγορία ανήκουν τα σενάρια τα οποία εκτελεί ο διαχειριστής/βιβλιοθηκονόμος της ψηφιακής βιβλιοθήκης μέσα από το web περιβάλλον που του προσφέρεται. Σε αυτά τα σενάρια συμπεριλαμβάνονται τα σενάρια διαχείρισης αντικειμένων (βιβλίων, ψηφιακού υλικού κλπ) και τα σενάρια διαχείρισης χρηστών. Αναλυτικότερα, ο διαχειριστής μπορεί να προσθέσει ένα αντικείμενο στην βάση δεδομένων και το αποθετήριο της βιβλιοθήκης, συμπληρώνοντας τα απαραίτητα πεδία τα οποία αποτελούν τα στοιχεία του αντικειμένου (τίτλος, περιγραφή και άλλα) και να ανεβάσει το ίδιο το αρχείο στον file server του συστήματος της ψηφιακής βιβλιοθήκης. Όταν ο διαχειριστής πατήσει αποθήκευση, το ενδιάμεσο λογισμικό θα εκτελέσει τις απαραίτητες ενέργειες για την ενημέρωση της βάσης δεδομένων και για την προσθήκη του αντικειμένου στον file server. Ακόμα, ο διαχειριστής μπορεί να τροποποιήσει ή και να διαγράψει πλήρως ένα αντικείμενο από την ψηφιακή βιβλιοθήκη. Σε αυτή την περίπτωση το ενδιάμεσο λογισμικό θα εκτελέσει τις αλλαγές στην βάση δεδομένων και στον

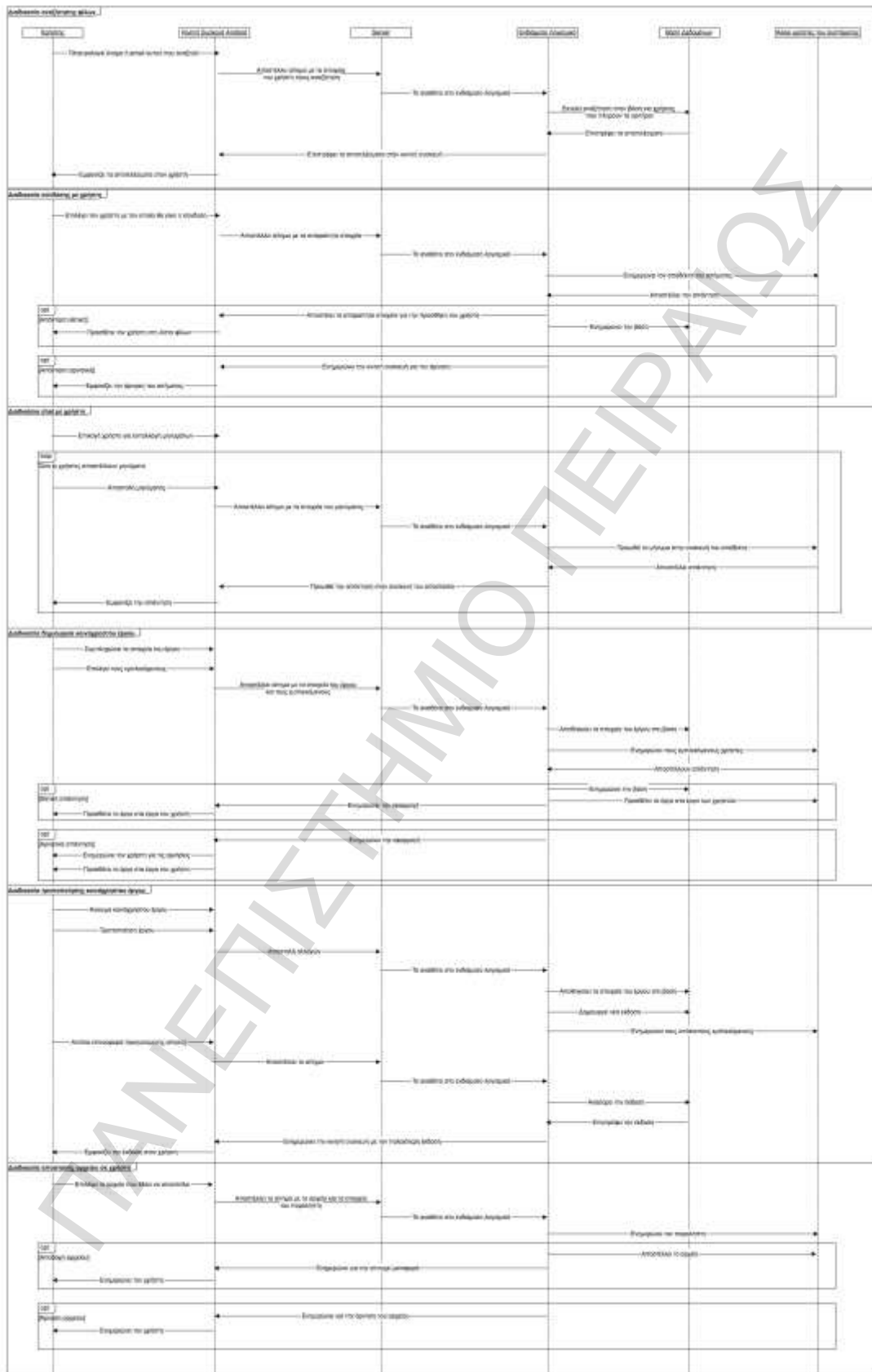
file server του συστήματος, αν αυτό είναι απαραίτητο. Στην περίπτωση διαχείρισης χρήστη, ο διαχειριστής μπορεί να εκτελέσει παρόμοιες ενέργειες, δηλαδή διαγραφή, προσθήκη και ενημέρωση χρήστη (πχ αλλαγή αναγνωριστικού ή αλλαγών στοιχείων, ή επαναφορά κωδικού). Ο διαχειριστής διαβιβάζει τα αιτήματα του στο ενδιάμεσο λογισμικό και αυτό με τη σειρά εκτελεί τις απαραίτητες ενέργειες στην βάση δεδομένων, και ενημερώνει τον χρήστη στο οποίο το προφίλ έγιναν οι αλλαγές, αν αυτό είναι αναγκαίο. Το διάγραμμα ακολουθίας αυτής της κατηγορίας εμφανίζεται στον πίνακα 8.



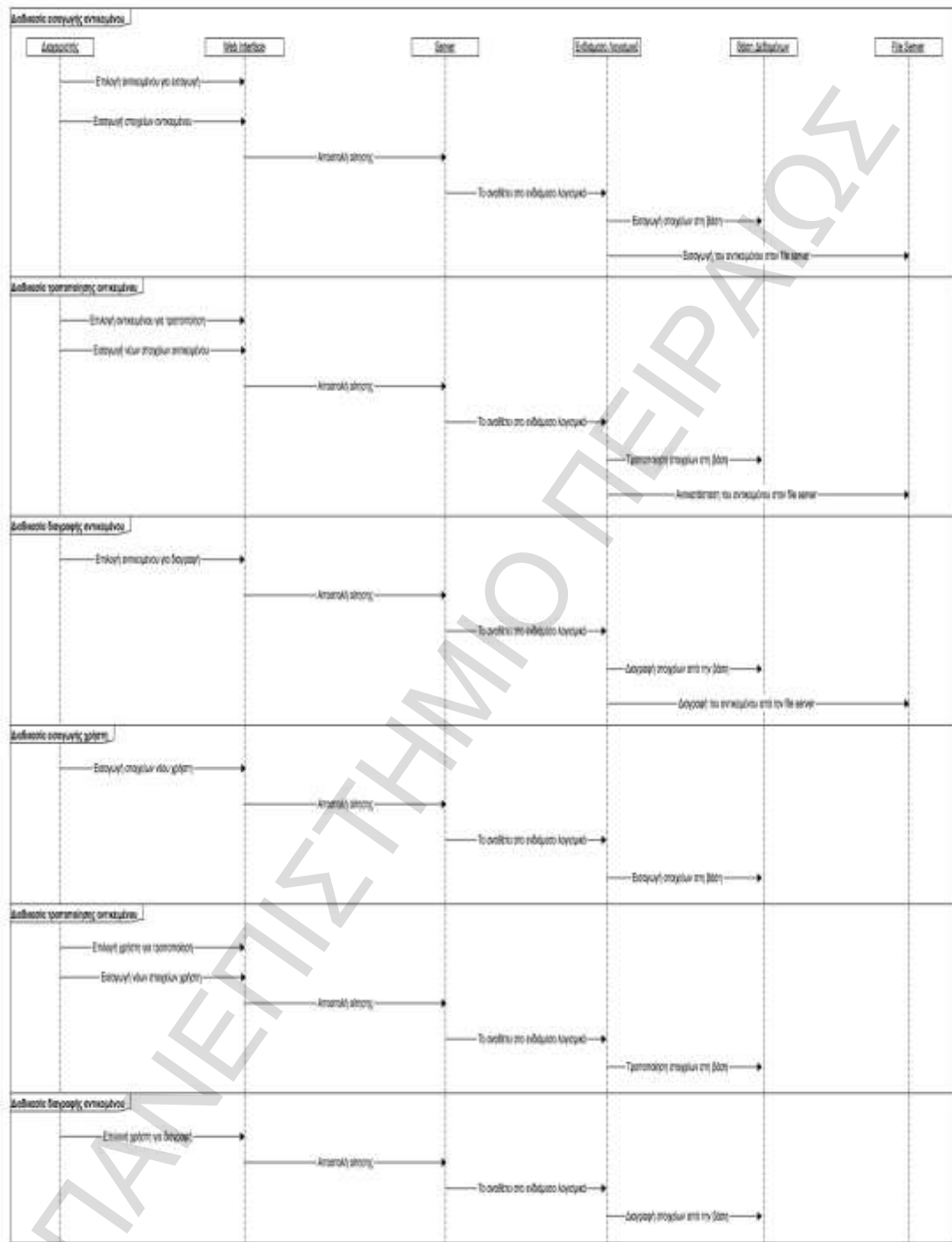
Εικόνα 7 Η αρχιτεκτονική του συστήματος



Πίνακας 5 Διάγραμμα ακολουθίας απλών ενεργειών χρήστη



Πίνακας 6 Διάγραμμα ακολουθίας συνεργατικών ενεργειών



Πίνακας 8 Διάγραμμα ακολουθίας ενεργειών διαχειριστή

5.Η εφαρμογή LibrApp

Σε αυτή την ενότητα θα αναφέρουμε πώς χρησιμοποιήθηκαν οι προαναφερθείσες τεχνολογίες στην εφαρμογή LibrApp, τόσο στην μεριά του client όσο και στην μεριά του server. Στην συνέχεια θα γίνει παρουσίαση της λειτουργίας της εφαρμογής, χρησιμοποιώντας μία πληθώρα από ενδεικτικές οθόνες (screenshots).

5.1.Χρήση των τεχνολογιών

5.1.1.Client

Για τις κινητές συσκευές των χρηστών της ψηφιακής βιβλιοθήκης χρησιμοποιήθηκε το λειτουργικό σύστημα Android, και για την ακρίβεια το Android API 18 (Android 4.3). Το λειτουργικό σύστημα Android παρέχει πολλές δυνατότητες και εργαλεία τα οποία χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής που εκτελείται στις κινητές συσκευές των χρηστών. Ένα από αυτά τα εργαλεία είναι η βάση δεδομένων SQLite, η οποία έρχεται μαζί με το Android για την αποθήκευση δεδομένων τοπικά στην συσκευή. Στην περίπτωση της LibrApp, η SQLite χρησιμοποιείται για να αποθηκεύσει τα δεδομένα του χρήστη, να κρατήσει τα στοιχεία του, κρατώντας τον έτσι συνδεδεμένο με τον απομακρυσμένο server, να αποθηκεύει πληροφορίες σχετικά με αντικείμενα που αφορούν τον χρήστη και γενικά να κρατάει πληροφορία τοπικά, ούτως ώστε η εφαρμογή να μην χρειάζεται να κάνει συνεχείς κλήσεις στον διακομιστή. Για την επικοινωνία με τον διακομιστή, χρησιμοποιήθηκε ένα ακόμη εργαλείο, το JJson, με την μορφή ενός Android JsonParser, το οποίο μεταφράζει αιτήματα JJson από και προς τον server. Ακόμα για την αποστολή ειδοποιήσεων προς την εφαρμογή του χρήστη (για αιτήματα φιλίας και λοιπές ενημερώσεις προς τον χρήστη), χρησιμοποιήθηκε η υπηρεσία Google Cloud Messaging της Google (GCM), η οποία αντικατέστησε το παλαιότερο C2DM, και η οποία χρησιμοποιείται για να «σπρώξει» ειδοποιήσεις από τον server στην κινητή συσκευή (push service). Για την χρήση αυτής της τεχνολογίας απαιτείται δήλωση της εφαρμογής στην Google (μέσω του Google Developer⁷), ώστε να γίνει λήψη του απαραίτητου αναγνωριστικού εφαρμογής που θα χρησιμοποιηθεί για την αποστολή των ειδοποιήσεων και μηνυμάτων.

5.1.2.Server

Στον απομακρυσμένο διακομιστή της εφαρμογής βρίσκονται τέσσερις οντότητες. Η βάση δεδομένων, το ενδιάμεσο λογισμικό (middleware), η εφαρμογή των διαχειριστών και ένας διακομιστής αρχείων σε μορφή ftp. Το ενδιάμεσο λογισμικό είναι γραμμένο στην γλώσσα διαδικτυακού προγραμματισμού PHP και κάνει χρήση των τεχνολογιών Ajax και Jjson, για να πραγματοποιήσει την επικοινωνία με την MySQL βάση δεδομένων, τον ftp server και την android εφαρμογή. Με την χρήση αυτών των τεχνολογιών το middleware επικοινωνεί με την βάση δεδομένων και την εφαρμογή και μεταβιβάζει πληροφορίες από την μία στην άλλη. Η εφαρμογή των διαχειριστών είναι επίσης γραμμένη στην γλώσσα PHP και χρησιμοποιεί ακόμα το JQuery, την βιβλιοθήκη της γλώσσας Javascript που αναλύθηκε σε προηγούμενη ενότητα. Η jquery βοηθάει στην εύκολη δημιουργία γραφικής διεπαφής, μαζί με την χρήση CSS, και στον έλεγχο και ανταπόκριση στις ενέργειες των χρηστών (για παράδειγμα, το πάτημα ενός κουμπιού). Και εδώ χρησιμοποιείται η τεχνολογία ajax για την εκτέλεση ταυτόχρονων λειτουργιών.

⁷ <https://developers.google.com>

5.2.Περιγραφή της λειτουργίας του προγράμματος

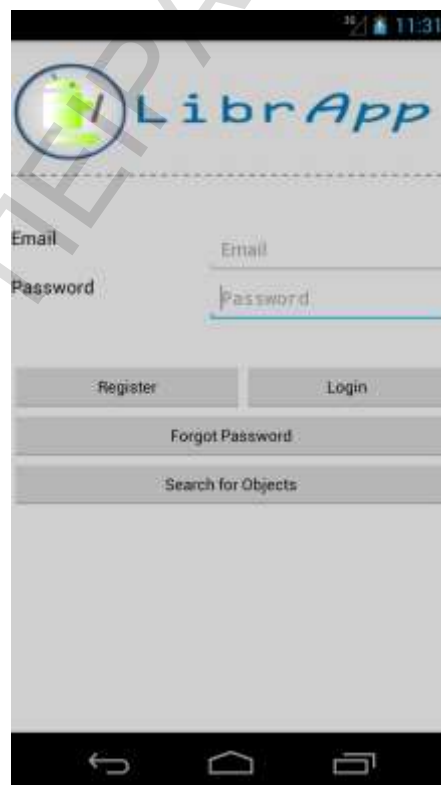
5.2.1.Η εφαρμογή LibrApp στην κινητή συσκευή

Στην παρούσα υποενότητα θα γίνει μια αναλυτική παρουσίαση των λειτουργιών της εφαρμογής που υλοποιήθηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής, με τη μορφή screenshots (στιγμιότυπα οθόνης). Η παρουσίαση αυτή λειτουργεί και ως εγχειρίδιο χρήσης της εφαρμογής.

Κατά την εκκίνηση της εφαρμογής από τον χρήστη εμφανίζεται για μερικά δευτερόλεπτα στην οθόνη της συσκευής το λογότυπο της εφαρμογής, έως ότου το σύστημα ολοκληρώσει την φόρτωση των απαραίτητων στοιχείων για την λειτουργία της εφαρμογής (Εικόνα 8).



Εικόνα 9 Οθόνη έναρξης εφαρμογής



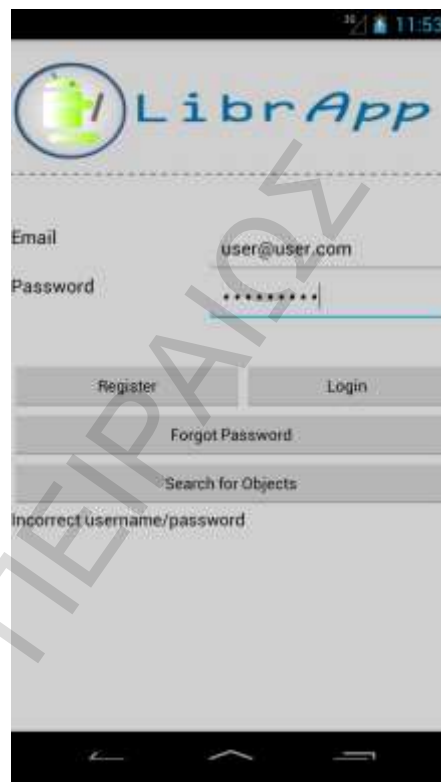
Εικόνα 8 Αρχική οθόνη

Στη συνέχεια ο χρήστης οδηγείται στην αρχική οθόνη (Εικόνα 9), στην οποία μπορεί να πραγματοποιήσει τις ακόλουθες εργασίες:

- Ήδη εγγεγραμμένος χρήστης, μπορεί να εισάγει το αναγνωριστικό του και τον κωδικό του στα αντίστοιχα πεδία και να εισέλθει στην εφαρμογή. (Εικόνα 10)
- Ήδη εγγεγραμμένος χρήστης, ο οποίος έχει ξεχάσει τον κωδικό του, μπορεί να εκκινήσει τη δυνατότητα επανάκτησης του κωδικού.
- Νέος χρήστης που επιθυμεί να δημιουργήσει λογαριασμό, μπορεί να εκκινήσει την αντίστοιχη λειτουργία (Register).
- Μη εγγεγραμμένος χρήστης που απλά επιθυμεί να περιηγηθεί στο περιεχόμενο της βιβλιοθήκης, μπορεί να το πραγματοποιήσει μέσω της αντίστοιχης επιλογής (Search for Objects).



Εικόνα 11 Εισαγωγή στοιχείων εγγεγραμμένου χρήστη



Εικόνα 10 Μήνυμα λάθους e-mail ή κωδικού

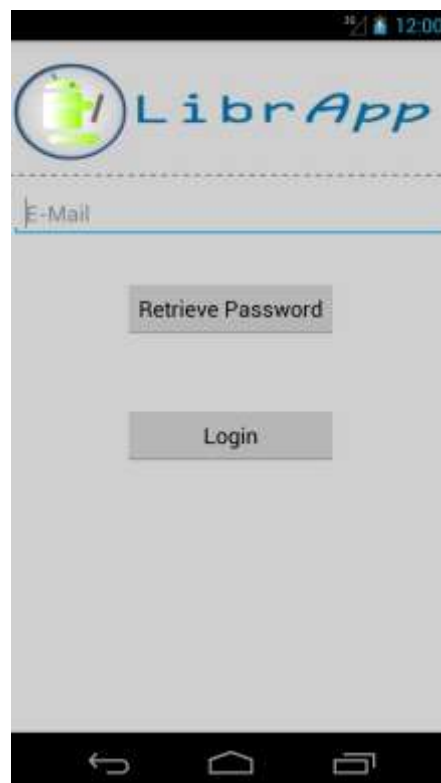
Στην περίπτωση που ο χρήστης πληκτρολογήσει λάθος είτε τον κωδικό του είτε το αναγνωριστικό του, τότε οδηγείται στην οθόνη που φαίνεται στην Εικόνα 11.

Για την ανάκτηση του κωδικού ο χρήστης θα επιλέξει το κουμπί ανάκτησης (Forgot Password) και θα οδηγηθεί στην Εικόνα 12, όπου και καλείται να συμπληρώσει την διεύθυνση ηλεκτρονικού ταχυδρομείου του (e-mail) στην οποία θα λάβει τον νέο του κωδικό κι σχετικές οδηγίες. Στην οθόνη της εφαρμογής ενημερώνεται ο χρήστης ότι το σχετικό μήνυμα έχει αποσταλεί (Εικόνα 13) και το e-mail που θα λάβει έχει ως κάτωθι:

“Hello User,

Your Password is successfully changed. Your new Password is cplv3hou. Login with your new Password and change it in the User Panel.

