

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ



ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ & ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΚΑΤΕΥΘΥΝΣΗ: ΗΛΕΚΤΡΟΝΙΚΗ ΜΑΘΗΣΗ

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

ΔΙΑΧΕΙΡΙΣΗΣ ΕΚΠΑΙΔΕΥΤΙΚΩΝ

ΣΥΓΓΡΑΜΜΑΤΩΝ

Αριστείδης Λυμπερόπουλος (Α.Μ.: 11025)

Επιβλέπων Καθηγητής: Συμεών Ρετάλης

Διπλωματική Εργασία υποβληθείσα στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης στην Διδακτική της Τεχνολογίας και των Ψηφιακών Συστημάτων με κατεύθυνση την Ηλεκτρονική Μάθηση.

Πειραιάς, Νοέμβριος 2015

*Η Διπλωματική Εργασία
αφιερώνεται στη σύζυγό μου Αθηνά
και στον υιό μου Κωνσταντίνο...*

Ευχαριστίες

Για την εκπόνηση της παρούσας εργασίας και την επιτυχή ολοκλήρωσή της μια βιαστική εκτίμηση θα ήταν ότι χρησιμοποιήθηκαν δεξιότητες και αναπτύχθηκαν ικανότητες οι οποίες υπερβαίνουν τα προτεινόμενα πλαίσια και τους διδακτικούς στόχους του μεταπτυχιακού προγράμματος. Ωστόσο με μια προσεκτικότερη ματιά αντιλαμβάνεται κάποιος ότι αυτός είναι ο απώτερος σκοπός ενός επιτυχημένου προγράμματος σπουδών, ακριβώς δηλαδή το να καταφέρει να προάγει τη μελέτη και την έρευνα διευρύνοντας τα πεδία αναζήτησης και εργασίας.

Με το πέρας αυτής της εργασίας κλείνει ένας κύκλος σπουδών πολύ σημαντικών για τη ζωή μου, αφού συνέβαλλαν σε μια βαθιά και ριζική αλλαγή στην επαγγελματική μου πορεία. Αυτή η διαδρομή με έχει οδηγήσει σε ένα σταυροδρόμι όπου μπροστά μου ξετυλίγονται αχαρτογράφητα μέρη γεμάτα προκλήσεις. Το σημαντικό είναι ότι κάθε βήμα μου υποστυλώνεται από την ικανοποίηση που νιώθω κάνοντας κάτι δημιουργικό και που πραγματικά με ευχαριστεί.

Στις περιπέτειές μου όμως δεν είμαι μόνος. Είναι μαζί η οικογένειά μου, η σύζυγος και ο υιός μου. Αθηνά, ευχαριστώ για την κατανόηση, την υποστήριξη και την υπομονή σου όλο αυτό το διάστημα. Κωνσταντίνε, σου υπόσχομαι να αναπληρώσω τον χρόνο που στερήθηκες μαζί μου. Ευχαριστώ τους γονείς μου για την αμέριστη συμπαράσταση όλα αυτά τα χρόνια.

Τέλος, θα ήθελα να ευχαριστήσω τον καθηγητή μου, κύριο Συμεών Ρετάλη για την ανατροφοδότηση και τις πολύτιμες συμβουλές που μου παρείχε όλο αυτό το διάστημα προκειμένου να ολοκληρωθεί η εργασία αυτή.

Ανάπτυξη Διαδικτυακού Συστήματος Διαχείρισης Εκπαιδευτικών Συγγραμμάτων

Σημαντικοί Όροι: Διαχείριση Έργου, Ευέλικτη Μεθοδολογία, Scrum, Ruby on Rails, MongoDB, Ruby, ElasticSearch, MVC, NoSQL,

Περίληψη

Η παρούσα εργασία αφορά στον σχεδιασμό και στην υλοποίηση ενός διαδικτυακού συστήματος διαχείρισης εκπαιδευτικών συγγραμμάτων. Επικεντρώνεται στην αποτελεσματική διαχείριση του εκπαιδευτικού περιεχομένου και απαντάει σε προκλήσεις σχετικά με την οργάνωση, δημιουργία, χρήση και συντήρηση αυτού με απώτερο σκοπό την ταχύτερη και ευχερέστερη εξυπηρέτηση του μαθητή στην προβολή της σχολικής ύλης.

Ταυτόχρονα εστιάζει στις βασικές έννοιες διαχείρισης έργου και στα μοντέλα διαχείρισης έργων πληροφορικής, ενώ παράλληλα εμβαθύνει στις ευέλικτες προσεγγίσεις με στόχο να εφαρμόσει διαδικασίες τις μεθοδολογίας Scrum στην ανάπτυξη της εν θέματι εφαρμογής.

Η εφαρμογή βασίζεται σε τεχνολογίες αιχμής, όπως τη γλώσσα προγραμματισμού Ruby, το πλαίσιο εφαρμογών Ruby on Rails και τη βάση δεδομένων MongoDB, για τις οποίες γίνεται εκτεταμένη ανάλυση τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο.

Περιεχόμενα

Περίληψη.....	II
Κατάλογος Διαγραμμάτων.....	VIII
Κατάλογος Πινάκων.....	VIII
Κατάλογος Εικόνων	IX
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ	1
1.1. Γενικά.....	1
1.2. Αντικείμενο & Στόχοι της Διπλωματικής Εργασίας.....	2
1.3. Δομή της Διπλωματικής Εργασίας	3
ΚΕΦΑΛΑΙΟ 2: ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ	5
2.1. Εισαγωγή.....	5
2.2. Ορισμός Έργου	5
2.3. Ορισμός Διαχείρισης Έργου.....	6
2.4. Ορισμός Απαιτήσεων	7
2.5. Χαρακτηριστικά Απαιτήσεων.....	7
2.6. Διαδικασίες διαχείρισης έργου.....	8
2.6.1. Διαδικασίες Αρχικοποίησης.....	9
2.6.2. Διαδικασίες Σχεδιασμού	9
2.6.3. Διαδικασίες Εκτέλεσης.....	9
2.6.4. Διαδικασίες Παρακολούθησης & Ελέγχου.....	9
2.6.5. Διαδικασίες Ολοκλήρωσης.....	10
2.7. Ορισμός Κύκλου Ζωής Διαχείρισης Έργου.....	10
2.8. Συμπεράσματα.....	10
ΚΕΦΑΛΑΙΟ 3: ΜΟΝΤΕΛΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ.....	11
3.1. Εισαγωγή.....	11
3.2. Προσεγγίσεις Διαχείρισης Έργου.....	11
3.2.1. Παραδοσιακή Διαχείριση Έργου	11
3.2.1.1. Γραμμικό Μοντέλο Κύκλου Ζωής Διαχείρισης Έργου.....	12
3.2.1.2. Αυξητικό Μοντέλο Κύκλου Ζωής Διαχείρισης Έργου	12
3.2.2. Ευέλικτη Διαχείριση Έργου	13
3.2.2.1. Επαναληπτικό Μοντέλο Κύκλου Ζωής Διαχείρισης Έργου.....	13
3.2.2.2. Προσαρμοστικό Μοντέλο Κύκλου Ζωής Διαχείρισης Έργου	14
3.2.3. Extreme Διαχείριση Έργου.....	14
3.3. Συμπεράσματα.....	15

ΚΕΦΑΛΑΙΟ 4: ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ	16
4.1. Εισαγωγή.....	16
4.2. Βασικές Αξίες Ευέλικτης Διαχείρισης Έργων.....	16
4.3. Βασικές Αρχές της Ευελιξίας.....	17
4.4. Μεθοδολογία Scrum.....	18
4.4.1 Βασικές Έννοιες.....	18
4.4.1.1. Sprint.....	19
4.4.1.2. Ρόλοι & Ευθύνες.....	19
4.4.1.3. Παραδοτέα Αντικείμενα.....	20
4.4.1.4. Γεγονότα.....	20
4.4.2. Διαδικασία.....	21
4.4.2.1. Στάδιο 1 – Όραμα.....	21
4.4.2.2. Στάδιο 2 – Οδικός Χάρτης Προϊόντος.....	21
4.4.2.3. Στάδιο 3 – Σχεδιασμός Εκδόσεων.....	22
4.4.2.4. Στάδιο 4 – Σχεδιασμός Sprint.....	23
4.4.2.5. Στάδιο 5 – Καθημερινή Συνάντηση.....	23
4.4.2.6. Στάδιο 6 – Ανασκόπηση Scrum.....	23
4.4.2.7. Στάδιο 7 – Ανατροφοδότηση Scrum.....	24
4.4.2.8. Στάδιο 8 – Δημοσίευση Έκδοσης.....	24
4.5. Συμπεράσματα.....	24
ΚΕΦΑΛΑΙΟ 5: ΠΑΡΟΥΣΙΑΣΗ ΤΕΧΝΟΛΟΓΙΩΝ	25
5.1. Εισαγωγή.....	25
5.2. Ruby.....	25
5.3. Ruby on Rails.....	26
5.3.1. Αρχές Σχεδίασης του RoR.....	26
5.3.2. Πρότυπο Σχεδίασης: Μοντέλο-Όψη-Ελεγκτής (Model-View-Controller).....	27
5.3.3. Πρότυπο REST (Representational State Transfer).....	29
5.3.4. RubyGems.....	29
5.4. Βάσεις Δεδομένων.....	29
5.4.1. Ορισμός της Βάσης Δεδομένων.....	30
5.4.2. Συστήματα Διαχείρισης Βάσεων Δεδομένων.....	30
5.4.2.1. Ιεραρχικό & Δικτυωτό Μοντέλο Βάσεων Δεδομένων.....	30
5.4.2.2. Σχεσιακό Μοντέλο Βάσεων Δεδομένων.....	31
5.4.2.3. Μετά-σχεσιακά Μοντέλα Βάσεων Δεδομένων.....	32
5.5. MongoDB.....	34

5.5.1. Χαρακτηριστικά της MongoDB.....	34
5.6. ElasticSearch.....	36
5.6.1. Βασικές έννοιες.....	36
5.6.2. Δημιουργία, Διαγραφή & Ανάκτηση Δεδομένων.....	38
ΚΕΦΑΛΑΙΟ 6: ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ: «ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΣΥΓΓΡΑΜΜΑΤΩΝ».....	40
6.1. Εισαγωγή.....	40
6.2. Διαδικασίες πριν την Εκκίνηση.....	41
6.2.1. Αίτηση Έναρξης Ανάπτυξης Εφαρμογής.....	41
6.2.2. Ανάθεση Ρόλων.....	42
6.2.3. Προσδιορισμός Κινδύνων.....	43
6.2.4. Προϋπολογισμός Έργου.....	43
6.3. Στάδιο 1 ^ο – Όραμα.....	44
6.4. Στάδιο 2 ^ο – Οδικός Χάρτης Έργου.....	45
6.4.1. Αρχικές Απαιτήσεις Πελάτη.....	46
6.4.2. Επιθυμητά Χαρακτηριστικά.....	46
6.4.3. Βασικό Σχέδιο Έργου.....	48
6.5. Στάδιο 3 ^ο – Προγραμματισμός Εκδόσεων.....	52
6.5.1. Product Backlog.....	52
6.5.2. Σχεδιασμός Εκδόσεων.....	54
6.6. Στάδιο 4 ^ο – Σχεδιασμός Sprint.....	55
6.7. Στάδιο 5 ^ο – Καθημερινές Συναντήσεις.....	59
6.8. Υλοποίηση Εκδόσεων.....	60
6.8.1. Έκδοση 1.0 – Αρχική Σελίδα.....	60
6.8.1.1. Sprint 1.0.1 – Σκελετός Εφαρμογής.....	61
6.8.1.2. Sprint 1.0.2 – Σύνδεση με Βάση Δεδομένων.....	63
6.8.1.3. Sprint 1.0.3 – Εγκατάσταση πακέτου bootstrap-sass.....	63
6.8.1.4. Sprint 1.0.4 – Εγκατάσταση πακέτου font-awesome-rails.....	64
6.8.1.5. Sprint 1.0.5 – Εγκατάσταση πακέτου Haml.....	65
6.8.1.6. Sprint 1.0.6 – Δημιουργία Controller & View για Στατικές Σελίδες.....	65
6.8.1.7 Sprint 1.0.7 – Συγγραφή Κώδικα Αρχικής Σελίδας.....	66
6.8.2. Έκδοση 1.1 – Καταχώρηση Βιβλίων.....	67
6.8.2.1. Sprint 1.1.1 – Καθορισμός Οντοτήτων.....	69
6.8.2.2. Sprint 1.1.2 – Δημιουργία Πόρου Book.....	70
6.8.2.3. Sprint 1.1.3 – Δημιουργία μοντέλου Publisher.....	73
6.8.2.4. Sprint 1.1.4 – Δημιουργία μοντέλου Author.....	73

6.8.2.5. Sprint 1.1.5 – Δημιουργία μοντέλου Preface	74
6.8.2.6. Sprint 1.1.6 – Δημιουργία μοντέλου Preface Paragraph.....	74
6.8.2.7. Sprint 1.1.7 – Δημιουργία μοντέλου School Grade.....	75
6.8.2.8. Sprint 1.1.8 – Δημιουργία μοντέλου Chapter.....	75
6.8.2.9. Sprint 1.1.9 – Δημιουργία μοντέλου Section	76
6.8.2.10. Sprint 1.1.10 – Δημιουργία μοντέλου Paragraph	76
6.8.2.11. Sprint 1.1.11 – Προσθήκη Admin namespace στο routes.rb	77
6.8.2.12. Sprint 1.1.12 – Παραμετροποίηση του books_controller.rb.....	78
6.8.2.13. Sprint 1.1.13 – Διαμόρφωση της όψεως books/new.html.haml.....	79
6.8.3. Έκδοση 1.2 – Προβολή όλων των Βιβλίων.....	82
6.8.3.1. Sprint 1.2.1 – Διαμόρφωση της Όψεως books/index.html.haml	82
6.8.4. Έκδοση 1.3 – Επεξεργασία Βιβλίων	83
6.8.4.1. Sprint 1.3.1 – Διαμόρφωση Όψεως books/edit.html.haml.....	84
6.8.4.2. Sprint 1.3.2 – Παραμετροποίηση Όψεως books/index.html.haml	86
6.8.5. Έκδοση 1.4 – Διαγραφή Βιβλίων	87
6.8.5.1. Sprint 1.4.1 – Παραμετροποίηση Όψεως books/index.html.haml	88
6.8.6. Έκδοση 1.5 – Κατηγοριοποίηση Βιβλίων	89
6.8.6.1. Sprint 1.5.1 – Δημιουργία Ελεγκτή student/books_controller.rb	90
6.8.6.2. Sprint 1.5.2 – Προσθήκη Student namespace στο routes.rb.....	90
6.8.6.3. Sprint 1.5.3 – Παραμετροποίηση Ελεγκτή student/books_controller.rb.....	91
6.8.6.4. Sprint 1.5.4 – Διαμόρφωση Όψεως student/books/index.html.haml	91
6.8.6.5. Sprint 1.5.5 - Παραμετροποίηση Ελεγκτή student/books_controller.rb	93
6.8.7. Έκδοση 1.6 – Προβολή Περιεχομένου Βιβλίου.....	94
6.8.7.1. Sprint 1.6.1 – Δημιουργία Όψεως student/books/show.html.haml	94
6.8.8. Έκδοση 1.7 – Είσοδος Μαθητή στην Εφαρμογή	95
6.8.8.1. Sprint 1.7.1. – Εγκατάσταση πακέτου ‘devise’	96
6.8.8.2. Sprint 1.7.2 – Δημιουργία μοντέλου User	97
6.8.8.3. Sprint 1.7.3 – Δημιουργία Student Controller	98
6.8.8.4. Sprint 1.7.4 – Παραμετροποίηση Ελεγκτή controllers/student.....	98
6.8.8.5. Sprint 1.7.5 – Διαμόρφωση Όψεως devise /registrations /new .html.haml .	99
6.8.8.6. Sprint 1.7.6 – Διαμόρφωση Όψεως devise/sessions/new.html.haml	100
6.8.8.7. Sprint 1.7.7 – Διαμόρφωση Όψεως devise/passwords/new.html.haml	101
6.8.9. Έκδοση 1.8 – Είσοδος Διαχειριστή στην Εφαρμογή.....	102
6.8.9.1. Sprint 1.8.1 – Δημιουργία μοντέλου Administrator	102
6.8.9.2. Sprint 1.8.2 – Δημιουργία Admin Controller.....	102

