

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εφαρμογή Πιστότητας / Επιβράβευσης Πελατών Βιβλιοπωλείου BookLoyalty application
Όνοματεπώνυμο Φοιτητή	Αποστολοπούλου Αθηνά
Πατρώνυμο	Νικόλαος
Αριθμός Μητρώου	ΜΠΣΠ/ 14007
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης: **Οκτώβριος 2016**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Μεταπτυχιακή Διατριβή

Αποστολοπούλου Αθηνά

Στην οικογένεια μου

Περιεχόμενα

Πρόλογος	5
Abstract	6
Ευχαριστίες	7
1. Εισαγωγή	8
1.1. Εφαρμογή Επιβράβευσης Πελατών (Mobile BookLoyalty application)	8
1.2. Εργαλείο διαχείρισης του προγράμματος Επιβράβευσης Πελατών	8
2. Ανασκόπηση πεδίου	9
2.1. National Bank of Greece - Go4more	10
2.2. Intersport - Score For More Greece	10
2.3. Shell Smart App	10
3. Παρουσίαση και χρήση εφαρμογής	11
3.1. Εφαρμογή BookLoyalty	11
3.1.1. Είσοδος και εγγραφή στην εφαρμογή	11
3.1.2. Αρχική Οθόνη	14
3.1.3. Προσφορές	20
3.1.4. Τηλεφωνική Κλήση στο κατάστημα	21
3.1.5. Διεύθυνση καταστήματος	21
3.1.6. Έξοδος από την εφαρμογή	22
3.2. Εργαλείο διαχείρισης του προγράμματος Πιστότητας / Επιβράβευσης Πελατών	24
3.2.1. Συναλλαγές συλλογής πόντων (Awards)	26
3.2.2. Συναλλαγές εξαργύρωσης πόντων (Redemptions)	27
3.2.3. Προσφορές (Offers)	28
3.2.4. Λογαριασμοί Χρηστών (Loyalty Profiles)	30
3.2.5. Βιβλία (Items)	31
4. Αρχιτεκτονική Συστήματος	32
4.1. Mobile Εφαρμογή	32
4.2. Κεντρικός Server	41
4.2.1. Models.py	41
4.2.2. Views.py	42
4.3. QR Code	48
5. Συμπεράσματα και Μελλοντικές Επεκτάσεις	49
6. Βιβλιογραφία	48

Ευρετήριο Εικόνων

Εικόνα 1: Είσοδος ή εγγραφή στην εφαρμογή.....	11
Εικόνα 2: Εγγραφή στην εφαρμογή.....	12
Εικόνα 3: Σύνδεση στην εφαρμογή.....	13
Εικόνα 4: Αρχική Οθόνη.....	15
Εικόνα 5: QR code scan.....	16
Εικόνα 6: Επιτυχής προσθήκη πόντων.....	17
Εικόνα 7: Επιτυχής εξαργύρωση πόντων.....	18
Εικόνα 8: Μη επιτυχής εξαργύρωση πόντων.....	19
Εικόνα 9: Προσφορές.....	20
Εικόνα 10: Διεύθυνση καταστήματος.....	22
Εικόνα 11: Έξοδος από την εφαρμογή.....	23
Εικόνα 12: Είσοδος στην σελίδα διαχείρισης.....	24
Εικόνα 13: Αρχική οθόνη της σελίδας διαχείρισης.....	25
Εικόνα 14: Συναλλαγές προσθήκης πόντων.....	26
Εικόνα 15: Συναλλαγές εξαργύρωσης πόντων.....	27
Εικόνα 16: Λίστα προσφορών στην σελίδα διαχείρισης.....	28
Εικόνα 17: Προσθήκη προσφορών στην σελίδα διαχείρισης.....	29
Εικόνα 18: Λογαριασμοί χρηστών στην σελίδα διαχείρισης.....	30
Εικόνα 19: Λίστα βιβλίων του καταστήματος.....	31

Πρόλογος

Τα Προγράμματα Πιστότητας/ Επιβράβευσης Πελατών (Loyalty Programs), βοηθούν τις επιχειρήσεις να κρατήσουν, να επιβραβεύσουν και να επεκτείνουν το πελατολόγιό τους. Έρευνες δείχνουν ότι κατά μέσο όρο 7 έως 10 φορές είναι πιο “ακριβό” για μία επιχείρηση να κερδίσει έναν πελάτη από κάποιον ανταγωνιστή συγκριτικά με το να ξανά-πουλήσει στους ήδη υπάρχοντες πελάτες της. Επιπρόσθετα, οι υπάρχοντες πελάτες δαπανούν κατά μέσο όρο 67% περισσότερα χρήματα σε σχέση με τους νέους.

Πάρα πολλές κατηγορίες επιχειρήσεων και οργανισμών μπορούν να ωφεληθούν από Προγράμματα Πιστότητας/Επιβράβευσης Πελατών. Επιχειρήσεις Λιανικής πώλησης, Ψυχαγωγίας, Αθλητικά Σωματεία, Εκδοτικές Επιχειρήσεις, Γυμναστήρια, Πολιτικοί Οργανισμοί, Μεταφορικές και Τουριστικές εταιρίες, είναι μερικές από αυτές που μπορούν να ωφεληθούν. Όσο μεγαλύτερος είναι ο ανταγωνισμός του εκάστοτε Κλάδου, τόσο πιο σημαντική γίνεται μία επένδυση σε Προγράμματα Επιβράβευσης και Πιστότητας. Απλά και μόνο η συλλογή των στοιχείων του πελάτη για την έκδοση της κάρτας, είναι πολύτιμη. Πόσο μάλλον αν η συνολική ενέργεια αποτελεί τμήμα μιας ολοκληρωμένης Στρατηγικής Marketing που δίνει οφέλη στους πελάτες για να προβούν σε επαναληπτικές αγορές.

Στην εποχή που ζούμε τα smartphones να έχουν καθιερωθεί ως το πιο απαραίτητο εργαλείο στην ζωή μας, γι’ αυτό το λόγο στόχος της παρούσας μεταπτυχιακής διατριβής είναι να υλοποιηθεί μια ολοκληρωμένη λύση συλλογής και εξαργύρωσης πόντων καθώς και μια σελίδα διαχείρισης αυτών για ένα βιβλιοπωλείο.

Abstract

Customer Loyalty Programs help businesses retain, reward and expand their customer base. Research shows that it is on average 7-10 times more "expensive" for a company to gain a customer from a competitor compared to reselling to existing customers. Additionally, existing customers spend on average 67% more money in comparison to new customers.

Numerous business and organizations categories can benefit from Loyalty / Customer Loyalty Programs. Business Retail, Entertainment, Sports Clubs, Publishing Companies, Gyms, Political Organizations, Transport and Tourism companies are among those who can benefit. The greater the competition in each Sector, the more important an investment in loyalty programs and loyalty becomes. Even simply collecting customer data for issuing the card is valuable. Especially if this act is part of an Integrated Marketing Strategy that may motivate to the customers to make repeat purchases.

In the era we live in Smartphones have been established as one of the most essential tools in our life. For this reason the objective of this master thesis is to implement a complete solution for the collection and redemption of points as well as an online management page for a bookstore.

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω ιδιαίτερα τον συνάδελφο μου Αλέξανδρο Νικολόπουλο που η συμβουλευτική του υποστήριξη έπαιξε σημαντικό ρόλο στην περάτωση της παρούσας μεταπτυχιακής διατριβής. Στη συνέχεια θα ήθελα να ευχαριστήσω τους προϊσταμένους μου που με διευκόλυναν, καθώς συμμετείχα στο μεταπτυχιακό παράλληλα με τη δουλειά μου.

Επίσης, θα ήθελα να ευχαριστήσω το Πανεπιστήμιο Πειραιά που με επέλεξε δίνοντας μου την ευκαιρία να συμμετάσχω στο μεταπτυχιακό πρόγραμμα και να αποκτήσω γνώσεις που θα με βοηθήσουν στην μετέπειτα επαγγελματική μου πορεία. Επιπλέον, θα ήθελα να ευχαριστήσω τον κο Ευθύμιο Αλέπη για την αμέριστη και άψογη συνεργασία μας.

Τέλος, θα επιθυμούσα να ευχαριστήσω την οικογένεια μου, η οποία μου συμπαραστάθηκε σε πολύ μεγάλο βαθμό, όχι μόνο κατά τη διάρκεια της διπλωματικής μου εργασίας, αλλά και σε όλη τη διάρκεια των σπουδών μου.

1. Εισαγωγή

Σε αυτή την εργασία θα γίνει η υλοποίηση μιας εφαρμογής συλλογής και εξαργύρωσης πόντων με την χρήση ενός smartphone ή tablet. Πιο συγκεκριμένα οι πελάτες του Βιβλιοπωλείου θα έχουν την δυνατότητα με ένα απλό «σκανάρισμα» του QR code που θα έχει τυπωθεί στην απόδειξη θα μπορούν να συλλέξουν τους πόντους τους. Με τον ίδιο τρόπο θα μπορούν να εξαργυρώσουν πόντους έχοντας έκπτωση στις αγορές τους. Επιπλέον, θα υπάρχει άμεση ενημέρωση για το υπόλοιπο των πόντων που αντιστοιχεί στο λογαριασμό τους, έλεγχος προσφορών, άμεση κλήση του καταστήματος καθώς και χάρτης της τοποθεσίας

Ωστόσο, θα μια υπάρχει μια σελίδα διαχείρισης όπου ο υπεύθυνος του καταστήματος θα έχει την δυνατότητα ελέγχου των χρηστών , συναλλαγών και προσφορών της Mobile BookLoyalty εφαρμογής.

Αυτή η λύση προορίζεται για όλες τις επιχειρήσεις και οργανισμούς προσφέροντας πολλές δυνατότητες, τόσο στην αξιοποίηση δεδομένων πελατών για εμπορικούς και διαφημιστικούς σκοπούς όσο και στο κομμάτι προσέλκυσης και διατήρησης πελατών. Το σύστημα αποτελείται από τα εξής βασικά κομμάτια:

1.1 Εφαρμογή Επιβράβευσης Πελατών (Mobile BookLoyalty application)

Όπως αναφέρθη προηγουμένως, οι πελάτες του Βιβλιοπωλείου θα έχουν την δυνατότητα να παίρνουν τους πόντους αυτόματα διαβάζοντας το QR code που θα έχει τυπωθεί στην απόδειξη που θα παραλάβουν κατά την αγορά των προϊόντων. Το QR code θα περιέχει τα στοιχεία της συναλλαγής κρυπτογραφημένα (γιατί αλλιώς θα μπορούσε οποιοσδήποτε να δημιουργήσει και να σκανάρει QR codes). Ο server θα επιβεβαιώνει ότι το QR είναι σωστό και θα δίνει τους σωστούς πόντους, ανάλογα με την προσφορά που θα υπάρχει εκείνη την ημέρα. Επίσης κατά τον ίδιο τρόπο θα μπορεί να εξαργυρώσει τους πόντους που έχει συλλέξει. Επιπλέον οι πελάτες θα έχουν την δυνατότητα να ελέγχουν το υπόλοιπο των πόντων τους, να λαμβάνουν τις προσφορές τους , να καλέσουν αυτόματα το κατάστημα καθώς και να δουν άμεσα την τοποθεσία στον χάρτη.

1.2 Εργαλείο διαχείρισης του προγράμματος Πιστότητας / Επιβράβευσης Πελατών

Σε αυτό το κομμάτι θα υπάρχει μια σελίδα διαχείρισης όπου ο υπεύθυνος του καταστήματος θα έχει την δυνατότητα να προσθέτει νέους χρήστες, προσφορές, νέα βιβλία κτλ. Επιπλέον ο διαχειριστής θα έχει την δυνατότητα παρακολούθησης ς στατιστικών δεδομένων ανά ημέρα / εβδομάδα/ μήνα τις συναλλαγές που πραγματοποιήθηκαν από την Mobile BookLoyalty εφαρμογή, πόσες συναλλαγές ήταν εξαργύρωσης και πόσες συλλογή πόντων. Με αυτό τον τρόπο οι επιχειρήσεις θα μπορούν να ελέγχουν τις αγοραστικές συνήθειες των πελατών και μπορούν να καταναίμουν τις ανταμοιβές σε

αγορές συγκεκριμένων προϊόντων και υπηρεσιών που θα οδηγήσουν στην ανάπτυξη του πελατολογίου και σε πολλαπλάσια έσοδα.