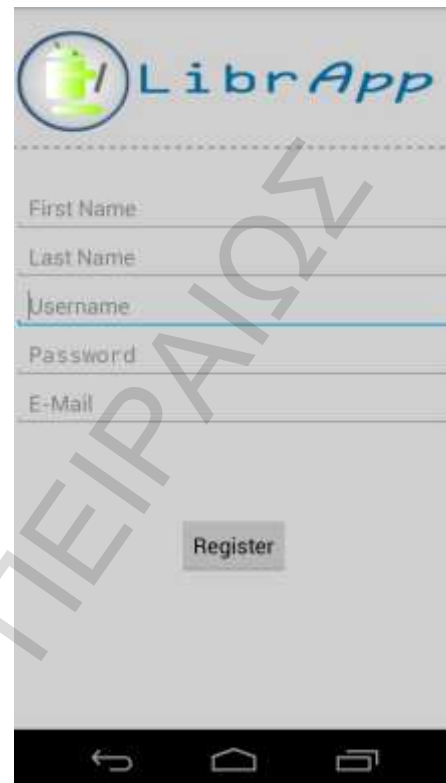
Regards,
LibrApp Team.”



Εικόνα 12 Ζήτηση ηλεκτρονικής διεύθυνσης για ανάκτηση κωδικού



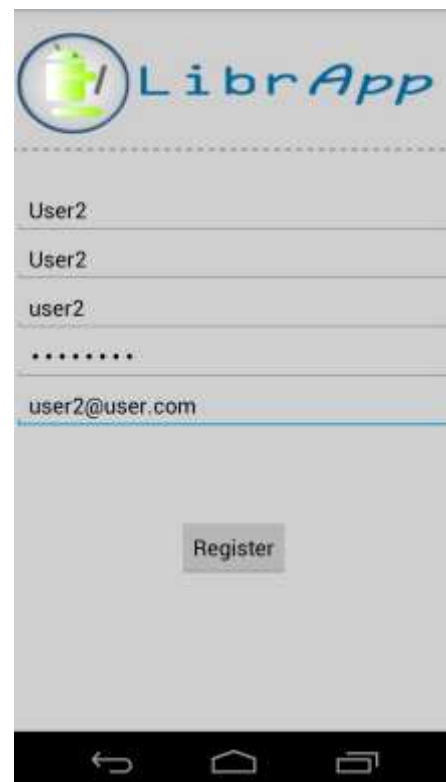
Εικόνα 13 Ενημέρωση αποστολή e-mail



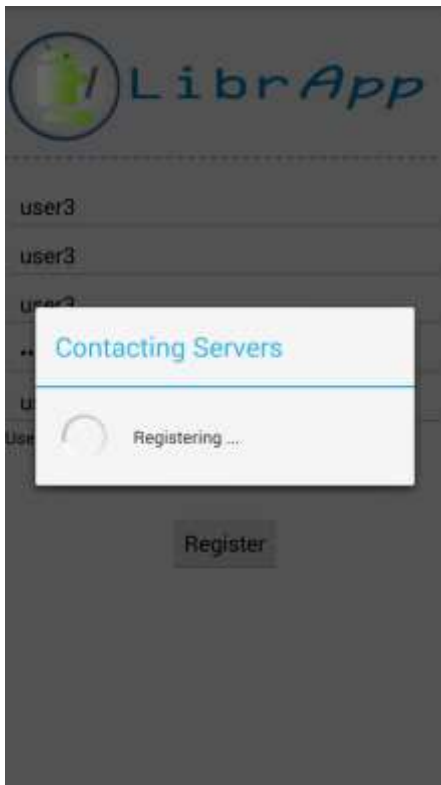
Εικόνα 14 Οθόνη εγγραφής νέου χρήστη

Η επόμενη περίπτωση είναι αυτή στην οποία ο χρήστης επιθυμεί να δημιουργήσει έναν νέο λογαριασμό στην εφαρμογή. Μετά την επιλογή Register οδηγείται στην οθόνη της Εικόνας 14 όπου και θα πρέπει να συμπληρώσει όλα τα απαραίτητα πεδία (ονοματεπώνυμο, όνομα χρήστη, κτλ) όπως φαίνεται στην Εικόνα 15.

Στην Εικόνα 16 φαίνεται η επικοινωνία της εφαρμογής με τον server(διακομιστή) για την καταχώρηση των στοιχείων του νέου χρήστη. Αν υπάρξει κάποιο σφάλμα στα στοιχεία του χρήστη θα λάβει μήνυμα λάθους το οποίο θα περιγράφει το πρόβλημα. Έτσι προκύπτουν οι οθόνες για ήδη χρησιμοποιούμενο όνομα χρήστη (Εικόνα 17), για μη αποδεκτό όνομα χρήστη (Εικόνα 18) και για μη αποδεκτό e-mail (Εικόνα 19.)



Εικόνα 15 Συμπληρωμένη φόρμα νέου χρήστη



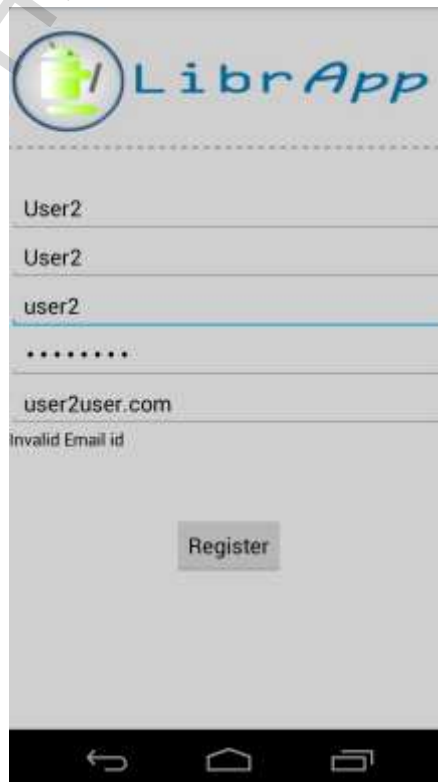
Εικόνα 16 Σύνδεση με server



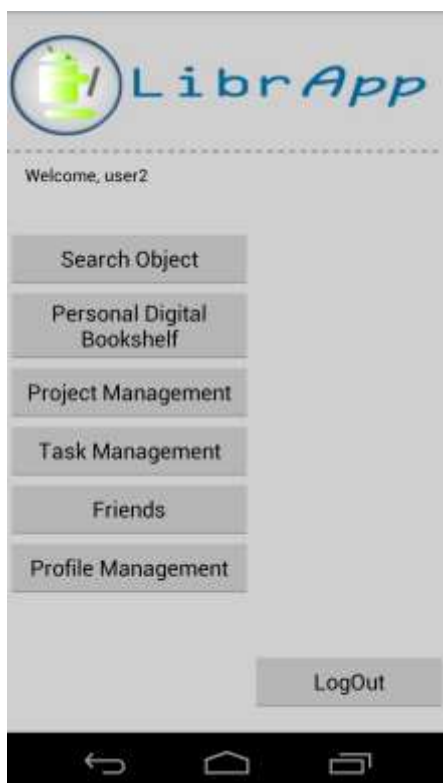
Εικόνα 17 Ήδη χρησιμοποιούμενο όνομα χρήστη



Εικόνα 19 Μη αποδεκτό όνομα χρήστη



Εικόνα 18 Μη αποδεκτό e-mail



Εικόνα 20 Κεντρικό μενού

Με την επιτυχή σύνδεση του χρήστη στην εφαρμογή, είτε για νέο είτε για παλιό χρήστη, εμφανίζεται η Εικόνα 20. Από εδώ ο χρήστης μπορεί να αναζητήσει υλικό στην βιβλιοθήκη, να διαχειριστεί το προσωπικό του «ράφι», τα έργα του, τις λίστες καθηκόντων, τους φίλους και τέλος να επεξεργαστεί το προφίλ του.

Με την επιλογή αναζήτησης, δίδεται στο χρήστη η δυνατότητα να εισάγει τα δεδομένα αναζήτησης στο αντίστοιχο πεδίο (Εικόνα 21) και στη συνέχεια εμφανίζονται στην οθόνη του τα σχετικά αποτελέσματα σε μορφή λίστας. (Εικόνα 22)

Αν ο χρήστης επιλέξει κάποιο από τα αντικείμενα της λίστας αυτό εμφανίζεται σε νέα οθόνη, η οποία περιέχει τα αναλυτικά στοιχεία, όπως αυτή της Εικόνας 23. Στην ίδια οθόνη δίνεται στον χρήστη η δυνατότητα να βαθμολογήσει το αντικείμενο (Εικόνα 24) μέσω ενός συστήματος five star rating.

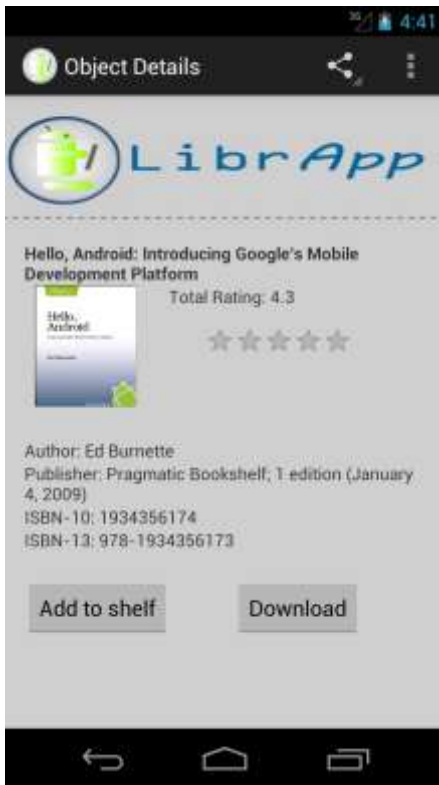
Επίσης ο χρήστης μπορεί να επιλέξει να «κατεβάσει» στη συσκευή του το αντικείμενο μέσω του πλήκτρου download, οπότε και η διαδικασία εκκινεί στο παρασκήνιο της εφαρμογής (Εικόνα 25) και ο χρήστης λαμβάνει ειδοποίηση στο notification bar στην οποία μπορεί να δει αναλυτικά την πρόοδο του download (Εικόνα 26).



Εικόνα 21 Οθόνη αναζήτησης



Εικόνα 22 Αποτελέσματα αναζήτησης



Εικόνα 24 Αναλυτική εμφάνιση αποτελέσματος



Εικόνα 23 Βαθμολόγηση υλικού



Εικόνα 26 Download στο background της εφαρμογής
LibrApp: Μία Android συνεργατική ψηφιακή βιβλιοθήκη.



Εικόνα 25 Πρόσδος download



Εικόνα 28 Share 1



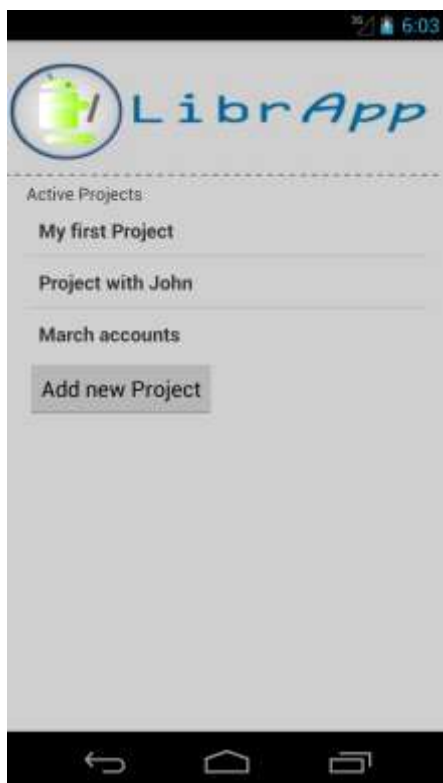
Εικόνα 27 Share 2



Εικόνα 29 Προσωπικό "ράφι"

Στην ίδια οθόνη ο χρήστης μπορεί μέσω του εικονιδίου διαμοιρασμού (share) να κοινοποιήσει το αντικείμενο μέσω ενός κοινωνικού δικτύου (Facebook, twitter, LinkedIn κ.α.) ή άλλου εργαλείου (e-mail, dropbox, drive, κ.α.) (Εικόνα 27 και Εικόνα 28).

Τέλος, αν στη οθόνη που εμφανίζεται το αντικείμενο, επιλέξει ο χρήστης το πλήκτρο για την προσθήκη στο προσωπικό ράφι, οδηγείται στο προσωπικό του ράφι όπου έχει προστεθεί και το αντικείμενο (Εικόνα 29). Στην συγκεκριμένη οθόνη θα οδηγηθεί και στην περίπτωση που επιλέξει το αντίστοιχο πεδίο από το κεντρικό μενού.



Εικόνα 31 Οθόνη διαχείρισης έργου



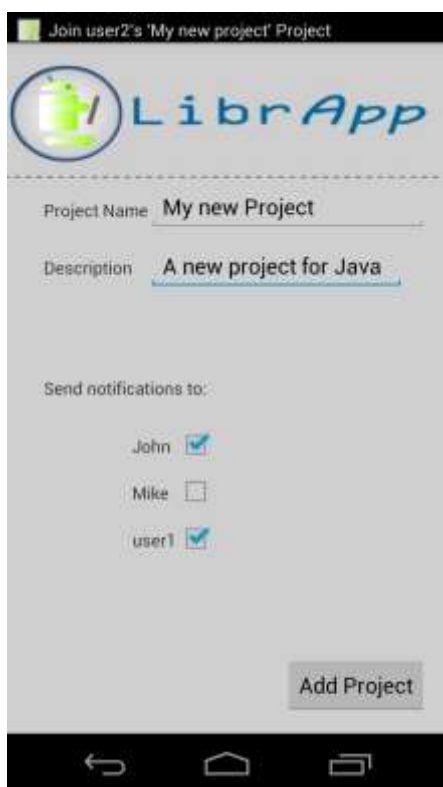
Εικόνα 30 Διαχείριση έργου



Εικόνα 32 Οθόνη προσθήκης νέου έργου

Με την επιλογή διαχείρισης έργου (Project Management) από το κεντρικό μενού, ο χρήστης οδηγείται στην οθόνη της Εικόνας 30.

Από εκεί μπορεί είτε να επιλέξει ένα από τα υπάρχοντα έργα, είτε να δημιουργήσει κάποιο νέο. Στην περίπτωση που επιλέξει ένα από τα υπάρχοντα, θα οδηγηθεί στην οθόνη της Εικόνας 31. Σε αυτήν την οθόνη μπορεί να επεξεργαστεί το έργο το οποίο είναι σε μορφή wiki, να δει ποιοι συμμετέχουν σε αυτό καθώς και να το διαγράψει.



Εικόνα 34 Ειδοποίηση για συμμετοχή σε νέο έργο 1

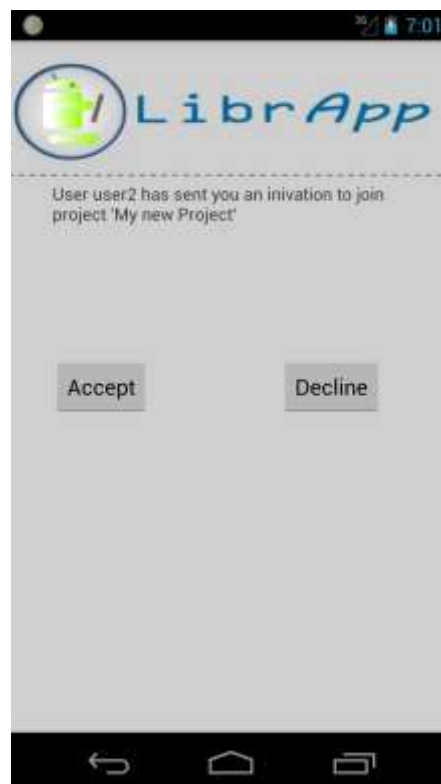


Εικόνα 33 Ειδοποίηση για συμμετοχή σε νέο έργο 2

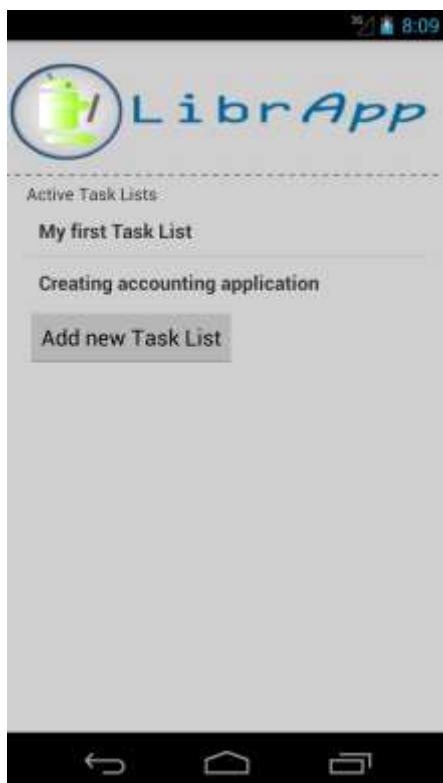
Στην περίπτωση που ο χρήστης επιλέξει να δημιουργήσει ένα νέο έργο, οδηγείται στην οθόνη της Εικόνας 32, όπου και καλείται να συμπληρώσει τα στοιχεία του έργου (όνομα, περιγραφή) και να επιλέξει, εφόσον το επιθυμεί, με ποιους χρήστες θα το μοιραστεί.

Πατώντας το πλήκτρο προσθήκης έργου, θα αποσταλούν ειδοποιήσεις στους αντίστοιχους χρήστες. (Εικόνα 33, Εικόνα 34)

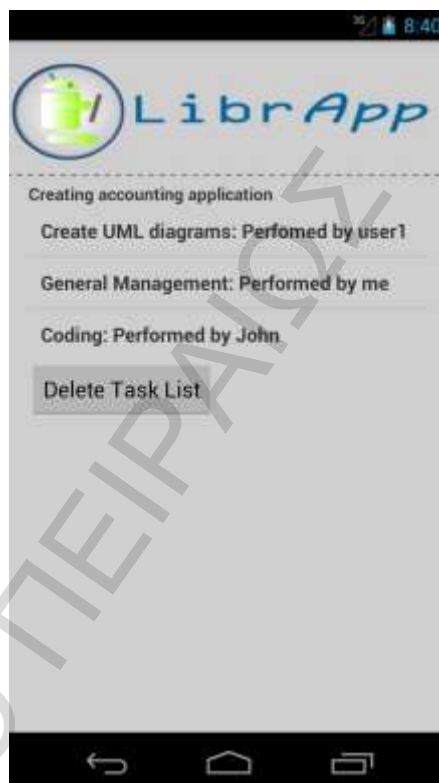
Πατώντας πάνω στην ειδοποίηση της Εικόνας 34 ο αποδέκτης μπορεί να δεχτεί ή να απορρίψει τη συμμετοχή στο εν λόγω έργο. (Εικόνα 35)



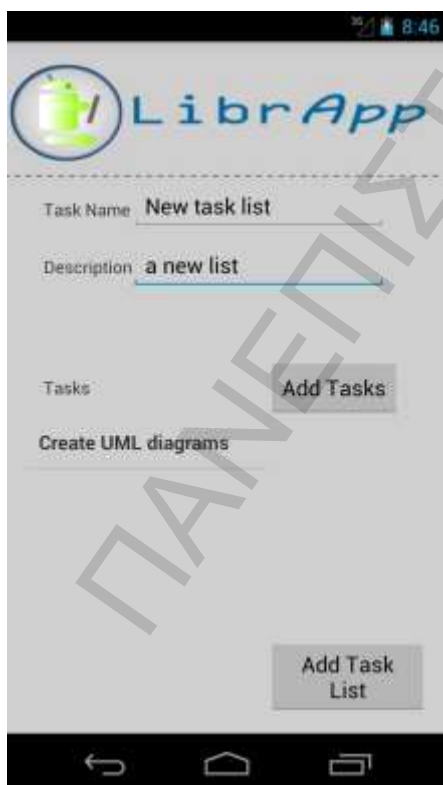
Εικόνα 35 Αποδοχή πρόσκλησης για συμμετοχή σε νέο έργο



Εικόνα 36 Οθόνη διαχείρισης λίστας καθηκόντων



Εικόνα 37 Λίστα καθηκόντων



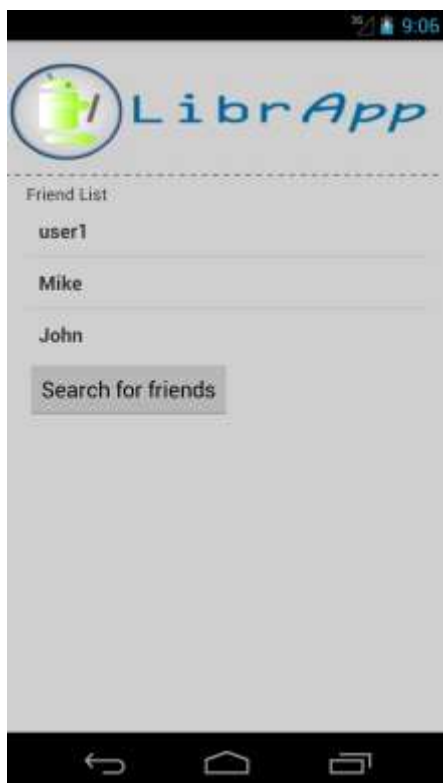
Εικόνα 38 Προσθήκη νέας λίστας καθηκόντων

Αν ο χρήστης επιλέξει την διαχείριση λίστας καθηκόντων (Tasks Management) από το κεντρικό μενού οδηγείται στην οθόνη της Εικόνας 36 όπου και μπορεί να δει τις ήδη υπάρχουσες λίστες καθώς και να προσθέσει μία νέα.