6.8.9.3. Sprint 1.8.3 – Παραμετροποίηση Ελεγκτή controllers/admin/books_controller.rb	103
6.9. Στάδιο 6^ο – Ανασκόπηση Scrum	103
6.9.1. Ανασκόπηση Έκδοσης 1.0	103
6.9.2. Ανασκόπηση Έκδοσης 1.1	104
6.9.3. Ανασκόπηση Έκδοσης 1.2	106
6.9.4. Ανασκόπηση Έκδοσης 1.3	107
6.9.5. Ανασκόπηση Έκδοσης 1.4	108
6.9.6. Ανασκόπηση Έκδοσης 1.5	109
6.9.7. Ανασκόπηση Έκδοσης 1.6	111
6.9.8. Ανασκόπηση Εκδόσεων 1.7 & 1.8	111
6.10. Στάδιο 7^ο – Ανατροφοδότηση Scrum.....	113
ΚΕΦΑΛΑΙΟ 7: ΕΠΙΣΚΟΠΗΣΗ ΠΑΡΟΜΟΙΩΝ ΕΦΑΡΜΟΓΩΝ.....	114
7.1. Εισαγωγή.....	114
7.2. Συστήματα Διαχείρισης Περιεχομένου.....	114
7.3. Παρόμοιες εφαρμογές.....	115
7.3.1. Creatavist	116
7.3.2. eBook Maestro Pro	117
7.3.3. iBooks Author	117
7.3.4. Lucidpress	118
7.3.5. Pressbooks	118
7.6. Συμπεράσματα	119
ΚΕΦΑΛΑΙΟ 8: ΣΥΜΠΕΡΑΣΜΑΤΑ	122
8.1. Εισαγωγή.....	122
8.2. Λειτουργικότητα & Ευχρηστία.....	122
8.3. Μελλοντικές Επεκτάσεις.....	123
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	125

Κατάλογος Διαγραμμάτων

Διάγραμμα 1: Γραμμικό Μοντέλο Διαχείρισης Έργων.....	12
Διάγραμμα 2: Αυξητικό Μοντέλο Διαχείρισης Έργων.....	12
Διάγραμμα 3: Επαναληπτικό Μοντέλο Διαχείρισης Έργων	13
Διάγραμμα 4: Προσαρμοστικό Μοντέλο Διαχείρισης Έργων	14
Διάγραμμα 5: Extreme Μοντέλο Διαχείρισης Έργων.....	15
Διάγραμμα 6: Μεθοδολογία SCRUM.....	18
Διάγραμμα 7: Αρχιτεκτονική Μοντέλου – Όψεως – Ελεγκτή	28
Διάγραμμα 8: Αναπαράσταση επικοινωνίας των browser, controller και view.....	67
Διάγραμμα 9: Οντότητες – Συσχετίσεις (Ο/Σ) της Εφαρμογής	69

Κατάλογος Πινάκων

Πίνακας 1: NoSQL Βάσεις Δεδομένων.....	33
Πίνακας 2: Αίτηση Έναρξης Ανάπτυξης Συστήματος	41
Πίνακας 3: Δήλωση Οράματος για το έργο BookYourWay	44
Πίνακας 4: Επιθυμητά Χαρακτηριστικά	47
Πίνακας 5: Βασικό Σχέδιο Έργου	48
Πίνακας 6: Product Backlog.....	52
Πίνακας 7: Εκδόσεις.....	54
Πίνακας 8: Sprint Backlog	56
Πίνακας 9: Έκδοση 1.0 – Αρχική Σελίδα.....	60
Πίνακας 10: Έκδοση 1.1 – Καταχώρηση Βιβλίων	68
Πίνακας 11: Έκδοση 1.2 – Προβολή όλων των Βιβλίων	82
Πίνακας 12: Έκδοση 1.3 – Επεξεργασία Βιβλίων.....	83
Πίνακας 13: Έκδοση 1.4 – Διαγραφή Βιβλίων	87
Πίνακας 14: Έκδοση 1.5 – Κατηγοριοποίηση Βιβλίων	89
Πίνακας 15: Έκδοση 1.6 – Προβολή Περιεχομένου Βιβλίου	94
Πίνακας 16: Έκδοση 1.7 – Είσοδος Μαθητή στην Εφαρμογή.....	95
Πίνακας 17: Έκδοση 1.8 – Είσοδος Διαχειριστή στην Εφαρμογή.....	102

Πίνακας 18: Σύγκριση Παρόμοιων Εφαρμογών	119
--	-----

Κατάλογος Εικόνων

Εικόνα 1: Δημιουργία & Διαγραφή Εγγράφου στην ElasticSearch	38
Εικόνα 2: Ανάκτηση Εγγράφου στην ElasticSearch	39
Εικόνα 3: Αρχική Σελίδα.....	104
Εικόνα 4: Πρώτη οθόνη καταχώρησης βιβλίων.....	104
Εικόνα 5: Δεύτερη οθόνη καταχώρησης βιβλίων	105
Εικόνα 6: Τρίτη οθόνη καταχώρησης βιβλίων.....	105
Εικόνα 7: Τέταρτη οθόνη καταχώρησης βιβλίων.....	106
Εικόνα 8: Προβολή όλων των Βιβλίων.....	106
Εικόνα 9: Πρώτη Οθόνη Επεξεργασίας Βιβλίου.....	107
Εικόνα 10: Δεύτερη Οθόνη Επεξεργασίας Βιβλίου	107
Εικόνα 11: Τρίτη Οθόνη Επεξεργασίας Βιβλίου.....	108
Εικόνα 12: Τέταρτη Οθόνη Επεξεργασίας Βιβλίου	108
Εικόνα 13: Πρώτη Οθόνη Διαγραφής Βιβλίων.....	109
Εικόνα 14: Δεύτερη Οθόνη Διαγραφής Βιβλίων.....	109
Εικόνα 15: Πρώτη Οθόνη Κατηγοριοποίησης Βιβλίων	110
Εικόνα 16: Δεύτερη Οθόνη Κατηγοριοποίησης Βιβλίων.....	110
Εικόνα 17: Τρίτη Οθόνη Κατηγοριοποίησης Βιβλίων.....	111
Εικόνα 18: Προβολή Περιεχομένου Βιβλίου	111
Εικόνα 19: Sign In.....	112
Εικόνα 20: Sign Up	112
Εικόνα 21: Επανεκδοση Κωδικού Πρόσβασης.....	113

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1. Γενικά

Το τέλος του 20^{ου} αιώνα σηματοδοτήθηκε από τη ραγδαία εξέλιξη μιας σειράς τεχνολογιών με αιχμή τους τομείς της πληροφορικής και των τηλεπικοινωνιών. Οι αλλαγές που επέφεραν και συνεχίζουν να επιφέρουν στο παγκόσμιο γίγνεσθαι είναι τόσο σημαντικές που από πολλούς χαρακτηρίζεται η σημερινή εποχή ως εποχή της «Πληροφορικής Επανάστασης». Η νέα αυτή εποχή του τεχνολογικού πολιτισμού έχει αναμορφώσει όχι μόνο την παραγωγική διαδικασία, τις εργασιακές σχέσεις και τις συναλλαγές, αλλά κυρίως την ανθρώπινη επαφή και επικοινωνία.

Δομικό στοιχείο της κοινωνίας που ζούμε, της «κοινωνίας της πληροφορίας», είναι ο Παγκόσμιος Ιστός. Ο Παγκόσμιος Ιστός αποτελούσε πάντα το μέσο διασύνδεσης σε μια τράπεζα πληροφοριών. Αρχικά αυτό σήμαινε εύκολη σύνδεση σε πολλαπλά έγγραφα και δυνατότητα εντοπισμού μοναδικών πληροφοριών ακολουθώντας έναν σύνδεσμο. Στην πορεία, το ψηφιακό περιεχόμενο, εμπλουτίστηκε με οπτικοακουστικό υλικό και αποτέλεσε το κύριο μέσο μαζικής ενημέρωσης. Ταυτόχρονα κατέστη διαδραστικό δίνοντας τη δυνατότητα στους χρήστες να συνδέονται σε διάφορες υπηρεσίες και να αλληλοεπιδρούν με διαδικτυακές εφαρμογές αναπτύσσοντας τη δημιουργικότητά τους. Σήμερα αποτελεί πλέον αναπόσπαστο κομμάτι της καθημερινότητας των ανθρώπων συμβάλλοντας στην αναδιαμόρφωση της ζωής τους και κατ' επέκταση της κοινωνίας. Αυτό συμπεριλαμβάνει τον τρόπο με τον οποίο οι άνθρωποι ταυτίζονται, συμμετέχουν, συνδέονται, επικοινωνούν, εργάζονται, ψυχαγωγούνται, ενημερώνονται, εκπαιδεύονται, γνωρίζουν και μαθαίνουν.

Στις μέρες μας χιλιάδες διαδικτυακές υπηρεσίες και εφαρμογές αναπτύσσονται σε καθημερινή βάση. Παρατηρώντας προσεκτικά το εφελτήριο των σχεδιαστών-προγραμματιστών στην ανάπτυξη νέων εφαρμογών, διαπιστώνεται ότι η πρόθεσή τους είναι κοινή, δηλαδή η οργάνωση των δεδομένων με τέτοιο τρόπο ώστε να εξυπηρετούν συγκεκριμένο σκοπό σε σχέση πάντα με τις απαιτήσεις των χρηστών. Αυτό που είναι σκόπιμο να επιτευχθεί είναι η απεικόνιση του πραγματικού κόσμου. Έτσι τα δεδομένα από διάσπαρτα στοιχεία συσχετίζονται μεταξύ τους ώστε να μπορούν να περιγράψουν

οντότητες ή πράγματα που υφίστανται στην πραγματικότητα. Το αποτέλεσμα που προκύπτει από αυτή τη συσχέτιση καθορίζει την επιτυχία της εφαρμογής.

Επιτυχημένη εφαρμογή όμως θεωρείται και εκείνη που έχει χαρακτηριστικά όπως η απλότητα, η πρακτικότητα και η χρηστικότητα. Παρατηρείται τα τελευταία χρόνια μια αλλαγή στην ανάπτυξη των εφαρμογών από την πολυπλοκότητα των πολλαπλών λειτουργιών σε ένα μοντέλο σχεδίασης απλών και χρήσιμων λειτουργιών για τον χρήστη.

1.2. Αντικείμενο & Στόχοι της Διπλωματικής Εργασίας

Η εργασία αυτή επικεντρώνεται σε πτυχές του Παγκόσμιου Ιστού που αφορούν στη διαχείριση του ψηφιακού περιεχομένου στον τομέα της εκπαίδευσης, αναπτύσσοντας μια διαδικτυακή εφαρμογή. Το ψηφιακό περιεχόμενο καθίσταται αναγκαίο στον χώρο της εκπαίδευσης σηματοδοτώντας μια νέα εποχή εύκολης πρόσβασης στη μάθηση και στη γνώση. Η αποτελεσματική διαχείριση της συλλογικής γνώσης δημιουργεί νέες προκλήσεις για την οργάνωση, δημιουργία, χρήση και συντήρηση του ψηφιακού περιεχομένου.

Το αντικείμενο της παρούσας εργασίας αφορά στην ανάπτυξη μιας διαδικτυακής εφαρμογής διαχείρισης εκπαιδευτικών συγγραμμάτων, με χρήση λογισμικών ανοιχτού κώδικα και με στόχο την ταχύτερη και ευχερέστερη εξυπηρέτηση του μαθητή στην αναζήτηση και προβολή της σχολικής ύλης. Απώτερος σκοπός της εφαρμογής είναι οι μαθητές να μπορούν να μελετήσουν και να κατανοήσουν τη σχολική ύλη εύκολα και γρήγορα, διαθέτοντας εποικοδομητικά τον πολύτιμο χρόνο τους.

Η θεωρητική προσέγγιση που ακολουθείται για την ανάπτυξη της εφαρμογής βασίζεται στις ευέλικτες μεθοδολογίες διαχείρισης έργων πληροφορικής και συγκεκριμένα στη μεθοδολογία Scrum. Στο στάδιο της θεωρητικής θεμελίωσης κρίνεται σκόπιμο να αναλυθούν: α) οι βασικές έννοιες και τα μοντέλα διαχείρισης έργων πληροφορικής, β) οι βασικές αρχές των ευέλικτων προσεγγίσεων και γ) οι διαδικασίες της μεθοδολογίας Scrum, με στόχο τη βαθύτερη κατανόηση του γνωστικού πεδίου της ανάπτυξης συστημάτων.

Στο στάδιο της τεχνολογικής τεκμηρίωσης της εφαρμογής γίνεται εκτεταμένη ανάλυση: α) της γλώσσας προγραμματισμού Ruby, β) του πλαισίου ανάπτυξης

εφαρμογών Ruby on Rails, γ) της βάσης δεδομένων MongoDB και δ) του συστήματος αναζήτησης Elasticsearch, με στόχο την κατάρτιση στις παραπάνω τεχνολογίες και γενικότερα την απόκτηση γνώσεων και δεξιοτήτων αιχμής, αναγκαίων στη σημερινή αγορά εργασίας.

1.3. Δομή της Διπλωματικής Εργασίας

Η διπλωματική εργασία οργανώνεται σε επτά κεφάλαια, συμπεριλαμβανομένου και του παρόντος πρώτου κεφαλαίου, όπου παρατίθεται η εισαγωγή, το αντικείμενο και οι στόχοι.

Στο δεύτερο Κεφάλαιο προσεγγίζεται το γνωστικό πεδίο της διαχείρισης έργων πληροφορικής μέσα από την περιγραφή βασικών εννοιών και την παράθεση ορισμών. Οι έννοιες αυτές αποτελούν τη βάση για τη μετέπειτα ανάλυση των μοντέλων διαχείρισης έργου.

Το τρίτο κεφάλαιο επιδίδεται στην καταγραφή των διαφορετικών προσεγγίσεων διαχείρισης έργου, εστιάζοντας στην κατηγοριοποίηση των μοντέλων διαχείρισης έργων πληροφορικής και στην βαθύτερη διερεύνησή τους.

Το τέταρτο κεφάλαιο επικεντρώνεται στην ευέλικτη προσέγγιση διαχείρισης έργων πληροφορικής παραθέτοντας τις βασικές αξίες και αρχές της, καθώς επίσης εξετάζει τη μεθοδολογία Scrum στην οποία βασίζεται η διαχείριση του έργου ανάπτυξης της εφαρμογής.

Το πέμπτο κεφάλαιο παραθέτει και αναλύει τις πιο σημαντικές τεχνολογίες που χρησιμοποιούνται για την ανάπτυξη της εφαρμογής διαχείρισης εκπαιδευτικών συγγραμμάτων, εστιάζοντας στην αντικειμενοστρεφή γλώσσα προγραμματισμού Ruby, στο πλαίσιο ανάπτυξης εφαρμογών Ruby on Rails, στη βάση δεδομένων MongoDB και στο σύστημα αναζήτησης κειμενικών δεδομένων Elasticsearch.

Στο έκτο κεφάλαιο υλοποιείται η ανάπτυξη της εφαρμογής ακολουθώντας τα στάδια της μεθοδολογίας Scrum. Καταγράφονται λεπτομερώς οι διαδικασίες, οι απαιτήσεις του συστήματος, τα επιθυμητά χαρακτηριστικά του και ταυτόχρονα προσδιορίζονται οι λειτουργικές εκδόσεις και οι εργασίες που απαιτούνται για την επιτυχή ολοκλήρωσή τους, παραθέτοντας και αναλύοντας τον κώδικα που αναπτύχθηκε.

Στο έβδομο κεφάλαιο εντοπίζονται οι τυχόν ομοιότητες της εφαρμογής με τα Συστήματα Διαχείρισης Περιεχομένου και παρατίθενται άλλες εφαρμογές με παραπλήσιες λειτουργικότητες.

Στο όγδοο Κεφάλαιο, η εργασία ολοκληρώνεται με την ανασκόπηση των συμπερασμάτων που προέκυψαν κατά τη διάρκεια της εκπόνησης της διπλωματικής εργασίας.

ΚΕΦΑΛΑΙΟ 2: ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

2.1. Εισαγωγή

Σε αυτό το κεφάλαιο αναλύονται οι βασικές έννοιες του πεδίου της διαχείρισης έργων πληροφορικής.