2. Ανασκόπηση πεδίου

Σε αυτή την ενότητα κάναμε μια ανασκόπηση σε διάφορες εφαρμογές οι οποίες εξυπηρετούν παρόμοιες ανάγκες στις επιχειρήσεις. Σχεδόν όλες οι επιχειρήσεις με ευρύ φάσμα δραστηριοτήτων χρησιμοποιούν προγράμματα επιβράβευσης πελατών (Loyalty programs) σε Ελλάδα και σε εξωτερικό. Ενδεικτικά αναφέρουμε:

- Τράπεζες
- Αεροπορικές Εταιρείες
- Φαρμακεία
- Κέντρα Αισθητικής
- Πρατήρια Υγρών Καυσίμων
- Καφέ και Εστίαση κτλ

Έπειτα από μια αναζήτηση στο διαδίκτυο πολλές επιχειρήσεις και οργανισμοί έχουν δημιουργήσει την δική τους mobile loyalty εφαρμογή. Σχεδόν όλες οι εφαρμογές έχουν πάρει την θέση της «πλαστικής κάρτας» (Loyalty card) και οι δυνατότητες που έχουν οι περισσότερες εφαρμογές παρουσιάζονται συνοπτικά παρακάτω:

- Έλεγχος διαθέσιμων πόντων
- Έλεγχος προσωπικών στοιχείων του λογαριασμού μας.
- Ενημέρωση προφορών
- Προνόμια του προγράμματος
- Άμεσα στοιχεία επικοινωνίας για την εξυπηρέτηση πελατών της αντίστοιχης επιχείρησης.
- Αναζήτηση στον χάρτη για τις κοντινότερες συνεργαζόμενες επιχειρήσεις του προγράμματος.
- Λίστα με τις αγαπημένες επιχειρήσεις ή τα προϊόντα της εκάστοτε επιχείρησης.

Στην συνέχεια θα αναφέρουμε ενδεικτικά κάποιες εφαρμογές επιβράβευσης πελατών που ήδη είναι σε λειτουργία:

2.1 National Bank of Greece - Go4more

Το συγκεκριμένο πρόγραμμα επιβράβευσης ανήκει στην Εθνική Τράπεζας, σε κάθε συναλλαγή οι χρήστες κερδίζουν πόντους. Με την εφαρμογή Go4more οι χρήστες μπορούν εύκολα και γρήγορα να ενημερωθούν για τα εξής:

- Αναζήτηση στον χάρτη τις κοντινότερες συνεργαζόμενες επιχειρήσεις και να ενημερωθούν για τις μοναδικές προσφορές του προγράμματος.
- Ενημερωθούν για το σύνολο των πόντων τους, καθώς και το σύνολο που έχουν συγκεντρώσει και εξαργυρώσει τον τελευταίο μήνα
- Αναλυτική λίστα με τους πόντους που συγκεντρώσατε από τις καθημερινές τραπεζικές συναλλαγές καθώς και τις αγορές .
- Δημιουργία λίστας με τις αγαπημένες επιχειρήσεις
- Προσφορές του προγράμματος

2.2 Intersport - Score For More Greece

Με αυτή την εφαρμογή mobile τα μέλη του προγράμματος SCORE for MORE μπορούν να πραγματοποιήσουν τα εξής:

- Να δουν τα προσωπικά στοιχεία του λογαριασμού τους στο SCORE for MORE.
- Να δουν τους πόντους τους συνολικά αλλά και αναλυτικά με ημερομηνία, ώρα και ποσό πόντων που κέρδισαν σε κάθε αγορά.
- Να έχουν ενημέρωση για τις κλίμακες εξαργύρωσης που μπορούν να επιλέξουν ανάλογα με τον αριθμό των κερδισμένων πόντων που διαθέτουν.
- Να ενημερώνονται για τις προσφορές της INTERSPORT.

2.3 Shell Smart App

Η εφαρμογή Shell Smart App έχει τις παρακάτω λειτουργίες:

- Η κάρτα επιβράβευσης Smart Club στο smart phone.
- Ενημέρωση για τους πόντους και τις πρόσφατες συναλλαγές της Smart Club κάρτας.
- Ενημέρωση για όλα τα δώρα του καταλόγου Smart Club.
- Βρείτε όποιο πρατήριο Shell χρειάζεστε ανάλογα με τις ανάγκες σας
- Ενημέρωση για όλες τις υπάρχουσες και νέες προσφορές των πρατηρίων και καυσίμων Shell.
- Πείτε τη γνώμη σας για το επίπεδο εξυπηρέτησης στο πρατήριο που επισκεφθήκατε και να κερδίσετε πόντους.

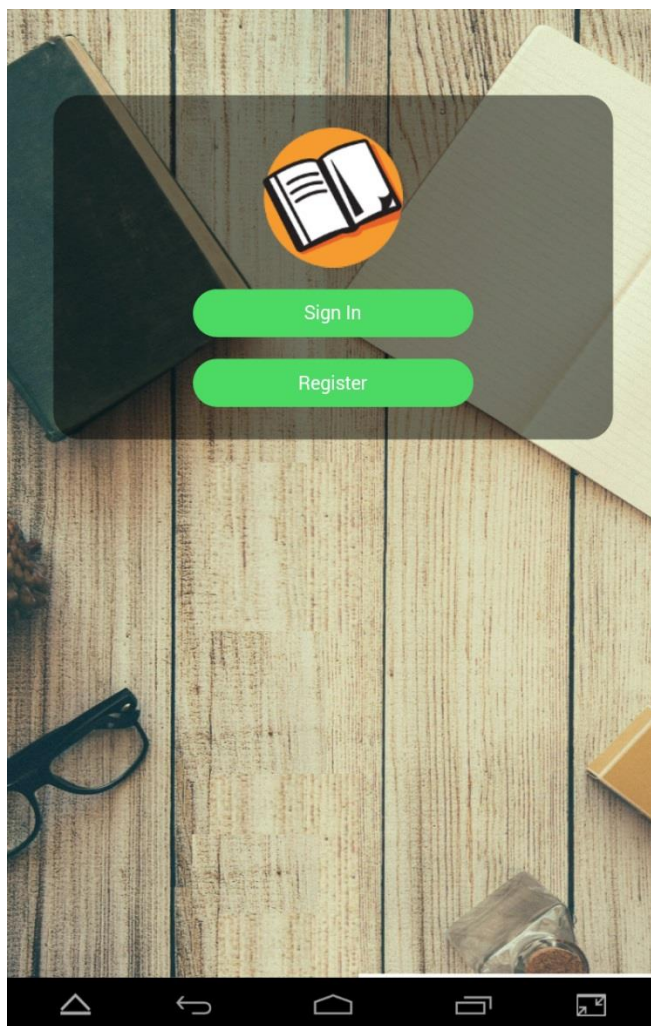
Τέλος, είναι αξιοσημείωτο να αναφερθεί ότι κατά την αναζήτηση να διερευνηθούν αντίστοιχες εφαρμογές παρατηρήθηκε πως σχεδόν όλες οι εφαρμογές έχουν πάρει την θέση της «πλαστικής κάρτας» (Loyalty card) με κύρια λειτουργικότητα τους να είναι ενημέρωση για το σύνολο πόντων και προσφορών. Ωστόσο, δεν έγινε αντιληπτή κάποια εφαρμογή που να έχει ρόλο «εκτελεστικό» όπως την εφαρμογή BookLoyalty που αναπτύχθηκε στην παρούσα εργασία.

3. Παρουσίαση και χρήση εφαρμογής

3.1 Εφαρμογή BookLoyalty

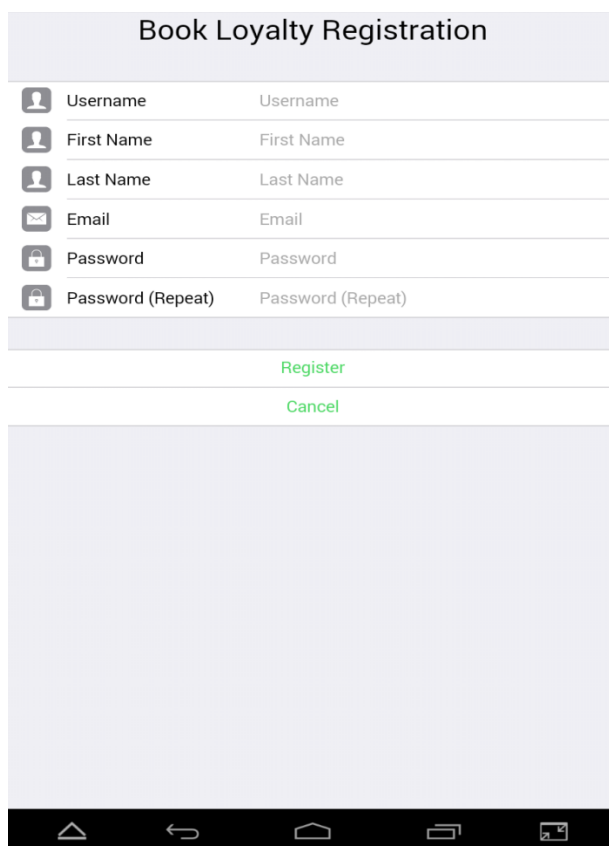
3.1.1. Είσοδος και εγγραφή στην εφαρμογή

Όταν ανοίγει η εφαρμογή εμφανίζεται η πρώτη οθόνη στην οποία ο χρήστης πρέπει να κάνει είσοδο ή εγγραφή στην εφαρμογή όπως φαίνεται παρακάτω.



Εικόνα 1: Είσοδος ή εγγραφή στην εφαρμογή

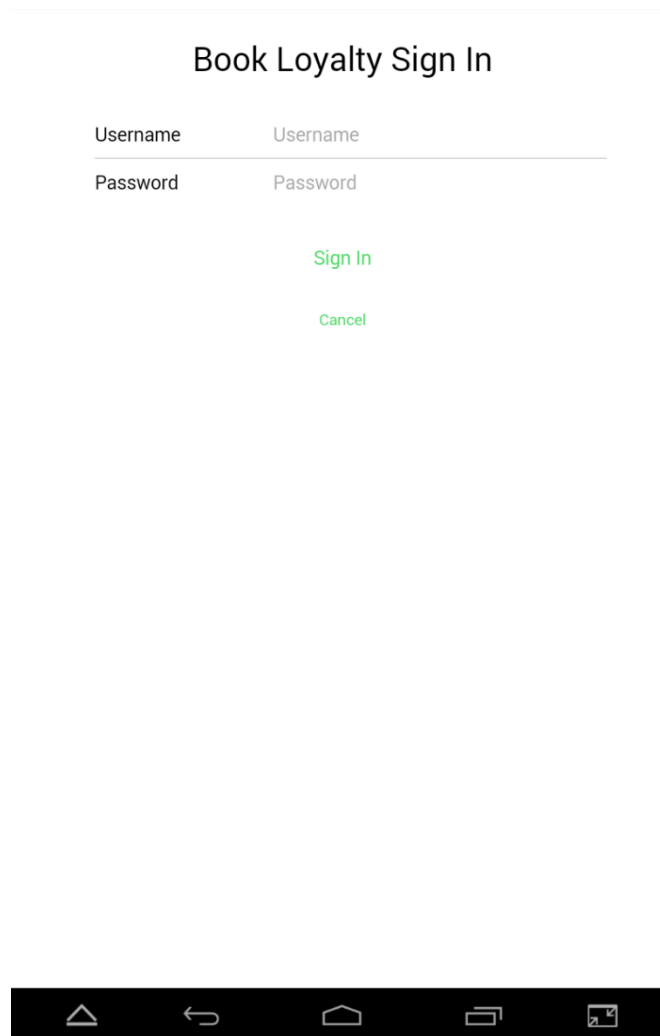
- Εάν ο χρήστης δεν έχει κάνει εγγραφή στην εφαρμογή τότε θα πρέπει να επιλέξει την επιλογή «Register» (από την εικόνα 1). Προκειμένου να δημιουργηθεί ο λογαριασμός επιβράβευσης στο κατάστημα βιβλιοπωλείου, ο χρήστης θα πρέπει να συμπληρώσει τα απαραίτητα πεδία και να επιλέξει «Register» στην εικόνα 2. Πατώντας την επιλογή «Cancel» ο χρήστης επιστέφει στην αρχική οθόνη (εικόνα 1).