Επιλέγοντας μία από τις λίστες, εμφανίζεται η οθόνη της Εικόνας 37, στην οποία του δίνεται η δυνατότητα προβολής των καθηκόντων και σε ποιους χρήστες έχουν αυτά ανατεθεί, καθώς και η δυνατότητα διαγραφής της λίστας.

Αν ο χρήστης δημιουργήσει μία νέα λίστα, θα οδηγηθεί στην Εικόνα 38 όπου και καλείται να συμπληρώσει τα στοιχεία της νέας λίστας καθώς και να προσθέσει καθήκοντα τα οποία θα αντιστοιχήσει σε χρήστες.

Με την δημιουργία της νέας λίστας θα αποσταλούν ειδοποιήσεις στους αντίστοιχους χρήστες, οι οποίες ακολουθούν την μορφή των ειδοποιήσεων για συμμετοχή σε νέο έργο, που παρουσιάστηκε παραπάνω (Εικόνα 34 και Εικόνα 35)



Εικόνα 40 Οθόνη διαχείρισης φίλων



Εικόνα 39 Οθόνη συνομιλίας με φίλο

Με την επιλογή διαχείρισης φίλων (Friends) ο χρήστης μπορεί να πλοηγηθεί στην οθόνη της Εικόνας 39 στην οποία του παρουσιάζεται η λίστα των χρηστών με τους οποίους είναι συνδεδεμένος.

Ο χρήστης επιλέγοντας έναν από τους φίλους του, οδηγείται σε ένα μενού στο οποίο μπορεί να εκτελέσει λειτουργίες, όπως να αποστείλει αρχείο. Ο παραλήπτης θα λάβει σχετική ειδοποίηση παρόμοια με αυτή της Εικόνας 26.

Ακόμα μπορεί να επιλέξει να εκκινήσει μια σύγχρονη επικοινωνία (chat) της οποίας η λειτουργία εμφανίζεται στην Εικόνα 40.

Τέλος στην επιλογή διαχείρισης προφίλ του κεντρικού μενού, δίδεται η δυνατότητα αλλαγής των στοιχείων του χρήστη σε μία οθόνη παρόμοια με αυτή της αρχικής εγγραφής. (Εικόνα 41)

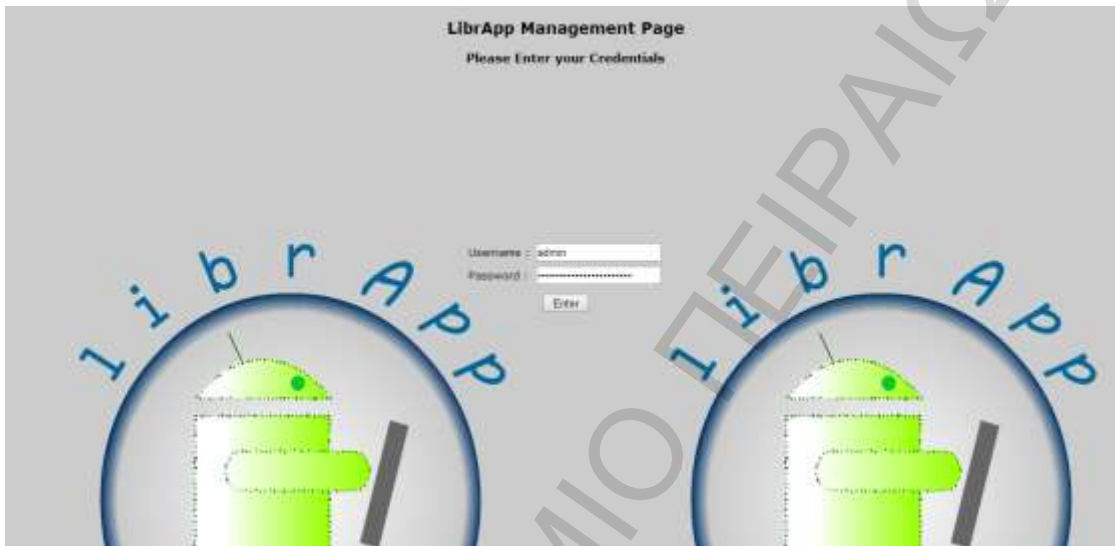


Εικόνα 41 Διαχείριση προφίλ

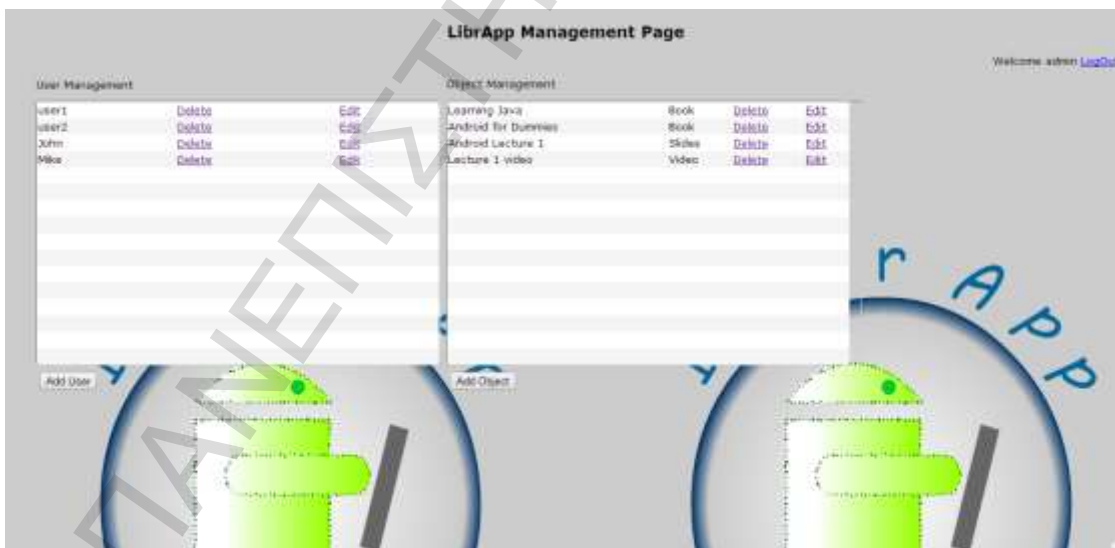
5.2.2. Η εφαρμογή των διαχειριστών

Σε αυτήν την ενότητα θα γίνει περιγραφή της εφαρμογής των διαχειριστών, η οποία είναι διαδικτυακή και βρίσκεται στη μεριά του διακομιστή. (server side) Η εφαρμογή αυτή όπως αναφέρθηκε και νωρίτερα, είναι γραμμένη στη γλώσσα php και χρησιμοποιεί τη βιβλιοθήκη JQuery της γλώσσας JavaScript.

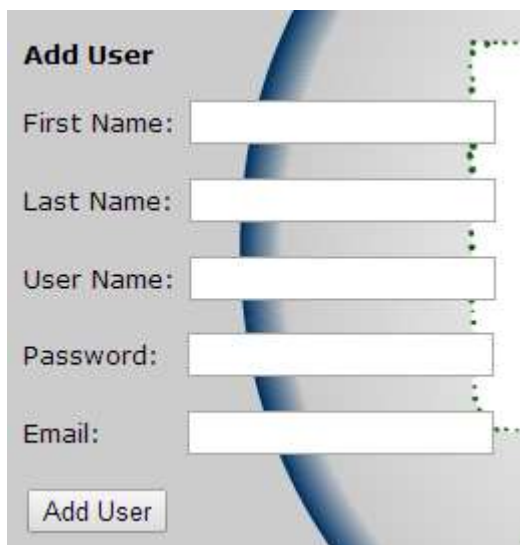
Η αρχική οθόνη, στην οποία μπορεί να βρεθεί ο διαχειριστής πληκτρολογώντας το απαραίτητο URL στον browser, είναι η οθόνη αυθεντικοποίησης. (Εικόνα 42) Σε αυτή την οθόνη ο διαχειριστής εισάγει το αναγνωριστικό του και τον κωδικό του και εφόσον αυτά είναι ορθά, οδηγείται στην οθόνη διαχείρισης (Εικόνα 43).



Εικόνα 43 Οθόνη αυθεντικοποίησης διαχειριστή



Εικόνα 42 Οθόνη διαχείρισης



Add User

First Name:

Last Name:

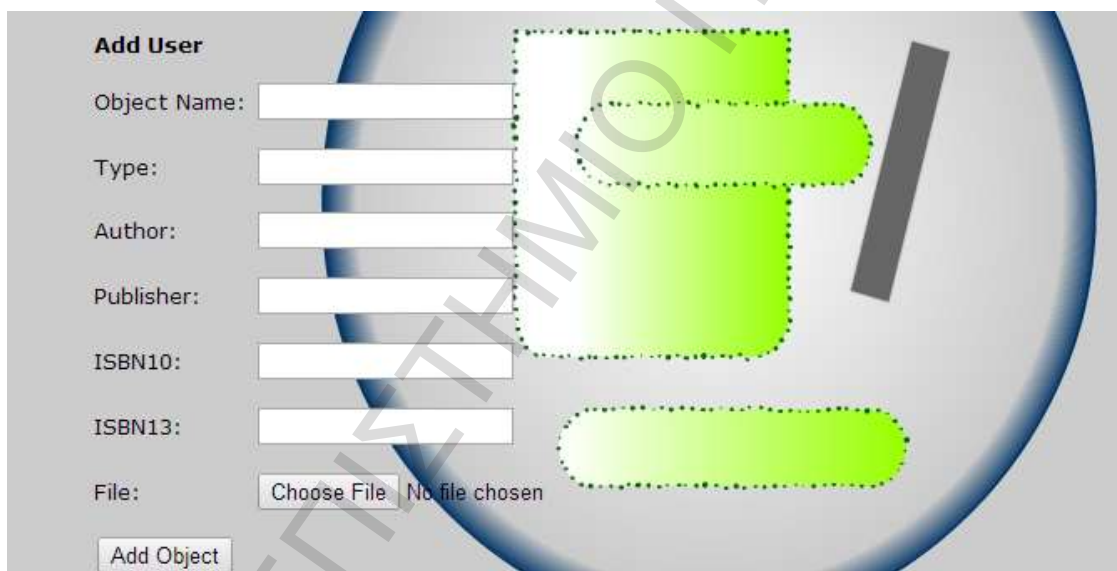
User Name:

Password:

Email:

Στην οθόνη διαχείρισης ο διαχειριστής βλέπει τους αποθηκευμένους χρήστες και τα αποθηκευμένα αντικείμενα και μπορεί να τα επεξεργαστεί ή να τα διαγράψει. Επίσης μπορεί να προχωρήσει στην προσθήκη ενός χρήστη ή αντικειμένου όπως φαίνεται στις Εικόνες 44 και 45.

Εικόνα 44 Προσθήκη χρήστη



Add Object

Object Name:

Type:

Author:

Publisher:

ISBN10:

ISBN13:

File: No file chosen

Εικόνα 45 Προσθήκη αντικειμένου

6.Αξιολόγηση εφαρμογής LibrApp

6.1.Περιγραφή ερωτηματολογίου αξιολόγησης

Για την αξιολόγηση την εφαρμογής ψηφιακής βιβλιοθήκης LibrApp, ώστε να διαπιστώσουμε κατά πόσο καλύπτει τις ανάγκες των χρηστών και αν και πόσο τις εξελίσει, χρησιμοποιήσαμε δύο επικαλυπτόμενες τεχνικές. Για την μέτρηση της χρηστικότητας της εφαρμογής χρησιμοποιήθηκε ένα ερωτηματολόγιο ευχρηστίας (usability questionnaire), το οποίο στην συνέχεια αναλύθηκε από ένα μαθηματικό μοντέλο, ώστε να μετρηθεί η αύξηση στην αποδοτικότητα που αναμένεται να προσφέρει η εφαρμογή σε αυτούς που θα την χρησιμοποιήσουν.

Η ευχρηστία είναι μια γενική έννοια που σχετίζεται στην περίπτωση μας με την παραγωγή ενός εργαλείου χρήσιμου προς τον χρήστη. Κατά την ανάπτυξη εφαρμογών, οι οποίες προορίζονται για χρήση από ευρύ κοινό και στοχεύουν στο να βοηθήσουν τους χρήστες να αυξήσουν την παραγωγικότητά τους σε κάποιον τομέα, είναι σημαντικό για τους προγραμματιστές να γνωρίζουν την γνώμη των χρηστών και να διαθέτουν προγνωστικά σχετικά με την άποψη των χρηστών για την αναμενόμενη αύξηση στην παραγωγικότητα [56], [57]. Για να πραγματοποιηθεί αυτή η μέτρηση, γίνεται χρήση ερωτηματολογίων ευχρηστίας, τα οποία είναι στοχευμένα στο να βολιδοσκοπήσουν την πρόθεση του κοινού να χρησιμοποιήσει την προσφερόμενη υπηρεσία, καθώς και να μετρήσουν το πόσο χρήσιμη θεωρείται.

Στην περίπτωση της LibrApp, δημιουργήθηκε ένα τέτοιο ερωτηματολόγιο, το οποίο προσπαθεί να αξιολογήσει την επίδοση της εφαρμογής σε πέντε βασικές κατηγορίες. Οι κατηγορίες αυτές είναι οι ακόλουθες:

1. Απλές λειτουργίες κινητών εφαρμογών: Σε αυτή την ενότητα μετριέται η χρηστικότητα των λειτουργιών όπως σύνδεση, εγγραφή κλπ.
2. Λειτουργίες ψηφιακής βιβλιοθήκης: Σε αυτή την ενότητα γίνεται μέτρηση της χρηστικότητας λειτουργιών που αφορούν την λειτουργία μίας ψηφιακής βιβλιοθήκης, όπως για παράδειγμα η αναζήτηση βιβλίων.
3. Λειτουργίες σύνδεσης με χρήστες: Σε αυτή την ενότητα γίνεται μέτρηση της χρηστικότητας «κοινωνικών» λειτουργιών, δηλαδή σύνδεση με φίλους κλπ.
4. Συνεργατικές λειτουργίες: Σε αυτή την ενότητα ερωτήσεων, ανήκουν αυτές που μετρούν την ευχρηστία λειτουργιών συνεργατικότητας, δηλαδή αυτές που αφορούν την διαχείριση έργων και γενικά όσες προάγουν την συνεργασία μεταξύ στους χρήστες.
5. Λειτουργίες διαχείρισης καθηκόντων: Σε αυτή την ενότητα ανήκουν οι λειτουργίες που σχετίζονται με την λειτουργία διαχείρισης καθηκόντων η οποία προσφέρεται από την εφαρμογή LibrApp.

Οι ενότητες αυτές, πέραν από την μέτρηση της ικανοποίησης των χρηστών από τις λειτουργίες της LibrApp, περιέχουν και ερωτήσεις, οι οποίες έχουν σκοπό να ανακαλύψουν τυχόν πρόσθετες ανάγκες των χρηστών, οι οποίες θα ήταν καλό να προστεθούν σε νεότερες εκδόσεις της εφαρμογής.

Πλήρης παρουσίαση του ερωτηματολογίου που χρησιμοποιήθηκε, καθώς και ανάλυση των ερωτήσεων δίνεται στο Παράρτημα 2.

Για την αξιολόγηση του ερωτηματολογίου χρησιμοποιήθηκε ένα μεικτό δείγμα 132 ατόμων, και τα αποτελέσματα από την στατιστική ανάλυση του παρουσιάζονται αναλυτικά στην επόμενη ενότητα.

6.1.1.Ανάλυση αποτελεσμάτων ερωτηματολογίου

Όπως αναφέρθηκε και στην προηγούμενη ενότητα το δείγμα το οποίο εξετάστηκε αποτελείται από 132 άτομα, των οποίων η κατανομή με βάση την ηλικία δίνεται στο Σχήμα 15. Παρατηρούμε ότι το ποσοστό των ανδρών είναι λίγο μεγαλύτερο από αυτό των γυναικών, κατά 20%. Επίσης το μορφωτικό επίπεδο του δείγματος ήταν αρκετά υψηλό, με το 89% του δείγματος να έχει ολοκληρώσει τουλάχιστον εκπαίδευση επιπέδου Bachelor (34,61% γυναίκες, 54,39% άνδρες). Η κατανομή του γένους ανά επίπεδο εκπαίδευσης, δίνεται στο Σχήμα 16, ενώ η γενική εικόνα του μορφωτικού επιπέδου του δείγματος στο Σχήμα 17. Το ότι το δείγμα μας έχει αρκετά υψηλό μορφωτικό επίπεδο, είναι αρκετά σημαντικό για τα αποτελέσματα σχετικά με

την αποτελεσματικότητα της εφαρμογής, δεδομένου ότι είναι στοχευμένη για την εξυπηρέτηση ατόμων που λειτουργούν σε ακαδημαϊκά ή παρόμοια περιβάλλοντα.



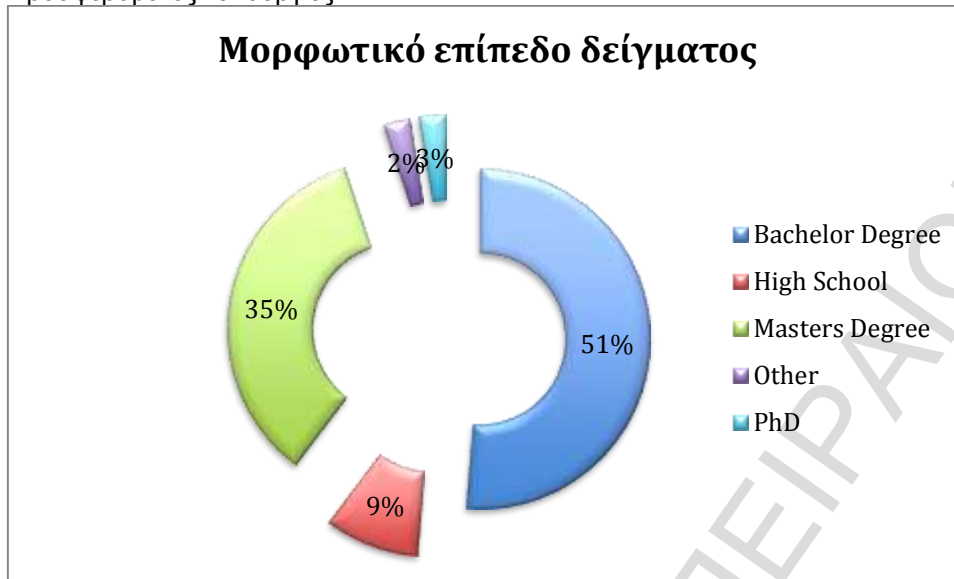
Σχήμα 15 Κατανομή δείγματος με βάση την ηλικία



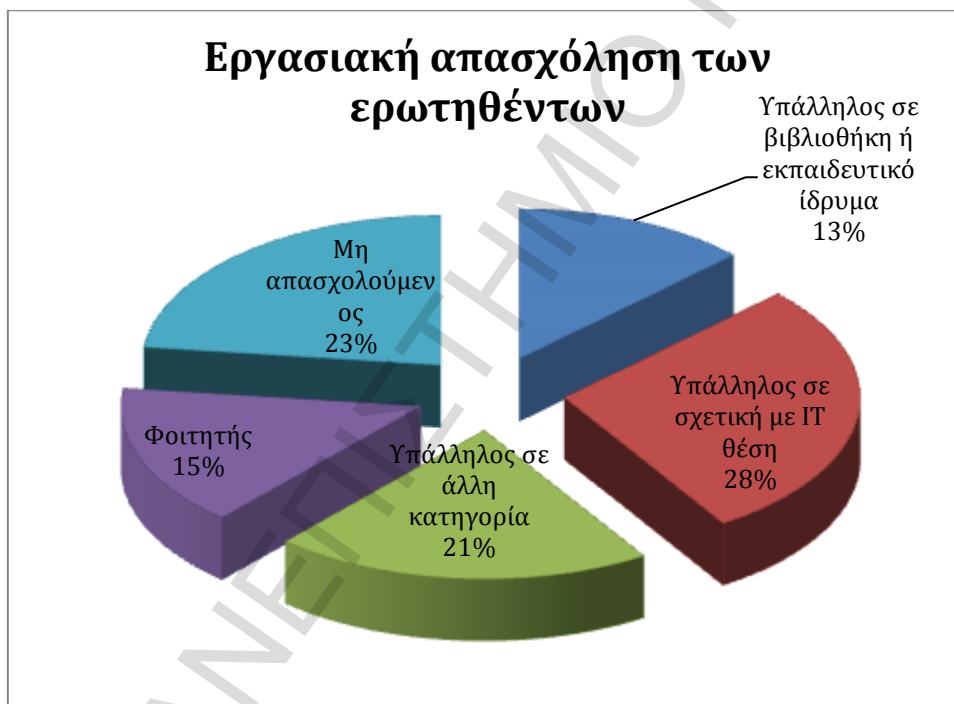
Σχήμα 16 Κατανομή γένους ανά επίπεδο εκπαίδευσης

Δύο ακόμα σημαντικά στοιχεία που αξίζει να αναλύσουμε σχετικά με το δείγμα, είναι η εργασιακή του κατάσταση καθώς και η εξοικείωση που παρουσιάζει με το λειτουργικό σύστημα Android, στο οποίο λειτουργεί η εφαρμογή LibrApp. Στο Σχήμα 18, μπορούμε να δούμε την κατανομή της εργασιακής απασχόλησης των ερωτηθέντων. Παρατηρούμε ότι το 41% έχει αρκετά μεγάλη επαφή με το αντικείμενο της εφαρμογής, είτε ως εμπλεκόμενος σε βιβλιοθήκη ή ακαδημαϊκό ίδρυμα (13%), είτε έχοντας εξοικείωση με νέες τεχνολογίες, εφόσον εργάζεται σε θέση σχετική με τις Τ.Π.Ε επιστήμες (28%). Στο Σχήμα 19, δίνεται η εξοικείωση των χρηστών με το σύστημα Android, και μπορούμε να παρατηρήσουμε ότι το 83,33% του δείγματος διαθέτει από μέτρια έως τέλεια εξοικείωση με συσκευές που φέρουν λειτουργικό σύστημα Android. Αυτό

είναι σημαντικό για την έρευνα μας, μιας και θα μπορέσουν να κατανοήσουν περισσότερο τις προσφερόμενες λειτουργίες.