2.2. Ορισμός Έργου

«Έργο είναι μια προσωρινή προσπάθεια που έχει στόχο τη δημιουργία ενός μοναδικού προϊόντος, υπηρεσίας ή αποτελέσματος». (PMI, 2013, σ. 2)

Αναλύοντας τον παραπάνω ορισμό του Ινστιτούτου Διαχείρισης Έργου διαφαίνεται ότι ένα έργο έχει πεπερασμένη διάρκεια του οποίου το τέλος εξαρτάται είτε από την επίτευξη των στόχων, είτε από την εξάντληση των διαθέσιμων πόρων.

Ένας εναλλακτικός ορισμός ο οποίος διατυπώνεται από τον Robert Wysocki στο βιβλίο του «Effective Project Management» είναι ο εξής:

«Έργο είναι μια ακολουθία μοναδικών, σύνθετων και αλληλένδετων διεργασιών οι οποίες έχουν ένα μόνο στόχο ή σκοπό και πρέπει να ολοκληρωθούν σε καθορισμένο χρόνο, εντός προϋπολογισμού και βάσει προδιαγραφών». (Wysocki, 2014, σ. 4)

Από τον ορισμό του Wysocki αρχικά συμπεραίνεται ότι οι διεργασίες ενός έργου έχουν μια ακολουθία και αλληλεξαρτώνται, δηλαδή το παραγόμενο αποτέλεσμα μίας διεργασίας αποτελεί τη βάση πάνω στην οποία θα στηριχτεί η επόμενη. Για παράδειγμα πρώτα σχεδιάζεται ένα λογισμικό και στη συνέχεια γράφεται ο κώδικας. Επίσης οι διεργασίες είναι μοναδικές, σαφείς, έχουν ξεκάθαρο στόχο, καθορισμένο χρονοδιάγραμμα και συγκεκριμένους πόρους. Τέλος ένα έργο πρέπει να πληροί τις προδιαγραφές που έχει θέσει ο πελάτης.

Αξίζει να σημειωθεί πως ο παραπάνω ορισμός έχει ένα σημαντικό μειονέκτημα: δεν είναι προσανατολισμένος στην επιχειρηματική αξία του πελάτη. Έτσι ενώ μπορεί να ικανοποιούνται όλοι οι περιορισμοί και οι συνθήκες που αναφέρονται ανωτέρω, ο

πελάτη ενδέχεται να μην είναι ικανοποιημένος από το αποτέλεσμα. Για τον λόγο αυτό ο Wysocki δίνει ένα δεύτερο ορισμό του έργου, στο ίδιο βιβλίο, προσδιορίζοντάς το με βάση την επιχειρηματική αξία που παράγεται σε αντίθεση με τις διαδικασίες που οδηγούν στο παραγόμενο αποτέλεσμα. Έργο λοιπόν είναι:

«μια ακολουθία από πεπερασμένες αλληλεξαρτώμενες διεργασίες των οποίων η επιτυχής ολοκλήρωση οδηγεί στην παράδοση της αναμενόμενης προστιθέμενης επιχειρηματικής αξίας, που διαμορφώνεται κατά την υλοποίηση του έργου». (Wysocki, 2014, σ. 4)

2.3. Ορισμός Διαχείρισης Έργου

«Η Διαχείριση έργου αφορά την εφαρμογή γνώσης, δεξιοτήτων, εργαλείων και τεχνικών στις δραστηριότητες του έργου, προκειμένου να πληρούνται οι προδιαγραφές του». (PMI, 2013, σ. 6)

Ο παραπάνω ορισμός του Ινστιτούτου Διαχείρισης Έργου (PMI) είναι γενικός και δεν προσδιορίζει την κύρια αιτία και το αποτέλεσμα ανάληψης ενός έργου. Ο Wysocki δίνει μια διαφορετική προοπτική στο θέμα ορίζοντας ως διαχείριση έργου:

«την οργανωμένη λογική προσέγγιση η οποία αξιοποιεί απαραίτητα τη συμμετοχή του πελάτη, με στόχο την υλοποίηση των απαιτήσεών του και την επίτευξη της αναμενόμενης προστιθέμενης επιχειρηματικής αξίας». (Wysocki, 2014, σ. 29)

Από τον παραπάνω ορισμό δηλώνεται ξεκάθαρα ότι η συμμετοχή του πελάτη σε ένα έργο είναι απαραίτητη, με σκοπό την ικανοποίησή του μέσω της απόδοσης μέγιστης αξίας. Η επιχειρηματική αξία προσδιορίζεται από τις δηλώσεις των απαιτήσεων του πελάτη και ο διαχειριστής έργου είναι υπεύθυνος για την ικανοποίηση και υλοποίηση αυτών των απαιτήσεων. Έτσι, η επιχειρηματική αξία καθορίζεται από τον πελάτη και αποτελεί την αιτία ανάληψης ενός έργου, ενώ υποχρέωση του διαχειριστή είναι η παράδοση ενός έργου, που θα έχει ως αποτέλεσμα την επίτευξη της αναμενόμενης επιχειρηματικής αξίας.

2.4. Ορισμός Απαιτήσεων

Στο πλαίσιο αυτό της πελατοκεντρικής προσέγγισης, μια απαίτηση αφορά στη δήλωση ενός πελάτη για

«μια επιθυμητή τελική κατάσταση, της οποίας η ενσωμάτωση στην τελική λύση, θα οδηγήσει στην παράδοση συγκεκριμένης, μετρήσιμης και αυξητικής επιχειρηματικής αξίας σε έναν οργανισμό». (Wysocki, 2014, σ. 36)

Σύμφωνα με το πρότυπο 610 του Ινστιτούτου Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE)

«ως απαίτηση ορίζεται:

- 1. Μια συνθήκη ή ικανότητα που χρειάζεται ένας χρήστης για να επιλύσει ένα πρόβλημα ή να επιτύχει έναν στόχο.*
- 2. Μια συνθήκη ή ικανότητα που ένα σύστημα ή ένα συστατικό του συστήματος πρέπει να πληροί ή να διαθέτει, έτσι ώστε να ικανοποιείται μια σύμβαση, μια προδιαγραφή, ένα πρότυπο ή άλλο επίσημο έγγραφο.*
- 3. Μια τεκμηριωμένη αναπαράσταση μιας συνθήκης ή ικανότητας, όπως περιεγράφηκε στη περίπτωση (1) ή (2)».*

(IEE Standard Glossary of Software Engineering Terminology, 1990, σ. 62)

Σε ένα έργο πληροφορικής οι απαιτήσεις προσδιορίζουν τις δυνατότητες ενός συστήματος, καθορίζοντας το τι πρέπει να κάνει και πώς αυτό πρέπει να λειτουργεί. Οι απαιτήσεις αρχικά, είναι γενικές και περιγράφονται από τη σκοπιά των χρηστών. Στη συνέχεια όμως γίνονται πιο συγκεκριμένες και περισσότερο τεχνικές. Όσο πιο σαφείς είναι οι απαιτήσεις τόσο λιγότερες αμφιβολίες και δυσκολίες παρουσιάζονται ως προς την υλοποίησή τους.

Οι απαιτήσεις κυρίως χωρίζονται σε δύο κατηγορίες: α) τις λειτουργικές απαιτήσεις, οι οποίες δηλώνουν τι πρέπει να κάνει το σύστημα και β) τις μη λειτουργικές απαιτήσεις, οι οποίες δηλώνουν ποιοι περιορισμοί υπάρχουν στο σύστημα και την ανάπτυξή του.

2.5. Χαρακτηριστικά Απαιτήσεων

Οι απαιτήσεις που καταγράφονται θα πρέπει να πληρούν κάποιες προϋποθέσεις. Με βάση λοιπόν την προσέγγιση SMART οι απαιτήσεις θα πρέπει να είναι:

- Συγκεκριμένες (specific): Να είναι σαφείς, ακριβείς και τεκμηριωμένες.
- Μετρήσιμες (measurable): Να μπορούν να ελεγχθούν σύμφωνα με το αναμενόμενο αποτέλεσμα τους.
- Εφικτές (achievable): Να μπορούν να υλοποιηθούν.
- Σχετικές (relevant): Να προσδίδουν επιχειρηματική αξία, η οποία να συνδέεται με τους αρχικούς στόχους.
- Έγκαιρες (timely): Να μπορούν να πραγματοποιηθούν στον καθορισμένο χρόνο.

2.6. Διαδικασίες διαχείρισης έργου

Σύμφωνα με τον «Οδηγό» του Ινστιτούτου Διαχείρισης Έργου (PMI, 2013) η επιτυχής διαχείριση ενός έργου προϋποθέτει την αποτελεσματική διαχείριση διαδικασιών.

«Μία διαδικασία είναι ένα σύνολο αλληλένδετων ενεργειών και δραστηριοτήτων που εκτελούνται με σκοπό τη δημιουργία ενός προκαθορισμένου προϊόντος, υπηρεσίας ή αποτελέσματος» (PMI, 2013, σ. 47).

Οι διαδικασίες χωρίζονται σε πέντε ομάδες:

- Αρχικοποίησης
- Σχεδιασμού
- Εκτέλεσης
- Παρακολούθησης και Ελέγχου
- Ολοκλήρωσης

Παρόλο που οι διαδικασίες διαχείρισης έργου χωρίζονται σε ομάδες, θα πρέπει να τονιστεί ότι στην πράξη αλληλοεπιδρούν μεταξύ τους και πολλές φορές επικαλύπτει η μία την άλλη. Παράλληλα υπογραμμίζεται ότι οι ανωτέρω ομάδες διαδικασιών δεν ταυτίζονται με τις φάσεις του κύκλου ζωής ενός έργου. Οι διαδικασίες είναι ανεξάρτητες από το μοντέλο διαχείρισης έργου που εφαρμόζεται και εκτελούνται στο σύνολό τους ή μέρος τους κατά τη διάρκεια ενός έργου, μία φορά η κατ' επανάληψη, σύμφωνα με την οργάνωση, τις ανάγκες και τα χαρακτηριστικά του έργου.

2.6.1. Διαδικασίες Αρχικοποίησης

Αναλυτικότερα οι διαδικασίες αρχικοποίησης περιλαμβάνουν τα ακόλουθα:

- Ορισμός ομάδας διαχείρισης έργου και προσδιορισμός ρόλων
- Εντοπισμός και τεκμηρίωση των αναγκών του πελάτη σε αρχικό επίπεδο
- Συγγραφή και κατάθεση σύντομης περιγραφής του έργου
- Λήψη έγκρισης για τον σχεδιασμό του έργου.

2.6.2. Διαδικασίες Σχεδιασμού

Οι διαδικασίες σχεδιασμού περιλαμβάνουν:

- Ανάπτυξη πλάνου διαχείρισης έργου
- Ανάλυση στόχων του έργου
- Συλλογή και καταγραφή απαιτήσεων
- Ανάπτυξη πλάνου εκτέλεσης
- Προσδιορισμός κινδύνων
- Εκτίμηση απαιτούμενων πόρων, κόστους και χρονοδιαγράμματος εκτέλεσης

2.6.3. Διαδικασίες Εκτέλεσης

Οι διαδικασίες εκτέλεσης αφορούν στην:

- Διαχείριση και εκτέλεση του πλάνου υλοποίησης
- Διαχείριση των ομάδων εργασίας και της επικοινωνίας μεταξύ των μελών τους
- Διαχείριση αλλαγών αρχικών στόχων
- Επικοινωνία με τον πελάτη.

2.6.4. Διαδικασίες Παρακολούθησης & Ελέγχου

Οι διαδικασίες παρακολούθησης και ελέγχου περιλαμβάνουν:

- Παρακολούθηση προόδου, εκτέλεσης διαδικασιών και κινδύνων
- Σύνταξη αναφορών προόδου και αποκλίσεων
- Εντοπισμό και επίλυση προβλημάτων

2.6.5. Διαδικασίες Ολοκλήρωσης

Τέλος, οι διαδικασίες ολοκλήρωσης αφορούν στην:

- Σύνταξη τελικής αναφορά έργου
- Διεξαγωγή ελέγχου των τελικών παραδοτέων
- Αρχαιοθέτηση υλικού
- Παράδοση έργου

2.7. Ορισμός Κύκλου Ζωής Διαχείρισης Έργου

«Ο κύκλος ζωής διαχείρισης έργου αφορά τις διαδοχικές φάσεις ή στάδια ενός έργου». (Wysocki, 2014, σ. 40)

Κατά τη διάρκεια αυτών των φάσεων εκτελούνται διαδικασίες οι οποίες ανήκουν στις προαναφερόμενες ομάδες. Ο τρόπος με τον οποίο εκτελούνται οι διαδικασίες αυτές καθορίζεται από τα μοντέλα διαχείρισης του κύκλου ζωής ενός έργου, τα οποία εξετάζονται στο επόμενο κεφάλαιο.

2.8. Συμπεράσματα

Το κεφάλαιο αυτό επικεντρώθηκε στον προσδιορισμό των βασικών εννοιών διαχείρισης έργων πληροφορικής πάνω στις οποίες θα βασιστεί η περαιτέρω ανάλυση του πεδίου. Ορίστηκαν οι έννοιες του έργου, της διαχείρισης έργου, των απαιτήσεων, των διαδικασιών και του κύκλου ζωής ενός έργου. Αναλύθηκε η σημασία της συμμετοχής του πελάτη στη διαχείριση ενός έργου με απώτερο σκοπό τον προσδιορισμό των απαιτήσεων του και την επίτευξη της προσδοκώμενης επιχειρηματικής αξίας. Τέλος ομαδοποιήθηκαν οι διαδικασίες που δύναται να εκτελεστούν κατά τη διάρκεια ενός έργου.

ΚΕΦΑΛΑΙΟ 3: ΜΟΝΤΕΛΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

3.1. Εισαγωγή

Το παρόν κεφάλαιο εστιάζει στα διαφορετικά μοντέλα διαχείρισης έργου πληροφορικής και στον τρόπο με τον οποίο χειρίζονται το σύνολο των διαδικασιών κατά τη διάρκεια ενός έργου.

3.2. Προσεγγίσεις Διαχείρισης Έργου

Τα μοντέλα διαχείρισης έργου κατηγοριοποιούνται σύμφωνα με την προσέγγιση που ακολουθείται. Υπάρχουν τρεις κύριες προσεγγίσεις διαχείρισης έργου:

1. Η Παραδοσιακή
2. Η Ευέλικτη και
3. Η Extreme

3.2.1. Παραδοσιακή Διαχείριση Έργου

Σύμφωνα με αυτή την προσέγγιση, της οποίας αναφορές εντοπίζονται από τη δεκαετία του 1950, η διαχείριση έργου καθορίζεται από συγκεκριμένες δομές εργασιών που βασίζονται σε πειθαρχημένες και σκόπιμες μεθόδους. Οι φάσεις του κύκλου ζωής ενός έργου είναι εύκολα διακριτές με τις δραστηριότητες να ολοκληρώνονται η μία μετά την άλλη, σε μια τακτική ακολουθία, απαιτώντας την εκ των προτέρων σχεδίαση ενός σημαντικού μέρους του έργου.

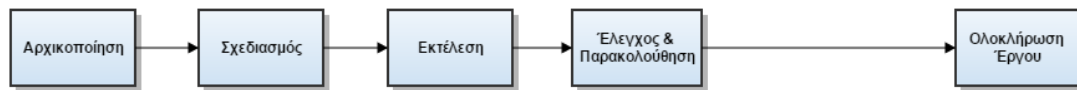
Τα μοντέλα διαχείρισης του κύκλου ζωής ενός έργου που ανήκουν σε αυτή τη προσέγγιση είναι (Wysocki, 2014, σ. 45-47):

- Το γραμμικό (linear) και
- Το Αυξητικό (Incremental)

3.2.1.1. Γραμμικό Μοντέλο Κύκλου Ζωής Διαχείρισης Έργου

Το γραμμικό μοντέλο, το οποίο ονομάζεται και σειριακό ή μοντέλο καταρράκτη (waterfall) χωρίζεται σε πέντε φάσεις που αποτυπώνονται στο παρακάτω διάγραμμα.