Book Loyalty Registration	
<input type="text"/>	Username
<input type="text"/>	First Name
<input type="text"/>	Last Name
<input type="text"/>	Email
<input type="password"/>	Password
<input type="password"/>	Password (Repeat)
Register	
Cancel	

Εικόνα 2: Εγγραφή στην εφαρμογή

- Εάν ο χρήστης έχει πραγματοποιήσει εγγραφή τότε επιλέγει «Sign in» από την εικόνα 1. Στην συνέχεια ο χρήστης καλείται να δώσει το όνομα χρήστη ,τον προσωπικό του κωδικό και να πατήσει «Sign in» προκειμένου να συνδεθεί στην εφαρμογή (εικόνα 3). Πατώντας την επιλογή «Cancel» ο χρήστης επιστρέφει στην αρχική οθόνη (εικόνα 1).



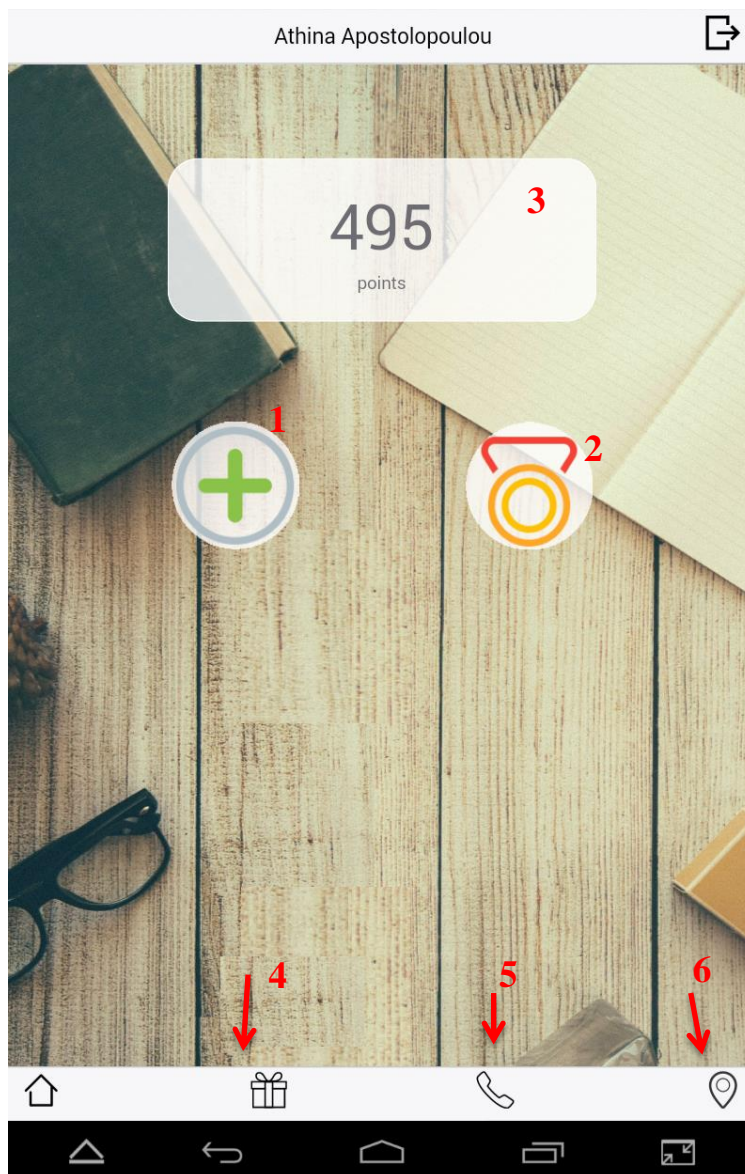
Εικόνα 3: Σύνδεση στην εφαρμογή

3.1.2. Αρχική Οθόνη

Όταν ο χρήστης συνδεθεί επιτυχώς στο σύστημα τότε έχει διαθέσιμες όλες τις λειτουργίες της BookLoalty εφαρμογής που είναι οι εξής:

- Προσθήκη πόντων [1]
- Εξαργύρωση πόντων [2]
- Σύνολο πόντων [3]
- Έλεγχος προσφορών [4]
- Άμεση κλήση στο κατάστημα [5]
- Διεύθυνση καταστήματος [6]

Η αρχική οθόνη της εφαρμογής φαίνεται στην παρακάτω εικόνα.



Εικόνα 4: Αρχική Οθόνη

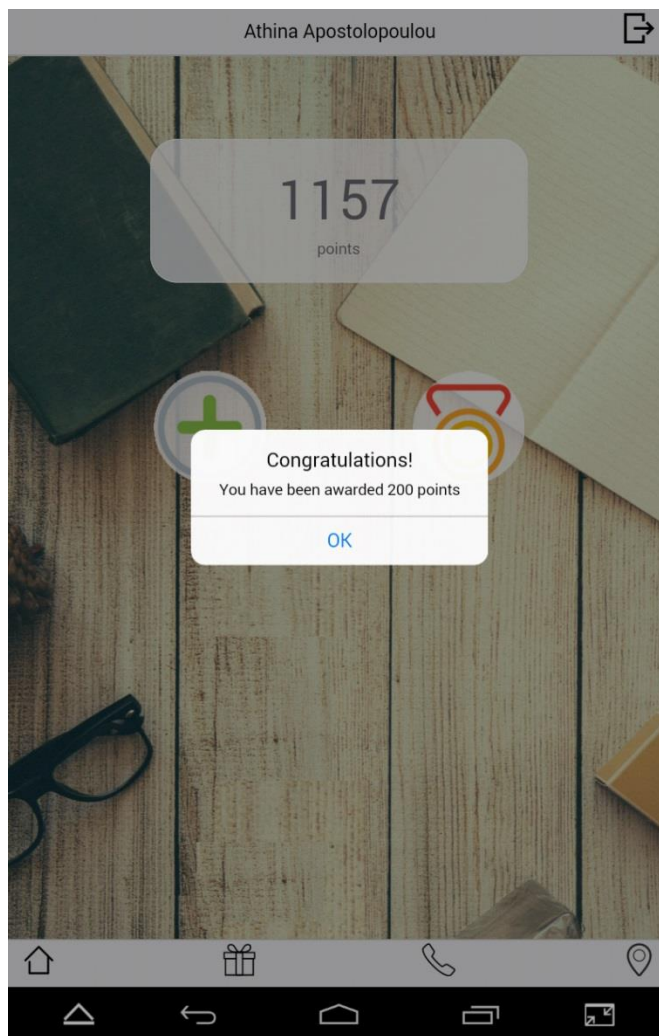
✓ Προσθήκη Πόντων

Για συλλογή πόντων ο χρήστης θα πρέπει να επιλέξει το κουμπί [1] από την αρχική οθόνη (εικόνα 4). Έπειτα θα πρέπει να “σκανάρει” το QR code της απόδειξης όπως φαίνεται παρακάτω



Εικόνα 5: QR code scan

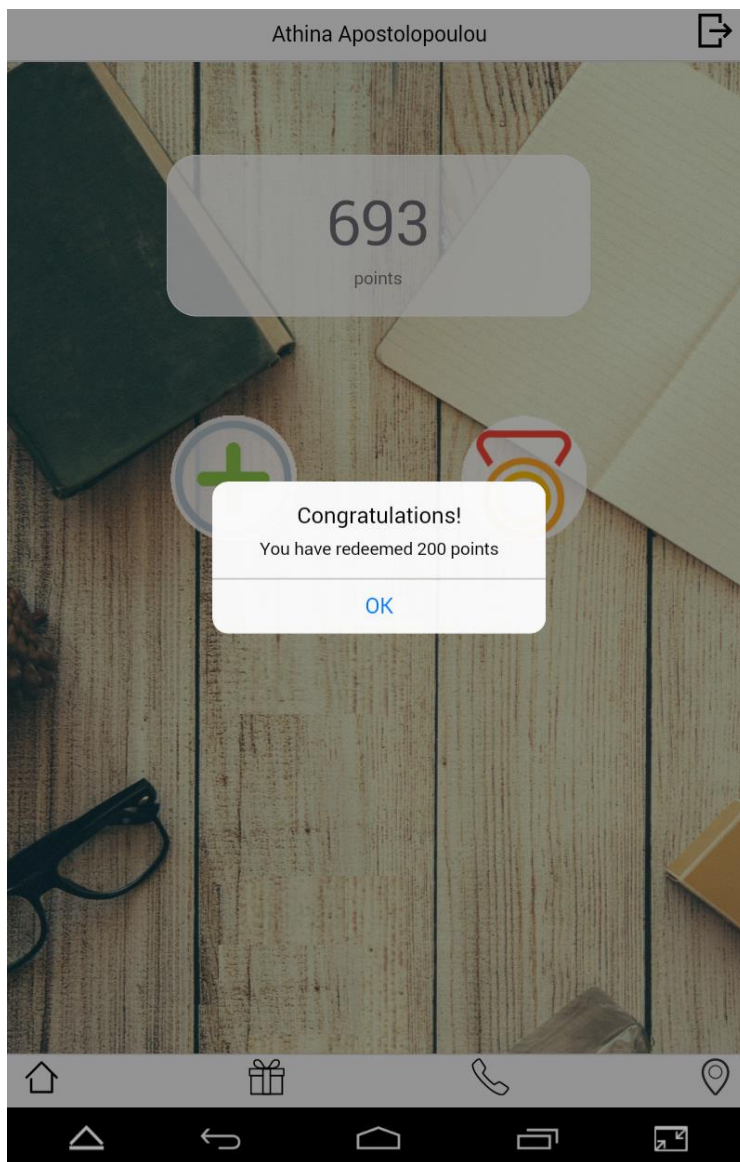
Εάν οι πόντοι προστεθούν με επιτυχία στο λογαριασμό του χρήστη τότε θα λάβει και το αντίστοιχο μήνυμα.