Σχήμα 17 Μορφωτικό επίπεδο δείγματος



Σχήμα 18 Εργασιακή απασχόληση των ερωτηθέντων

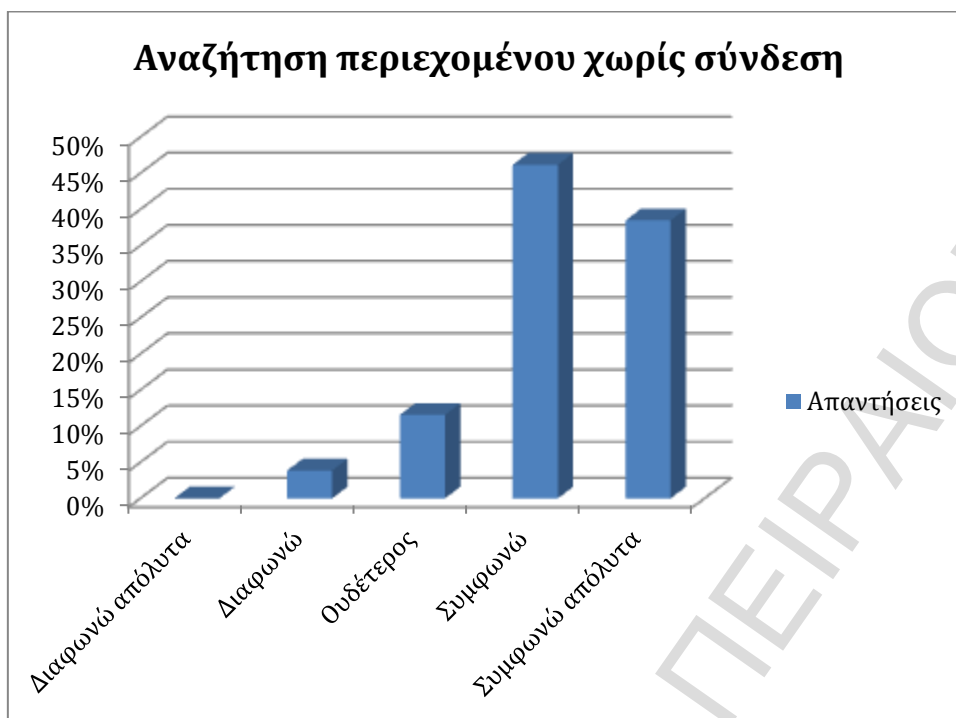
Έχοντας αναλύσει το δείγμα, στην συνέχεια θα παρουσιάσουμε τα γενικά αποτελέσματα από καθεμία από τις 5 κατηγορίες λειτουργιών της εφαρμογής, στις οποίες επικεντρώθηκε η έρευνα. Αφού δώσουμε την γενική εικόνα, θα αναλύσουμε κάποιες ενδιαφέρουσες περιπτώσεις και τέλος θα αναφέρουμε αποτελέσματα σχετικά με την μελλοντική ανάπτυξη της εφαρμογής τα οποία προκύπτουν από την ανάλυση των απαντήσεων του δείγματος.

Για την πρώτη κατηγορία λειτουργιών, η οποία αφορά λειτουργίες που εκτελούν οι περισσότερες εφαρμογές σε smartphone (login, register, απλή αναζήτηση κλπ), παρατηρούμε τα αποτελέσματα στις ερωτήσεις που έχουν να κάνουν με την αναζήτηση περιεχομένου χωρίς σύνδεση και με το download αρχείων. Η εφαρμογή προσφέρει στους χρήστες την δυνατότητα

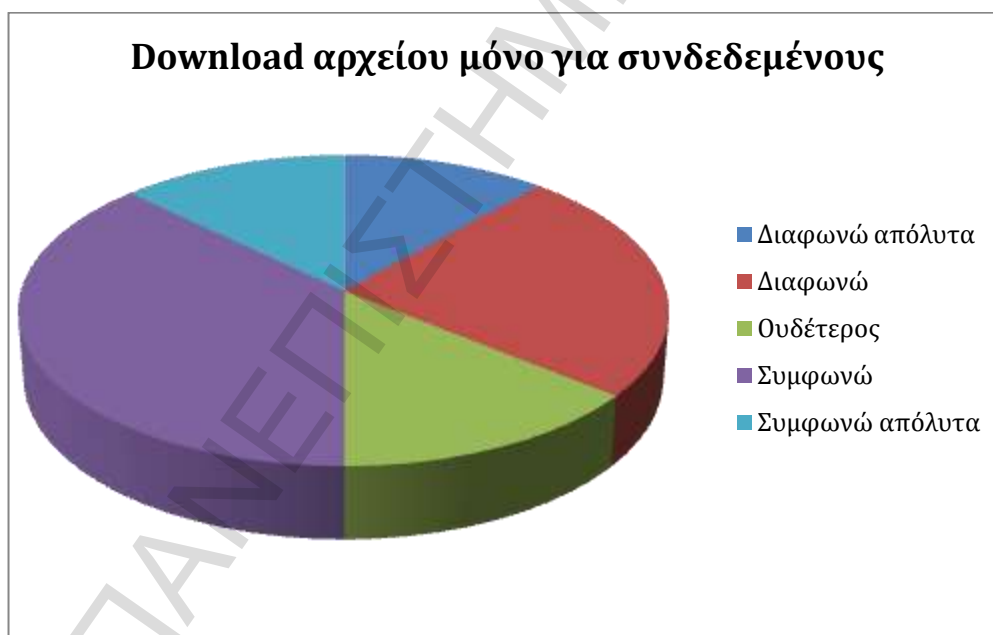
να αναζητήσουν περιεχόμενο χωρίς να έχουν προηγουμένως δημιουργήσει προφίλ στην εφαρμογή, κάνοντας έτσι την απλή αναζήτηση πιο γρήγορη. Παρατηρούμε, πως το 85% περίπου του δείγματος συμφωνεί με αυτή την λειτουργία, ενώ ένα 10% παραμένει ουδέτερο. Δηλαδή, η λειτουργία αυτή της εφαρμογής βρίσκει τους χρήστες σύμφωνους, αποτέλεσμα το οποίο ήταν το αναμενόμενο (Σχήμα 20). Ακόμα η εφαρμογή, επιτρέπει τη μεταφόρτωση περιεχομένου μόνο από εγγεγραμμένους χρήστες, ώστε να υπάρχει ένας έλεγχος το που πηγαίνουν τα αρχεία. Αυτή η λειτουργία αντιμετωπίζεται με θετική αντίδραση από το 65% του δείγματος (Σχήμα 21). Από τα παραπάνω μπορούμε να συμπεράνουμε ότι γενικά αυτή η κατηγορία λειτουργιών αφήνει αρκετά ικανοποιημένο το δείγμα μας.



Σχήμα 19 Βαθμός εξοικείωσης με το Android

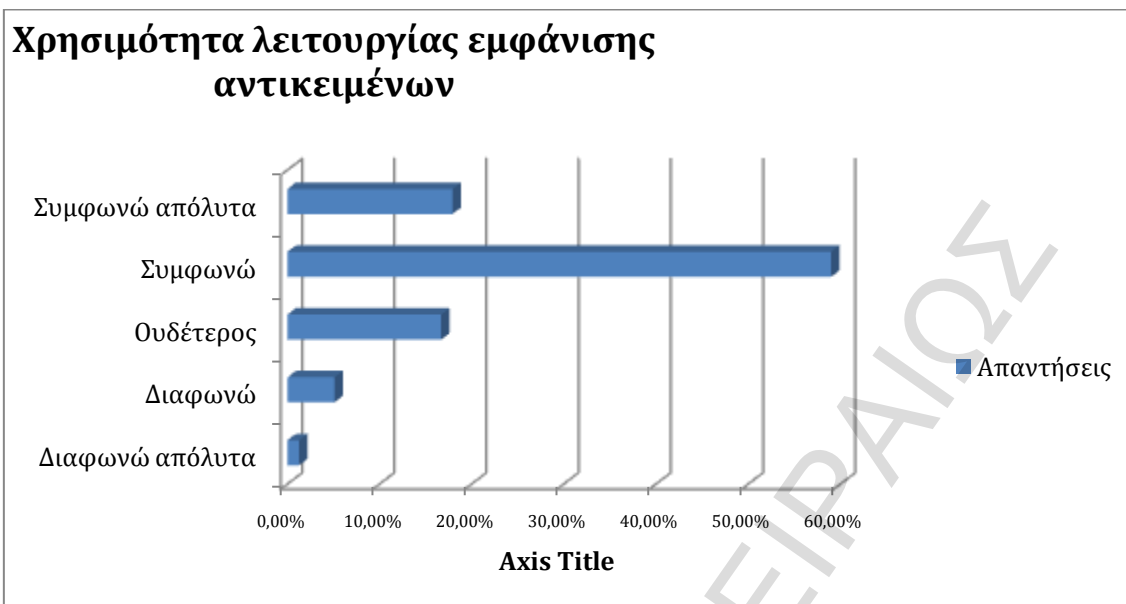


Σχήμα 20 Αναζήτηση περιεχομένου χωρίς σύνδεση



Σχήμα 21 Αναζήτηση περιεχομένου μόνο για συνδεδεμένους χρήστες

Η επόμενη κατηγορία λειτουργιών που αξιολογήθηκε ήταν αυτή των γενικών λειτουργιών ψηφιακής βιβλιοθήκης. Για την γενική αξιολόγηση αυτής της κατηγορία επιλέξαμε την ερώτηση που μετράει την γενική ικανοποίηση των χρηστών σχετικά με την λειτουργία αναζήτησης και εμφάνισης αντικειμένων. Όπως μπορούμε να δούμε και στο Σχήμα 22, πάνω από το 70% των ερωτηθέντων είναι αρκετά ικανοποιημένο με αυτή την ομάδα λειτουργιών, με ένα 15% να παραμένει ουδέτερο.



Σχήμα 22 Χρησιμότητα λειτουργίας εμφάνισης αντικειμένων

Στην συνέχεια έγινε αξιολόγηση των «κοινωνικών» λειτουργιών της εφαρμογής LibrApp, δηλαδή αυτών που σχετίζονται με ενέργειες όπως σύνδεση με φίλους, διαδραστικότητα με αυτούς κ.ά. Για την γενική αξιολόγηση, θα αναλύσουμε την ερώτηση που αφορά το πόσο χρήσιμη φαίνεται αυτή η ομάδα λειτουργιών στο δείγμα μας. Πάνω από το 60% του δείγματος είναι ικανοποιημένο με αυτές τις λειτουργίες, με ένα 25% περίπου να είναι ουδέτερο (Σχήμα 23). Δεδομένου του πολύ μικρού ποσοστού των αρνητικών απαντήσεων, μπορούμε να συμπεράνουμε ότι αυτή η κατηγορία λειτουργιών αξιολογείται ως χρήσιμη και μπορεί να συνδράμει στην αύξηση της παραγωγικότητας των χρηστών της εφαρμογής.



Σχήμα 23 Χρησιμότητας της σύνδεσης με φίλους

Οι επόμενες δύο κατηγορίες που εξετάστηκαν στα πλαίσια της αξιολόγησης της εφαρμογής είναι αυτές που αφορούν τις συνεργατικές λειτουργίες που προσφέρονται και αυτές που αφορούν τις λειτουργίες διαχείρισης καθηκόντων (task management). Σχετικά με την LibrApp: Μία Android συνεργατική ψηφιακή βιβλιοθήκη.

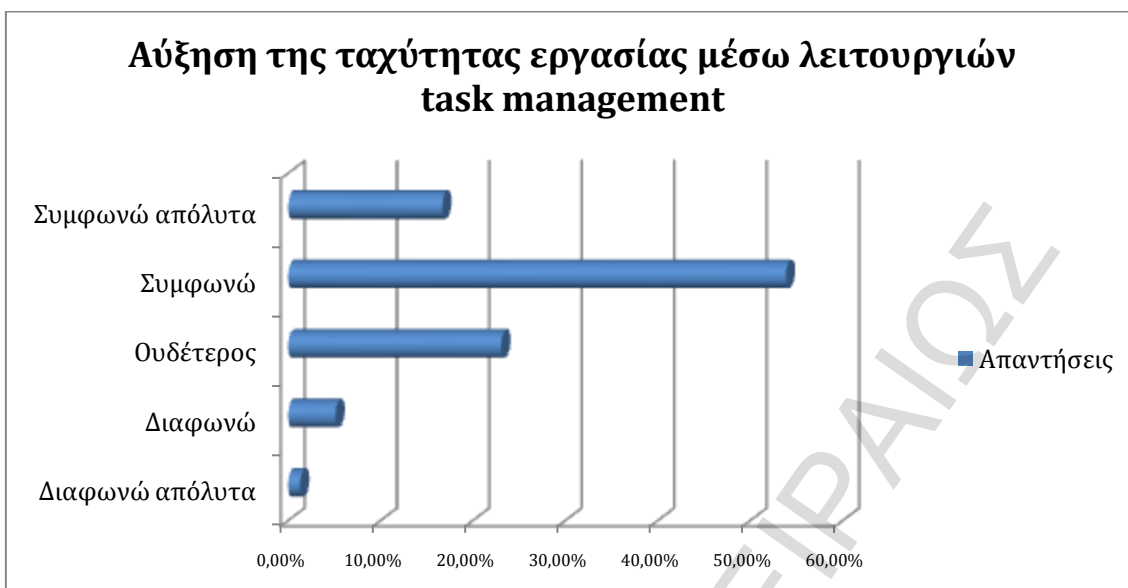
πρώτη κατηγορία, μπορούμε να δούμε στο Σχήμα 24, ότι δεν υπάρχουν αρνητικές αξιολογήσεις, το οποίο μας οδηγεί στο συμπέρασμα ότι το δείγμα είναι ικανοποιημένο από τις λειτουργίες, και μάλιστα τις θεωρεί εξαιρετικά χρήσιμες και απαραίτητες στην εφαρμογή.

Στην τελευταία κατηγορία, μπορούμε να παρατηρήσουμε επίσης θετική τάση, με τις αρνητικές απαντήσεις να βρίσκονται κάτω από το 5% (Σχήμα 25). Άρα, θα μπορούσαμε να πούμε ότι ειδικά για τις δύο τελευταίες κατηγορίες το δείγμα εκφράζει την ανάγκη ύπαρξής τους. Ξεκινώντας την εκπόνηση αυτή της διατριβής, θεωρήσαμε ότι είναι σημαντικό να δώσουμε στο κοινό που θα χρησιμοποιήσει την εφαρμογή ψηφιακής βιβλιοθήκης LibrApp, μία πληθώρα εργαλείων στοχευμένων στην αύξηση της συνεργατικότητας και της παραγωγικότητας. Με βάση τα αποτελέσματα της ανατροφοδότησης που λαμβάνουμε από την ανάλυση του ερωτηματολογίου, θα μπορούσαμε χωρίς μεγάλη απόκλιση να συμπεράνουμε, ότι όντως το κοινό λαμβάνει αυτές τις λειτουργίες με θετική αντιμετώπιση.

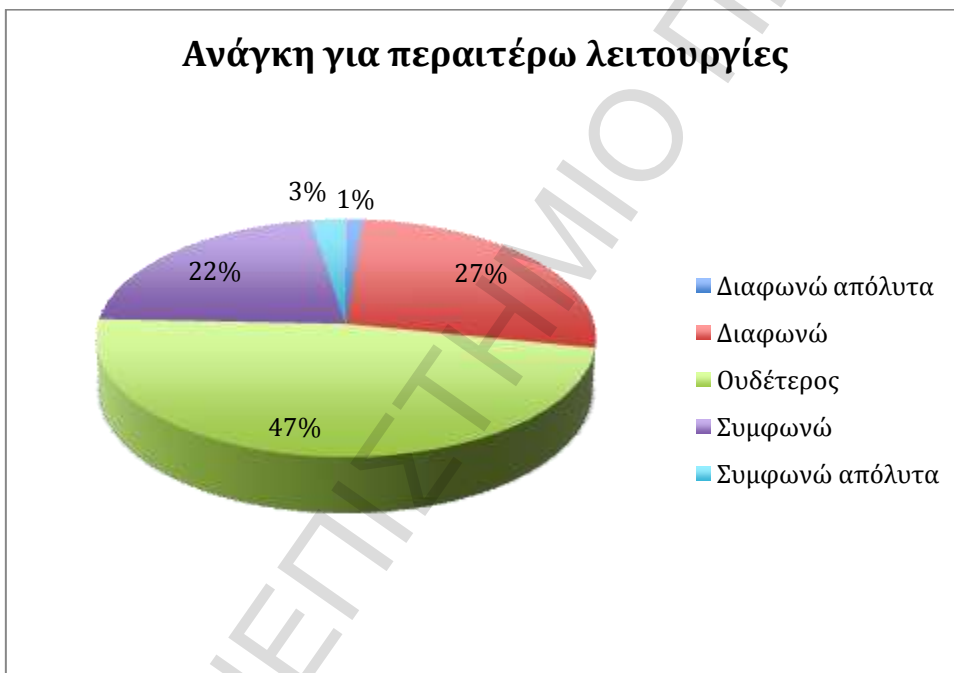
Επειδή το δείγμα παρουσιάζει μία γενική ομοιομορφία, όπως φαίνεται και στα αρχικά σχήματα, δεν παρατηρούνται αξιοσημείωτες τάσεις σχετικά με τα δημογραφικά στοιχεία. Μπορούμε όμως να παρατηρήσουμε κάποιες ανάγκες που προκύπτουν σχετικά με την μελλοντική εξέλιξη της εφαρμογής. Για παράδειγμα, στο Σχήμα 26, διαφαίνεται πως το μεγαλύτερο ποσοστό των ερωτηθέντων είναι ικανοποιημένο από τις προσφερόμενες λειτουργίες και δεν επιθυμεί κάποια προσθήκη, ωστόσο υπάρχει ένα ποσοστό που τις επιθυμεί (περίπου 25%), το οποίο κρίνεται ικανό να μας οδηγήσει σε περαιτέρω ανάπτυξη στο μέλλον. Μία από τις λειτουργίες που θα ήταν καλό να αναπτυχθεί στο μέλλον, όπως φαίνεται από το Σχήμα 27, είναι η είσοδος στην εφαρμογή χρησιμοποιώντας έναν ήδη υπάρχων λογαριασμό από κάποιο κοινωνικό δίκτυο, μιας και ζητήθηκε από ένα ποσοστό του δείγματος μεγαλύτερο από το 35%.