Διάγραμμα 1: Γραμμικό Μοντέλο Διαχείρισης Έργων



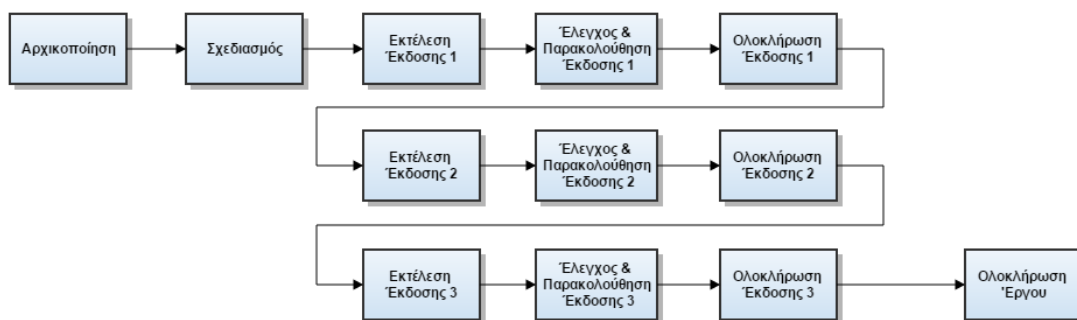
Οι φάσεις αυτές εκτελούνται μία μόνο φορά, στη σειρά και δεν υπάρχει ανάδραση. Για τον λόγο αυτό θεωρείται ένα από τα πιο ανελαστικά μοντέλα.

Το γραμμικό μοντέλο χρησιμοποιείται κυρίως σε έργα που έχουν σαφή και καθορισμένο στόχο, είναι επαναλαμβανόμενα, δε χρειάζονται αλλαγές και στηρίζονται σε συγκεκριμένα πρότυπα.

3.2.1.2. Αυξητικό Μοντέλο Κύκλου Ζωής Διαχείρισης Έργου

Το αυξητικό μοντέλο είναι μια παραλλαγή του γραμμικού, ωστόσο χρησιμοποιείται σε έργα που κατά τη διάρκειά τους χρειάζεται να παραδίδονται λειτουργικά κομμάτια. Ο κύκλος ζωής του μοντέλου έχει τις ίδιες φάσεις με το γραμμικό, με τη μόνη διαφορά ότι οι φάσεις της Εκτέλεσης, Ελέγχου - Παρακολούθησης και Ολοκλήρωσης μπορεί να επαναληφθούν διαδοχικά για κάθε παραδοτέο τμήμα. Τα στάδια του μοντέλου αποτυπώνονται στο παρακάτω διάγραμμα.

Διάγραμμα 2: Αυξητικό Μοντέλο Διαχείρισης Έργων



3.2.2. Ευέλικτη Διαχείριση Έργου

Η ευέλικτη προσέγγιση διαχείρισης έργου θεωρείται ως η μετεξέλιξη της παραδοσιακής προσέγγισης. Παρόλο που στον τομέα διαχείρισης έργων πληροφορικής έχει εφαρμοστεί τα τελευταία χρόνια, έχει τις ρίζες της στα συστήματα παραγωγής της γιαπωνέζικης αυτοκινητοβιομηχανίας. Στη βιβλιογραφία υποστηρίζεται ότι μειώνει τους κινδύνους σε έργα πληροφορικής και επιταχύνει την ανάπτυξή τους.

Τα ευέλικτα μοντέλα χρησιμοποιούνται για τη σταδιακή ανάπτυξη συστημάτων λογισμικού ενσωματώνοντας επαναληπτικούς κύκλους σχεδιασμού και προσαρμόζοντας αδιάλειπτα τα δεδομένα των χρηστών και των ομάδων ενδιαφέροντος μέσα από τη διαρκή ανατροφοδότηση (Hass, 2007).

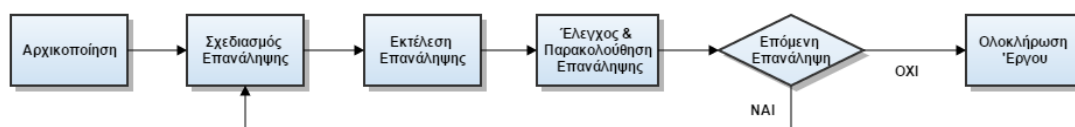
Τα μοντέλα διαχείρισης του κύκλου ζωής ενός έργου που ανήκουν στην ευέλικτη προσέγγιση είναι (Wysocki, 2014, σ. 47-52):

- Επαναληπτικό (Iterative)
- Προσαρμοστικό (Adaptive)

3.2.2.1. Επαναληπτικό Μοντέλο Κύκλου Ζωής Διαχείρισης Έργου

Το επαναληπτικό μοντέλο αποτελείται από φάσεις οι οποίες επαναλαμβάνονται διαδοχικά, με στόχο την υλοποίηση μικρών τμημάτων του έργου μέσα από συνεχή ανατροφοδότηση από τα ενδιαφερόμενα μέλη. Τα τμήματα του έργου χωρίζονται βάσει των απαιτήσεων του πελάτη. Η ουσιαστική διαφορά του επαναληπτικού μοντέλου από το αυξητικό είναι ότι σε κάθε επανάληψη αναμένονται αλλαγές σε λειτουργίες και χαρακτηριστικά, έτσι ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα. Ο κύκλος ζωής του μοντέλου περιγράφεται στο παρακάτω διάγραμμα.

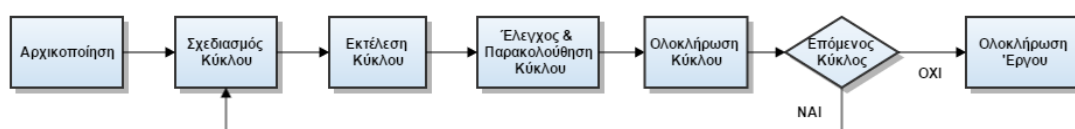
Διάγραμμα 3: Επαναληπτικό Μοντέλο Διαχείρισης Έργων



3.2.2.2. Προσαρμοστικό Μοντέλο Κύκλου Ζωής Διαχείρισης Έργου

Το προσαρμοστικό μοντέλο είναι παραλλαγή του επαναληπτικού μοντέλου με τη μόνη διαφορά ότι σε κάθε επανάληψη διερευνάται και η πιθανότητα άλλων λειτουργιών οι οποίες θα μπορούσαν να είναι συμπληρωματικές μιας λύσης ή θα μπορούσαν να την αναδείξουν με διαφορετικό τρόπο, ίσως πιο πρωτοποριακό, έχοντας πάντα ως αποτέλεσμα την προσδοκώμενη επιχειρηματική αξία. Τα στάδια του μοντέλου αποτυπώνονται στο παρακάτω διάγραμμα.

Διάγραμμα 4: Προσαρμοστικό Μοντέλο Διαχείρισης Έργων



Το προσαρμοστικό μοντέλο χρησιμοποιείται κυρίως για έργα με υψηλό βαθμό αβεβαιότητας και πολυπλοκότητας.

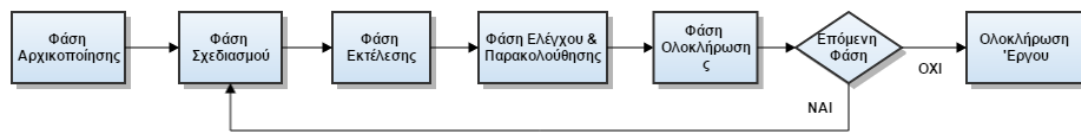
Στα πλαίσια αυτής της εργασίας χρησιμοποιείται μία από τις ευέλικτες μεθοδολογίες η οποία βασίζεται στο προσαρμοστικό μοντέλο και ονομάζεται Scrum. Η ανάλυσή της πραγματοποιείται στο επόμενο κεφάλαιο.

3.2.3. Extreme Διαχείριση Έργου

Η extreme προσέγγιση σε πολλές βιβλιογραφικές πηγές κατατάσσεται στην κατηγορία της ευέλικτης προσέγγισης. Ωστόσο η extreme προσέγγιση συναντάται κυρίως σε έργα που αφορούν στον χώρο της Έρευνας και Τεχνολογίας (R&D), όπου το αποτέλεσμα είναι αβέβαιο ή άγνωστο (DeCarlo, 2004).

Το extreme μοντέλο διαχείρισης του κύκλου ζωής ενός έργου ακολουθεί τις ίδιες φάσεις με τα μοντέλα της ευέλικτης προσέγγισης με τη μόνη διαφορά ότι οποιαδήποτε επανάληψη ξεκινάει από το πρώτο στάδιο, δηλαδή τη φάση της αρχικοποίησης. Το παρακάτω διάγραμμα απεικονίζει τον κύκλο ζωής του extreme μοντέλου.

Διάγραμμα 5: Extreme Μοντέλο Διαχείρισης Έργων



3.3. Συμπεράσματα

Στο κεφάλαιο αυτό ερευνήθηκαν οι διαφορετικές προσεγγίσεις διαχείρισης έργων πληροφορικής και αναλύθηκαν τα μοντέλα της κάθε προσέγγισης. Είναι σημαντικό να αποσαφηνιστεί ότι δεν υπάρχει ιδανική προσέγγιση ή μοντέλο που να ανταποκρίνεται επιτυχώς και να παρέχει λύσεις στις ανάγκες οποιουδήποτε έργου πληροφορικής. Υπάρχουν παράμετροι όπως το συνολικό κόστος, η διάρκεια, η σταθερότητα της αγοράς καθώς και η τεχνολογία, που πρέπει να λαμβάνονται υπόψη για την επιτυχή διεκπεραίωση ενός έργου.

ΚΕΦΑΛΑΙΟ 4: ΕΥΕΛΙΚΤΕΣ ΜΕΘΟΔΟΛΟΓΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

4.1. Εισαγωγή

Η παρούσα μελέτη, για την ανάπτυξη της διαδικτυακής εφαρμογής εκπαιδευτικών συγγραμμάτων, επικεντρώνεται στην ευέλικτη προσέγγιση διαχείρισης έργων και συγκεκριμένα στο προσαρμοστικό μοντέλο.

Παρακάτω αναφέρονται οι βασικές αξίες και αρχές της ευέλικτης διαχείρισης έργων. Επιπλέον αναλύεται η μεθοδολογία Scrum στην οποία βασίζεται η διαχείριση του έργου ανάπτυξης της εφαρμογής.

4.2. Βασικές Αξίες Ευέλικτης Διαχείρισης Έργων

Η Ευέλικτη προσέγγιση ανάπτυξης λογισμικού στηρίζεται στις κάτωθι βασικές αξίες, οι οποίες περικλείουν τη φιλοσοφία της (Beck, Kent et al, 2001):

- Τα άτομα και η μεταξύ τους αλληλεπίδραση είναι πάνω από τις διαδικασίες και τα εργαλεία.
- Ένα λογισμικό που λειτουργεί έχει μεγαλύτερη αξία από μια εκτενή τεκμηρίωση.
- Η συνεργασία με τους πελάτες είναι πιο σημαντική από τις συμβατικές διαπραγματεύσεις.
- Η ανταπόκριση στις αλλαγές είναι πιο σημαντική από την τήρηση ενός προδιαγεγραμμένου σχεδίου.

Αυτές οι βασικές αξίες δηλώνουν ξεκάθαρα ότι η ευέλικτη προσέγγιση εστιάζει στο ανθρώπινο δυναμικό, την επικοινωνία, το προϊόν καθαυτό και την ελαστικότητα, σε αντίθεση με τα παραδοσιακά μοντέλα που υιοθετούν άκαμπτα πλάνα, αποφεύγουν την αλλαγή και επιμένουν στην εξαντλητική τεκμηρίωση.

4.3. Βασικές Αρχές της Ευελιξίας

Σύμφωνα με την Agile Alliance υπάρχουν δώδεκα αρχές οι οποίες διέπουν την ευέλικτη προσέγγιση και έχουν ως ακολούθως (Beck, Kent et al, 2001):

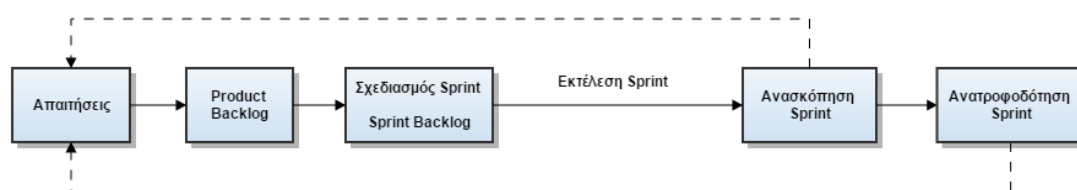
1. Βασική προτεραιότητα είναι η ικανοποίηση του πελάτη μέσω της έγκαιρης και συνεχούς παράδοσης χρήσιμου λογισμικού.
2. Οι αλλαγές στις απαιτήσεις είναι ευπρόσδεκτες, ακόμα και σε προχωρημένα στάδια της ανάπτυξης. Οι ευέλικτες διαδικασίες αξιοποιούν την αλλαγή προς ανταγωνιστικό όφελος του πελάτη.
3. Η παράδοση λειτουργικού λογισμικού συχνά γίνεται σε διαστήματα μερικών εβδομάδων ή μηνών, με προτίμηση τη συντομότερη χρονική κλίμακα.
4. Οι προγραμματιστές και οι ειδικοί της αγοράς πρέπει να συνεργάζονται καθημερινά καθ' όλη τη διάρκεια του έργου.
5. Τα έργα θεμελιώνονται γύρω από άτομα με πάθος και ενδιαφέρον. Διαμορφώνεται το κατάλληλο περιβάλλον και παρέχεται η αναγκαία υποστήριξη και εμπιστοσύνη στην ικανότητά τους να φέρουν εις πέρας την αποστολή τους.
6. Η πιο αποδοτική και αποτελεσματική μέθοδος για τη μετάδοση πληροφορίας προς και εντός της ομάδας είναι η διαπροσωπική επαφή και η συζήτηση.
7. Το λογισμικό που λειτουργεί είναι το βασικό μέτρο αποτίμησης της προόδου.
8. Οι ευέλικτες διαδικασίες προάγουν την απρόσκοπτη ανάπτυξη. Οι χρηματοδότες, η ομάδα ανάπτυξης λογισμικού και οι χρήστες θα πρέπει να είναι σε θέση να διατηρήσουν ένα σταθερό ρυθμό επ' αόριστον.
9. Η διαρκής έμφαση στην τεχνική αρτιότητα και στον καλό σχεδιασμό ενισχύει την ευελιξία.
10. Η απλότητα είναι ουσιώδης.
11. Οι καλύτερες αρχιτεκτονικές, απαιτήσεις και σχέδια προκύπτουν από ομάδες που αυτό-οργανώνονται.
12. Σε τακτά χρονικά διαστήματα, η ομάδα αξιολογεί τους τρόπους με τους οποίους θα γίνει πιο αποτελεσματική, ρυθμίζοντας και προσαρμόζοντας τη συμπεριφορά της αναλόγως.

4.4. Μεθοδολογία Scrum

Η Scrum θεωρείται από τις πιο δημοφιλείς ευέλικτες μεθόδους ανάπτυξης λογισμικού (State of Agile Survey, 2011). Στη μέθοδο αυτή δίνεται προτεραιότητα στον τρόπο οργάνωσης της ομάδας με σκοπό την επίτευξη όσο το δυνατόν καλύτερης συνεργασίας και συντονισμού των εργασιών τους. Πρωταρχικός στόχος της μεθόδου είναι να μειώσει το χρονικό διάστημα που απαιτείται έτσι ώστε να ολοκληρωθεί ο κύκλος ανάπτυξης και παράδοσης τμημάτων κώδικα που έχουν συμφωνηθεί. Χρησιμοποιείται για την οργάνωση και λειτουργία των ομάδων αλλά και για τον σχεδιασμό και διαχείριση ευέλικτων έργων. Ο όρος Scrum προέρχεται από το ράγκμπι καθώς παραλληλίζεται η μεθοδολογία με τον τρόπο που η ομάδα, σε ένα παιχνίδι ράγκμπι, προσπαθεί να διανύσει την απόσταση περνώντας την μπάλα μπρος-πίσω (Kohn, 2009). Έτσι και στην ανάπτυξη ενός λογισμικού γίνεται μια επαναληπτική διαδικασία κατά την οποία παραδίδονται σταδιακά λειτουργικά μέρη του λογισμικού μέχρι «να διανυθεί όλη η απόσταση», δηλαδή όλα τα διαφορετικά τμήματα κώδικα να ενωθούν για τη δημιουργία του τελικού «προϊόντος». Η μεθοδολογία Scrum υποστηρίζεται από συγκεκριμένους ρόλους, παραδοτέα αντικείμενα και γεγονότα.

Το παρακάτω διάγραμμα απεικονίζει τη μεθοδολογία Scrum.