Εικόνα 6: Επιτυχής προσθήκη πόντων

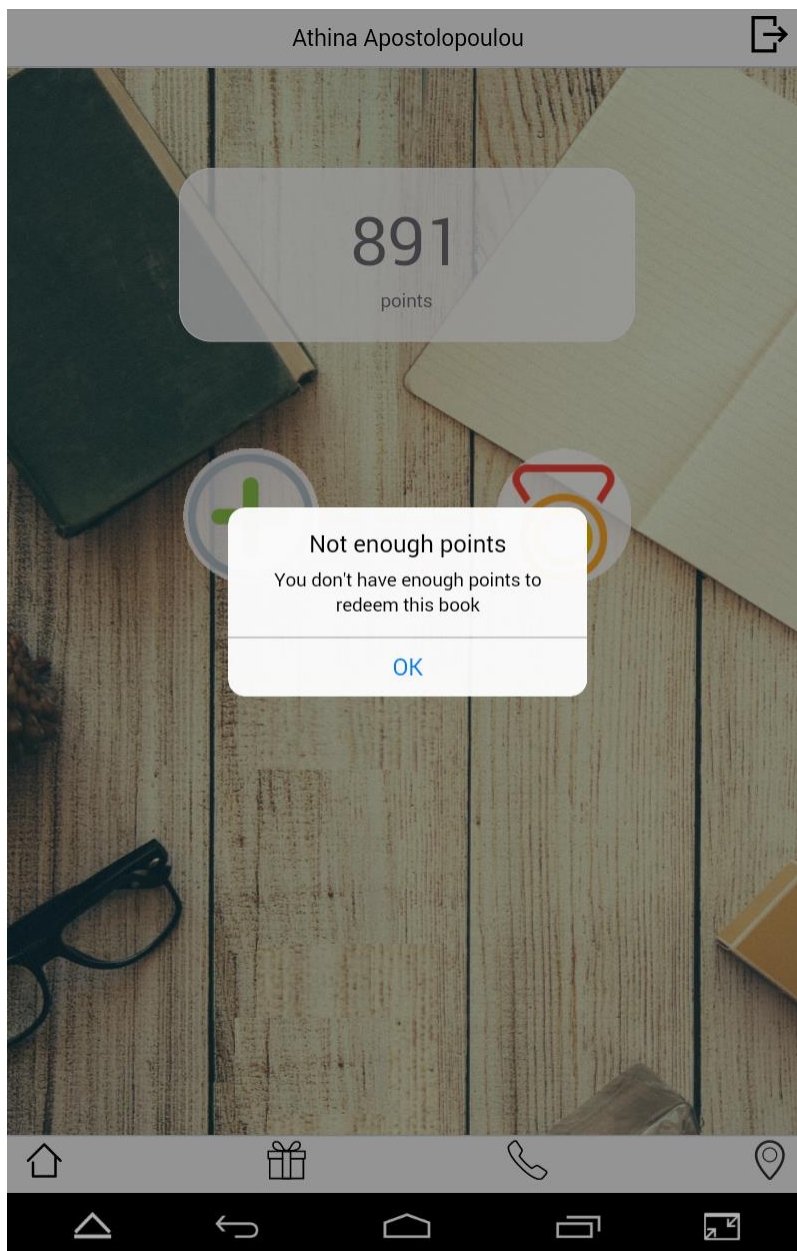
✓ Εξαργύρωση Πόντων

Για εξαργύρωση πόντων ο χρήστης θα πρέπει να επιλέξει το κουμπί [2] από την αρχική οθόνη (εικόνα 4). Έπειτα θα πρέπει να "σκανάρει" το αντίστοιχο QR code που θα του υποδείξει ο υπάλληλος του καταστήματος (εικόνα 5). Εάν ο χρήστης έχει τους απαραίτητους πόντους τότε θα λάβει και το αντίστοιχο μήνυμα.



Εικόνα 7: Επιτυχής εξαργύρωσης πόντων

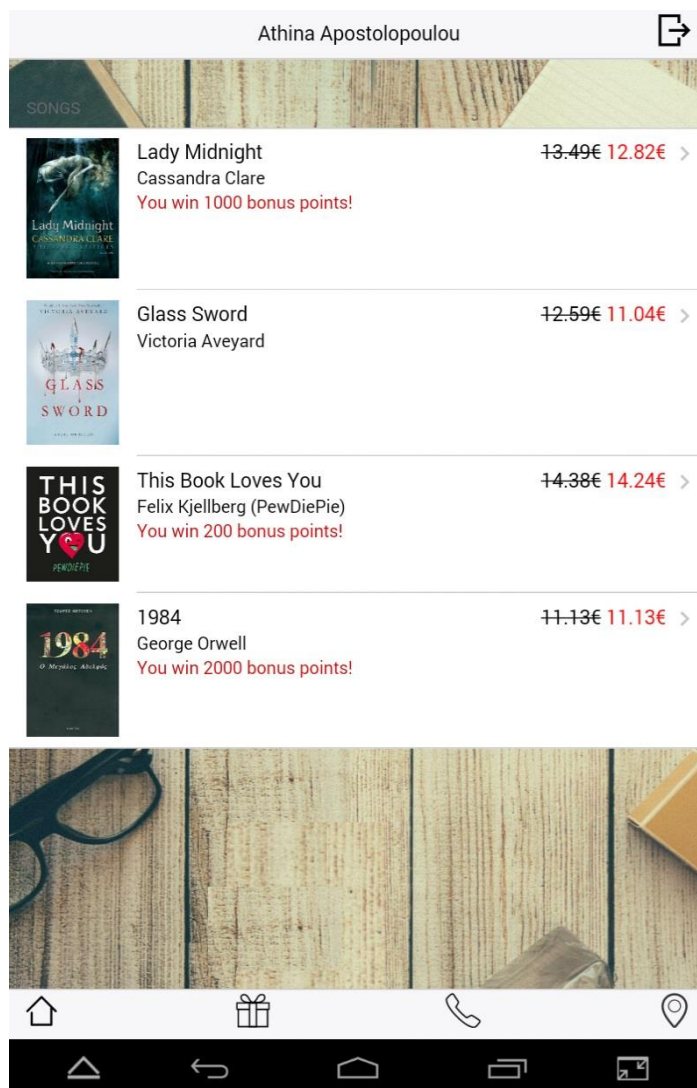
Εάν ο χρήστης ΔΕΝ έχει τους απαραίτητους πόντους τότε θα λάβει ένα ενημερωτικό μήνυμα ότι οι πόντοι δεν είναι αρκετοί για να πραγματοποιηθεί η εξαργύρωση.



Εικόνα 8: Μη επιτυχής εξαργύρωση πόντων

3.1.3. Προσφορές

Ο χρήστης προκειμένου να κάνει έλεγχο των διαθέσιμων προσφορών που υπάρχουν θα πρέπει να επιλέξει από την αρχική οθόνη (εικόνα 4) το κουμπί [4]. Στην συνέχεια θα εμφανιστεί η παρακάτω οθόνη με όλες τις ενεργές προσφορές.



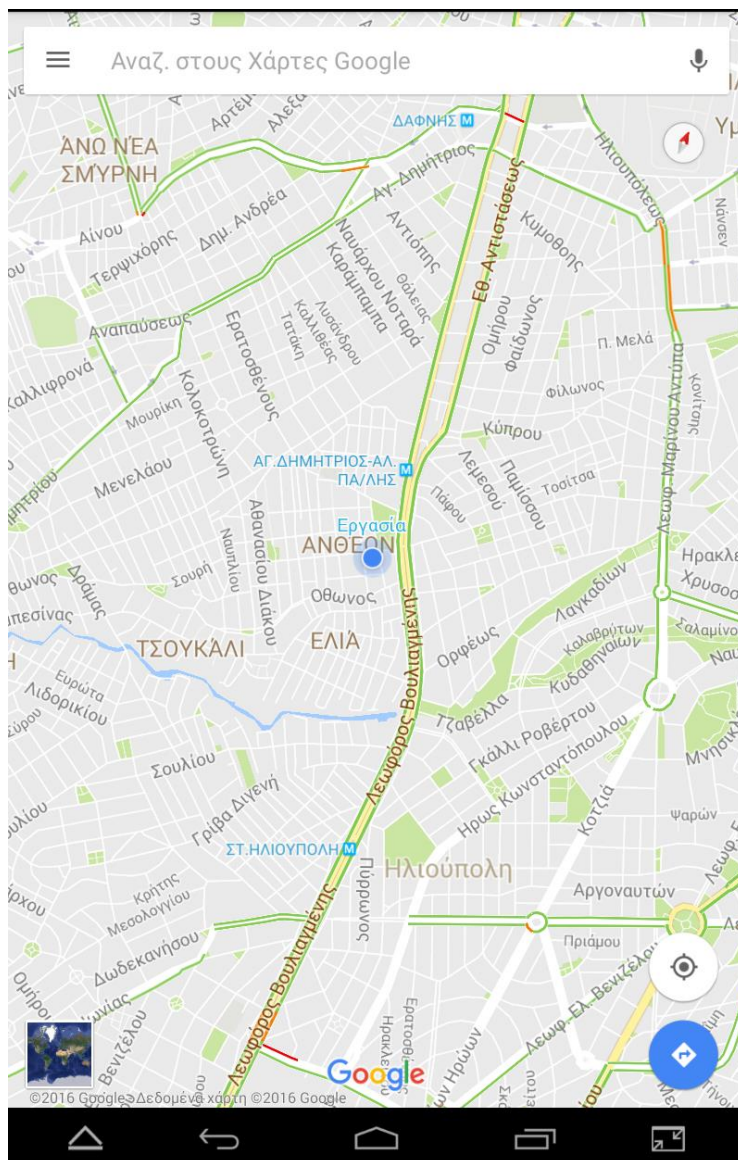
Εικόνα 9: Προσφορές

3.1.4. Τηλεφωνική Κλήση στο κατάστημα

Ο χρήστης προκειμένου να επικοινωνήσει με το κατάστημα θα πρέπει να επιλέξει από την αρχική οθόνη (εικόνα 4) το κουμπί **[5]**. Στην συνέχεια θα εμφανιστεί η παρακάτω οθόνη με όλες τις ενεργές προσφορές.

3.1.5. Διεύθυνση καταστήματος

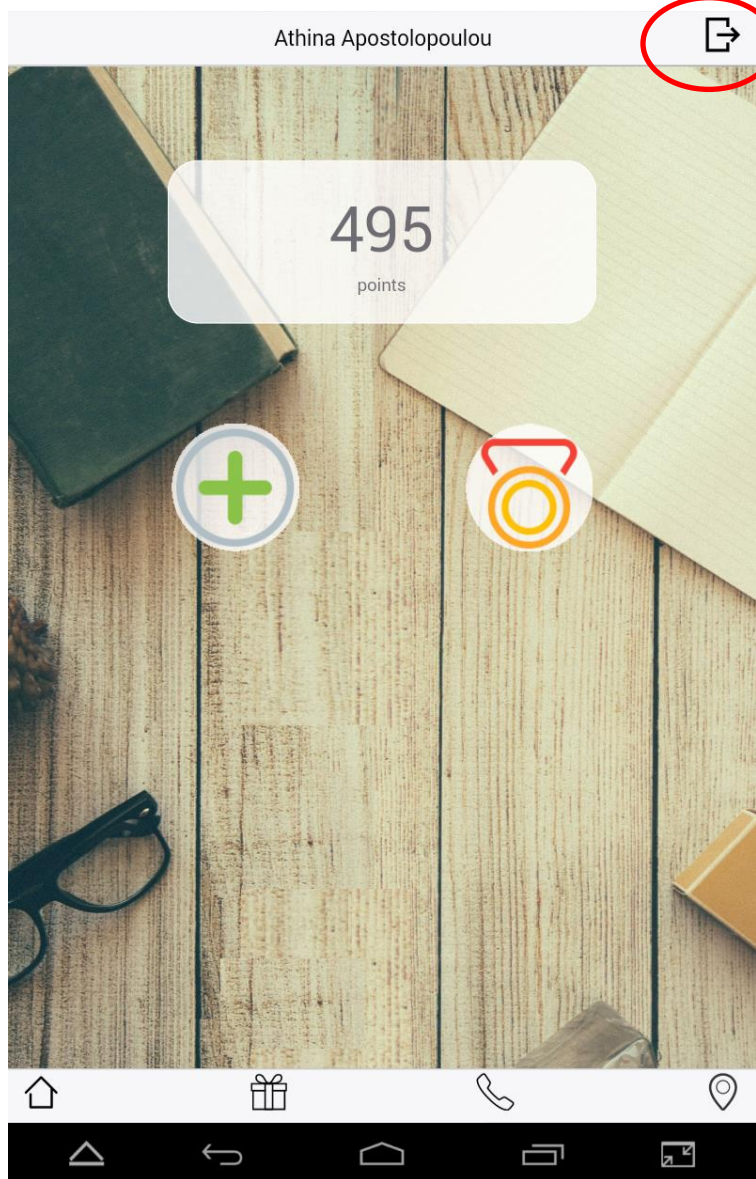
Ο χρήστης προκειμένου να βρει άμεσα την διεύθυνση του καταστήματος στον χάρτη θα πρέπει να επιλέξει από την αρχική οθόνη (εικόνα 4) το κουμπί **[6]**. Στην συνέχεια θα εμφανιστεί η παρακάτω οθόνη.