Σχήμα 24 Αξιολόγηση συνεργατικών λειτουργιών



Σχήμα 25 Αξιολόγηση λειτουργιών task management



Σχήμα 26 Αξιολόγηση ανάγκης για περαιτέρω ανάπτυξη



Σχήμα 27 Αξιολόγηση εισόδου με social account

6.2. Περιγραφή του μοντέλου

Για την συνέχεια της αξιολόγησης της εφαρμογής, κρίθηκε σωστό να μετρηθεί το κατά πόσο αυτή συμβάλλει στην αύξηση της παραγωγικότητας των χρηστών της. Για να γίνει αυτή η μέτρηση, δημιουργήθηκε ένα μαθηματικό μοντέλο πάνω στο ερωτηματολόγιο που αναλύθηκε στην προηγούμενη ενότητα, ακολουθώντας μία σειρά από παραδοχές και χρησιμοποιώντας μία τεχνική που έχει πρωτίστως χρησιμοποιηθεί σε παρόμοια μελέτη στον ιατρικό τομέα [58].

Για να μπορέσουμε να οδηγηθούμε σε ένα μετρήσιμο αποτέλεσμα για την αύξηση της παραγωγικότητας, θα κάνουμε την παραδοχή ότι καθένας από τους ερωτηθέντες στην έρευνα ξεκινάει από ένα επίπεδο μέτριας ικανοποίησης, στο οποίο δίνουμε την αποτίμηση 3. Στην συνέχεια για κάθε μία από τις κατηγορίες στις οποίες αναλύσαμε την έρευνα, θα διαλέξουμε τις ερωτήσεις που αντιπροσωπεύουν την παραγωγικότητα και θα υπολογίσουμε τα σκορ τους χρησιμοποιώντας την αντιστοιχία τιμή-απάντηση που δίνεται στον παρακάτω πίνακα.

	Απαντήσεις				
	Διαφωνώ απόλυτα	Διαφωνώ	Ουδέτερος	Συμφωνώ	Συμφωνώ απόλυτα
Αποτίμηση	1	$S_i = \sum_{j=1}^{132} v_j$	3	4	5

Πίνακας 9 Αποτίμηση απαντήσεων ερωτηματολογίου

Να σημειωθεί ότι μπορεί η αντιστοιχία των αποτιμήσεων να είναι αντίστροφη σε περιπτώσεις που οι αρνητικές απαντήσεις θεωρούνται ως οι θετικές για την παραγωγικότητα, όπως για παράδειγμα σε ερωτήσεις τύπου «Θεωρώ ότι δεν ωφελούμαι».

Το σκορ για καθεμία από τις ερωτήσεις θα υπολογιστεί με την χρήση του τύπου:

$$S_i = \sum_{j=1}^{132} v_j$$

όπου:

- i = ο αριθμός της ερώτησης
- j = το πλήθος των απαντήσεων
- v = η αποτίμηση της απάντησης

Στην συνέχεια, για κάθε σκορ που προκύπτει για την συγκεκριμένη κατηγορία, θα υπολογιστεί το συνολικό σκορ, χρησιμοποιώντας τον παρακάτω τύπο:

$$T_k = \sum_{i=1}^l w_i S_i$$

όπου:

- κ = το πλήθος των κατηγοριών
- l = το πλήθος των ερωτήσεων
- i = ο αριθμός της ερώτησης
- w_i = το βάρος που δίνεται στην κάθε ερώτηση i
- S = το σκορ της κάθε ερώτησης

Τέλος, για να υπολογίσουμε το συνολικό σκορ και να μπορέσουμε να υπολογίσουμε αν υπάρχει αύξηση σε σχέση με το αρχικό σκορ παραγωγικότητας, θα χρησιμοποιήσουμε τον τύπο:

$$P = \sum_{p=1}^5 w_p T_p$$

Στην υποενότητα που ακολουθεί γίνεται παρουσίαση των αποτελεσμάτων του μοντέλου.

6.2.1. Αποτελέσματα του μοντέλου

Για την πρώτη κατηγορία λειτουργιών ως ερωτήσεις που επηρεάζουν την παραγωγικότητα επιλέχθηκαν οι :

- «Θέλω να είμαι σε θέση να ψάξω για περιεχόμενο χωρίς να διαθέτω λογαριασμό στην εφαρμογή» με βάρος 0,7
- «Συμφωνώ με το να είμαι συνδεδεμένος για να κατεβάσω περιεχόμενο από την εφαρμογή» με βάρος 1,5
- Θέλω να είμαι σε θέση να ανεβάσω το δικό μου περιεχόμενο στην εφαρμογή με βάρος 0,8

Η δεύτερη ερώτηση διαθέτει το μεγαλύτερο βάρος μιας και η λειτουργία αυτή θεωρείται ως η σημαντικότερη αυτής της κατηγορίας. Το σκορ υπολογίζεται ως εξής:

$$T_1 = \sum_{i=1}^3 w_i S_i = \frac{0,7S_{11} + 1,5S_{12} + 0,8S_{13}}{3} = \frac{0,7 * 3,27 + 1,5 * 4,19 + 0,8 * 3,27}{3} = 3,73$$

Η αύξηση που παρατηρείται σε σχέση με την αρχική παραδοχή της μέσης αποτίμησης, υπολογίζεται περίπου στο 24,3%. Αποτελεί δηλαδή ένα αρκετά μεγάλο ποσοστό, αποτέλεσμα που μας οδηγεί στο συμπέρασμα ότι αυτή η κατηγορία προσδίδει αρκετά μεγάλη αύξηση στην παραγωγικότητα των χρηστών.

Για την δεύτερη κατηγορία λειτουργιών ως ερωτήσεις που επηρεάζουν την παραγωγικότητα επιλέχθηκαν οι :

- «Θεωρώ ότι η συγκεκριμένη κατηγορία λειτουργιών είναι χρήσιμη» με βάρος 1,5.
- «Θέλω να είμαι σε θέση να έχω το προσωπικό μου ράφι διαθέσιμο τοπικά στη συσκευή μου» με βάρος 1,2.
- «Θέλω να είμαι σε θέση να μοιράζομαι το προσωπικό μου ράφι με τους χρήστες με τους οποίους είμαι συνδεδεμένος» με βάρος 1,3.

Τα βάρη σε αυτές τις ερωτήσεις είναι αρκετά υψηλά, μιας και οι λειτουργίες είναι αρκετά σημαντικές, με την πρώτη ερώτηση να διαθέτει λίγο μεγαλύτερο βάρος. Το σκορ υπολογίζεται ως εξής:

$$T_2 = \sum_{i=1}^3 w_i S_i = \frac{1,5S_{21} + 1,2S_{22} + 1,3S_{23}}{4} = \frac{1,5 * 3,87 + 1,2 * 3,97 + 1,3 * 3,81}{4} = 3,88$$

Η αύξηση που παρατηρείται σε σχέση με την αρχική παραδοχή της μέσης αποτίμησης, υπολογίζεται περίπου στο 29.3%. Η αύξηση συνεχίζει να είναι αρκετά μεγάλη σε σχέση με την αρχική αποτίμηση και για αυτήν την κατηγορία. Το αποτέλεσμα είναι σημαντικό ειδικά για την κατηγορία αυτή που ελέγχει τις δυνατότητες ψηφιακής βιβλιοθήκης της εφαρμογής, γιατί μας οδηγεί στο συμπέρασμα ότι η εφαρμογή λειτουργεί αποδοτικά ως ψηφιακή βιβλιοθήκη.

Για την τρίτη κατηγορία λειτουργιών ως ερωτήσεις που επηρεάζουν την παραγωγικότητα επιλέχθηκαν οι :

- «Βρίσκω τη σύνδεση με άλλους χρήστες χρήσιμη» με βάρος 1,6.
- «Θέλω να είμαι σε θέση να ανεβάσω περιεχόμενο απευθείας στην προσωπικό ράφι των φίλων μου» με βάρος 1.
- «Πιστεύω ότι έχοντας συνδέσεις με άλλους χρήστες θα ωφεληθεί το έργο μου» με βάρος 1,6.

Τα βάρη σε αυτές τις ερωτήσεις είναι αρκετά υψηλά, μιας και οι λειτουργίες είναι αρκετά σημαντικές όπως και της προηγούμενης κατηγορίας, με την πρώτη και την τρίτη ερώτηση να διαθέτουν λίγο μεγαλύτερο βάρος. Το σκορ υπολογίζεται ως εξής:

$$T_3 = \sum_1^3 w_i S_i = \frac{1,6S_{31} + 1S_{32} + 1,6S_{33}}{4,2} = \frac{1,6 * 3,60 + 1 * 3,53 + 1,6 * 3,62}{4,2} = 3,59$$

Η αύξηση που παρατηρείται σε σχέση με την αρχική παραδοχή της μέσης αποτίμησης, υπολογίζεται περίπου στο 19,66%. Η αύξηση είναι σημαντική και δείχνει την αύξηση στην αποδοτικότητα που προσδίδουν οι «κοινωνικές» λειτουργίες της LibrApp στους τελικούς χρήστες.

Για την τέταρτη κατηγορία λειτουργιών ως ερωτήσεις που επηρεάζουν την παραγωγικότητα επιλέχθηκαν οι :

- «Πιστεύω ότι η λειτουργία chat με τους συναδέλφους μου, θα κάνει τη δουλειά μου πιο αποτελεσματική» με βάρος 1,1.
- «Πιστεύω ότι το να είμαι σε θέση να επεξεργαστώ το ίδιο κείμενο με τους συναδέλφους μου με ωφελεί» με βάρος 0,9.
- «Θεωρώ ότι το να χρησιμοποιώ wikis είναι εύκολο» με βάρος 1,1.
- «Πιστεύω ότι οι δυνατότητες που παρέχονται είναι ικανοποιητικές» με βάρος 1,5.

Τα βάρη σε αυτές κυμαίνονται γύρω από την τιμή 1, με την τέταρτη ερώτηση να διαθέτει λίγο μεγαλύτερο βάρος. Το σκορ υπολογίζεται ως εξής:

$$T_4 = \sum_1^4 w_i S_i = \frac{1,1S_{41} + 0,9S_{42} + 1,1S_{43} + 1,5S_{44}}{4,6} = \frac{1,1 * 3,74 + 0,9 * 2,84 + 1,1 * 3,35 + 1,5 * 3,65}{4,6} = 3,44$$

Η αύξηση που παρατηρείται σε σχέση με την αρχική παραδοχή της μέσης αποτίμησης, υπολογίζεται περίπου στο 14.66%. Η αύξηση αυτή είναι μικρότερη από αυτές των προηγούμενων κατηγοριών, αλλά είναι εξίσου σημαντική και σημαίνει πως οι συνεργατικές λειτουργίες της εφαρμογής, προσδίδουν στην αύξηση της αποδοτικότητας σε αρκετά ικανοποιητικό βαθμό.

Για την πέμπτη και τελευταία κατηγορία λειτουργιών ως ερωτήσεις που επηρεάζουν την παραγωγικότητα επιλέχθηκαν οι :

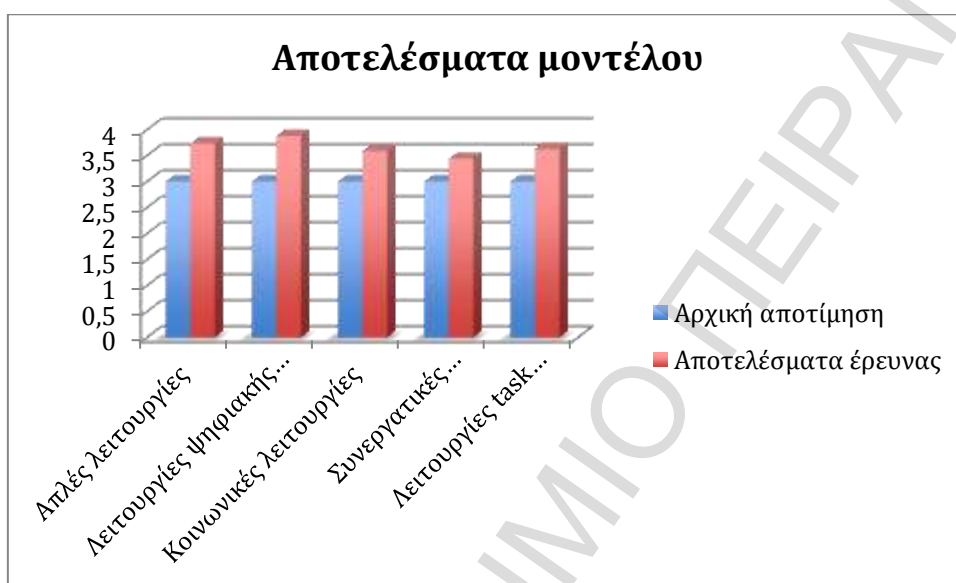
- «Δεν βρίσκω χρήσιμο αυτό το χαρακτηριστικό» με βάρος 1,2.
- «Πιστεύω ότι ένα χαρακτηριστικό όπως αυτό, θα βοηθήσει να επιταχυνθεί το έργο μου» με βάρος 1,4.
- «Θα χρησιμοποιούσα μία τέτοια λειτουργία στον οργανισμό μου» με βάρος 1,4.
- «Δεν θεωρώ ότι αυτό το χαρακτηριστικό είναι περίπλοκο» με βάρος 1,1.

Τα βάρη σε αυτές κυμαίνονται στις ίδιες τιμές, με την τρίτη και τέταρτη ερώτηση να διαθέτουν λίγο μεγαλύτερο βάρος. Το σκορ υπολογίζεται ως εξής:

$$T_5 = \sum_{i=1}^5 w_i S_i = \frac{1,2S_{51} + 1,4S_{52} + 1,4S_{53} + 1,1S_{54}}{5,1}$$

$$= \frac{1,2 * 3,45 + 1,4 * 3,79 + 1,4 * 3,76 + 1,1 * 3,49}{5,1} = 3,63$$

Η αύξηση που παρατηρείται σε σχέση με την αρχική παραδοχή της μέσης αποτίμησης, υπολογίζεται περίπου στο 21%. Η αύξηση αυτή αφορά τις λειτουργίες task management που προσφέρονται από την εφαρμογή και ουσιαστικά μας δικαιώνει στην επιλογή της προσθήκης του συγκεκριμένου χαρακτηριστικού. Στο Σχήμα 28 βλέπουμε την αύξηση στην αποδοτικότητα που εμφανίζει καθεμία από τις πέντε κατηγορίες λειτουργιών.



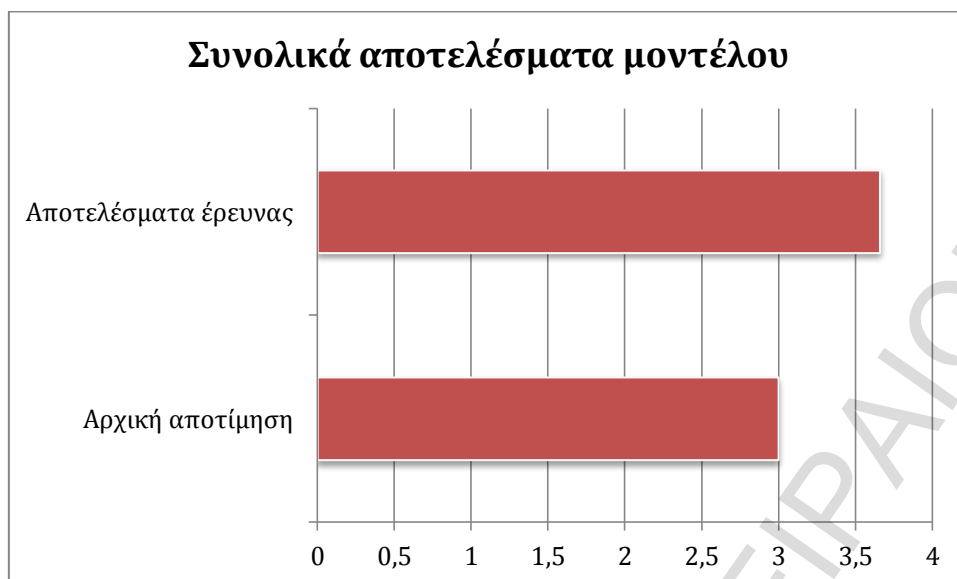
Σχήμα 28 Αποτελέσματα μοντέλου ανά κατηγορία

Για να αποκτήσουμε και μία συνολική εικόνα της αύξησης της αποδοτικότητας από την χρήση της εφαρμογής LibrApp, θα υπολογίσουμε και το συνολικό σκορ, αθροίζοντας τα επιμέρους αφού πρώτα τους δώσουμε τα απαραίτητα βάρη. Το σκορ υπολογίζεται ως εξής:

$$P = \sum_{p=1}^5 w_p T_p = \frac{0,9T_1 + 1,3T_2 + 1T_3 + 1,2T_4 + 1,2T_5}{5,6}$$

$$= \frac{0,9 * 3,73 + 1,3 * 3,88 + 1 * 3,44 + 1,2 * 3,63}{5,6} = 3,66$$

Το συνολικό σκορ του 3,66, μας δίνει μία σημαντική αύξηση της τάξης του 22%. Το αποτέλεσμα φαίνεται και στο Σχήμα 29.



Σχήμα 29 Συνολικά αποτελέσματα μοντέλου

6.3 Συμπεράσματα

Από την ανάλυση που έγινε στην ενότητα της αξιολόγησης μπορούμε να βγάλουμε αρκετά και χρήσιμα συμπεράσματα. Από τα αποτελέσματα του ερωτηματολογίου, μπορούμε να συμπεράνουμε πως η εφαρμογή εμφανίζει θετικές κριτικές, και γίνεται γενικά αποδεκτή από τους χρήστες της, κυρίως από άτομα υψηλής μόρφωσης, που εργάζονται σε θέσεις Τ.Π.Ε ή ακαδημαϊκού ενδιαφέροντος. Βλέπουμε ότι το δείγμα μας είναι γενικά ικανοποιημένο με τις προσφερόμενες υπηρεσίες, ειδικά με αυτές που αφορούν την συνεργατικότητα κατά την εκπόνηση εργασιών και έργων, η οποία μάλιστα δεν συγκεντρώνει αρνητικές απαντήσεις.

Ακόμα, από την ανάλυση του ερωτηματολογίου με την χρήση του μοντέλου που περιγράφηκε, καταλήγουμε σε ένα τελικό συγκεντρωτικό σκορ 3,66 μονάδων. Αυτό το σκορ, από σκοπιά αποδοτικότητας και παραγωγικότητας, σημαίνει ότι η εφαρμογή είναι ικανή να προκαλέσει αύξηση 0,66 από την αρχική παραδοχή της τιμής 3. Ουσιαστικά, παρατηρούμε μία αύξηση περίπου 22%, αν θεωρήσουμε ότι όσοι απάντησαν στο ερωτηματολόγιο βρίσκονται σε μία μέση απόδοση.

Η μεγαλύτερη αύξηση της παραγωγικότητας προσδίδεται στις λειτουργίες που αφορούν τις απλές λειτουργίες ψηφιακής βιβλιοθήκης, καθώς και την καινούρια λειτουργία διαχείρισης καθηκόντων που αναπτύχθηκε και παρουσιάστηκε από την παρούσα μεταπτυχιακή διατριβή. Επίσης σημαντική αύξηση δίνονται και από τις συνεργατικές λειτουργίες, με την δυνατότητα εξέλιξης όμως, να είναι εμφανής.

Γενικά αυτό που συμπεραίνουμε είναι πως η εφαρμογή φαίνεται να προσδίδει αύξηση στην παραγωγικότητα των χρηστών και να τους διευκολύνει κατά την δημιουργία και διαχείριση εργασιών, ταυτόχρονα με την παροχή πληροφοριών, θα ήταν όμως ωφέλιμο να υπάρξει βελτίωση σε σημεία όπως τα παρακάτω:

- Χρήση ήδη υπαρχόντων λογαριασμών για σύνδεση στην εφαρμογή.
- Μεγαλύτερη χρήση των κοινωνικών δικτύων.
- Παροχή περισσότερων συνεργατικών λειτουργιών (forum κ.α.).
- Περαιτέρω μελέτη και ανάπτυξη των λειτουργιών διαχείρισης καθηκόντων.
- Εφαρμογή ακόμα πιο σύγχρονων λειτουργιών και τεχνολογιών.

7.Μελλοντικές προτάσεις

Όπως αναφέρθηκε και στον προηγούμενη υποενότητα, υπάρχουν κάποια σημεία στην εφαρμογή τα οποία χρήζουν βελτίωσης. Είναι σημαντικό για την ανάπτυξη και για την διάδοση μίας τέτοιου είδους εφαρμογής, να δοθεί έμφαση στην χρήση των κοινωνικών δικτύων, τόσο στο κομμάτι της χρήσης υπάρχοντων λογαριασμών για την είσοδο στην εφαρμογή, τόσο και σε άλλους τομείς.

Ένα ακόμα κομμάτι που θα χρήζει ενδιαφέροντος, είναι η μεταφορά της εφαρμογής LibrApp και σε άλλα λειτουργικά συστήματα κινητών συσκευών, όπως για παράδειγμα το Apple iOS ή το Windows Phone 7. Επίσης, θα ήταν χρήσιμη η δημιουργία ενός διαδικτυακού portal, το οποίο θα μπορούσε να επεκτείνει και τις συνεργατικές λειτουργίες που προβάλλει η εφαρμογή.