Διάγραμμα 6: Μεθοδολογία SCRUM



4.4.1 Βασικές Έννοιες

Σύμφωνα με τον επίσημο οδηγό της Scrum (Schwaber & Sutherland, 2013), οι βασικές έννοιες της μεθοδολογίας αναλύονται στις παρακάτω ενότητες.

4.4.1.1. Sprint

Το Sprint ορίζεται ως ένας επαναληπτικός κύκλος, με καθορισμένη διάρκεια, από μια εβδομάδα έως ένα μήνα. Μέσα σε αυτό το χρονικό διάστημα αναπτύσσεται ένα ολοκληρωμένο τμήμα του τελικού προϊόντος. Κατά τη διάρκεια ενός sprint εκτελούνται οι ακόλουθες ενέργειες:

- Συνεχής επιθεώρηση για την αξιολόγηση της προόδου
- Συναντήσεις σε καθημερινή βάση για την ανασκόπηση της προηγούμενης και προγραμματισμός της τρέχουσας ημέρας
- Συνάντηση στο τέλος του sprint για αξιολόγηση του αποτελέσματος, σχεδιασμό τυχόν αναπροσαρμογών και διορθωτικών κινήσεων

4.4.1.2. Ρόλοι & Ευθύνες

Σύμφωνα με τη μεθοδολογία Scrum υπάρχουν τρεις διακριτοί ρόλοι που εμπλέκονται στην ανάπτυξη ενός συστήματος:

- **Διαχειριστής της Scrum (Scrum Master):** Είναι υπεύθυνος για την εφαρμογή των πρακτικών της Scrum και την ορθή ανάπτυξη του συστήματος. Πρέπει να επισημαίνει τυχόν εμπόδια και να φροντίζει για την εξάλειψή τους έτσι ώστε να διατηρήσει όσο το δυνατόν ψηλότερα το επίπεδο παραγωγικότητας της ομάδας.
- **Διαχειριστής του Προϊόντος (Product Owner):** Είναι υπεύθυνος για τον εντοπισμό και την καταγραφή των απαιτήσεων του συστήματος. Σε συνεργασία με την ομάδα ανάπτυξης, επιλέγει ποιες απαιτήσεις θα υλοποιηθούν και με ποια σειρά. Επίσης κατά τη διάρκεια της ανάπτυξης συμβάλλει αποφασιστικά στην εξεύρεση λύσεων σε τυχόν προβληματικές καταστάσεις.
- **Ομάδα Ανάπτυξης Scrum (Scrum Team):** Η ομάδα αναλαμβάνει την ανάπτυξη του συστήματος και δίνει ιδέες για την βελτίωσή του. Επίσης αποφασίζει και εκτελεί τις κατάλληλες εργασίες που απαιτούνται για την υλοποίησή του. Συμβάλλει στην ανακάλυψη των εμποδίων που θα πρέπει να αντιμετωπιστούν έτσι ώστε η διαδικασία ανάπτυξης να συνεχιστεί ομαλά. Είναι αυτοελεγχόμενη, αρκετά αυτόνομη και με υψηλό βαθμό υπευθυνότητας.

Αξίζει να σημειωθεί ότι στα πλαίσια της διπλωματικής εργασίας οι ανωτέρω ρόλοι δεν είναι ευδιάκριτοι καθώς οι δραστηριότητές τους αναλαμβάνονται από ένα μόνο άτομο, τον συγγραφέα.

4.4.1.3. Παραδοτέα Αντικείμενα

Κατά τη διάρκεια του έργου, χρησιμοποιώντας τη μεθοδολογία Scrum, ορίζονται τα εξής παραδοτέα αντικείμενα:

- **Backlog Προϊόντος (Product Backlog):** Αφορά στην καταγραφή των απαιτήσεων, με τη μορφή ιστοριών χρήστη, σε λίστα ανεκτέλεστων εργασιών με σειρά προτεραιότητας,
- **Sprint Backlog:** Αφορά στη λίστα, με σειρά προτεραιότητας των εργασιών που πρόκειται να εκτελεστούν κατά τη διάρκεια του sprint,
- **Λειτουργικό Τμήμα Προϊόντος (Product Increment):** Αφορά σε ένα λειτουργικό τμήμα του τελικού προϊόντος.

4.4.1.4. Γεγονότα

Κατά τη διάρκεια του έργου, χρησιμοποιώντας τη μεθοδολογία Scrum, συμβαίνουν τα εξής γεγονότα:

- **Συνάντηση Σχεδιασμού Sprint (Sprint Planning Meeting):** Διεξάγεται με την έναρξη του Sprint και καθορίζονται οι εργασίες που πρέπει να υλοποιηθούν, βάσει του Product Backlog, κατά τη διάρκεια του Sprint.
- **Καθημερινή Συνάντηση Scrum:** Πραγματοποιείται στην αρχή της ημέρας με σκοπό την ανασκόπηση της προηγούμενης και προγραμματισμό της τρέχουσας ημέρας. Τυχόν εμπόδια και προβληματισμοί γνωστοποιούνται προκειμένου να αντιμετωπιστούν.
- **Συνάντηση Ανασκόπησης Sprint:** Τελείται στο τέλος κάθε Sprint με σκοπό την ανασκόπηση των εργασιών και την επίδειξη στον πελάτη του λειτουργικού τμήματος που αναπτύχθηκε κατά τη διάρκεια του Sprint.
- **Ανατροφοδότηση Sprint:** Πραγματοποιείται στο τέλος κάθε Sprint με σκοπό την αποτίμηση των εργασιών, τον εντοπισμό λαθών και παραλείψεων καθώς

και την κατάθεση απόψεων και προτάσεων για τη βελτίωση της όλης διαδικασίας.

4.4.2. Διαδικασία

Η μεθοδολογία Scrum ακολουθεί τα εξής στάδια:

1. **Όραμα:** Καθορισμός των στόχων σε σχέση με τη στρατηγική της επιχείρησης
2. **Οδικός χάρτης Προϊόντος:** Καταγραφή των χαρακτηριστικών που συνθέτουν το όραμα
3. **Σχεδιασμός Εκδόσεων:** Σχεδιασμός παράδοσης έκδοσης με δεδομένη λειτουργικότητα
4. **Σχεδιασμός Sprint:** Σχεδιασμός στόχων και εργασιών της επανάληψης
5. **Καθημερινή Συνάντηση:** Συντονισμός προτεραιοτήτων ημέρας
6. **Ανασκόπηση Scrum:** Επίδειξη λειτουργιών προϊόντων
7. **Ανατροφοδότηση Scrum:** Αποτίμηση επανάληψης και βελτιστοποίηση αποτελέσματος
8. **Δημοσίευση Έκδοσης (Release):** Έκδοση προϊόντος

4.4.2.1. Στάδιο 1 – Όραμα

Στο πρώτο στάδιο της μεθοδολογία Scrum γίνεται ο καθορισμός των βασικών στόχων του τελικού προϊόντος, δηλαδή αποτυπώνεται η προστιθέμενη επιχειρηματική αξία. Ο καθορισμός των βασικών στόχων είναι αποκλειστική ευθύνη του Product Owner. Στο στάδιο αυτό μπορούν να πραγματοποιηθούν και διαδικασίες αρχικοποίησης, οι οποίες όπως έχει ήδη επισημανθεί είναι ανεξάρτητες από τα στάδια της μεθοδολογίας.

4.4.2.2. Στάδιο 2 – Οδικός Χάρτης Προϊόντος

Σε αυτό το στάδιο εντοπίζονται και καταγράφονται οι αρχικές απαιτήσεις, οργανώνονται σε λογικές ομάδες και κατατάσσονται σε σειρά προτεραιότητας. Κατά τη δημιουργία του οδικού χάρτη δεν απαιτείται λεπτομερής αποτύπωση των απαιτήσεων. Εξάλλου, σε επόμενα στάδια, μέσω της επαναληπτικής διαδικασίας και της ενεργής εμπλοκής του πελάτη, οι απαιτήσεις αναθεωρούνται αρκετές φορές. Στη

συνέχεια οι αρχικές απαιτήσεις ομαδοποιούνται και δημιουργείται μια λίστα με τα επιθυμητά χαρακτηριστικά (features).

4.4.2.3. Στάδιο 3 – Σχεδιασμός Εκδόσεων

Στο στάδιο αυτό τα επιθυμητά χαρακτηριστικά (features) εξειδικεύονται και αποσυντίθενται σε ιστορίες χρηστών, δηλαδή σε περιγραφικά κείμενα που εστιάζουν σε μεμονωμένες αλληλεπιδράσεις του χρήστη με το σύστημα καθώς και στην προστιθέμενη αξία που προσδίδουν. Έτσι δημιουργείται το Product Backlog όπου οι ιστορίες χρήστη καταγράφονται κατά σειρά προτεραιότητας με τη μορφή λίστας.

Οι ιστορίες χρήστη (User Stories) θα πρέπει να πληρούν τα κάτωθι (Wake, 2003):

- Να είναι ανεξάρτητες (Independent). Είναι πιο εύκολο να προγραμματιστούν χρονικά και να υλοποιηθούν ιστορίες ανεξάρτητες μεταξύ τους. Ωστόσο αυτό δεν είναι πάντα εφικτό.
- Να είναι διαπραγματεύσιμες (Negotiable). Μια καλή ιστορία θα πρέπει να περνά στην ουσία του αιτήματος και να μη διευρύνεται με πολλές λεπτομέρειες. Οι λεπτομέρειες θα προστεθούν αργότερα κατόπιν συνεργασίας με τον χρήστη.
- Να έχουν αξία για τον χρήστη (Valuable). Κάθε ιστορία που υλοποιείται θα πρέπει να θεωρείται σημαντική από την πλευρά του χρήστη έτσι ώστε να προσδίδεται αξία και στο τελικό αποτέλεσμα.
- Να μπορούν να εκτιμηθούν αξιόπιστα (Estimable). Οι ιστορίες θα πρέπει να είναι κατανοητές έτσι ώστε να μπορεί να εκτιμηθεί το μέγεθος αλλά και ο χρόνος που χρειάζεται για την υλοποίησή τους.
- Να είναι μικρές (Small). Οι ιστορίες δε θα πρέπει να είναι εκτενείς και να περιέχουν πολλές λεπτομέρειες. Αν μια ιστορία δεν είναι υλοποιήσιμη από λίγα άτομα σε μικρό χρονικό διάστημα σημαίνει ότι υπάρχει δυνατότητα να χωριστεί σε μικρότερα τμήματα.
- Να είναι ελέγξιμες (Testable). Αν ο χρήστης γνωρίζει τον τρόπο με τον οποίο μπορεί να ελέγξει την υλοποίηση κάποιας απαίτησης, σημαίνει ότι η ιστορία, που περιγράφει την απαίτησή αυτή, είναι ξεκάθαρη και υψηλής σπουδαιότητας.

Μετά τη δημιουργία του Product Backlog σχεδιάζονται οι εκδόσεις, όπου καταγράφεται ο στόχος της έκδοσης, η ημερομηνία δημοσίευσης καθώς και η ιστορία χρήστη που θα υλοποιηθεί στα πλαίσια της συγκεκριμένης έκδοσης.

4.4.2.4. Στάδιο 4 – Σχεδιασμός Sprint

Στο στάδιο αυτό σχεδιάζονται οι εργασίες που πρέπει να υλοποιηθούν σε κάθε Sprint. Έτσι δημιουργείται το Sprint Backlog που περιλαμβάνει όλες τις εργασίες, οι οποίες πρόκειται να αναπτυχθούν σε κάθε Sprint. Επίσης υπολογίζεται ο χρόνος που απαιτείται για τη διεκπεραίωση των εργασιών που συγκαταλέγονται σε κάθε Sprint.

4.4.2.5. Στάδιο 5 – Καθημερινή Συνάντηση

Καθημερινά γίνεται συνάντηση των μελών της ομάδας προκειμένου να αποτιμηθούν οι εργασίες της προηγούμενης ημέρας, να προγραμματιστούν οι εργασίες της τρέχουσας ημέρας και να γνωστοποιηθούν τυχόν θέματα που εμποδίζουν την ανάπτυξη.

Η συχνή συνάντηση και επικοινωνία των μελών της ομάδας αποτελεί κύριο χαρακτηριστικό της ευέλικτης φιλοσοφίας.

4.4.2.6. Στάδιο 6 – Ανασκόπηση Scrum

Στο στάδιο αυτό, που διενεργείται στο τέλος του κάθε Sprint, παρουσιάζεται στον πελάτη η λειτουργικότητα που αναπτύχθηκε βάσει των ιστοριών χρήστη. Με τον τρόπο αυτό ενισχύεται η συνεργατικότητα και ο πελάτης συμμετέχει ενεργά στη διαδικασία ανάπτυξης παρέχοντας ανάδραση.

Υπάρχουν τέσσερα κριτήρια τα οποία θα πρέπει να πληρούνται προτού ξεκινήσει το στάδιο της ανασκόπησης. Ένα λειτουργικό τμήμα θα πρέπει να είναι (Swhaber & Sutherland, 2013):

- Αναπτυγμένο (Developed)
- Δοκιμασμένο (Tested)
- Ενσωματωμένο με άλλα λειτουργικά τμήματα (Integrated)
- Τεκμηριωμένο (Documented)

Να σημειωθεί ότι στο στάδιο αυτό μπορεί να προκύψουν αλλαγές ή νέες απαιτήσεις από τον πελάτη, οι οποίες θα πρέπει να μετατραπούν σε ιστορίες χρήστη και να ταξινομηθούν με σειρά προτεραιότητας στο Product Backlog, έτσι ώστε να αναπτυχθούν σε επόμενα Sprints.

4.4.2.7. Στάδιο 7 – Ανατροφοδότηση Scrum

Στο στάδιο αυτό αποτιμάται η όλη διαδικασία και προτείνονται τρόποι αντιμετώπισης τυχόν εμποδίων που προέκυψαν, καθώς και διαφορετικές πρακτικές που θα μπορούσαν να βοηθήσουν στην βελτιστοποίηση του αποτελέσματος. Έτσι δημιουργείται ουσιαστικά μια διαδικασία αυτό-διόρθωσης με πρωταρχικό στόχο την επίτευξη της προσδοκώμενης επιχειρηματικής αξίας.

4.4.2.8. Στάδιο 8 – Δημοσίευση Έκδοσης

Στο στάδιο αυτό όλα τα λειτουργικά τμήματα που αναπτύχθηκαν σε κάθε επανάληψη Sprint ενώνονται και οδηγούν στη δημοσίευση συγκεκριμένης έκδοσης λογισμικού, η οποία μπορεί να διατεθεί στους χρήστες ως τελικό προϊόν.

Για τη δημοσίευση της έκδοσης εκτελείται ένα ακόμα Sprint, το οποίο ονομάζεται Release Sprint και περιλαμβάνει τις κάτωθι ενέργειες (Layton, 2012, σ. 180):

- Συγγραφή υλικού τεκμηρίωσης των λειτουργιών και χαρακτηριστικών της έκδοσης
- Εκτέλεση δοκιμών ασφαλείας
- Ενσωμάτωση έκδοσης με άλλα συστήματα, εφόσον απαιτείται
- Εγκατάσταση της έκδοσης σε περιβάλλον παραγωγής.

4.5. Συμπεράσματα

Στο κεφάλαιο αυτό αναλύθηκαν οι βασικές αρχές και αξίες της ευέλικτης προσέγγισης διαχείρισης έργων πληροφορικής. Πραγματοποιήθηκε εκτενής ανάλυση της ευέλικτης μεθοδολογίας Scrum, εστιάζοντας στις βασικές της έννοιες καθώς και στα στάδια υλοποίησής της για τη διαχείριση ενός έργου. Σε επόμενα κεφάλαια ακολουθείται η μεθοδολογία αυτή για την ανάπτυξη της εφαρμογής που πραγματεύεται η εργασία.

ΚΕΦΑΛΑΙΟ 5: ΠΑΡΟΥΣΙΑΣΗ ΤΕΧΝΟΛΟΓΙΩΝ

5.1. Εισαγωγή

Στόχος αυτού του κεφαλαίου είναι να παραθέσει και να αναλύσει τις πιο σημαντικές τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Αρχικά περιγράφεται η αντικειμενοστρεφής γλώσσα προγραμματισμού Ruby και το πλαίσιο ανάπτυξης εφαρμογών Ruby on Rails. Στη συνέχεια αναλύονται οι βασικές έννοιες των βάσεων δεδομένων, τα διάφορα μοντέλα τους και εξετάζεται διεξοδικά ένα συγκεκριμένο σύστημα διαχείρισης βάσεων δεδομένων, η MongoDB. Τέλος διερευνάται ένα σύστημα αναζήτησης κειμενικών δεδομένων, το Elasticsearch.