Εικόνα 10: Διεύθυνση καταστήματος

3.1.6. Έξοδος από την εφαρμογή

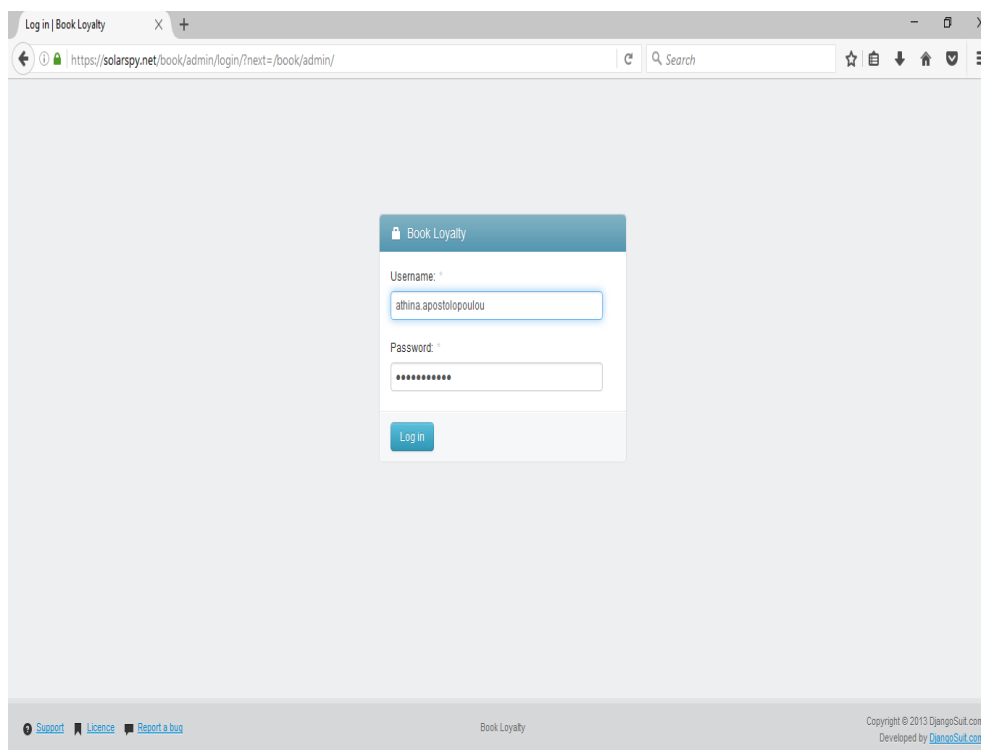
Για να πραγματοποιήσουμε έξοδο από την εφαρμογή επιλέγουμε αρχική οθόνη (εικόνα 4) το κουμπί της εξόδου πάνω δεξιά τότε αυτόματα θα εμφανιστεί η οθόνη εισόδου (εικόνα 1).



Εικόνα 11: Έξοδος από την εφαρμογή

3.2 Εργαλείο διαχείρισης του προγράμματος Πιστότητας / Επιβράβευσης Πελατών

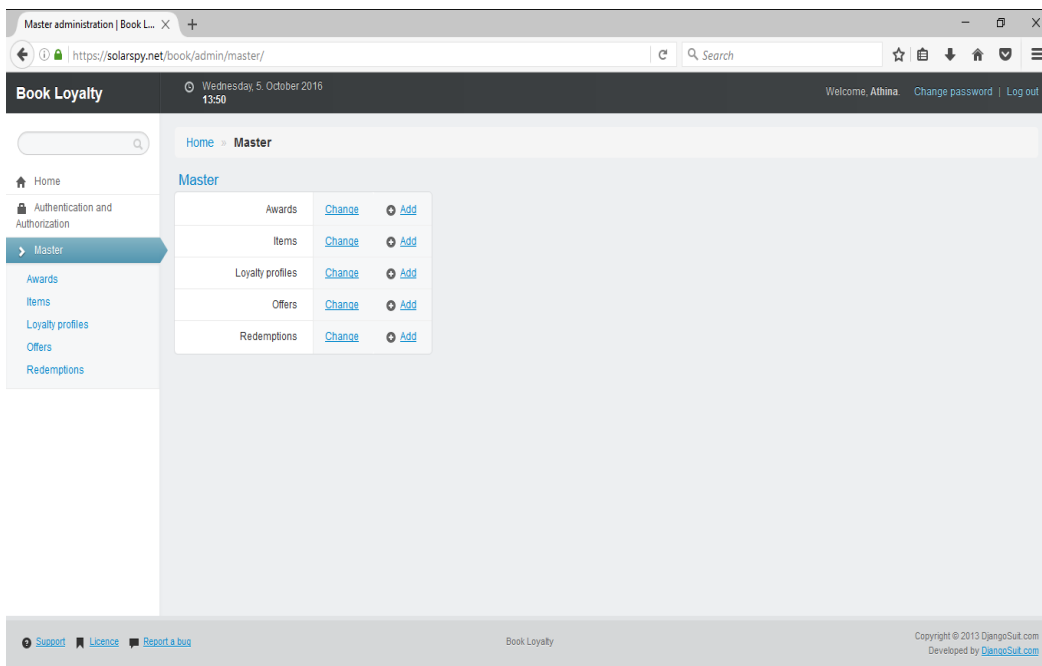
Ο υπεύθυνος του βιβλιοπωλείου θα έχει την δυνατότητα να διαχειρίζεται τους χρήστες, προσφορές, νέα βιβλία καθώς και την παρακολούθηση των συναλλαγών ανά ημέρα / εβδομάδα/ μήνα. Η πρώτη οθόνη στην οποία ο χρήστης πρέπει να κάνει είσοδο είναι η παρακάτω.



Εικόνα 12: Είσοδος στην σελίδα διαχείρισης

Έπειτα από την επιτυχή σύνδεση του χρήστη η κεντρική οθόνη της σελίδας είναι η παρακάτω και έχει τις εξής λειτουργίες:

- Συναλλαγές συλλογής πόντων (Awards).
- Συναλλαγές εξαργύρωσης πόντων (Redemptions)
- Προσφορές (Offers)
- Λογαριασμοί Χρηστών (Loyalty Profiles)
- Βιβλία (Items)



Εικόνα 13: Αρχική οθόνη της σελίδας διαχείρισης

3.2.1 Συναλλαγές συλλογής πόντων (Awards)

Στην επιλογή Awards ο χρήστης θα έχει την δυνατότητα να δει αναλυτικά όλες τις συναλλαγές προσθήκης πόντων ανά ημερομηνία , χρήστη και τους πόντους.

<input type="checkbox"/>	Award Timestamp	Profile	Points
<input type="checkbox"/>	Oct. 4, 2016, 10:30 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 4, 2016, 10:27 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 4, 2016, 10:27 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 4, 2016, 10:25 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 4, 2016, 10:07 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 4, 2016, 10:06 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 3, 2016, 2:43 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 2, 2016, 11:04 a.m.	athinapos	100
<input type="checkbox"/>	Oct. 2, 2016, 11:03 a.m.	athinapos	100
<input type="checkbox"/>	Oct. 2, 2016, 10:57 a.m.	athinapos	500
<input type="checkbox"/>	Oct. 2, 2016, 10:56 a.m.	athinapos	1000
<input type="checkbox"/>	Oct. 2, 2016, 10:56 a.m.	athinapos	500
<input type="checkbox"/>	Oct. 2, 2016, 10:56 a.m.	athinapos	200
<input type="checkbox"/>	Oct. 2, 2016, 10:55 a.m.	athinapos	5000
<input type="checkbox"/>	Oct. 2, 2016, 10:55 a.m.	athinapos	2000
<input type="checkbox"/>	Oct. 2, 2016, 10:49 a.m.	athinapos	100
<input type="checkbox"/>	Oct. 2, 2016, 10:48 a.m.	athinapos	100
<input type="checkbox"/>	Oct. 2, 2016, 10:46 a.m.	athinapos	1000
<input type="checkbox"/>	July 9, 2016, 5:54 p.m.	athinapos	200
<input type="checkbox"/>	July 9, 2016, 5:51 p.m.	athinapos	200

Εικόνα 14: Συναλλαγές προσθήκης πόντων

3.2.2 Συναλλαγές εξαργύρωσης πόντων (Redemptions)

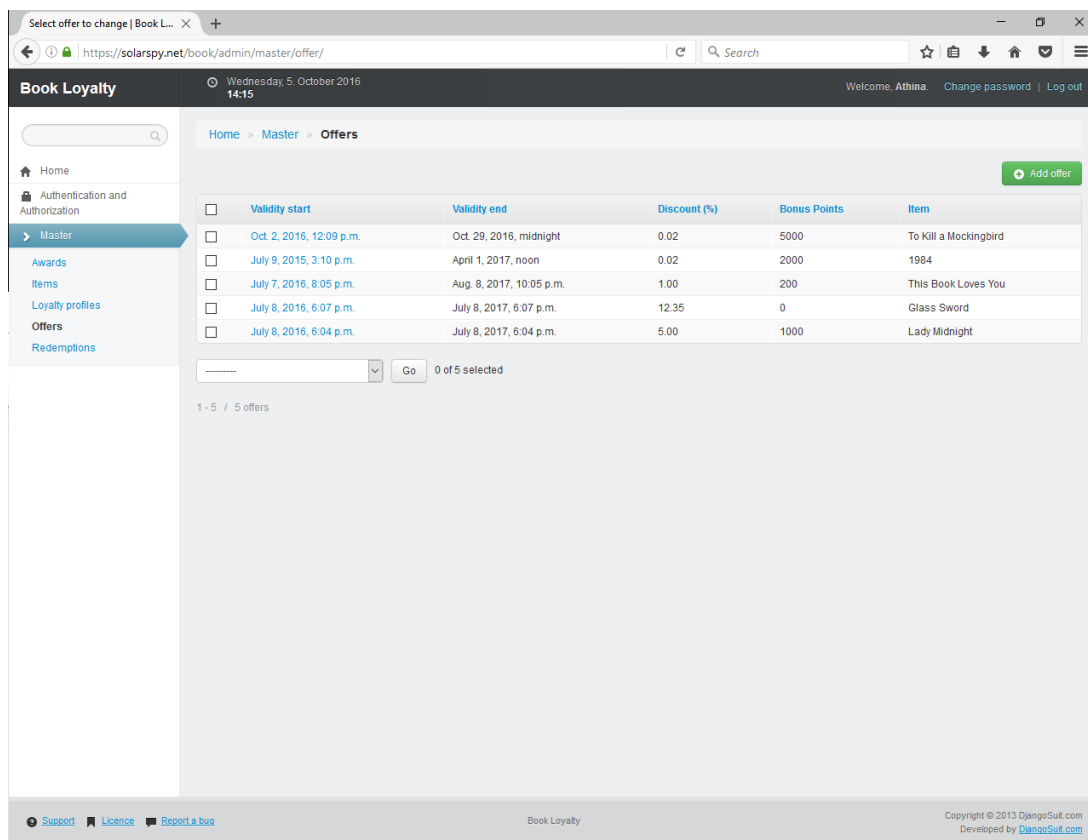
Στην επιλογή Redemptions ο χρήστης θα έχει την δυνατότητα να δει αναλυτικά όλες τις συναλλαγές εξαργύρωσης πόντων ανά ημερομηνία , χρήστη και τους πόντους.