Ένα άλλο σημαντικό στοιχείο, είναι η εξερεύνηση περαιτέρω λειτουργιών που θα μπορούσαν να προστεθούν στην εφαρμογή, για να τονώσουν κυρίως τον συνεργατικό της χαρακτήρα και να εξελίξουν σε μεγαλύτερο επίπεδο τις λειτουργίες διαχείρισης καθηκόντων που προτάθηκαν. Μία τέτοια έρευνα θα μπορούσε να επικεντρωθεί σε συνεργατικά στοιχεία, όπως ασύγχρονες πλατφόρμες επικοινωνίας ή ακόμα και κάποιο σύστημα ταυτόχρονης συγγραφής και διαμοιρασμού κώδικα, για χρήστες οι οποίοι ασχολούνται με την ανάπτυξη εφαρμογών. Αυτό το σύστημα θα μπορούσε να έχει την μορφή ενός συστήματος subversioning ή ακόμα και να χρησιμοποιεί ένα ήδη υπάρχον, το οποίο θα ενσωματωθεί στην εφαρμογή.

Τέλος, από την σκοπιά της αξιολόγησης του συστήματος που προτάθηκε, μία ενδιαφέρουσα ιδέα να ακολουθηθεί θα ήταν η εξής. Η εφαρμογή, μαζί με το σύστημα διαχείρισης να εγκατασταθεί σε ένα ήδη υπάρχον φυσικό βιβλιοθηκονομικό περιβάλλον και να λαμβάνεται ανατροφοδότηση από τους διαχειριστές και χρήστες της, ανά τακτά χρονικά διαστήματα. Έτσι θα μπορούσαμε να έχουμε μία πλήρη χαρτογράφηση των δυνατοτήτων που προσδίδει η εφαρμογή στους χρήστες της και θα μπορούσαμε να μετρήσουμε με μεγαλύτερη ακρίβεια την μεταβολή στην αποδοτικότητα των εμπλεκομένων με την εφαρμογή ατόμων.

Βιβλιογραφία

- [1] Witten, Ian H., Bainbridge, David Nichols. How to Build a Digital Library. Morgan Kaufmann Publishers (2010) ISBN 978-0123748577. Accessed January 31,2014.
- [2] Edward A. Fox. The Digital Libraries Initiative - Update and Discussion, Bulletin of the America Society of Information Science, Vol. 26, No 1, October/November 1999.
- [3] L. Candela, G. Athanasopoulos, D. Castelli, K. El Raheb, P. Innocenti, Y. Ioannidis, A. Katifori, A. Nika, G. Vullo, S. Ross: *The Digital Library Reference Model*. April 2011 (PDF)
- [4] L. Candela et al.: *The DELOS Digital Library Reference Model - Foundations for Digital Libraries*. Version 0.98, February 2008 (PDF)
- [5] Yi, Esther, Inside the Quest to Put the World's Libraries Online, The Atlantic, July 26, 2012.
- [6] Nürnberg, P. J., Furuta, R., Leggett, J. J., Marshall, C. C., & Shipman III, F. M. (1995, June). Digital Libraries: Issues and Architectures. In *DL* (p. 0).
- [7] Gonçalves, Marcos André, et al. "Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries." *ACM Transactions on Information Systems (TOIS)* 22.2 (2004): 270-312.
- [8] Lannom, Laurence. *Handle system overview*. ERIC Clearinghouse, 2000.
- [9] William Y. Arms Christophe Blanchi Edward A. Overly Corporation for National Research Initiatives Reston, Virginia, An Architecture for Information in Digital Libraries, D-Lib Magazine, February 1997
- [10] Adam, Nabil R., et al. "Digital library task force." *Computer* 29.8 (1996): 89-91.
- [11] Juliano Lopes de Oliveira, Marcos Andre Goncalves, Claudia BauzerMedeiros. A framework for designing and implementing the user interface of a geographic digital library. *International Journal on Digital Libraries*, 1999, issue 2 page 190-206.
- [12] Hong W, Thong J.Y.L, Tam K.Y. Understanding user acceptance of digital libraries: What are the roles of interface characteristics, organizational context, and individual differences? *International Journal of Human-Computer Studies*, 2002. 57; 215-42.
- [13] Mulhem p, Nigay L. Interactive Information Retrieval Systems: From User Centered Interface Design to Software design. Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in information Retrieval , New York 1996. 326-334.
- [14] Fox, Edward A, Hix D, Nowell L.T, Brueni D.J, Wake W.C, Heath L.S. Users, User Interfaces, and Objects: Envision, a Digital Library. *Journal of the American Society for Information Science*, 1993. 44:8; 480-491.
- [15] Savage-Knepshield P, Belkin N. Interaction in Information Retrieval: Trends Over Time. *Journal of the American Society for Information Science*, 1999. 50:12; 1067-1082.
- [16] Mastidoro N. The Intratext Project. The University of Edinburg, 2002.
- [17] Druin A. Designing a Digital Library for Young Children, Association for Computing Machinery. Proceedings of the ACM Conference, Virginia, 2002.

- [18] Sastry, Hanumat G., and Lokanatha C. Reddy. "User Interface Design Principles for Digital Libraries." *International Journal of Web Applications*, 1 (2) (2009).
- [19] Maness, J. M. (2006), 'Library 2.0 Theory: Web 2.0 and Its Implications for Libraries', MLS, University of Colorado at Boulder Libraries, 1720 Pleasant St., Boulder, CO, USA, Available from <http://www.webology.ir/2006/v3n2/a25.html#24>
- [20] Murugesan, S. (2007), 'Understanding Web 2.0', IT Professional 9(4), 34-41
- [21] O'Reilly, T. (2005), 'What is web 2.0?', Available from <http://oreilly.com/web2/archive/what-is-web-20.html>
- [22] Casey, M. (2006), 'Born in the biblioblogosphere', LibraryCrunch, διαθέσιμο στον σύνδεσμο http://www.librarycrunch.com/2006/01/post_1.html
- [23] Habib, M. (2006), 'Toward Academic Library 2.0: Development and Application of a Library 2.0 Methodology', Master's thesis, University of North Carolina at Chapel Hill, School of Information and Library Science
- [24] Chalon, P. X., Di Pretoro, E. and Kohn, L. (2008), 'OPAC 2.0: Opportunities, development and analysis', 11th European Conference of Medical and Health Libraries
- [25] Gavrilis, D., Kakali, C. and Papatheodorou, C. (2008), 'Enhancing Library Services with Web 2.0 Functionalities', Proceedings of the 12th European Conference Research and Advanced Technology for Digital Libraries, ECDL 2008, Aarhus, Denmark, September 2008, Lecture Notes in Computer Science (LNCS) No. 5173: Springer-Verlag, pp. 148-159
- [26] Mitrelis, A., Papachristopoulos, L., Tsakonas G., Papatheodorou, C. (2007), 'Requirement Investigation for Digital Library Services through Qualitative Methodological Approaches', Proceedings of 16th Panhellenic Conference of Academic Libraries
- [27] Ginsparg, Paul (2011). "It was twenty years ago today ..". **arXiv**:1108.2700 [cs.DL].
- [28] Caffaro, Jerome, and Samuele Kaplun. *Invenio: A modern digital library for grey literature*. No. CERN-OPEN-2010-027. 2010.
- [29] Giles, C. Lee, Kurt D. Bollacker, and Steve Lawrence. "CiteSeer: An automatic citation indexing system." *Proceedings of the third ACM conference on Digital libraries*. ACM, 1998.
- [30] Li, Huajing, et al. "CiteSeerx: an architecture and web service design for an academic document search engine." *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006.
- [31] Schlanger, Margo, and Denise Lieberman. "Using Court Records for Research, Teaching, and Policymaking: The Civil Rights Litigation Clearinghouse." *University of Missouri-Kansas City Law Review* (2006).
- [32] Berners-Lee, Tim, Robert Cailliau, and Ari Luotonen. "Henrik Frystyk Nielsen, and Arthur Secret." *The world-wide web. Communications of the ACM* 37.8 (1994): 76-82.
- [33] Lanagan, J., Smeaton, F. (2012), Video digital libraries: contributive and decentralized
- [34] Meghini, M., Spiratos, N. and Yang, J. (2010), 'A data model for digital libraries', *International Journal on Digital Libraries*, Vol.11, No.1, pp.41-56

- [35] Marketakis, Y. and Tzitzikas, Y. (2009), 'Dependancy management for digital preservation using semantic web technologies', *International Journal on Digital Libraries*, Vol.10, No.4, pp. 135-157
- [36] Koutsomitropoulos, A., Solomou, G., Alexopoulos, A. and Papatheodorou, T. (2009), 'Semantic Web enabled digital repositories', *International Journal on Digital Libraries*, Vol.10, No.4, pp.179-199
- [37] Langxian, D., Junxia, Q. and Pengfei, G. (2012), *The Application of Semantics Web in DigitalLibrary Knowledge Management*
- [38] García-Crespo, Á., Gómez-Berbís, J.M., Colomo-Palacios, R. and García-Sánchez, F. (2012), *Digital libraries and Web 3.0. The CallimachusDL approach*
- [39] Romero, N. (2012), "Cloud computing" in library automation. Benefits and drawbacks.", *Bottom Line: Managing Library Finances*, The, Vol. 25 Iss: 3
- [40] "Android Overview". Open Handset Alliance. Retrieved 2012-02-15.
- [41] "Industry Leaders Announce Open Platform for Mobile Devices" (Press release). Open Handset Alliance. November 5, 2007. Retrieved 2012-02-17.
- [42] Developer Economics Q3 2013 analyst report—<http://www.visionmobile.com/DevEcon3Q13> – Retrieved July 2013
- [43] Android Application Framework , ανακτήθηκε από την διεύθυνση <http://developer.android.com/about/index.html> στις 14/3/2014.
- [44] Butler, Timothy; Coleman, David: "Models of Collaboration", September 2003.
- [45] http://www.collaborate.com/publication/newsletter/publications_newsletter_september03.html
- [46] Shustitskiy, Maria. "Collaboration of digital libraries." *Vienna University of Economics and Business Administration, Vienna* (2004).
- [47] Saltzer, Jerome H., and Michael D. Schroeder. "The protection of information in computer systems." *Proceedings of the IEEE* 63.9 (1975): 1278-1308.
- [48] "Distributed Application Architecture". Sun Microsystem. Retrieved 2009-06-16.
- [49] Jing, Jin, Abdelsalam Sumi Helal, and Ahmed Elmagarmid. "Client-server computing in mobile environments." *ACM computing surveys (CSUR)* 31.2 (1999): 117-157.
- [50] <http://martinfowler.com/eaDev/uiArchs.html> προσπελάθηκε 21/3/2014
- [51] Reenskaug, Trygve, and James O. Coplien. "The DCI architecture: A new vision of object-oriented programming." *An article starting a new blog:(14pp)* http://www.artima.com/articles/dci_vision.html (2009).
- [52] Burbeck, Steve. "Applications programming in smalltalk-80 (tm): How to use model-view-controller (mvc)." *Smalltalk-80 v2 5* (1992).

- [53] <http://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design> προσπελάστηκε στις 21/3/2014
- [54] Buschmann, Frank, Kelvin Henney, and Douglas Schimdt. *Pattern-oriented Software Architecture: On Patterns and Pattern Language*. Vol. 5. John Wiley & Sons, 2007.
- [55] Leff, Avraham, and James T. Rayfield. "Web-application development using the model/view/controller design pattern." *Enterprise Distributed Object Computing Conference, 2001. EDOC'01. Proceedings. Fifth IEEE International*. IEEE, 2001.
- [56] Rahimi, Bahlol, and Vivian Vimarlund. "Methods to evaluate health information systems in healthcare settings: a literature review." *Journal of medical systems* 31.5 (2007): 397-432.
- [57] Sauro, Jeff, and Joseph S. Dumas. "Comparison of three one-question, post-task usability questionnaires." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009.
- [58] Stephan, Paula E., et al. "Who's patenting in the university? Evidence from the survey of doctorate recipients." *Econ. Innov. New Techn.* 16.2 (2007): 71-99.
- [59] Tang, Jie, et al. "ArnetMiner: An Expertise Oriented Search System for Web Community." *Semantic Web Challenge*. 2007.
- [60] Mitropoulos Sarandis, Baltasis George, Rodios Michalis, Douligeris Christos. SocialLib: A Collaborative Digital Library Model Platform Using Web 2.0, Accepted in The Electronic Library by Emerald Insight, to be published in Vol.32 Iss.5 (2014)

Παράρτημα 1

Α.Επεξήγηση κώδικα

Α.1.Κώδικας εφαρμογής διαχειριστών

Η εφαρμογή των διαχειριστών της ψηφιακής βιβλιοθήκης, είναι γραμμένη στην γλώσσα PHP και χρησιμοποιεί την βιβλιοθήκη JQuery, της JavaScript. Χρησιμοποιήθηκε επίσης η γλώσσα φύλλων στυλ CSS (Cascading Style Sheets) για τον έλεγχο της εμφάνισης της κεντρικής εφαρμογής καθώς και η τεχνική AJAX (Asynchronous Javascript and XML) για την επικοινωνία με την βάση δεδομένων MySQL.

Παρακάτω θα δούμε κάποια από τα βασικά αρχεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής των διαχειριστών.

Το αρχείο που εκτελείται κατά την εκκίνηση της εφαρμογής είναι το *login.php* το οποίο είναι η σελίδα στην οποία ο διαχειριστής αυθεντικοποιεί τον εαυτό του με την χρήση του συνθηματικού του. Το αρχείο αυτό χρησιμοποιεί την βιβλιοθήκη PDO για να επικοινωνήσει με την βάση. Ο κώδικας δίνεται παρακάτω.

login.php

Στην αρχή του αρχείου υπάρχει ένα block κώδικα PHP το οποίο χρησιμοποιεί την μέθοδο HTTP POST για να πάρει τα στοιχεία που χρησιμοποιεί ο διαχειριστής για να αυθεντικοποιηθεί. Στη συνέχεια αρχικοποιεί ένα αντικείμενο τύπου PDO το οποίο πραγματοποιεί την σύνδεση με την βάση δεδομένων. Μέσω αυτού ελέγχει τα στοιχεία του χρήστη και αν αυτά είναι έγκυρα τον οδηγεί στην κεντρική σελίδα της εφαρμογής.

Στο HTML κομμάτι της εφαρμογής, ορίζονται τα δομικά στοιχεία που απαρτίζουν την φόρμα εισόδου.

```

session_start();
if($_SERVER["REQUEST_METHOD"] == "POST")
{
    // username and password sent from Form
    $myusername=addslashes($_POST['username']);
    echo $myusername;
    $mypassword=addslashes($_POST['password']);
    $db = new PDO('mysql:host=localhost;dbname=librappc_librapp;charset=utf8', 'librappc', 'pass');
    //$con = new DBConnect();
    //$dbh = $db->getConnection();
    $sql = $db -> prepare("select uid from users where username = ? and password = ?");
    $sql -> execute(array($myusername, $mypassword));
    if ($row = $sql->fetch(PDO::FETCH_ASSOC)) {
        $count = 1;
    }
    else {
        $count = 0;
    }
    print_r($db->errorInfo());
    $db= null;

    // If result matched $myusername and $mypassword, table row must be 1 row
    if($count==1)
    {
        session_register("myusername");
        $_SESSION["login_user"]=$myusername;
        header("location: librapp_management.php");
    }
    else
    {
        $error="Your Login Name or Password is invalid";
    }
}

```

}

lock.php

Αρχείο το οποίο χρησιμοποιείται για να κρατάει συνδεδεμένο τον χρήστη. Χρησιμοποιείται σαν include σε όλα.php αρχεία είναι απαραίτητο.

```
session_start();
$user_check=$_SESSION['login_user'];

$dbh = new PDO('mysql:host=localhost;dbname=librappc_librapp;charset=utf8', 'librappc', 'password');
$sql = $dbh -> prepare("select username from users where username =".$user_check."");
$sql -> execute();
$row = $sql->fetch(PDO::FETCH_ASSOC);

$login_session=$row['USERNAME'];

if(!isset($login_session))
{
    header("Location: login.php");
}
```

proc_management.php

Αυτό είναι το κεντρικό αρχείο που εκτελεί τις διαχειριστικές λειτουργίες της εφαρμογής. Χωρίζεται σε δύο κομμάτια, το HTML κομμάτι το οποίο ορίζει την δομή της σελίδας, και το κομμάτι JQuery, το οποίο εκτελεί τις λειτουργίες που ζητούνται. Παρακάτω θα δούμε μερικά παραδείγματα κώδικα JQuery.

Στα document ready functions, εκτελούνται οι λειτουργίες που πρέπει να γίνουν κατά την εκκίνηση της σελίδας. Όπως για παράδειγμα, η συμπλήρωση των πινάκων που διαθέτουν τους χρήστες της εφαρμογής και τα αντικείμενα που είναι αποθηκευμένα στην βάση.

```
$( document ).ready(function() {
    var auto_refreshs_contents = 2;

    var uname = '<?php echo $login_session?>';
    var authn = '<?php echo $_SESSION['admin_user']?>';
    $.ajax({
        url: "findAllUsers.php",
        success: function(data) {
            var tmp = data.split("^");
            count = tmp.length;
            $('#folders_table').empty();
            var spool = "";
            if (count >= 16) {
                for (var index =0; index < count; index++)
                {
                    if (index%2 == 0)
                    {
                        spool+= '<tr>';
                    }
                    else
                    {
                        spool+= '<tr class =\"alt\">';
                    }

                    spool+= '<td>'+tmp[index]+'</td>';
                    spool+= '</tr>';
                }
            }
        }
    });
```

Ο παραπάνω κώδικας δείχνει πως το JQuery, εκτελεί ένα αίτημα ajax για να επικοινωνήσει με την βάση δεδομένων χρησιμοποιώντας ένα.php αρχείο που ουσιαστικά χρησιμοποιεί το PDO με τον τρόπο που είδαμε προηγουμένως και τροφοδοτεί τα αποτελέσματα

στο αίτημα ajax, το οποίο και τα επιστρέφει στο JQuery, ώστε να γίνει populate ο πίνακας. Με αντίστοιχο τρόπο εκτελούνται και οι υπόλοιπες διαχειριστικές λειτουργίες.

A.2.Κώδικας ενδιάμεσου λογισμικού

Ο κώδικας του ενδιάμεσου λογισμικού είναι επίσης γραμμένος στην γλώσσα PHP. Για κάθε αίτηση από την εφαρμογή υπάρχει και μία λειτουργία του ενδιάμεσου λογισμικού, η οποία είναι σχεδιασμένη να την εξυπηρετεί, είτε επικοινωνώντας με την βάση, είτε πραγματοποιώντας κάποιο push service, μέσω του GCM, είτε γράφοντας και διαβάζοντας από τον FTP.