5.2. Ruby

Η Ruby είναι μια δυναμική, ευέλικτη, αντικειμενοστρεφής γλώσσα προγραμματισμού γενικής χρήσης. Δημιουργήθηκε από τον Ιάπωνα Yukihiro Matsumoto τη δεκαετία του 90' και η πρώτη εμφάνισή της έγινε τον Φεβρουάριο του 1993. Ωστόσο το ενδιαφέρον για την Ruby εξαπλώθηκε το 2005 με τη δημιουργία ενός πλαισίου ανάπτυξης λογισμικού ιστού, γραμμένο σε Ruby, του Ruby on Rails. Η γλώσσα έχει επιρροές από άλλες γλώσσες προγραμματισμού όπως την Eiffel, Lisp, Perl, Python και Smalltalk.

Η Ruby έχει τα εξής γενικά χαρακτηριστικά:

- Είναι γλώσσα **υψηλού επιπέδου**, δηλαδή η ανάγνωση και η γραφή της είναι αρκετά εύκολη, σε σχέση με άλλες γλώσσες προγραμματισμού, καθώς ομοιάζει με τη φυσική γλώσσα της Αγγλικής.
- Είναι **ερμηνευμένη**, που σημαίνει ότι δεν χρειάζεται μεταγλωττιστή για τη γραφή και την εκτέλεση ενός προγράμματος Ruby.
- Είναι πλήρως **αντικειμενοστρεφής**, δηλαδή επιτρέπει στους χρήστες να χειριστούν τα δεδομένα και τις διαδικασίες που ενεργούν σε αυτά, μέσω μιας δομής δεδομένων¹ που ονομάζεται αντικείμενο².

¹ **Δομή δεδομένων** αναφέρεται στους διαφορετικούς δυνατούς τρόπους οργάνωσης και αποθήκευσης δεδομένων ([Βικιπαίδεια](#))

² **Αντικείμενο** καλείται το στιγμιότυπο ενός σύνθετου τύπου δεδομένων που ονομάζεται κλάση ([Βικιπαίδεια](#))

- Είναι **εύχρηστη** καθώς «εστιάζει στους ανθρώπους, στο πως προγραμματίζουν ή χειρίζονται τις εφαρμογές των υπολογιστών» (Venpers, 2003), δηλαδή δίνει έμφαση στις ανθρώπινες ανάγκες και όχι σε αυτές του υπολογιστή.
- Είναι **ανοιχτού κώδικα**, δηλαδή οι χρήστες μπορούν να εξετάσουν και να τροποποιήσουν τον πηγαίο της κώδικα ανάλογα με τις ανάγκες τους.

5.3. Ruby on Rails

Το Ruby on Rails (RoR) είναι ένα πλαίσιο³ (framework) ανοιχτού κώδικα που χρησιμοποιείται για την ανάπτυξη λογισμικού ιστού (web applications) και βασίζεται στην αντικειμενοστρεφή γλώσσα προγραμματισμού Ruby. Περιλαμβάνει εργαλεία που διευκολύνουν κοινές προγραμματιστικές εργασίες και βοηθούν στην ταχεία ανάπτυξη λογισμικού.

Το RoR αναπτύχθηκε από τον Δανό προγραμματιστή David Heinemeier Hansson. Δημοσιεύτηκε για πρώτη φορά στον παγκόσμιο ιστό τον Ιούλιο του 2004. Η πρώτη έκδοση κυκλοφόρησε στις 13 Δεκεμβρίου του 2005.

5.3.1. Αρχές Σχεδίασης του RoR

Το RoR βασίζει την φιλοσοφία του σε δύο αρχές:

1. Στην αρχή του «Μη Επαναλαμβάνεστε» (Don't Repeat Yourself – DRY)
2. Στην αρχή της «Προτυποποίησης αντί της Διαμόρφωσης» (Convention over Configuration – CoC)

Η πρώτη αρχή (DRY) σημαίνει ότι η ίδια πληροφορία δεν θα πρέπει να επαναλαμβάνεται σε πολλά σημεία, δηλαδή θα πρέπει να αποφεύγεται η επανάληψη του ίδιου κώδικα σε διάφορα μέρη της εφαρμογής. Αυτό είναι ιδιαίτερα χρήσιμο στην τεχνολογία λογισμικού και ειδικότερα στις πολυεπίπεδες αρχιτεκτονικές. Έτσι μειώνονται τα σφάλματα, ο κώδικας παραμένει ευανάγνωστος και η συντήρησή του

³ Πλαίσιο (framework) είναι μια υποστηρικτική δομή που χρησιμοποιείται ως βάση για κάτι το οποίο κατασκευάζεται. Μπορεί να περιέχει υποστηρικτικά προγράμματα, βιβλιοθήκες, άλλο λογισμικό, και είναι στοχευμένο σε συγκεκριμένη τεχνολογία ακολουθώντας μια καθορισμένη ροή εργασιών.

γίνεται πιο εύκολη. Επιπρόσθετα οποιαδήποτε αλλαγή μπορεί να γίνει με τρόπο προβλέψιμο, ομοιόμορφο και συγχρονισμένα.

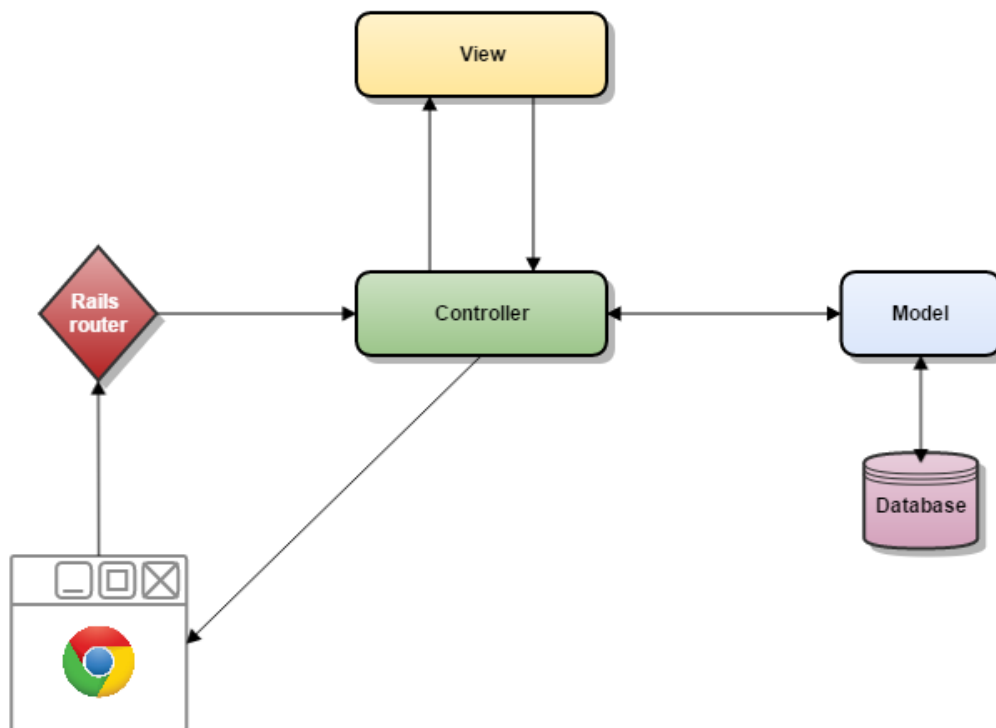
Η αρχή «Προτυποποίηση αντί Διαμόρφωση» (CoC) επιδιώκει να περιορίσει τον αριθμό των αποφάσεων που πρέπει να πάρουν οι προγραμματιστές, αφού καλούνται να ορίσουν μόνο εκείνες τις παραμέτρους που δεν είναι δυνατό να προτυποποιηθούν. Σε όλα τα πλαίσια ανάπτυξης λογισμικού, όπως είναι και το RoR, υπάρχουν πρότυπα σχεδίασης, δηλαδή «αποδεδειγμένες καλές λύσεις που έχουν εφαρμοστεί με επιτυχία στην επίλυση επαναλαμβανόμενων προβλημάτων σχεδίασης» (Βικιπαίδεια, 2014). Έτσι μειώνεται αισθητά ο κώδικας που πρέπει να γραφεί και αποφεύγονται οι πολλαπλές ρυθμίσεις, χωρίς όμως να χάνεται η ευελιξία.

5.3.2. Πρότυπο Σχεδίασης: Μοντέλο-Όψη-Ελεγκτής (Model-View-Controller)

Το RoR χρησιμοποιεί μια πολυεπίπεδη αρχιτεκτονική λογισμικού για τη δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Διαιρεί μια εφαρμογή σε τρία διασυνδεδεμένα τμήματα: το μοντέλο (model), την όψη (view) και τον ελεγκτή (controller). Με τον τρόπο αυτό διαχωρίζονται οι εσωτερικές διεργασίες της εφαρμογής που αφορούν την ανάκτηση, αποθήκευση και διαχείριση της πληροφορίας, σε σχέση με το πώς παρουσιάζεται η ίδια η πληροφορία.

Η παραπάνω διαίρεση ωστόσο, δεν αποκλείει, αλλά ενισχύει την αλληλεπίδραση μεταξύ των τριών τμημάτων. Το μοντέλο (model) αφορά διεργασίες που σχετίζονται με τη βάση δεδομένων και τους κανόνες που συνδέονται με το πεδίο του προβλήματος. Ενημερώνει τις όψεις και τους ελεγκτές όταν υφίσταται αλλαγή στην κατάστασή του. Η όψη (view) επικεντρώνεται στην παρουσίαση και αναπαράσταση της πληροφορίας με σκοπό την παραγωγή ενημερωμένων αποτελεσμάτων. Ο ελεγκτής (controller) ασχολείται με τη λειτουργικότητα της εφαρμογής και την οργάνωση των δεδομένων, στέλνοντας εντολές αφενός στο μοντέλο για να ενημερώσει την κατάστασή του και αφετέρου στην όψη για να επιτευχθεί η αναπαράσταση της νέας, ενημερωμένης κατάστασης του μοντέλου. Το παρακάτω διάγραμμα απεικονίζει την MVC φιλοσοφία.

Διάγραμμα 7: Αρχιτεκτονική Μοντέλου – Όψεως – Ελεγκτή



Όταν ο χρήστης αιτηθεί μια ενέργεια, τα βήματα που ακολουθούνται είναι τα εξής:

1. Το αίτημα μεταβιβάζεται στον Ελεγκτή μέσω του router. Ο ελεγκτής αποφασίζει είτε α) να επικοινωνήσει απευθείας με την Όψη, εφόσον δεν απαιτούνται δεδομένα που βρίσκονται αποθηκευμένα στη Βάση Δεδομένων, είτε β) να επικοινωνήσει με το Μοντέλο από το οποίο θα απαιτήσει την πρόσβαση στη βάση για την ανάκτηση αποθηκευμένων δεδομένων.
2. Στη συνέχεια η βάση στέλνει τα δεδομένα στο Μοντέλο
3. Το Μοντέλο απαντάει στέλνοντας τα αποτελέσματα πίσω στον Ελεγκτή.
4. Όταν ο Ελεγκτής λάβει τα αποτελέσματα καλεί την αντίστοιχη Όψη που θα αναλάβει να τα εμφανίσει.
5. Η Όψη διαμορφώνει την παρουσίαση και την στέλνει πίσω στον Ελεγκτή σε μορφή HTML.
6. Τέλος ο Ελεγκτής στέλνει την HTML σελίδα στον browser και παρουσιάζονται τα αποτελέσματα στον χρήστη.

5.3.3. Πρότυπο REST (Representational State Transfer)

Ένα άλλο πρότυπο που χρησιμοποιείται από το πλαίσιο Ruby on Rails είναι το REST. Το REST αποτελεί ένα πρότυπο επικοινωνίας για την ανάπτυξη υπηρεσιών ιστού. Χρησιμοποιείται κυρίως πάνω από το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HTTP) παρέχοντας καλύτερη διασύνδεση για τη μεταφορά δεδομένων μεταξύ πελάτη (client) και διακομιστή (server). Το REST βασίζεται στην οργάνωση της πληροφορίας με τη μορφή πόρων. Κάθε πόρος αντιδρά μοναδικά σε κάθε ενέργεια που επιτελείται επάνω του. Οι ενέργειες αυτές προσδιορίζονται από έναν Ενιαίο Εντοπιστή Πόρων⁴ (Uniform Resource Locator – URL) και από τις μεθόδους HTTP. Έτσι κάθε μοντέλο σε μια RoR εφαρμογή θεωρείται ως ένα σύνολο πόρων πάνω στο οποίο επιτελούνται διάφορες ενέργειες (actions) μέσω του ελεγκτή. Οι ενέργειες του ελεγκτή δρομολογούνται από τη διεύθυνση του πόρου (URL) και τις HTTP μεθόδους GET, POST, PUT, PATCH και DELETE. Το πρότυπο REST προσφέρει πλεονεκτήματα που αφορούν στην επίδοση, επεκτασιμότητα, απλότητα, δυνατότητα τροποποίησης, φορητότητα και αξιοπιστία των υπηρεσιών ιστού (Βικιπαίδεια, 2015).

5.3.4. RubyGems

Το Ruby on Rails χρησιμοποιεί ένα σύστημα διαχείρισης πακέτων λογισμικού ruby, που ονομάζεται RubyGems. Το σύστημα αυτό αναπτύχθηκε με σκοπό να αυτοματοποιηθεί η διαδικασία της εγκατάστασης, αναβάθμισης, ρύθμισης και αφαίρεσης προγραμμάτων και βιβλιοθηκών της Ruby. Με τη βοήθεια του συστήματος RubyGems το πλαίσιο RoR, εγκαθιστά χρήσιμες βιβλιοθήκες, διαθέσιμες ως πακέτα λογισμικού (gems), οι οποίες διευκολύνουν τις εργασίες του και διευρύνουν τη λειτουργικότητά του. Θα πρέπει να σημειωθεί ότι το ίδιο το RoR αποτελεί ένα πακέτο λογισμικού ruby.

5.4. Βάσεις Δεδομένων

Αυτή η ενότητα επικεντρώνεται στις θεμελιώδεις αρχές των βάσεων δεδομένων και στη συνέχεια αναλύει τη MongoDB, η οποία θα χρησιμοποιηθεί στην ανάπτυξη της εφαρμογής που πραγματεύεται η εργασία.

⁴ Ο Ενιαίος Εντοπιστής Πόρων (URL) δηλώνει τη διεύθυνση ενός πόρου.

5.4.1. Ορισμός της Βάσης Δεδομένων

Η Βάση Δεδομένων (Database) είναι μια οργανωμένη συλλογή μεγάλου όγκου δεδομένων (Merriam Webster Dictionary, 2015), σχεδιασμένη για να υποστηρίζει διαδικασίες που απαιτούν πρόσβαση στις πληροφορίες που διαθέτει. Μια βάση δεδομένων μπορεί να περιέχει οποιαδήποτε πληροφορία σχετίζεται με μια πραγματική επιχείρηση, αφού χρησιμοποιείται για τη μοντελοποίηση των δραστηριοτήτων της. Έτσι, για παράδειγμα μπορεί να περιέχει καταλόγους διευθύνσεων, λίστες με λεπτομέρειες προϊόντων, ονόματα πελατών, επωνυμίες επιχειρήσεων, διαθέσιμα ξενοδοχεία, αριθμούς λογαριασμών και πολλά άλλα δεδομένα που αφορούν ιδιώτες και οργανισμούς. Τα δεδομένα αυτά είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις και μπορούν να χρησιμοποιηθούν από πολλές εφαρμογές και χρήστες ταυτόχρονα.

5.4.2. Συστήματα Διαχείρισης Βάσεων Δεδομένων

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (Database Management Systems – DBMS) είναι εφαρμογές λογισμικού υπολογιστή οι οποίες επιτρέπουν στους χρήστες να ορίζουν, να δημιουργούν, να διαχειρίζονται και να συντηρούν τις βάσεις δεδομένων παρέχοντας αποτελεσματική, αξιόπιστη, εύκολη και ασφαλή πρόσβαση σε αυτές. Τα πιο γνωστά συστήματα διαχείρισης βάσεων δεδομένων είναι η MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase και IBM DB2.