<input type="checkbox"/>	Redemption Timestamp	Profile	Points
<input type="checkbox"/>	Oct. 4, 2016, 10:26 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 4, 2016, 10:07 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 3, 2016, 2:42 p.m.	athinapos	200
<input type="checkbox"/>	Oct. 2, 2016, 10:58 a.m.	athinapos	5000
<input type="checkbox"/>	Oct. 2, 2016, 10:58 a.m.	athinapos	5000
<input type="checkbox"/>	Oct. 2, 2016, 10:57 a.m.	athinapos	200
<input type="checkbox"/>	Oct. 2, 2016, 10:57 a.m.	athinapos	500
<input type="checkbox"/>	Oct. 2, 2016, 10:55 a.m.	athinapos	100
<input type="checkbox"/>	Oct. 2, 2016, 10:45 a.m.	athinapos	2000
<input type="checkbox"/>	Oct. 2, 2016, 10:44 a.m.	athinapos	500
<input type="checkbox"/>	July 9, 2016, 5:55 p.m.	athinapos	100
<input type="checkbox"/>	July 9, 2016, 5:50 p.m.	athinapos	100
<input type="checkbox"/>	July 9, 2016, 5:41 p.m.	athinapos	100
<input type="checkbox"/>	July 9, 2016, 5:10 p.m.	athinapos	200
<input type="checkbox"/>	July 9, 2016, 5 p.m.	athinapos	5000
<input type="checkbox"/>	July 9, 2016, 5 p.m.	athinapos	5000
<input type="checkbox"/>	July 9, 2016, 5 p.m.	athinapos	5000
<input type="checkbox"/>	July 9, 2016, 11:20 a.m.	tina	200
<input type="checkbox"/>	July 3, 2016, 6:06 a.m.	themapos	1000
<input type="checkbox"/>	July 3, 2016, 6:05 a.m.	themapos	200

Εικόνα 15: Συναλλαγές εξαργύρωσης πόντων

3.2.3 Προσφορές (Offers)

Στην επιλογή offers ο χρήστης θα έχει την δυνατότητα να δει αναλυτικά το ιστορικό των προσφορών καθώς και να προσθέσει νέες προσφορές. Η παρακάτω οθόνη (εικόνα 16) εμφανίζει όλες τις προσφορές που έχουν προστεθεί από τον χρήστη.



The screenshot shows the 'Offers' management page in the Book Loyalty application. The page includes a sidebar with navigation options like Home, Authentication and Authorization, Master, Awards, Items, Loyalty profiles, Offers, and Redemptions. The main content area displays a table of offers with columns for Validity start, Validity end, Discount (%), Bonus Points, and Item. There are five offers listed, each with a checkbox for selection. Below the table, there is a search bar and a 'Go' button. The footer contains links for Support, Licence, and Report a bug, along with copyright information for DjangoSui.com.

	Validity start	Validity end	Discount (%)	Bonus Points	Item
<input type="checkbox"/>	Oct. 2, 2016, 12:09 p.m.	Oct. 29, 2016, midnight	0.02	5000	To Kill a Mockingbird
<input type="checkbox"/>	July 9, 2015, 3:10 p.m.	April 1, 2017, noon	0.02	2000	1984
<input type="checkbox"/>	July 7, 2016, 8:05 p.m.	Aug. 8, 2017, 10:05 p.m.	1.00	200	This Book Loves You
<input type="checkbox"/>	July 8, 2016, 6:07 p.m.	July 8, 2017, 6:07 p.m.	12.35	0	Glass Sword
<input type="checkbox"/>	July 8, 2016, 6:04 p.m.	July 8, 2017, 6:04 p.m.	5.00	1000	Lady Midnight

Εικόνα 16: Λίστα προσφορών στην σελίδα διαχείρισης

Επιπλέον, ο χρήστης θα μπορεί να προσθέσει μια νέα προσφορά πατώντας το κουμπί «Add Offer», τότε θα εμφανιστεί η παρακάτω οθόνη (εικόνα 17). Ο διαχειριστής μπορεί να δημιουργήσει προσφορά με ημερομηνία έναρξης και λήξης συγκεκριμένου βιβλίου καθώς και τους πόντους που θα κερδίζει ο κάθε πελάτης.

Book Loyalty

Wednesday, 5. October 2016 14:22

Welcome, Athina. [Change password](#) | [Log out](#)

Home » Master » Offers » Add offer

Validity start:	Date: 2016-10-02 Today
	Time: 00:00:00 Now
Validity end:	Date: 2018-06-28 Today
	Time: 00:00:00 Now
Item:	Les Misérables
Discount (%):	0.02
Bonus Points:	200

[Save](#)

[Save and continue editing](#)

[Save and add another](#)

[Support](#) | [Licence](#) | [Report a bug](#)

Book Loyalty

Copyright © 2013 DjangoSuit.com
Developed by [DjangoSuit.com](#)

Εικόνα 17: Προσθήκη προσφορών στην σελίδα διαχείρισης

3.2.4 Λογαριασμοί Χρηστών (Loyalty Profiles)

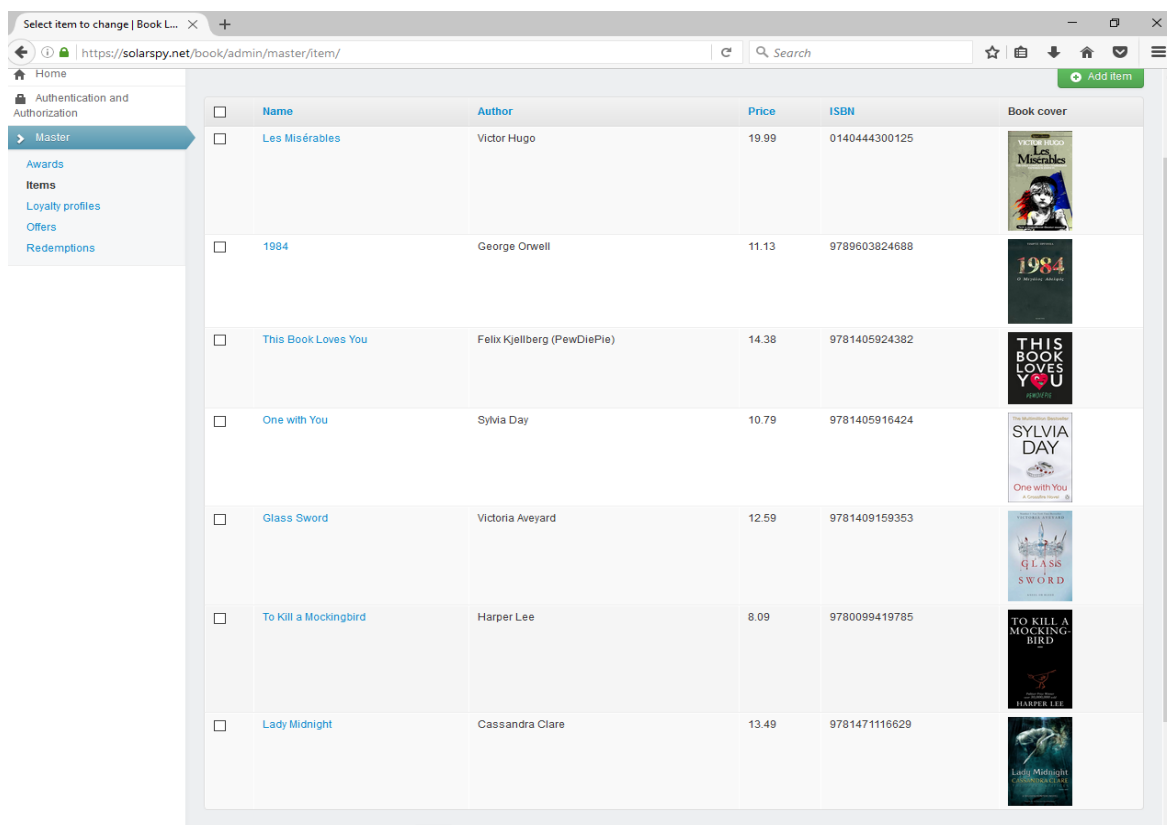
Στην επιλογή Loyalty Profiles ο διαχειριστής θα έχει την δυνατότητα να δει αναλυτικά όλους τους λογαριασμούς χρηστών και τους πόντους τους.








User	Points
<input type="checkbox"/> tzina	162
<input type="checkbox"/> themapros	4869
<input type="checkbox"/> athinapos	1251
<input type="checkbox"/> abnik	34944

Εικόνα 18: Λογαριασμοί χρηστών στην σελίδα διαχείρισης

3.2.5 Βιβλία (Items)

Στην επιλογή Items ο διαχειριστής θα έχει την δυνατότητα να δει αναλυτικά όλα τα βιβλία που έχει στο κατάστημα του.



<input type="checkbox"/>	Name	Author	Price	ISBN	Book cover
<input type="checkbox"/>	Les Misérables	Victor Hugo	19.99	0140444300125	
<input type="checkbox"/>	1984	George Orwell	11.13	9789603824688	
<input type="checkbox"/>	This Book Loves You	Felix Kjellberg (PewDiePie)	14.38	9781405924382	
<input type="checkbox"/>	One with You	Sylvia Day	10.79	9781405916424	
<input type="checkbox"/>	Glass Sword	Victoria Aveyard	12.59	9781409159353	
<input type="checkbox"/>	To Kill a Mockingbird	Harper Lee	8.09	9780099419785	
<input type="checkbox"/>	Lady Midnight	Cassandra Clare	13.49	9781471116629	

Εικόνα 19: Λίστα βιβλίων του καταστήματος

4. Αρχιτεκτονική Συστήματος

Σε αυτή την ενότητα θα γίνει η ανάλυση της αρχιτεκτονικής της εφαρμογής . Θα παρουσιαστεί ο τρόπος με τον οποίο υλοποιήθηκε η εφαρμογή, ποια εργαλεία χρησιμοποιήθηκαν, ποια γλώσσα προγραμματισμού, ποια βάση δεδομένων καθώς και μεθοδολογίες ανάπτυξης κώδικα.

Η εφαρμογή μας είναι μια client εφαρμογή κινητών συσκευών που επικοινωνεί με έναν κεντρικό server, ο οποίος έχει πάνω του μια βάση δεδομένων MySql. Για λόγους ασφαλείας την μεγαλύτερη διαχείριση της εφαρμογής είναι υπεύθυνος ο κεντρικός σέρβερ , για παράδειγμα το authentication των χρηστών, προσθήκη /αφαίρεση πόντων κτλ.

Παρακάτω θα αναλύσουμε με περισσότερη λεπτομέρεια σχετικά με το τι έχει υλοποιηθεί σε κάθε κομμάτι ξεχωριστά:

4.1 Mobile Εφαρμογή

Η εφαρμογή του κινητού (mobile client) έχει δημιουργηθεί σε PhoneGap. Το PhoneGap είναι ένα open-source λογισμικό το οποίο χρησιμοποιείται για την ανάπτυξη cross-platform εφαρμογών για Android, iPhone, iTouch, Blackberry, iPad, Symbian και Plam. Οι γλώσσες προγραμματισμού που χρησιμοποιούνται είναι HTML, CSS και JavaScript. Επιπλέον, χρησιμοποιήσαμε το plugin "phonegap-plugin-barcodescanner" προκειμένου να υποστηρίζεται το scanner για το QR code. Οι βασικές συναρτήσεις που εκτελούνται στον client είναι RequestOffers, PopulateOffers, Login Register, RegisterScanners, AwardPoints, RedeemPoints και SetUserPoints. Παρακάτω παρατίθεται ο κώδικας για όλες τις συναρτήσεις που αναφέραμε:

bookloyalty.js

```
// Initialize your app
var myApp = new Framework7();

// Export selectors engine
var $$ = Dom7;

var mainView = myApp.addView('.view-main', { animatePages: false });

if(window.localStorage.getItem("loggedIn") == 1) {
  mainView.router.loadPage('main.html');
}
else {
  mainView.router.loadPage('login.html');
}

var Host = 'https://www.solarspy.net'
var HostURL = Host + '/book/client'

$$$(document).on('pageBeforeInit', function (e) {
  $$$('#NameTag').html(window.localStorage.getItem("Name"));
});
```

```

myApp.onPageInit('Login', function (e) {
  mainView.hideNavbar();
  mainView.hideToolbar();
});

myApp.onPageInit('main', function (e) {
  SetUserPoints();
  RegisterScanners();
});

$$('#GoHome').on('click', function(){
  mainView.router.loadPage('main.html');
});

$$('#GoGift').on('click', function(){
  mainView.router.loadPage('gift.html');
  RequestOffers();
});

$$('#GoLogout').on('click', function(){
  window.localStorage.setItem("loggedIn", 0);
  mainView.router.loadPage('login.html');
});

function RequestOffers()
{
  var requestData = '{ "request" : "offers" }';

  AjaxRequest(
    HostURL,
    requestData,

    function(data) {
      Reply = JSON.parse(data);
      if(Reply['result'] == 'NOK')
      {
        Code = Reply['respcode'];

        if(Reply['respcode'] == 2)
        {
          myApp.alert('You are not logged in, 'Connection error');
          window.localStorage.setItem("loggedIn", 0);
          mainView.router.loadPage('login.html');
        }
        else
        {
          myApp.alert('Try again later (' + Code + ')', 'Error');
        }
      }
      else
      {
        populateOffers(Reply['offerdata']);
      }
    }
  );
}