Τα αρχεία config.php και DB_Connect.php, χρησιμοποιούνται από σχεδόν όλα τα αρχεία του middleware. Η ανάλυσή τους φαίνεται παρακάτω.

config.php

Αυτό το αρχείο κώδικα περιέχει τα απαραίτητα στοιχεία για την σύνδεση με την βάση και το Google API key που είναι απαραίτητο για την εγγραφή της συσκευής ώστε να λαμβάνει push messages.

```
define("DB_HOST", "localhost");
define("DB_USER", "librappc");
define("DB_PASSWORD", "password");
define("DB_DATABASE", "librappc_librapp");

/*
 * Google API Key
 */
define("GOOGLE_API_KEY", "AIzaSyBQ81K7ofp3vulENfQM1QdF-YTbVN_kdc"); // Place your Google
API Key
```

DB_Connect.php

Πραγματοποιεί την σύνδεση με την βάση δεδομένων καλώντας το config.php.

```
class DB_Connect {

    // constructor
    function __construct() {

    }

    // destructor
    function __destruct() {
        // $this->close();
    }

    // Connecting to database
    public function connect() {
        require_once 'include/config.php';
        // connecting to mysql
        $con = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
        // selecting database
        mysql_select_db(DB_DATABASE);

        // return database handler
        return $con;
    }

    // Closing database connection
    public function close() {
        mysql_close();
    }
}
```

}

DB_Functions.php

Περιέχει τις απαραίτητες συναρτήσεις για την επικοινωνία με την βάση. Πραγματοποιεί λειτουργίες αυθεντικοποίησης, ελέγχου εγκυρότητας e-mail κ.α. Για παράδειγμα, η συνάρτηση `random_string` που ορίζεται σε αυτό το αρχείο, χρησιμοποιείται όταν ζητηθεί νέος κωδικός από τον χρήστη, σε περίπτωση που τον έχει ξεχάσει. Δημιουργείται ένα τυχαίο αλφαριθμητικό, το οποίο επιστρέφεται στο `middleware`, το οποίο με τη σειρά του το αποστέλλει στην κινητή συσκευή.

```
public function random_string()
{
    $character_set_array = array();
    $character_set_array[] = array('count' => 7, 'characters' => 'abcdefghijklmnopqrstuvwxyz');
    $character_set_array[] = array('count' => 1, 'characters' => '0123456789');
    $temp_array = array();
    foreach ($character_set_array as $character_set) {
        for ($i = 0; $i < $character_set['count']; $i++) {
            $temp_array[] = $character_set['characters'][rand(0, strlen($character_set['characters']) - 1)];
        }
    }
    shuffle($temp_array);
    return implode("", $temp_array);
}
```

Η συνάρτηση `storeUser`, χρησιμοποιείται για να αποθηκεύσει έναν νέο χρήστη στην βάση δεδομένων. Λαμβάνει όλα τα στοιχεία του χρήστη ως παράμετρο και στην συνέχεια τα αποθηκεύει εκτελώντας ένα `mysql query`.

```
public function storeUser($fname, $lname, $email, $username, $password, $regld) {
    $uuid = uniqid("", true);
    $hash = $this->hashSSHA($password);
    $encrypted_password = $hash["encrypted"]; // encrypted password
    $salt = $hash["salt"]; // salt
    $result = mysql_query("INSERT INTO users(unique_id, firstname, lastname, email, username,
    encrypted_password, salt, created_at, gcm_regid) VALUES('$uuid', '$fname', '$lname', '$email', '$username',
    '$encrypted_password', '$salt', NOW(), '$regld')");
    // check for successful store
    if ($result) {
        // get user details
        $uid = mysql_insert_id(); // last inserted id
        $result = mysql_query("SELECT * FROM users WHERE uid = $uid");
        // return user details
        return mysql_fetch_array($result);
    } else {
        return false;
    }
}
```

Για να μην εισαχθούν λάθος στοιχεία στην βάση δεδομένων, είναι απαραίτητο να εκτελεστούν κάποιοι έλεγχοι, όπως για παράδειγμα αν το e-mail το οποίο εισήγαγε ο χρήστης είναι της αποδεκτής μορφής, ή αν το αναγνωριστικό χρήστη υπάρχει ήδη στην βάση. Ορίζονται συγκεκριμένες συναρτήσεις η οποίες εκτελούν τους παραπάνω ελέγχους, όπως η `validEmail` και η `isUserExisted`. Η πρώτη εκτελεί διάφορους ελέγχους στο e-mail με την μορφή `regular expression`, όπως φαίνεται παρακάτω.

```
$atIndex = strrpos($email, "@");
```

```
$domain = substr($email, $atIndex+1);
```

```
$local = substr($email, 0, $atIndex);
```

```

preg_match('/^[A-Za-z0-9\-\.\_]+$/'
preg_match('/^\(\|\|\.[A-Za-z0-9!#%&`_=\|$\!*+?^{|~.-})+$/'
public function isUserExisted($email) {
    $result = mysql_query("SELECT email from users WHERE email = '$email'");
    $no_of_rows = mysql_num_rows($result);
    if ($no_of_rows > 0) {
        // user existed
        return true;
    } else {
        // user not existed
        return false;
    }
}
}

```

GCM.php

Αυτό το αρχείο ασχολείται με την εγγραφή της συσκευής στο σύστημα Google Cloud Messaging, ώστε να μπορεί να λαμβάνει push messages, και την αποστολή των μηνυμάτων. Για να μπορεί ο κάθε χρήστης να λαμβάνει ειδοποιήσεις από την εφαρμογή στη συσκευή του, όπως για παράδειγμα αιτήματα φιλίας ή ενημερώσεις από την εφαρμογή, πρέπει η συσκευή να εγγραφεί στο GCM service της Google, χρησιμοποιώντας το μοναδικό GCM id της εφαρμογής και ένα κλειδί για την κάθε συσκευή.

Για παράδειγμα, όταν θέλουμε να στείλουμε ένα μήνυμα, θα πρέπει να διαθέτουμε το κλειδί και να πραγματοποιήσουμε κλήση της υπηρεσίας GCM, όπως φαίνεται στην μέθοδο send_notification.

```

public function send_notification($registatoin_ids, $message) {
    include_once 'include/config.php';

    $url = 'https://android.googleapis.com/gcm/send';

    $fields = array(
        'registration_ids' => $registatoin_ids,
        'data' => $message,
    );

    $headers = array(
        'Authorization: key=' . GOOGLE_API_KEY,
        'Content-Type: application/json'
    );

    $ch = curl_init();

    curl_setopt($ch, CURLOPT_URL, $url);

    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));

    $result = curl_exec($ch);
    if ($result === FALSE) {
        die('Curl failed: ' . curl_error($ch));
    }

    // Close connection
    curl_close($ch);
    echo $result;
}
}

```


}

index.php

Το *index.php* είναι το αρχείο *php*, το οποίο διαχειρίζεται αρκετές από τις λειτουργίες του ενδιαμέσου λογισμικού, όπως την αυθεντικοποίηση και την εγγραφή νέου χρήστη ή την αίτηση για εγγραφή στην λειτουργία GCM.

Μπορεί να εκτελέσει μία μεγάλη σειρά από λειτουργίες όπως έλεγχος για ύπαρξη χρήστη:

```
$user = $db->getUserByEmailAndPassword($email, $password);
```

Ανάκτηση κωδικού:

```
$forgotpassword = $_POST['forgotpassword'];
```

```
$randomcode = $db->random_string();
```

```
$hash = $db->hashSHA($randomcode);
```

```
    $encrypted_password = $hash["encrypted"]; // encrypted password
```

```
    $salt = $hash["salt"];
```

```
    $subject = "Password Recovery";
```

```
    $message = "Hello User,\n\nYour Password is successfully changed. Your new Password is $randomcode . Login with your new Password and change it in the User Panel.\n\nRegards,\nLibrApp Team.";
```

Το παρακάτω κομμάτι κώδικα παραγματοποιεί αυθεντικοποίηση χρήστη με ταυτόχρονη δημιουργία *encrypted password* σε συνεργασία με την κινητή εφαρμογή. Στο παρακάτω παράδειγμα φαίνεται και η χρήση του *Json* για την προετοιμασία του μηνύματος το οποίο θα αποσταλεί από την εφαρμογή στην μεριά της κινητής συσκευής ή θα ληφθεί από αυτήν:

```
$email = $_POST['email'];
```

```
$password = $_POST['password'];
```

```
$user = $db->getUserByEmailAndPassword($email, $password);
```

```
if ($user != false) {
```

```
    $response["success"] = 1;
```

```
    $response["user"]["fname"] = $user["firstname"];
```

```
    $response["user"]["lname"] = $user["lastname"];
```

```
    $response["user"]["email"] = $user["email"];
```

```
    $response["user"]["uname"] = $user["username"];
```

```
    $response["user"]["uid"] = $user["unique_id"];
```

```
    $response["user"]["created_at"] = $user["created_at"];
```

```
    echo json_encode($response);
```

```
}
```

```
else {
```

```
    $response["error"] = 1;
```

```
    $response["error_msg"] = "Incorrect email or password!";
```

```
    echo json_encode($response);
```

```
}
```

A.3.Κώδικας εφαρμογής κινητής συσκευής

Η εφαρμογή του κινητού είναι υλοποιημένη με τη βοήθεια της γλώσσας προγραμματισμού *Java* και τη βοήθεια αρχείων *xml*, τα οποία ουσιαστικά αποτελούν αυτό που εμφανίζεται από την πλατφόρμα *Android*. Η εφαρμογή αποτελείται από αρχεία *.java* και αρχεία *.xml*. Συγκεκριμένα

αποτελείται από:

28 κλάσεις Java χωρισμένες σε 3 πακέτα:

- Πακέτο com.example.libraryapp
 - AcceptActivity.java
 - AlertDialogManager.java
 - ConnectionDetector.java
 - DatePickerFragment.java
 - FriendListActivity.java
 - GCMIntentService.java
 - HTTPClientConnect.java
 - LibrApp.java
 - LoginScreen.java
 - Main.java
 - MessagingActivity.java
 - NewProjectActivity.java
 - NewTaskActivity.java
 - ObjectDetailsActivity.java
 - PasswordRetrieve.java
 - PDBActivity.java
 - ProgManagementActivity.java
 - ProjectPageActivity.java
 - RegisterScreen.java
 - SearchObjectActivity.java
 - TaskListPageActivity.java
 - TaskManagementActivity.java
 - UserManagementActivity.java
 - UserMenu.java
- Πακέτο com.example.libraryapp.book_handle
 - BookHandler.java
- Πακέτο com.example.libraryapp.php_con
 - DatabaseHandler.java
 - JSONParser.java
 - UserFunctions.java

1 φάκελο drawable με 13 αρχεία εικόνας

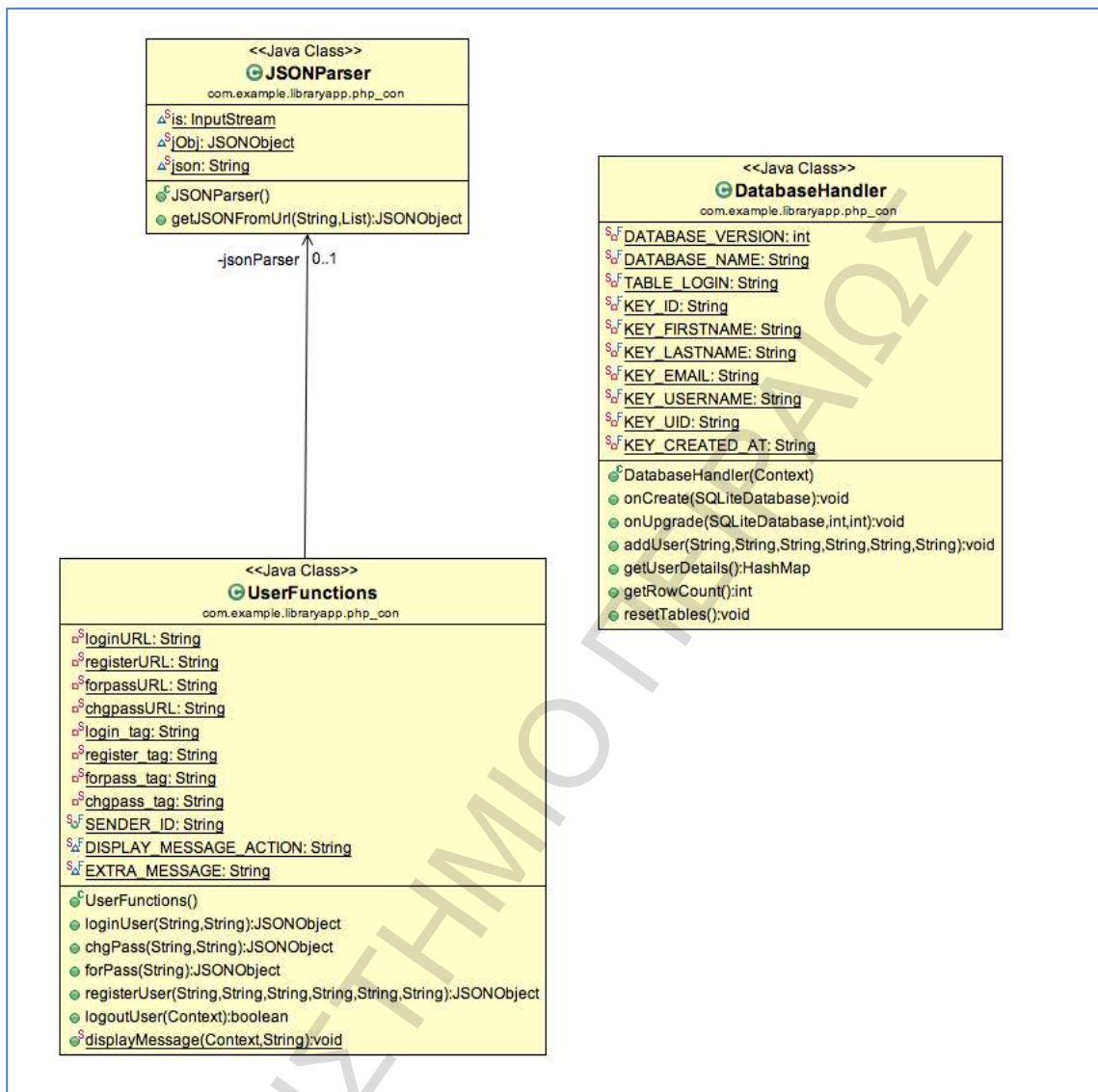
- book.jpg
- butl.png
- button_net_p.png
- button_net.png
- download.jpg
- ic_action_search.png
- ic_launcher.png
- la_launcher.png
- log_but.xml
- logbut.png
- logbut2.png
- logbut3.png
- main.png

1 φάκελο layout με 26 αρχεία .xml

- activity_accept.xml
- activity_friend_list.xml
- activity_libr_app.xml
- activity_login_screen.xml
- activity_main.xml
- activity_messaging.xml
- activity_new_project.xml
- activity_new_task.xml

- activity_object_details.xml
 - activity_password_retrieve.xml
 - activity_pdb.xml
 - activity_prog_management.xml
 - activity_project_page.xml
 - activity_register_screen.xml
 - activity_search_object.xml
 - activity_task_list_page.xml
 - activity_task_management.xml
 - activity_user_management.xml
 - activity_user_menu.xml
 - list_item.xml
 - list_item2.xml
 - list_item3.xml
 - list_item4.xml
 - list_item5.xml
 - list_item6.xml
 - list_item7.xml
- 1 φάκελο menu με 20 αρχεία .xml
- accept.xml
 - activity_libr_app.xml
 - activity_login_screen.xml
 - activity_main.xml
 - activity_splash.xml
 - friend_list.xml
 - messaging.xml
 - new_project.xml
 - new_task.xml
 - object_details.xml
 - password_retrieve.xml
 - pdb.xml
 - prog_management.xml
 - project_page.xml
 - register_screen.xml
 - search_object.xml
 - task_list_page.xml
 - task_management.xml
 - user_management.xml
 - user_menu.xml
- 1 φάκελο values με 4 αρχεία .xml
- colors.xml
 - dimens.xml
 - strings.xml
 - styles.xml

Για να κατανοήσουμε καλύτερα το πώς λειτουργούν κάποια από τα κομμάτια της εφαρμογής, παρακάτω παρατίθενται δύο διαγράμματα κλάσης. Αυτό που απεικονίζει τις βοηθητικές λειτουργίες σύνδεσης με τον απομακρυσμένο διακομιστή και αυτό που απεικονίζει το πώς γίνονται οι λειτουργίες login και register.



Σχήμα 30 Διάγραμμα κλάσης των λειτουργιών με τον server

DatabaseHandler.java

Η κλάση αυτή χρησιμοποιείται για να διαχειρίζεται την εσωτερική βάση δεδομένων SQLite της συσκευής. Πραγματοποιεί λειτουργίες, όπως δημιουργία ή ενημέρωση χρήστη κ.α.

Επειδή υπερφορτώνει την κλάση `SQLiteOpenHelper`, είναι υποχρεωτική να υλοποιεί τις μεθόδους `onCreate` και `onUpgrade`.

@Override

```

public void onCreate(SQLiteDatabase db) {
    String CREATE_LOGIN_TABLE = "CREATE TABLE " + TABLE_LOGIN + "("
        + KEY_ID + " INTEGER PRIMARY KEY,"
        + KEY_FIRSTNAME + " TEXT,"
        + KEY_LASTNAME + " TEXT,"
        + KEY_EMAIL + " TEXT UNIQUE,"
        + KEY_USERNAME + " TEXT,"
        + KEY_UID + " TEXT,"
        + KEY_CREATED_AT + " TEXT" + ")";
    db.execSQL(CREATE_LOGIN_TABLE);
}
  
```

```

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    // Drop older table if existed
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_LOGIN);

    // Create tables again
    onCreate(db);
}

```

Για την δημιουργία νέου χρήστη χρησιμοποιείται η μέθοδος `addUser`, η οποία αφού λάβει ένα στιγμιότυπο της βάσης μέσω της μεθόδου `getWritableDatabase` του πακέτου χειρισμού της SQLite, θα κάνει εγγραφή των στοιχείων στην τοπική βάση.

```

public void addUser(String fname, String lname, String email, String uname, String uid, String created_at) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(KEY_FIRSTNAME, fname); // FirstName
    values.put(KEY_LASTNAME, lname); // LastName
    values.put(KEY_EMAIL, email); // Email
    values.put(KEY_USERNAME, uname); // UserName
    values.put(KEY_UID, uid); // Email
    values.put(KEY_CREATED_AT, created_at); // Created At

    // Inserting Row
    db.insert(TABLE_LOGIN, null, values);
    db.close(); // Closing database connection
}

```

JsonParser.java

Η κλάση αυτή επεξεργάζεται αιτήματα JSON από και προς την εφαρμογή. Για να λάβει κάποιο JSON από τον απομακρυσμένο server, εκτελείται η παρακάτω μέθοδος.

```

public JSONObject getJSONFromUrl(String url, List params) {

    // Making HTTP request
    try {
        // defaultHttpClient
        DefaultHttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(url);
        httpPost.setEntity(new UrlEncodedFormEntity(params));

        HttpResponse httpResponse = httpClient.execute(httpPost);
        HttpEntity httpEntity = httpResponse.getEntity();
        is = httpEntity.getContent();

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

try {
    BufferedReader reader = new BufferedReader(new InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
}

```

```

        is.close();
        json = sb.toString();
        Log.e("JSON", json);
    } catch (Exception e) {
        Log.e("Buffer Error", "Error converting result " + e.toString());
    }

    // try parse the string to a JSON object
    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e("JSON Parser", "Error parsing data " + e.toString());
    }

    // return JSON String
    return jsonObj;
}
}

```

Η λειτουργία της είναι αρκετά απλή. Προσπαθεί να ανοίξει το αρχείο JSon από την απομακρυσμένη τοποθεσία και στην συνέχεια να το κάνει parse και να το επιστρέψει στην κλάση που το κάλεσε. Πάντα υπάρχουν τα αντίστοιχα μηνύματα σφάλματος για κάθε περίπτωση.

UserFunctions.java

Η κλάση αυτή εκτελεί ενέργειες που ζητάει ο χρήστης όπως login, register κλπ. Για κάθε αίτημα δημιουργείται ένα JSon request, το οποίο περιέχει τα απαραίτητα για τον διακομιστή στοιχεία ώστε να εξυπηρετηθεί από το ενδιάμεσο λογισμικό. Στην αρχή γίνεται δήλωση των βασικών στοιχείων που χρειάζεται να ξέρει η κλάση σχετικά με τον server.

```

private JSONParser jsonParser;

//URL of the PHP API
private static String loginURL = "http://librapp.com/librapp_login_api/";
private static String registerURL = "http://librapp.com/librapp_login_api/";
private static String forpassURL = "http://librapp.com/librapp_login_api/";
private static String chgpassURL = "http://librapp.com/librapp_login_api/";

private static String login_tag = "login";
private static String register_tag = "register";
private static String forpass_tag = "forpass";
private static String chgpass_tag = "chgpass";

public static final String SENDER_ID = "213751505438";

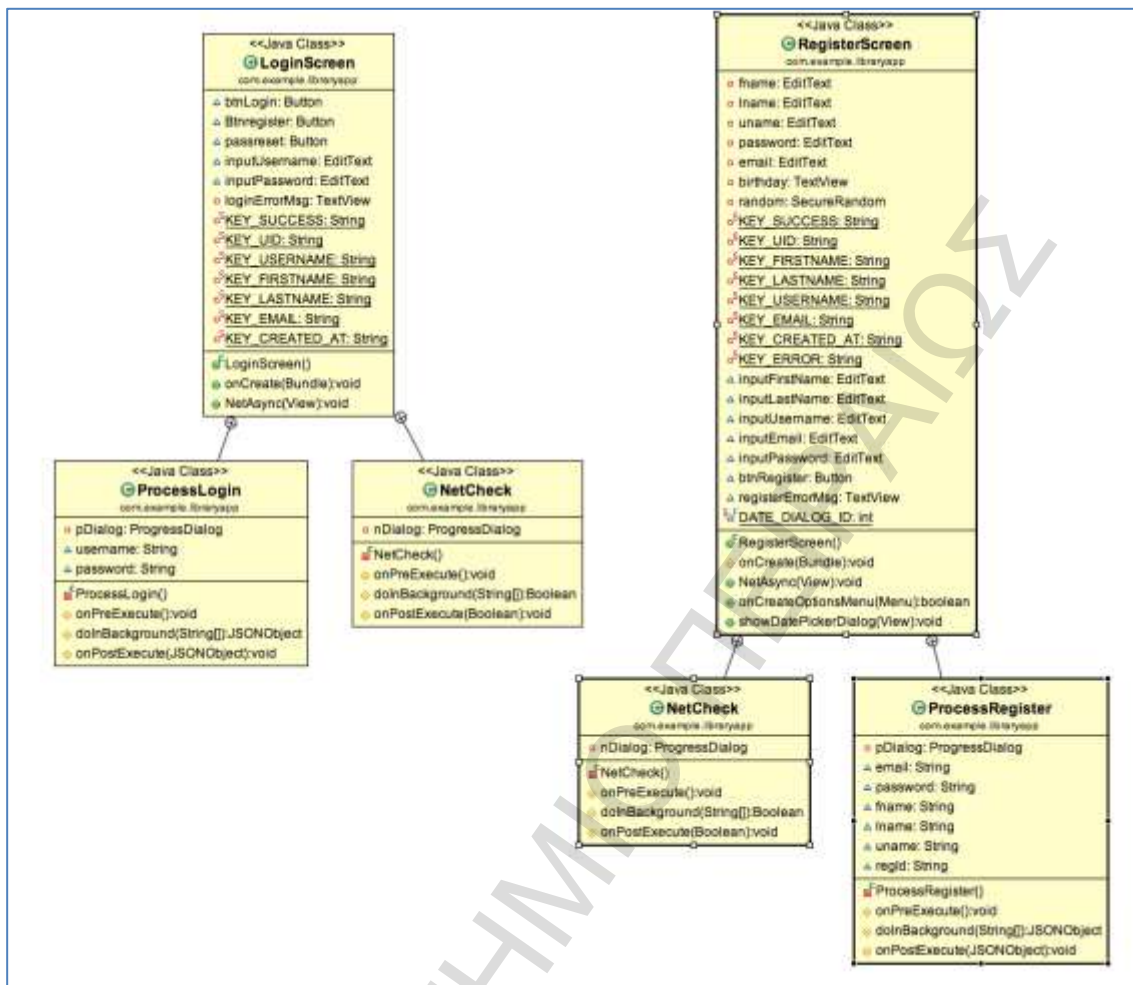
```

Για παράδειγμα, η λειτουργία login, δημιουργεί μία λίστα με τα απαραίτητα στοιχεία, την οποία την περνάει σε ένα JSon request.

```

public JSONObject loginUser(String email, String password){
    // Building Parameters
    List params = new ArrayList();
    params.add(new BasicNameValuePair("tag", login_tag));
    params.add(new BasicNameValuePair("email", email));
    params.add(new BasicNameValuePair("password", password));
    JSONObject json = jsonParser.getJSONFromUrl(loginURL, params);
    return json;
}

```



Σχήμα 31 Διάγραμμα κλάσης για της λειτουργίες login και register

LoginScreen.java

Η κλάση αυτή λειτουργεί ως controller για τις λειτουργίες της αρχικής οθόνης. Δηλαδή, υλοποιεί λειτουργίες οι οποίες διέπουν το login_screen_activity.xml, δηλαδή την οθόνη της εφαρμογής που λειτουργεί ως Home Page.