Τα συστήματα διαχείρισης βάσεων δεδομένων κατηγοριοποιούνται ανάλογα με το μοντέλο βάσης δεδομένων που χρησιμοποιούν, δηλαδή ανάλογα με τη δομή και τον τρόπο με τον οποίο τα δεδομένα αποθηκεύονται, οργανώνονται και διαχειρίζονται. Τα πρώτα μοντέλα δεδομένων αναπτύχθηκαν πριν μισό αιώνα.

5.4.2.1. Ιεραρχικό & Δικτυωτό Μοντέλο Βάσεων Δεδομένων

Τη δεκαετία του 1960 αναπτύχθηκε το ιεραρχικό μοντέλο IMS. Βάσει του ιεραρχικού μοντέλου τα δεδομένα οργανώνονται σε διάφορα επίπεδα ιεραρχικά, με αντιστοιχίες μεταξύ τους. Ωστόσο στοιχεία που βρίσκονται σε χαμηλότερα επίπεδα μπορούν να αντιστοιχηθούν σε ένα και μόνο ένα στοιχείο ανωτέρου επιπέδου, με αποτέλεσμα, εξαιτίας αυτού του περιορισμού, το μοντέλο να θεωρείται δύσχρηστο.

Την ίδια περίοδο αναπτύχθηκε το δικτυωτό μοντέλο IDS. Στο δικτυωτό μοντέλο ομοίως με το ιεραρχικό, τα στοιχεία οργανώνονται σε επίπεδα ιεραρχίας, αλλά κάθε στοιχείο μπορεί να συσχετισθεί με πολλά στοιχεία σε ανώτερο ή κατώτερο επίπεδο.

Τα συστήματα διαχείρισης βάσεων δεδομένων, που στηρίζονται στα παραπάνω μοντέλα ονομάζονται και «βάσεις πλοήγησης» διότι επιτρέπουν την ανεύρεση μιας εγγραφής ακολουθώντας παραπομπές από άλλες εγγραφές, αγνοώντας όμως άλλα κριτήρια.

5.4.2.2. Σχεσιακό Μοντέλο Βάσεων Δεδομένων

Τη δεκαετία του 1970 αναπτύχθηκε το σχεσιακό μοντέλο με το οποίο εισήχθη ένας σαφής, απλός και κατανοητός τρόπος αναπαράστασης και διαχείρισης των δεδομένων. Το σχεσιακό μοντέλο βασίστηκε στην ιδέα του Edgar Codd, ο οποίος υποστήριξε ότι τα δεδομένα θα πρέπει να καταχωρούνται σε προκαθορισμένους πίνακες προκειμένου να είναι εφικτή η αναπαράσταση οποιασδήποτε πληροφορίας. Με την πάροδο του χρόνου η ιδέα αυτή εξελίχθηκε, με αποτέλεσμα να δημιουργηθούν οι σχεσιακές βάσεις δεδομένων (relational databases), οι οποίες χρησιμοποιούνται μέχρι σήμερα σε κάθε είδους εφαρμογή.

Το σχεσιακό μοντέλο, που ονομάζεται και Μοντέλο Οντοτήτων – Συσχετίσεων (ΟΣ), ορίζεται από τρεις θεμελιώδεις έννοιες: τις οντότητες, τις ιδιότητες (χαρακτηριστικά) και τις συσχετίσεις. Οι οντότητες αναφέρονται στα αντικείμενα, τα πρόσωπα, τις καταστάσεις και γενικά σε οτιδήποτε μπορεί να προσδιορισθεί ως ανεξάρτητη ύπαρξη. Οι ιδιότητες αναφέρονται στα δομικά στοιχεία που προσδιορίζουν μια οντότητα και οι συσχετίσεις αφορούν στον τρόπο σύνδεσης των οντοτήτων, ώστε να αντλούνται πληροφορίες από τον συνδυασμό τους.

Έτσι, σύμφωνα με το σχεσιακό μοντέλο τα δεδομένα περιγράφονται από οντότητες και συσχετίσεις με τη μορφή πινάκων (tables) και σχέσεων (relations) αντίστοιχα. Ένας πίνακας αποτελείται από γραμμές και στήλες. Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και ονομάζεται χαρακτηριστικό (attribute) ή πεδίο (field), ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες που αφορούν ένα στοιχείο της οντότητας και ονομάζεται εγγραφή (record). Κάθε πεδίο του πίνακα δύναται να πάρει ορισμένες μόνο τιμές, οι οποίες καθορίζονται από τον τύπο των δεδομένων της ιδιότητας. Το σύνολο των αποδεκτών τιμών μιας οντότητας ονομάζεται

πεδίο ορισμού (domain). Από τις σημαντικότερες ιδιότητες των οντοτήτων είναι το πρωτεύον κλειδί, δηλαδή το πεδίο εκείνο του οποίου η τιμή είναι μοναδική και προσδιορίζει την κάθε εμφάνιση της οντότητας. Τέλος, οι συσχετίσεις συνδέουν δύο ή και περισσότερες οντότητες έτσι ώστε να είναι εφικτή η άντληση πληροφοριών.⁵

5.4.2.3. Μετά-σχεσιακά Μοντέλα Βάσεων Δεδομένων

Την τελευταία δεκαετία, με τη ραγδαία ανάπτυξη της τεχνολογίας αλλά και τη διόγκωση των δεδομένων του διαδικτύου με εκθετικούς ρυθμούς, αναπτύχθηκε μια νέα γενιά μοντέλων δεδομένων, με σκοπό να αντιμετωπιστούν οι κύριες αδυναμίες των σχεσιακών βάσεων δεδομένων. Τα μετά-σχεσιακά, όπως ονομάζονται, μοντέλα, δεν απαιτούν προκαθορισμένα σχήματα πινάκων, αλλά χρησιμοποιούν μια ελεύθερη δομή αποθήκευσης στοιχείων, απορρίπτοντας ουσιαστικά τη σχεσιακή δόμηση των δεδομένων. Επιπλέον τα συστήματα που βασίζονται στα μετά-σχεσιακά μοντέλα έχουν την ικανότητα διαχείρισης μεγάλου όγκου δεδομένων, είναι καταναμημένα, υποστηρίζουν την εύκολη αντιγραφή των δεδομένων και έχουν τη δυνατότητα οριζόντιας αναβάθμισης.

Αναλυτικότερα τα μη-σχεσιακά ή όχι-μόνο-σχεσιακά (NoSQL ή Not Only SQL) συστήματα δημιουργήθηκαν κυρίως με γνώμονα την δυνατότητα διαχείρισης μεγάλου όγκου δεδομένων χωρίς κατ' ανάγκη να διατηρούν μια συγκεκριμένη δομή (schema). Έτσι επιτρέπουν την εισροή δεδομένων χωρίς να χρειάζεται από πριν να υπάρχει κάποιο προκαθορισμένο μοντέλο, δίνοντας τη δυνατότητα να πραγματοποιούνται αλλαγές στη βάση σε πραγματικό χρόνο. Επιπλέον αξιοποιούν μια αρχιτεκτονική που επιτρέπει την καταναμημένη λειτουργία τους και την οριζόντια κλιμάκωση. Έτσι το σύστημα μπορεί να μεγιστοποιήσει τις επιδόσεις του προσθέτοντας επιπλέον εξυπηρετητές (servers) όπου καταναμημένα θα επεξεργάζονται τα δεδομένα του συστήματος. Αυτό γίνεται αυτόματα χωρίς εκ των προτέρων να έχει οριστεί η διάταξη αυτών των εξυπηρετητών. Επίσης, τα περισσότερα NoSQL συστήματα υποστηρίζουν αντιγραφή δεδομένων δημιουργώντας πολλαπλά αντίγραφα, για να διασφαλιστεί η υψηλή διαθεσιμότητα και η ανάκαμψη από ενδεχόμενη καταστροφή.

⁵ Κέντρο Πληροφορίας & Νέων Τεχνολογιών Ν. Φλώρινας. *Θεωρία Βάσεων Δεδομένων*. Ανακτήθηκε 8 Ιουνίου, 2015, από <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-DataBasesTheory.html>

Οι βάσεις δεδομένων που αναπτύχθηκαν χρησιμοποιώντας τα μετά-σχεσιακά μοντέλα διακρίνονται στις εξής κατηγορίες:

- Βάσεις Ζεύγους Κλειδιού – Τιμής (Key – Value)
- Βάσεις Αποθήκευσης σε Πίνακες κατά Στήλες (Column Oriented / Tabular)
- Βάσεις Εγγράφων (Document Oriented)
- Βάσεις Γράφου (Graph)

Παρακάτω παρατίθεται πίνακας που συνοψίζει το βαθμό επιδόσεων, επεκτασιμότητας, ευελιξίας, πολυπλοκότητας και λειτουργικότητας των NoSQL βάσεων δεδομένων ανάλογα με το μετά-σχεσιακό μοντέλο δεδομένων που τις χαρακτηρίζει.

Πίνακας 1: NoSQL Βάσεις Δεδομένων

Μοντέλο Δεδομένων	Επιδόσεις	Επεκτασιμότητα	Ευελιξία	Πολυπλοκότητα	Λειτουργικότητα
Βάσεις Ζεύγους Κλειδιού – Τιμής	Υψηλές	Υψηλή	Υψηλή	Καμία	Ποικίλει
Βάσεις Αποθήκευσης σε Πίνακες κατά Στήλες	Υψηλές	Υψηλή	Μέτρια	Χαμηλή	Ελάχιστη
Βάσεις Εγγράφων	Υψηλές	Ποικίλει	Υψηλή	Χαμηλή	Ποικίλει
Βάσεις Γράφου	Ποικίλουν	Ποικίλει	Υψηλή	Υψηλή	Θεωρία Γράφου

Για τους σκοπούς της παρούσας εργασίας αξιοποιούνται οι βάσεις εγγράφων (document oriented) και πιο συγκεκριμένα η MongoDB.

5.4.2.3.1. Βάσεις Οργανωμένες σε Έγγραφα

Οι βάσεις εγγράφων αποτελούν την κύρια κατηγορία των NoSQL βάσεων. Η ιδιαίτερη αρχιτεκτονική τους, δηλαδή η αποθήκευση των δεδομένων σε έγγραφα, βελτιστοποιεί

την απόδοση τους προσδίδοντας ταχύτητα, ευκολία και επεκτασιμότητα. Τα έγγραφα έχουν ένα μοναδικό αναγνωριστικό ID, ώστε να γίνεται γρήγορα η αναζήτηση και η ανάκτησή τους χρησιμοποιώντας ευρετήρια (indexes). Είναι συνήθως τύπου JSON (JavaScript Object Notation) ή παραλλαγές όπως BSON (Binary JSON) και περιέχουν δεδομένα με τη μορφή ζευγών πεδίου και τιμής. Η τιμή ενός πεδίου μπορεί να είναι άλλα έγγραφα, πίνακες (arrays) ή πίνακες εγγράφων (arrays of documents). Η απουσία σχήματος δίνει τη δυνατότητα ελεύθερης προσθήκης πεδίων σε αυτά, χωρίς να απαιτείται πρώτα καθορισμός των αλλαγών. Οι βάσεις εγγράφων συνήθως χρησιμοποιούνται σε συνδυασμό με αντικειμενοστρεφείς γλώσσες προγραμματισμού.

5.5. MongoDB

Η MongoDB είναι μια σχετικά νέα βάση δεδομένων, η οποία ανήκει στην οικογένεια των μη-σχεσιακών βάσεων δεδομένων και συγκεκριμένα στις βάσεις εγγράφων. Αναπτύχθηκε το 2007 και είναι ανοικτού λογισμικού γραμμένη σε γλώσσα C++. Αποτελεί μία από τις πιο διαδεδομένες λύσεις NoSQL καθώς είναι σχεδιασμένη με τέτοιο τρόπο ώστε να ανταπεξέρχεται στις απαιτήσεις των σύγχρονων εφαρμογών, χωρίς να έχει χαρακτηριστικά των σχεσιακών βάσεων όπως: πίνακες, σειρές, σχήματα, ερωτήματα SQL, συναλλαγές, ενώσεις και ξένα κλειδιά. Τα δεδομένα της αποθηκεύονται σε αρχεία BSON, με αποτέλεσμα να επιτυγχάνεται η γρήγορη κωδικοποίηση και αποκωδικοποίηση τους, προσφέροντας έτσι ένα μεγάλο πλεονέκτημα αποδοτικότητας έναντι άλλων βάσεων εγγράφων.

5.5.1. Χαρακτηριστικά της MongoDB

Τα κυριότερα χαρακτηριστικά της MongoDB είναι τα εξής (Hows, Membrey & Plugge, 2014):

- **Απουσία σχήματος.** Η MongoDB μπορεί να ανταπεξέρχεται σε μεγάλη ποσότητα δεδομένων. Η αποθήκευση των δεδομένων γίνεται σε έγγραφα, το σύνολο των οποίων ονομάζεται συλλογές (collections). Μια συλλογή δεν προαπαιτεί την ύπαρξη κάποιου σχήματος, δηλαδή τα έγγραφα που την αποτελούν δεν είναι απαραίτητο να έχουν τα ίδια πεδία ή την ίδια δομή. Τα έγγραφα οφείλουν μόνο να περιέχουν πεδία σχετικά με την οντότητα ή το

αντικείμενο που αντιπροσωπεύουν έτσι ώστε να υπάρχει συνάφεια μεταξύ της βάσης δεδομένων και των μοντέλων μιας εφαρμογής.

- **Αναδιπλασιασμός.** Ο Αναδιπλασιασμός (replication) αναφέρεται στην διαδικασία συγχρονισμού των δεδομένων μέσω πολλαπλών εξυπηρετητών έτσι ώστε τα δεδομένα να είναι διαθέσιμα κάθε στιγμή. Στη MongoDB τα δεδομένα αποθηκεύονται σε σύνολα από αντίγραφα (replica sets). Το κάθε σύνολο αντιγράφων αποτελείται από έναν πρωτεύοντα κόμβο και έναν ή περισσότερους δευτερεύοντες κόμβους. Ο πρωτεύων κόμβος αναλαμβάνει την εξυπηρέτηση των εγγραφών και των αναγνώσεων για τα δεδομένα που επεξεργάζεται, ενώ ταυτόχρονα αποθηκεύει τις κινήσεις που εκτελούνται και κάθε μετατροπή των δεδομένων σε ένα αρχείο καταγραφής ενεργειών, που ονομάζεται oplog (operations log). Στη συνέχεια οι δευτερεύοντες κόμβοι εκτελούν μία προς μία τις καταγεγραμμένες ενέργειες στο oplog και ενημερώνουν τα αποθηκευμένα δεδομένα τους. Συνεπώς με αυτή την ασύγχρονη διαδικασία τα δεδομένα είναι πλήρως ενημερωμένα σε κάθε αντίγραφο και ταυτόχρονα εξασφαλίζεται η διαθεσιμότητά τους σε περίπτωση ανάγκης.
- **Κατακερματισμός.** Αφορά στη διαδικασία κατά την οποία τα δεδομένα διαιρούνται και διανέμονται σε πολλούς εξυπηρετητές. Καθώς ο όγκος των δεδομένων αυξάνεται ένας μεμονωμένος εξυπηρετητής ενδέχεται να μην είναι επαρκής για την αποθήκευση των δεδομένων και για τις διαδικασίες εγγραφής και ανάγνωσης. Η MongoDB αναλαμβάνει να διαιρέσει τα δεδομένα και να τα καταλείψει σε διαφορετικούς εξυπηρετητές. Τα διαφορετικά κομμάτια ή «θραύσματα» που δημιουργούνται από την κατάτμηση ονομάζονται shards. Σε κάθε shard θα δημιουργηθεί ένα σύνολο αντιγράφων και θα ξεκινήσει μια διαδικασία ισοκατανομής των δεδομένων έτσι ώστε να έχουν τον ίδιο όγκο δεδομένων, μοιράζοντας ταυτόχρονα τον φόρτο εργασίας. Η τεχνική αυτή ονομάζεται και οριζόντια κλιμάκωση.
- **Ευρετήρια.** Η MongoDB χρησιμοποιεί ευρετήρια (index), δηλαδή δομές δεδομένων, για τον γρήγορο εντοπισμό εγγραφών βάσει των αποθηκευμένων τιμών σε συγκεκριμένα πεδία. Με τον τρόπο αυτό αυξάνεται η απόδοσή της κατά τη διαδικασία ανάγνωσης, όταν μάλιστα υπάρχουν χιλιάδες έγγραφα αποθηκευμένα στη βάση. Η MongoDB δημιουργεί ευρετήρια αυτόματα για όλα

τα έγγραφα αναθέτοντας στο καθένα ένα αναγνωριστικό `_id`. Το αναγνωριστικό αυτό εξασφαλίζει ότι κάθε τιμή είναι μοναδική και εντοπίσιμη. Επιπλέον η MongoDB δίνει τη δυνατότητα δημιουργίας προσαρμοσμένων ευρετηρίων αλλά και σύνθετων ευρετηρίων στην περίπτωση που πρόκειται να αναζητηθούν έγγραφές οι οποίες βασίζονται σε παραπάνω από ένα πεδίο.