```

```

    }
  },

  function(xhr) {
    myApp.alert('Connection failed. Please try again', 'Connection error');
  });
}

function populateOffers(data)
{
  TotalContent = "";
  for (var i in data) {
    if(parseInt(data[i]['bonuspoints']) > 0)
    {
      BonusText = '<div class="item-text" style="color: #cc0000;">You win ' + data[i]['bonuspoints'] + '
bonus points!</div>';
    }
    else
    {
      BonusText = "";
    }

    Content =
      '<li>' +
      '<a href="#" class="item-link item-content">' +
      '<div class="item-media"></div>' +
      '<div class="item-inner">' +
      '<div class="item-title-row">' +
      '<div class="item-title">' + data[i]['name'] + '</div>' +
      '<div class="item-after"><s style="color: black;"> + data[i]['startprice'] + '€</s>&nbsp;<span
style="color: red;"> + data[i]['endprice'] + '€</span></div>' +
      '</div>' +
      '<div class="item-subtitle">' + data[i]['author'] + '</div>' +
      BonusText +
      '</div>' +
      '</a>' +
      '</li>';

    TotalContent = TotalContent + Content;
  }

  $$('#giftList').html(TotalContent);
}

// Login Functionality
$$('.LoginToBackend').on('click', function(){

  var formData = myApp.formToJSON('#LoginForm');

  AjaxRequest(

```

```

HostURL,
JSON.stringify(formData),

function(data) {
  Reply = JSON.parse(data);
  if(Reply['result'] == 'NOK')
  {
    myApp.alert("Please check your username and password and try again", 'Sign In failed');
  }
  else
  {
    window.localStorage.setItem("loggedIn", 1);
    window.localStorage.setItem("Name", Reply['name']);
    mainView.router.loadPage('main.html');
    mainView.showNavbar();
    mainView.showToolbar();
    myApp.closeModal();
  }
},

function(xhr) {
  myApp.alert('Connection failed. Please try again', 'Sign In failed');
});
});

// Registration functionality
$$('.RegisterToBackend').on('click', function(){

var formData = myApp.formToJSON('#RegisterForm')

AjaxRequest(
  HostURL,
  JSON.stringify(formData),

function(data) {
  Reply = JSON.parse(data);
  if(Reply['result'] == 'NOK')
  {
    Code = Reply['respcode'];

    if(Code == 5)
      myApp.alert("User with this username already exists", 'Registration failed');
    else if(Code == 6)
      myApp.alert("User with this email already exists", 'Registration failed');
    else if(Code == 7)
      myApp.alert('Invalid email', 'Registration failed');
    else if(Code == 8)
      myApp.alert('Passwords do not match', 'Registration failed');
    else if(Code == 9)
      myApp.alert('Please check your data. All fields are mandatory', 'Registration failed');
    else

```

```

        myApp.alert('Try again later (' + Code + ')', 'Registration failed');
    }
    else
    {
        Name = Reply['name'];

        myApp.alert('Congratulations! You are now a member of BookLoyalty', 'Registration
Successful', function () {
            window.localStorage.setItem("loggedIn", 1);
            window.localStorage.setItem("Name", Reply['name']);
            myApp.closeModal();
            mainView.router.loadPage('main.html');
            mainView.showNavbar();
            mainView.showToolbar();
        });
    }
},

function(xhr) {
    myApp.alert('Connection failed. Please try again', 'Registration failed');
});
});

function RegisterScanners()
{
    $$('#AwardIcon').on('mousedown', function(){
        $$('#AwardIcon').attr('src','img/award-press.png');
    }).on('mouseup', function(){
        $$('#AwardIcon').attr('src','img/award.png');
    });

    $$('#RedeemIcon').on('mousedown', function(){
        $$('#RedeemIcon').attr('src','img/redeem-press.png');
    }).on('mouseup', function(){
        $$('#RedeemIcon').attr('src','img/redeem.png');
    });

    $$('#AwardIcon').on('click', function(){
        cordova.plugins.barcodeScanner.scan(
            function (result) {
                if(result.cancelled == true)
                    return;

                AwardPoints(result.text);
            },
            function (error) {
                myApp.alert("Scanning failed: " + error, 'Scanning error');
            },
            {

```

```

        "preferFrontCamera" : false, // iOS and Android
        "prompt" : "Place book barcode inside the scan area", // supported on Android only
        "orientation" : "landscape" // Android only (portrait|landscape), default unset so it rotates with
the device
    }
    );
});

$$('#RedeemIcon').on('click', function(){
cordova.plugins.barcodeScanner.scan(
function (result) {
if(result.cancelled == true)
return;

RedeemPoints(result.text);
},
function (error) {
myApp.alert("Scanning failed: " + error, 'Scanning error');
},
{
"preferFrontCamera" : false, // iOS and Android
"prompt" : "Place book barcode inside the scan area", // supported on Android only
"orientation" : "landscape" // Android only (portrait|landscape), default unset so it rotates with
the device
}
);
});
}

```

```

function AwardPoints(Points)
{
var requestData = '{ "request" : "award", "points" : "' + Points + "' }';

AjaxRequest(
HostURL,
requestData,

function(data) {
Reply = JSON.parse(data);
if(Reply['result'] == 'NOK')
{
Code = Reply['respcode'];

if(Reply['respcode'] == 2)
{
myApp.alert('You are not logged in', 'Connection error');
window.localStorage.setItem("loggedIn", 0);
mainView.router.loadPage('login.html');
}
else if(Reply['respcode'] == 2)
{

```

```

        myApp.alert('The book you have scanned is not in our database', 'Unknown book');
    }
    else
    {
        myApp.alert('Try again later (' + Code + ')', 'Error');
    }
}
else
{
    myApp.alert('You have been awarded ' + Reply['award'] + ' points', 'Congratulations!');
    countUp($$('#PointsCounter'), Reply['points']);
}
},

function(xhr) {
    myApp.alert('Connection failed. Please try again', 'Connection error');
});
}

function RedeemPoints(Points)
{
    var requestData = '{ "request" : "redeem", "points" : "' + Points + "' }';

    AjaxRequest(
        HostURL,
        requestData,

        function(data) {
            Reply = JSON.parse(data);
            if(Reply['result'] == 'NOK')
            {
                Code = Reply['respcode'];

                if(Reply['respcode'] == 2)
                {
                    myApp.alert('You are not logged in', 'Connection error');
                    window.localStorage.setItem("loggedIn", 0);
                    mainView.router.loadPage('login.html');
                }
                else if(Reply['respcode'] == 10)
                    myApp.alert('The book you have scanned is not in our database', 'Unknown book');
                else if(Reply['respcode'] == 11)
                    myApp.alert('You don\'t have enough points to redeem this book', 'Not enough points');
                else
                {
                    myApp.alert('Try again later (' + Code + ')', 'Error');
                }
            }
            else
            {
                myApp.alert('You have redeemed ' + Reply['redemption'] + ' points', 'Congratulations!');
                countUp($$('#PointsCounter'), Reply['points']);
            }
        }
    );
}

```

```

    }
  },

  function(xhr) {
    myApp.alert('Connection failed. Please try again', 'Connection error');
  });
}

function SetUserPoints()
{
  var requestData = '{ "request" : "getpoints" }';

  AjaxRequest(
    HostURL,
    requestData,

    function(data) {
      Reply = JSON.parse(data);
      if(Reply['result'] == 'NOK')
      {
        Code = Reply['respcode'];

        if(Reply['respcode'] == 2)
        {
          myApp.alert('You are not logged in', 'Connection error');
          window.localStorage.setItem("loggedIn", 0);
          mainView.router.loadPage('login.html');
        }
        else
        {
          myApp.alert('Try again later (' + Code + ')', 'Error');
        }
      }
      else
      {
        countUp($('#PointsCounter'), Reply['points']);
      }
    }
  ),

  function(xhr) {
    SetUserPoints();
    // myApp.alert('Connection failed. Please try again', 'Connection error');
  });
}

// Generic Ajax request wrapper
function AjaxRequest(URL, ReqData, SuccessFunction, FailFunction)
{
  $$$.ajax({
    url: URL,
    method: "POST",

```



```
    contentType: 'application/json',
    dataType: 'application/json',
    processData: false,
    xhrFields: { withCredentials: true },
    crossDomain: true,

    data: ReqData,
    success: SuccessFunction,
    error: FailFunction
  });
}
```

```
// Simple animator that starts counting up to a value 'count'
function countUp(object, count)
{
  var div_by = 100,
      speed = Math.round(count / div_by),
      $display = object,
      run_count = 1,
      int_speed = 24;

  var int = setInterval(function() {
    if(run_count < div_by){
      $display.text(speed * run_count);
      run_count++;
    } else if(parseInt($display.text()) < count) {
      var curr_count = parseInt($display.text()) + 1;
      $display.text(curr_count);
    } else {
      clearInterval(int);
    }
  }, int_speed);}
```

4.2 Κεντρικός Server

Όπως αναφέραμε ο κεντρικός σέρβερ είναι υπεύθυνος για την διαχείριση όλης της λειτουργίας της mobile εφαρμογής. Η γλώσσα που αναπτύχθηκε είναι Python , Django web framework. Οι λειτουργίες που πραγματοποιούνται από το models.py και views.py και θα αναλυθούν παρακάτω.

4.2.1 Models.py

Το Models που έχουμε υλοποιήσει στην εφαρμογή μας, είναι κλάσεις που δημιουργούν αυτόματα την βάση, τους πίνακες και τα foreign keys. Παρακάτω θα δούμε τις κλάσεις που δημιουργούν τους πίνακες LoyaltyProfile, Item, Award, Redemption, Offers στην βάση BookLoyalty.

```
from django.db import models
from django.conf import settings
from django.core.exceptions import ValidationError

# Create your models here.

class LoyaltyProfile(models.Model):
    Points = models.IntegerField('Points', default=0)
    User = models.OneToOneField(settings.AUTH_USER_MODEL)

class Item(models.Model):
    Name = models.CharField('Name', max_length = 100)
    Author = models.CharField('Author', max_length = 30)
    Price = models.DecimalField('Price', max_digits=10, decimal_places=2)
    Description = models.TextField('Description', null=True)
    Image = models.ImageField(upload_to='BookImages')
    ISBN = models.CharField(max_length = 13)

class Award(models.Model):
    Timestamp = models.DateTimeField('Award Timestamp')
    Points = models.IntegerField('Points', default=0)
    # Item = models.ForeignKey(Item)
    Profile = models.ForeignKey(LoyaltyProfile)

class Redemption(models.Model):
    Timestamp = models.DateTimeField('Redemption Timestamp')
```

```
Points = models.IntegerField('Points', default=0)

# Item = models.ForeignKey(Item)
Profile = models.ForeignKey(LoyaltyProfile)

class Offer(models.Model):
    FromDateTime = models.DateTimeField('Validity start')
    ToDateTime = models.DateTimeField('Validity end')

    Item = models.ForeignKey(Item)

    DiscountPercentage = models.DecimalField('Discount (%)'

    BonusPoints = models.IntegerField('Bonus Points')
```

4.2.2 Views.py

Σε αυτό το σημείο γίνεται το validation των βασικότερων λειτουργιών της εφαρμογής όπως αποκρυπτογράφηση πόντων που θα διαβαστούν από το QR code, το login , register, προσθήκη πόντων καθώς και εξαργύρωση πόντων. Οι βασικές κλήσεις που γίνονται από την εφαρμογή του κινητού (client) προς τον κεντρικό σέρβερ είναι οι εξής:

- ProcessLogin
- ProcessRegister
- ProcessGetPoints
- ProcessGetAwards
- ProcessGetRedemptions
- ProcessRedeem
- ProcessAward
- ProcessOffers

Παρακάτω παραθέεται κομμάτι κώδικα που δείχνει αυτή την λειτουργία της κάθε συνάρτησης ξεχωριστά.