Στην μέθοδο onCreate, γίνεται σύνδεση με τα στοιχεία της οθόνης (κουμπιά κλπ), και ορίζονται listeners για γεγονότα όπως το πάτημα ενός κουμπιού.

```

inputUsername = (EditText) findViewById(R.id.username);
inputPassword = (EditText) findViewById(R.id.password);
Btnregister = (Button) findViewById(R.id.button2);
btnLogin = (Button) findViewById(R.id.btnToLogin);
passreset = (Button) findViewById(R.id.passres);
loginErrorMsg = (TextView) findViewById(R.id.loginErrorMsg);

//On click event for password retrieve
passreset.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        Intent myIntent = new Intent(view.getContext(), PasswordRetrieve.class);
        startActivityForResult(myIntent, 0);
        finish();
    }
});

```

Η επικοινωνία με τον server θα εκτελεστεί μέσω μίας ασύγχρονης κλάσης, η οποία κληρονομεί από την κλάση AsyncTask, και χρησιμοποιείται για να εκτελέσει σε παράλληλο νήμα την λειτουργία επικοινωνίας, χωρίς να επηρεάζει την εφαρμογή. Επειδή τα δεδομένα με τα οποία θα γίνει αλληλεπίδραση δεν είναι πολλά προτιμάται η χρήση της AsyncTask, από την κλάση Thread. Παρακάτω δίνεται ένα παράδειγμα.

```
//Async task that checks whether internet connection is working
private class NetCheck extends AsyncTask<String, Void, Boolean>
{
    private ProgressDialog nDialog;

    @Override
    protected void onPreExecute(){
        super.onPreExecute();
        nDialog = new ProgressDialog(LoginScreen.this);
        nDialog.setTitle("Checking Network");
        nDialog.setMessage("Loading..");
        nDialog.setIndeterminate(false);
        nDialog.setCancelable(true);
        nDialog.show();
    }

    @Override
    protected Boolean doInBackground(String... args){
        //Checks the connection by hitting Google
        ConnectivityManager cm = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo netInfo = cm.getActiveNetworkInfo();
        if (netInfo != null && netInfo.isConnected()) {
            try {
                URL url = new URL("http://www.google.com");
                HttpURLConnection urlc = (HttpURLConnection) url.openConnection();
                urlc.setConnectTimeout(3000);
                urlc.connect();
                if (urlc.getResponseCode() == 200) {
                    return true;
                }
            } catch (MalformedURLException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
        return false;
    }

    @Override
    protected void onPostExecute(Boolean th){
        if(th == true){
            nDialog.dismiss();
            new ProcessLogin().execute();
        }
        else{
            nDialog.dismiss();
            loginErrorMsg.setText("Error in Network Connection");
        }
    }
}
}
```


Β.Σχεδίαση Βάσης Δεδομένων

Η βάση δεδομένων της εφαρμογής είναι σχετικά απλή και αποτελείται από 8 πίνακες.

Πίνακας *users*

Κρατάει τα προφίλ των χρηστών

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	uid	int(11)			No	None	AUTO_INCREMENT
2	unique_id	varchar(23)	latin1_swedish_ci		No	None	
3	firstname	varchar(50)	latin1_swedish_ci		No	None	
4	lastname	varchar(50)	latin1_swedish_ci		No	None	
5	username	varchar(20)	latin1_swedish_ci		No	None	
6	email	varchar(100)	latin1_swedish_ci		No	None	
7	encrypted_password	varchar(80)	latin1_swedish_ci		No	None	
8	salt	varchar(10)	latin1_swedish_ci		No	None	
9	created_at	datetime			Yes	NULL	
10	updated_at	datetime			Yes	NULL	
11	gcm_regid	text	latin1_swedish_ci		No	None	
12	auth	int(11)			No	0	

Πίνακας *user_connections*

Αποθηκεύει τις συνδέσεις μεταξύ χρηστών

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	user_id	int(11)			No	None	
2	connection_id	int(11)			No	None	

Πίνακας *books*

Αποθηκεύει πληροφορίες σχετικά με τα αντικείμενα της εφαρμογής.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	book_id	int(11)			No	None	AUTO_INCREMENT
2	book_name	varchar(200)	latin1_swedish_ci		No	None	
3	author	varchar(300)	latin1_swedish_ci		No	None	
4	publisher	varchar(300)	latin1_swedish_ci		No	None	
5	year_published	int(11)			No	None	
6	ISBN-10	varchar(200)	latin1_swedish_ci		No	None	
7	ISBN-13	varchar(200)	latin1_swedish_ci		No	None	
8	edition	int(11)			No	None	
9	book_type	varchar(30)	latin1_swedish_ci		No	None	
10	stars	int(11)			No	None	
11	info	varchar(1000)	latin1_swedish_ci		No	None	
12	tags	varchar(600)	latin1_swedish_ci		No	None	
13	file_id	int(11)			No	None	

Πίνακας *user_projects*

Διαθέτει πληροφορίες για τα έργα που έχουν δημιουργήσει οι χρήστες.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 user_project_id	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2 creator_id	int(11)			No	None	
<input type="checkbox"/>	3 project_name	varchar(20)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	4 description	varchar(200)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	5 current_version	int(11)			No	None	

Πίνακας *project_version*

Διαθέτει τις διάφορες εκδόσεις των κειμένων των έργων.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 project_version	int(11)			No	None	
<input type="checkbox"/>	2 user_project_id	int(11)			No	None	
<input type="checkbox"/>	3 project_text	varchar(1000)	latin1_swedish_ci		No	None	

Πίνακας *project_participants*

Διαθέτει πληροφορίες σχετικά με το ποιοί χρήστες είναι σε κάθε έργο.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 user_project_id	int(11)			No	None	
<input type="checkbox"/>	2 user_id	int(11)			No	None	

Πίνακας *task_lists*

Έχει τις πληροφορίες για τις λίστες καθηκόντων.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 task_list_id	int(11)			No	None	
<input type="checkbox"/>	2 creator_id	int(11)			No	None	
<input type="checkbox"/>	3 task_list_name	varchar(20)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	4 description	varchar(200)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	5 completed	int(11)			No	None	

Πίνακας *tasks*

Έχει τις πληροφορίες για τα καθήκοντα

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 task_id	int(11)			No	None	
<input type="checkbox"/>	2 task_list_id	int(11)			No	None	
<input type="checkbox"/>	3 description	varchar(200)	latin1_swedish_ci		No	None	

Πίνακας *task_asign*

Έχει τις πληροφορίες για το ποιος χρήστης έχει ανάλαβει κάποιο καθήκον.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	task_id	int(11)			No	None	
2	user_id	int(11)			No	None	
3	completed	int(11)			No	None	

Γ. Οδηγίες εγκατάστασης

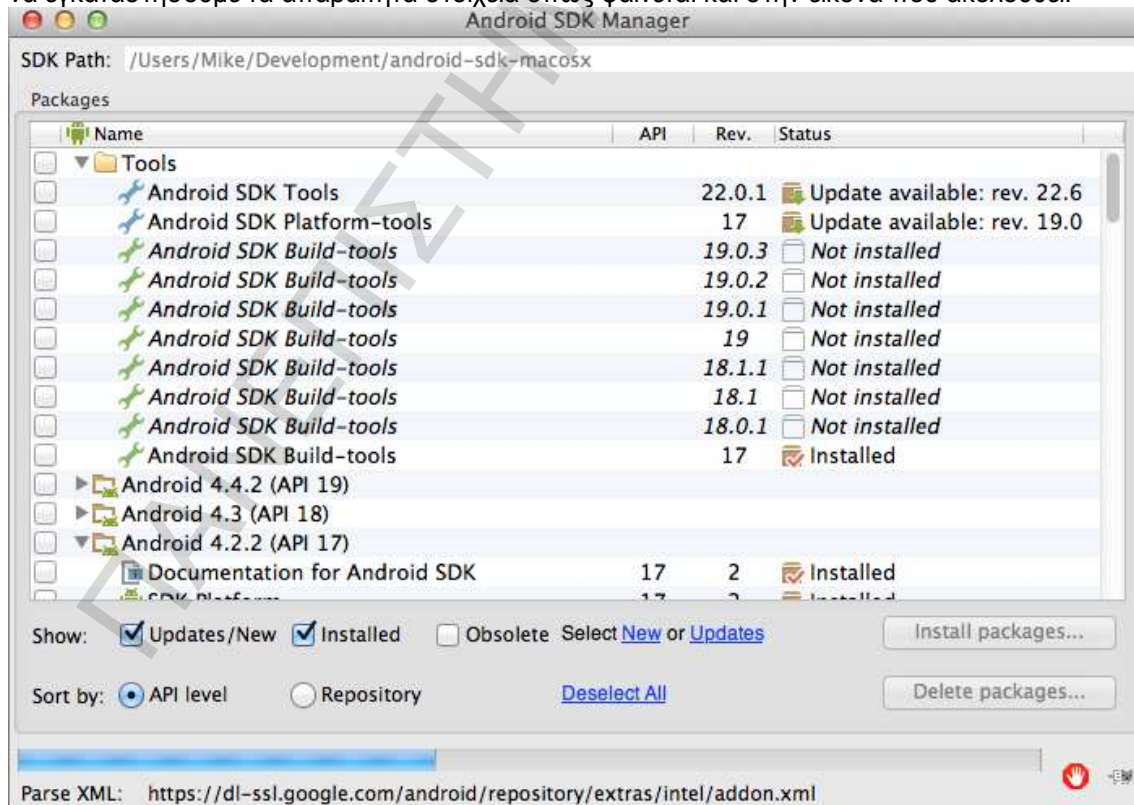
Γ.1.Εφαρμογή διαχειριστών και ενδιάμεσο λογισμικό

Στην περίπτωση της LibrApp, η εφαρμογή των διαχειριστών του συστήματος, καθώς και το ενδιάμεσο λογισμικό βρίσκεται σε απομακρυσμένο εξυπηρετητή. Για την ανάγκη όμως της εγκατάστασης σε τοπικό περιβάλλον μπορεί να χρησιμοποιηθεί οποιοσδήποτε server, ο οποίος υποστηρίζει php και mysql, όπως για παράδειγμα ο server XAMPP, ο οποίος είναι διαθέσιμος για download από την σελίδα <http://www.apachefriends.org/index.html>.

Γ.2.Εφαρμογή κινητού τηλεφώνου

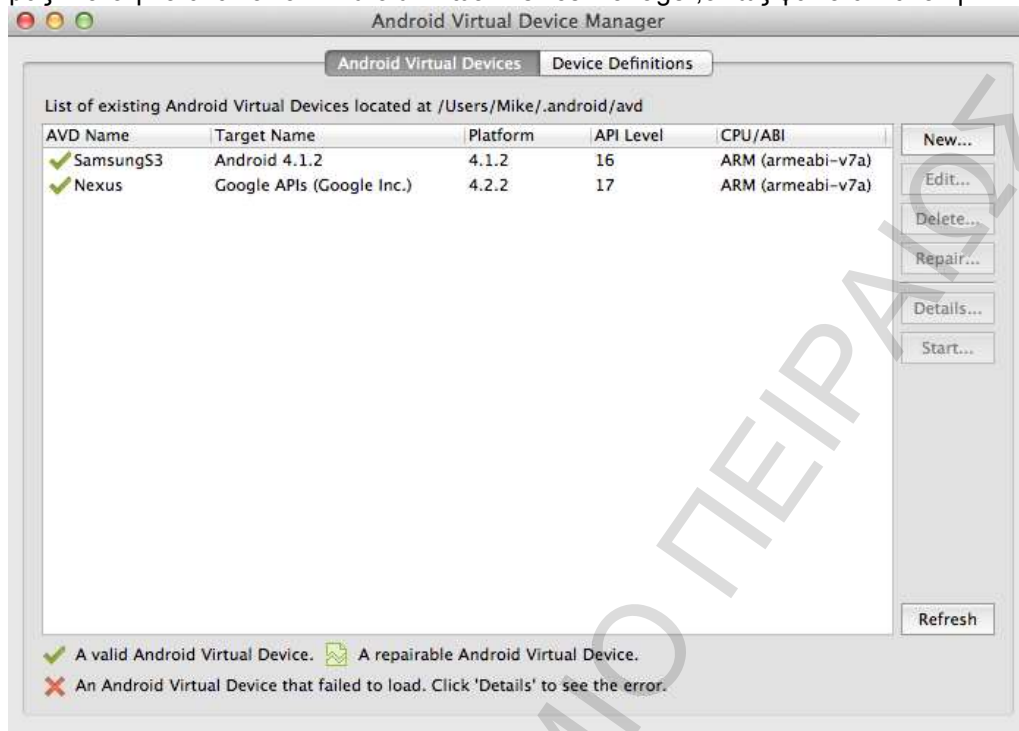
Για την υλοποίηση της εφαρμογής σε σύστημα Android, έγινε χρήση του εργαλείου IDE Eclipse, το οποίο είναι διαθέσιμο για download από την σελίδα <https://www.eclipse.org/downloads/>. Επίσης, χρειάζεται να υπάρχει εγκατεστημένο το πακέτο SDK της Java, διαθέσιμο στην ιστοσελίδα <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>.

Για να μπορέσουμε να χρησιμοποιήσουμε τις λειτουργίες του Android API, πρέπει να κατεβάσουμε και να εγκαταστήσουμε στο Eclipse τα Android SDK Tools, τα οποία μπορούμε να βρούμε στο <http://developer.android.com/tools/sdk/tools-notes.html>. Από εκεί θα μπορέσουμε να εγκαταστήσουμε τα απαραίτητα στοιχεία όπως φαίνεται και στην εικόνα που ακολουθεί.



Εικόνα 46 Android SDK Manager

Αφού ολοκληρωθεί η εγκατάσταση και γίνει η εισαγωγή του κώδικα της εφαρμογής, θα πρέπει να δηλωθεί ένα Android Virtual Device, το οποίο είναι ουσιαστικά ένας εξομοιωτής μίας κανονικής κινητής συσκευής στο οποίο θα γίνει η εγκατάσταση και εκτέλεση της εφαρμογής μας. Αυτό γίνεται από τον Android Virtual Device Manager, όπως φαίνεται και στην Εικόνα 47.



Εικόνα 47 Android Virtual Device Manager

Παράρτημα 2

Σε αυτό το παράρτημα δίνεται το ερωτηματολόγιο που χρησιμοποιήθηκε για την αξιολόγηση της εφαρμογής LibrApp, η οποία αναπτύχθηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής. Το ερωτηματολόγιο στάλθηκε σε ένα τυχαίο δείγμα ατόμων διαφορετικής ηλικίας, μορφωτικού επιπέδου και επαγγελματικού πεδίου και το ποσοστό απάντησης ήταν περίπου στο 63%. Το ερωτηματολόγιο και η συλλογή των απαντήσεων έγιναν με την χρήση των Google Live Forms, της εφαρμογής Google Drive.

Android Digital Library Evaluation Survey

We present this questionnaire in order to evaluate whether a mobile application that functions as a Digital Library, also covering some collaborative functionalities, will be helpful to academic or work environments. The application is made on the Android platform and offers the users the possibility to find and download books, research papers or other material from the system's database, find colleagues or co-students, share material with them, create working groups, create and manage tasks etc.

You will be guided through a series of questions that will evaluate how useful each feature of the application will be to you, if the applications is installed at your work or academic environment.

*Required

Section A. Personal Information

Gender *

- Male
 Female
-

Age *

- 18-24
 25-29
 30-34
 35-39
 40-44
 45+
-

Education *

- High School
 Bachelor Degree
 Masters Degree
 PhD
 Other
-

Employment Type

- Student
 Employee at a IT related post or company
 Employee at a educational environment or library
 Employee at other industry
 Unemployed

Continue »



Android Digital Library Evaluation Survey

*Required

Section B. Technical Background

*How familiar are you with smartphone devices? **

1 2 3 4 5

Not at all Absolutely

*How familiar are you with the Android operating system? **

1 2 3 4 5

Not at all Absolutely

*Are you familiar with one or more collaborative systems at your work or academic environment? **

- No
 1-2
 3-4
 5+

*How often do you use the functions of a library (physical or digital) for work or academic purposes? **

1 2 3 4 5


Never Very often

*Where do you find resources for work or academic assignments? **

You can choose more than one answer

- Internet (General search)
 Physical Libraries
 Forums
 Q&A sites
 Electronic repositories of libraries (academic or other)
 I do not perform any search

« Back Continue »

 50% completed

Android Digital Library Evaluation Survey

*Required

Section C. Evaluation of the Applications standard features

Our application to be evaluated is an mobile application targeted on Android devices (smartphones, tablets). A user can login, create an account or search for books without login. After that a user can download content from the application if that is available for free use.

For this section you will be asked to evaluate the usefulness of the basic features of the application.

*Regarding the basic login and search features **

Note that only logged in users can download content

	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
I want to be able to login with existing social media accounts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
i want to be able to search for books without authenticating myself	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I agree with being logged in order to download content	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I want to be able to upload my own content to the application	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Regarding books a user has viewed *

Each user can store books he has viewed in his own personal digital bookshelf inside the application. He can rate the books, review them and share his thoughts at the social media (Facebook, Twitter, LinkedIn etc)

	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
I find the above mentioned feature usefull	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
I want to be able to have my personal digital bookshelf available locally on my device	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I want to be able to share my personal digital bookshelf with users i am connected	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I do not want anyone to be able to view my personal digital bookshelf	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Regarding connecting with users *

Each user can find and connect with friends and colleagues, giving the app a collaborative feature.

	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
I want to be able to connect with my social media friends	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I do not find connecting with others usefull	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I want to be able to upload content directly to my friends' personal digital bookshelf	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I believe that having connections will benefit my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

« Back Continue »



75% completed

*Required

Section D. Evaluation of the application's collaborative and task assignment features

The application provides users with collaborative tools such as chats and wikis. A user can create a project, share it with his connected colleagues or edit it.

He also can create it as a task and assign jobs to his colleagues, manage the deadlines and view the progress.

Regarding the collaborative features *

You will be asked your opinion regarding features like chats, wikis and the possibility of adding extra features

	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
I believe that chatting with my colleagues will make my work more efficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I believe that not being able to edit the same text as my colleagues is better	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I find working with wikis difficult	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I would like to be able to post my problems in a forum like feature	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I believe that the features provided are satisfactory	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Regarding the task assignment and management feature *

A user can create a task and assign specific jobs to other colleagues and view their progress. They can also share documents inside the task.

	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
I do not find this feature usefull	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I believe that a feature like that will help speed up my work	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I would use such a feature in my organization	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I do not need any extra features added.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I find this feature complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

If you have any suggestions on the improvement of this application please feel free to write them below