- **ProcessLogin**

```
def ProcessLogin(request):
    ReqData = json.loads(request.body)

    user = authenticate(username = ReqData['username'], password =
    ReqData['password'])
```

```
if user != None:

    if user.is_active:
        login(request, user)
        Name = user.first_name + ' ' + user.last_name
        return ApproveRequest({ 'name' : Name })
    else:
        return DeclineRequest(3)

return DeclineRequest(2)
```

- **ProcessRegister**

```
def ProcessRegister(request):
    ReqData = json.loads(request.body)
    print "Registration process called"

    RequiredFields = ('username', 'email', 'firstname', 'lastname', 'password',
                      'password-repeat')

    for field in RequiredFields:
        if field not in ReqData.keys() or len(ReqData[field]) == 0:
            return DeclineRequest(9) # mandatory fields are missing

    if ReqData['password'] != ReqData['password-repeat']:
        return DeclineRequest(8)

    if validateEmail(ReqData['email']) == False:
        return DeclineRequest(7)

    try:
        user = User.objects.create_user(ReqData['username'], ReqData['email'],
                                        ReqData['password'])

        user.first_name = ReqData['firstname']
        user.last_name = ReqData['lastname']

        profile = models.LoyaltyProfile(User=user, Points=0)

        user.save()
```

```
profile.save()
```

```
user = authenticate(username = ReqData['username'], password =
ReqData['password'])
login(request, user)
Name = user.first_name + ' ' + user.last_name

return ApproveRequest({ 'name' : Name })

except IntegrityError:
return DeclineRequest(5)
except Exception as ex:
template = "An exception of type {0} occurred. Arguments:\n{1!r}"
message = template.format(type(ex).__name__, ex.args)
print message

return DeclineRequest(99)
```

- **ProcessGetPoints**

```
def ProcessGetPoints(request):
print('client requested points')

if not request.user.is_authenticated():
return DeclineRequest(2)

Profile = models.LoyaltyProfile.objects.get(User=request.user)

return ApproveRequest( { 'points': Profile.Points } )
```

- **ProcessGetAwards**

```
def ProcessGetAwards(request):
print('client requested award history')

if not request.user.is_authenticated():
return DeclineRequest(2)
```

```
AwardHistory =
models.Award.objects.filter(Profile__User=request.user).order_by('-Timestamp')
```

```
AwardHistoryJson = ()
for Award in AwardHistory:
    AwardJson = {}
    AwardJson['timestamp'] = str(Award.Timestamp)
    AwardJson['item'] = Award.Item.Name
    AwardJson['points'] = Award.Points

    AwardHistoryJson += AwardJson,

ReplyData = {
    'data': AwardHistoryJson
}

return ApproveRequest(ReplyData)
```

- **ProcessGetRedemptions**

```
def ProcessGetRedemptions(request):
    print('client requested redemption history')

    if not request.user.is_authenticated():
        return DeclineRequest(2)

    RedemptionHistory =
models.Redemption.objects.filter(Profile__User=request.user).order_by('-
Timestamp')

    RedemptionHistoryJson = ()
    for Redemption in RedemptionHistory:
        RedemptionJson = {}
        RedemptionJson['timestamp'] = str(Redemption.Timestamp)
        RedemptionJson['item'] = Redemption.Item.Name
        RedemptionJson['points'] = Redemption.Points

        RedemptionHistoryJson += RedemptionJson,

    ReplyData = {
```

```
        'data': RedemptionHistoryJson
    }

    return ApproveRequest(ReplyData)
```

- **ProcessRedeem**

```
def ProcessRedeem(request):
    print('client requested redemption')

    if not request.user.is_authenticated():
        return DeclineRequest(2)

    ReqData = json.loads(request.body)

    try:
        PointsToRedeem = DecryptPoints(ReqData['points'])
    except:
        return DeclineRequest(9)

    Profile = models.LoyaltyProfile.objects.get(User=request.user)

    if Profile.Points < PointsToRedeem:
        return DeclineRequest(11)

    Profile.Points -= PointsToRedeem
    Profile.save()

    Redemption=models.Redemption(Timestamp=datetime.utcnow().replace(tzinfo=utc),
    Points = PointsToRedeem, Profile = Profile)
    Redemption.save()

    ReplyData = {
        'points': int(Profile.Points),
        'redemption' : int(PointsToRedeem),
    }

    return ApproveRequest( ReplyData )
```

- **ProcessAward**

```
def ProcessAward(request):
    print('client requested award')

    if not request.user.is_authenticated():
        return DeclineRequest(2)
    ReqData = json.loads(request.body)

    try:
        AwardedPoints = DecryptPoints(ReqData['points'])
    except:
        return DeclineRequest(9)

    Profile = models.LoyaltyProfile.objects.get(User=request.user)

    Profile.Points += AwardedPoints
    Profile.save()

    Award = models.Award(Timestamp =
datetime.utcnow().replace(tzinfo=utc),Points=AwardedPoints, Profile=Profile)
    Award.save()

    ReplyData = {
        'points': int(Profile.Points),
        'award' : int(AwardedPoints),
    }
    return ApproveRequest( ReplyData )
```

- **ProcessOffers**

```
def ProcessOffers(request):
    print('client requested offers')
    if not request.user.is_authenticated():
        return DeclineRequest(2)
    offers = models.Offer.objects.filter(FromDate__lte =
datetime.utcnow().replace(tzinfo=utc)).filter(ToDate__gte =
datetime.utcnow().replace(tzinfo=utc))
    OffersJs = {}
    Id = 0
    for offer in offers:
```



```

OfferJs = {
    'name': offer.Item.Name,
    'author': offer.Item.Author,
    'ends' : str(offer.ToDateTime),
    'startprice': str(offer.Item.Price),
    'endprice' : str(round(offer.Item.Price - (offer.Item.Price
* ( offer.DiscountPercentage / 100 )), 2)),
    'percentage' : str(offer.DiscountPercentage),
    'bonuspoints' : str(offer.BonusPoints),
    'image'      : settings.MEDIA_URL + str(offer.Item.Image),
}
OffersJs[Id] = OfferJs
Id+=1
ReplyData = { 'offerdata' : OffersJs }
return ApproveRequest(ReplyData)

```

4.3 QR Code

Σε αυτή την ενότητα θα περιγράψουμε πως αποκρυπτογραφείτε το 12ψήφιο string που αντιστοιχεί σε κάθε QR code που θα σκανάρει η εφαρμογή. Αρχικά οι πόντοι κρυπτογραφούνται 3DES CBC και με ένα κλειδί που αναγνωρίζει μόνο ο server. Το string έχει την εξής μορφή [6 digits points][Current date in DD/MM/YY]. Όταν η εφαρμογή σκανάρει το κρυπτογραφημένο μήνυμα το στέλνει στον server και τότε ο server το αποκρυπτογραφεί και το κάνει award ή redemption εάν είναι σωστό.

Οι συναρτήσεις που πραγματοποιούν την κρυπτογράφηση και την αποκρυπτογράφηση στον server είναι οι παρακάτω:

```

def DecryptPoints(EncPoints):

    from Crypto.Cipher import DES

    from binascii import hexlify, unhexlify

    cipher = DES.new(unhexlify(settings.ENC_KEY), DES.MODE_CBC,
unhexlify('0000000000000000'))

    plain = cipher.decrypt(unhexlify(EncPoints))

    EncDate = datetime.strptime(hexlify(plain)[:6], '%d%m%y').date()

    if settings.STRICT_PTS == True and EncDate !=
datetime.utcnow().replace(tzinfo=utc).date():

        raise

    return int(hexlify(plain)[6:])

```

```
def EncryptPoints(Points):  
  
    from Crypto.Cipher import DES  
  
    from binascii import hexlify, unhexlify  
  
    cipher = DES.new(unhexlify(settings.ENC_KEY), DES.MODE_CBC,  
                    unhexlify('0000000000000000'))  
  
    DateStr = datetime.utcnow().replace(tzinfo=utc).strftime('%d%m%y')  
  
    plaintext = "{0}{1:0>10}".format(DateStr, str(Points))  
  
    return hexlify(cipher.encrypt(unhexlify(plaintext)))
```

5. Συμπεράσματα και Μελλοντικές Επεκτάσεις

Έχοντας κάνει ανασκόπηση του πεδίου της εφαρμογής, έχοντας δει τις λύσεις που υπάρχουν στην αγορά για εφαρμογές επιβράβευσης και έχοντας δει συνολικά την υλοποίηση της εφαρμογής αυτής της εργασίας μπορούμε εύκολα να καταλάβουμε ότι οι περισσότερες επιχειρήσεις έχουν το δικό τους πρόγραμμα ανταποδοτικής πίστης (loyalty programs). Μια τέτοια εφαρμογή μπορεί να βοηθήσει πολύ στην ανάπτυξη μιας οποιαδήποτε επιχείρησης με την προσέλκυση νέων πελατών καθώς και να δώσει κίνητρο νέων αγορών. Η αυξανόμενη χρήση κινητών συσκευών (smartphones και tablets) καθιστά αναγκαία την δημιουργία και εξέλιξη τέτοιου είδους εφαρμογών.

Οι μελλοντικές επεκτάσεις που θα μπορούσαν να γίνουν στο σύστημά μας είναι πολλές. Παρακάτω θα αναφερθούν ενδεικτικά κάποιες από τις επεκτάσεις της εφαρμογής:

- Ενσωμάτωση μέσω κοινωνικής δικτύωσης, για να μπορούν οι πελάτες να μοιράζονται προσφορές και νέα του βιβλιοπωλείου. Επίσης οι διοικούντες της επιχείρησης να γνωρίζουν τι συζητείται στο διαδίκτυο για το κατάστημα τους.
- Ο πελάτης να λέει τη γνώμη του για το επίπεδο εξυπηρέτησης του καταστήματος και να κερδίζει πόντους.
- Ενσωμάτωση Email, όπου το ημερολόγιο του χρήστη θα μπορούσε να συγχρονίζεται με κάποιον mail server (Microsoft Exchange Server κα) και οι χρήστες να μπορούν να στείλουν emails κατευθείαν μέσω της εφαρμογής προκειμένου να κάνουν αγορές βιβλίων.
- Δυνατότητα συλλογής και εξαργύρωσης πόντων αυτόματα από EFTPOS τερματικό κάνοντας χρήση NFC.



6. Βιβλιογραφία

- <https://play.google.com/store/search?q=loyalty%20app&c=apps>
- <http://www.mirror.co.uk/lifestyle/family/top-10-loyalty-cards-help-4020713>
- <http://technologyadvice.com/blog/marketing/why-customers-participate-loyalty-programs/>
- <http://www.forbes.com/forbes/welcome/?toURL=http://www.forbes.com/sites/hbsworkingknowledge/2014/02/24/six-myths-about-customer-loyalty-programs/4/&refURL=&referrer=#23ce366a2a7c>
- <http://marketingland.com/8-reasons-loyalty-programs-imperative-marketers-109077>
- https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm
- https://www.tools4noobs.com/online_tools/encrypt/
- <https://docs.djangoproject.com/el/1.10/ref/contrib/auth/>