5.6. Elasticsearch

Η Elasticsearch είναι μία ανοικτού λογισμικού μηχανή αναζήτησης κειμενικών δεδομένων, με δυνατότητα κλιμάκωσης σε εκατοντάδες εξυπηρετητές και αποθήκευσης petabytes δομημένων και αδόμητων δεδομένων. Δημιουργήθηκε από τον Shay Banon το 2010 και έκτοτε χρησιμοποιείται σε πολλές εφαρμογές. Λόγω των δυνατοτήτων που παρέχει σε πραγματικό χρόνο πολλοί τη χρησιμοποιούν και ως αποθήκη εγγράφων. Είναι γραμμένη σε γλώσσα προγραμματισμού Java και βασίζεται στη βιβλιοθήκη Apache Lucene. Είναι ιδιαίτερα εύχρηστη προγραμματιστικά, καθώς το κυρίως πρωτόκολλο που χρησιμοποιεί βασίζεται σε HTTP/JSON και ταυτόχρονα παρέχει διαλειτουργικότητα με γλώσσες προγραμματισμού πέραν της Java.

5.6.1. Βασικές έννοιες

Οι βασικές έννοιες της Elasticsearch είναι οι εξής (Paro, 2015):

- **Ευρετήριο (Index).** Το ευρετήριο είναι ο χώρος όπου η Elasticsearch αποθηκεύει τα δεδομένα. Σε αντιπαράβολή με τη MongoDB το ευρετήριο στην Elasticsearch είναι ότι η συλλογή (collection) στη MongoDB. Τα ευρετήρια μπορεί να είναι όλα μαζί σε ένα εξυπηρετητή ή να είναι διαμοιρασμένα σε πολλούς εξυπηρετητές. Κάθε ευρετήριο διαμορφώνεται από ένα ή περισσότερα κομμάτια (θραύσματα) που ονομάζονται shards (όπως και στη MongoDB) και κάθε shard μπορεί να έχει πολλά αντίγραφα.
- **Έγγραφο (Documents).** Η κύρια αποθηκευτική οντότητα στην Elasticsearch είναι ένα έγγραφο. Ένα έγγραφο μπορεί να έχει διαφορετικές δομές, αλλά, σε αντίθεση με τα έγγραφα της MongoDB, πρέπει να έχει τον ίδιο τύπο δεδομένων για όλα τα κοινά πεδία. Για παράδειγμα όλα τα έγγραφα που έχουν πεδίο title θα πρέπει να έχουν τον ίδιο τύπο δεδομένων, που στην περίπτωση αυτή έχει

οριστεί να είναι ένα `string`. Τα έγγραφα αποτελούνται από πεδία και κάθε πεδίο μπορεί να εμφανίζεται περισσότερες από μια φορές μέσα σε ένα έγγραφο (τότε ονομάζεται πεδίο πολλαπλών τιμών). Κάθε πεδίο έχει ένα τύπο δεδομένων όπως κείμενο, αριθμό, ημερομηνία ή μπορεί να έχει ένα τύπο πιο σύνθετο όπως κάποιο έγγραφο ή πίνακα. Ο τύπος του πεδίου είναι πολύ σημαντικός στην ElasticSearch διότι παρέχει πληροφορίες για τον τρόπο με τον οποίο λειτουργίες όπως η ανάλυση και η ταξινόμηση θα εκτελεστούν.

- **Χαρτογράφηση (Mapping).** Η διαδικασία της χαρτογράφησης είναι απαραίτητη έτσι ώστε η ElasticSearch να μπορεί να διεκπεραιώσει αποτελεσματικά την οποιαδήποτε αναζήτηση δεδομένων. Η ElasticSearch αποθηκεύει πληροφορίες για τα πεδία κατά τη διαδικασία της χαρτογράφησης. Κάθε πεδίο ενός εγγράφου αναλύεται και στη συνέχεια χαρτογραφείται σύμφωνα με τον τύπο του. Για παράδειγμα διαφορετική ανάλυση θα γίνει για τα πεδία που περιέχουν αριθμούς, εφόσον οι αριθμοί δεν ταξινομούνται αλφαβητικά και διαφορετική ανάλυση θα γίνει για τα πεδία που περιέχουν κείμενο.
- **Κατακερματισμός.** Η ElasticSearch μπορεί να λειτουργήσει σε έναν μόνο εξυπηρετητή, ωστόσο προκειμένου να επεξεργαστεί μεγάλο όγκο δεδομένων και να αντιμετωπίσει τυχόν σφάλματα είναι δυνατόν να λειτουργήσει σε ένα σύνολο συνεργαζόμενων εξυπηρετητών. Το σύνολο των συνεργαζόμενων εξυπηρετητών ονομάζεται σύμπλεγμα (cluster) και ο καθένας από αυτούς ξεχωριστά ονομάζεται κόμβος (node). Έτσι, στην περίπτωση που ένας κόμβος δεν επαρκεί για την επεξεργασία των δεδομένων, τα δεδομένα διαιρούνται σε μικρότερα κομμάτια τα `shards`⁶. Κάθε `shard` μπορεί να ανήκει σε διαφορετικό εξυπηρετητή με αποτέλεσμα τα δεδομένα να απλώνονται σε όλους τους κόμβους ενός συμπλέγματος. Στην περίπτωση που τεθούν ερωτήματα στο ευρετήριο, τα αποτελέσματα που επιστρέφονται ουσιαστικά συντίθενται από πολλαπλά θραύσματα.
- **Αναδιπλασιασμός.** Για να αυξηθεί η απόδοση και η διαθεσιμότητα των δεδομένων η ElasticSearch δημιουργεί αντίγραφα των `shards`. Τα `shards` χωρίζονται στο πρωτεύον και τα αντίγραφα. Οι λειτουργίες που αφορούν αλλαγές στο ευρετήριο αποθηκεύονται στο πρωτεύον `shard`. Στην περίπτωση

⁶ Κάθε `shard` είναι ένα ξεχωριστό ευρετήριο της Apache Lucene.

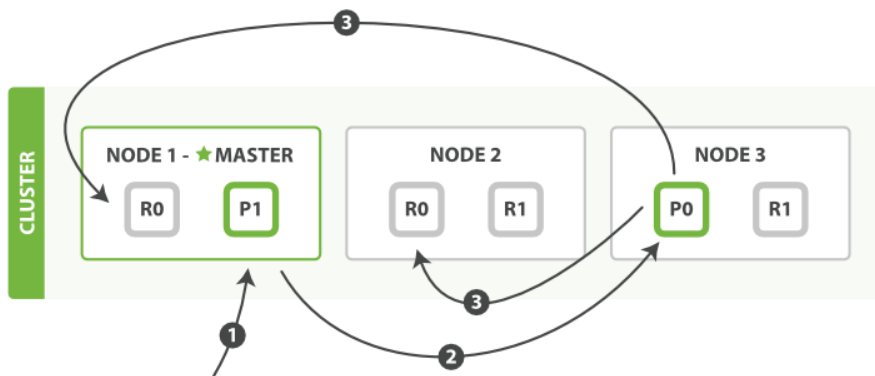
που το πρωτεύον shard δεν ανταποκρίνεται, για παράδειγμα λόγω αδυναμίας του εξυπηρετητή, τότε το σύμπλεγμα (σύνολο των εξυπηρετητών) θα ορίσει ένα αντίγραφο ως το πρωτεύον shard.

5.6.2. Δημιουργία, Διαγραφή & Ανάκτηση Δεδομένων

Για τη δημιουργία ή διαγραφή ενός εγγράφου στο σύμπλεγμα (cluster) ορίζεται ένα ευρετήριο σε έναν κόμβο (node). Ο κόμβος γνωρίζει πόσα shards έχει το ευρετήριο και καθορίζει ποιο shard θα χρησιμοποιηθεί για να αποθηκεύσει το έγγραφο. Κάθε έγγραφο έχει ένα μοναδικό αναγνωριστικό (`_id`) έτσι ώστε να η Elasticsearch να γνωρίζει σε ποιο shard είναι καταχωρημένο ή πού θα καταχωρηθεί, αλλά και για να μπορεί να το ανακτήσει όταν της ζητηθεί. Κατά τη διαδικασία της ανάκτησης ο κόμβος δέχεται ένα ερώτημα από τον πελάτη. Στη συνέχεια απευθύνεται στο shard που περιέχει το έγγραφο, μεταβιβάζει το ερώτημα, ανακτά το αποτέλεσμα και το στέλνει πίσω στον πελάτη.

Το παρακάτω διάγραμμα αναπαριστά τον τρόπο δημιουργίας και διαγραφής ενός εγγράφου (Gormley & Tong, 2014).

Εικόνα 1: Δημιουργία & Διαγραφή Εγγράφου στην Elasticsearch



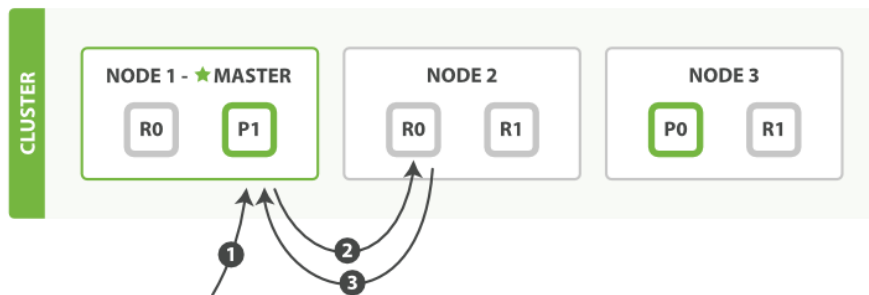
Τα βήματα που ακολουθούνται είναι τα εξής:

1. Ο πελάτης στέλνει αίτημα δημιουργίας ή διαγραφής στον κόμβο 1 (NODE 1)
2. Ο κόμβος 1 χρησιμοποιεί το αναγνωριστικό του εγγράφου (`_id`) και καθορίζει ότι το έγγραφο ανήκει στο shard 0. Μεταβιβάζει το αίτημα στον κόμβο 3 (NODE 3) διότι εκεί βρίσκεται το πρωτεύον shard 0 (P0).

3. Ο κόμβος 3 (NODE 3) εκτελεί το αίτημα στο πρωτεύον shard 0 (P0). Στην περίπτωση επιτυχίας στέλνει τα αποτελέσματα του αιτήματος στα αντίγραφα shards (R0) που βρίσκονται στον κόμβο 1 και 2 προκειμένου να ενημερωθούν. Εφόσον ολοκληρωθεί η ενημέρωσή τους, ο κόμβος 3 ενημερώνει τον κόμβο 1, όπου αρχικά υποδέχτηκε το αίτημα από τον πελάτη, ο οποίος με τη σειρά του ενημερώνει τον πελάτη για την επιτυχή έκβαση του αποτελέσματος.

Ομοίως το παρακάτω διάγραμμα αναπαριστά τον τρόπο ανάκτησης ενός εγγράφου.

Εικόνα 2: Ανάκτηση Εγγράφου στην Elasticsearch



Τα βήματα που ακολουθούνται έχουν ως εξής:

1. Ο πελάτης στέλνει αίτημα ανάκτησης στον κόμβο 1 (NODE 1).
2. Ο κόμβος 1 χρησιμοποιεί το αναγνωριστικό του εγγράφου (`_id`) και καθορίζει ότι το έγγραφο ανήκει στο shard 0. Αντίγραφα του shard 0 υπάρχουν και στους 3 κόμβους. Στην περίπτωση αυτή μεταβιβάζει το αίτημα στον κόμβο 2, σε ένα αντίγραφο shard, του οποίου η επιλογή έγινε ανάλογα με το φόρτο εργασίας.
3. Ο κόμβος 2 στέλνει το έγγραφο στον κόμβο 1, ο οποίος με τη σειρά του το μεταβιβάζει στον πελάτη.

ΚΕΦΑΛΑΙΟ 6: ΔΙΑΧΕΙΡΙΣΗ ΕΡΓΟΥ: «ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΣΥΓΓΡΑΜΜΑΤΩΝ»

6.1. Εισαγωγή

Για τους σκοπούς του συγκεκριμένου κεφαλαίου δημιουργήθηκε το παρακάτω σενάριο:

- Η διαδικτυακή εφαρμογή διαχείρισης εκπαιδευτικών συγγραμμάτων που θα αναπτυχθεί ονομάζεται BookYourWay. Την εφαρμογή αναλαμβάνει να υλοποιήσει η εταιρεία ALTA Designs Ε.Π.Ε για λογαριασμό του φροντιστηρίου «Εξέλιξη». Αντικείμενο εργασιών της εταιρείας είναι η ανάπτυξη λογισμικού. Η ALTA Designs δεν έχει πολλά χρόνια στον χώρο, απασχολεί ελάχιστο προσωπικό και διοικείται, ως επί των πλείστων από νέους ανθρώπους με όραμα και διάθεση για δουλειά. Ωστόσο το μοντέλο της διοίκησης που ακολουθείται είναι ιεραρχικό. Για όλες τις αποφάσεις τον τελικό λόγο έχει ο ιδιοκτήτης της εταιρείας, χωρίς να αφήνει πολλά περιθώρια ευελιξίας. Παρόλα αυτά είναι γνώστης του αντικειμένου και πλήρως ενημερωμένος για τις τελευταίες εξελίξεις. Για την ανάπτυξη της εφαρμογής αποφάσισε να χρησιμοποιήσει μια νέα μεθοδολογία, για τα δεδομένα της εταιρείας, την Scrum, η οποία ξεφεύγει από τον παραδοσιακό τρόπο ανάπτυξης λογισμικού. Καθώς όμως δε γνώριζε τίποτα για τη μεθοδολογία αυτή, προσέλαβε έναν ειδικό, τον Κωνσταντίνο, και του ανέθεσε καθήκοντα senior project manager στην εταιρεία. Ο Κωνσταντίνος έχει το δύσκολο έργο να εφαρμόσει ένα ευέλικτο μοντέλο, σε μια εταιρεία που στηρίζεται σε παραδοσιακές δομές.

Η δομή του περιεχομένου αυτού του κεφαλαίου ακολουθεί τον κύκλο ζωής διαχείρισης έργου της μεθοδολογίας Scrum.

6.2. Διαδικασίες πριν την Εκκίνηση

Πριν την εκκίνηση του έργου εκτελούνται διαδικασίες εντός της εταιρείας που θα βοηθήσουν στον καθορισμό του μεγέθους, του πεδίου εφαρμογής και της πολυπλοκότητας της εφαρμογής.

6.2.1. Αίτηση Έναρξης Ανάπτυξης Εφαρμογής

Αρχικά καταρτίζεται η **αίτηση έναρξης ανάπτυξης εφαρμογής**, η οποία υποβάλλεται στη διοίκηση της εταιρείας προς έγκριση. Σε συνεργασία με τον πελάτη, συλλέγονται, αναλύονται και οργανώνονται διάφορα στοιχεία έτσι ώστε να καθορισθούν η προβληματική κατάσταση καθώς και οι πρώιμες προτάσεις για την αντιμετώπισή της. Παρακάτω παρουσιάζεται η αίτηση έναρξης ανάπτυξης συστήματος που συντάχθηκε για τις ανάγκες του υπό εξέταση έργου προς τον ιδιοκτήτη της ALTA Designs.

Πίνακας 2: Αίτηση Έναρξης Ανάπτυξης Συστήματος

Αίτηση Έναρξης Ανάπτυξης Συστήματος	
Είδος Αίτησης	
<input checked="" type="checkbox"/>	Νέο Σύστημα
<input type="checkbox"/>	Βελτίωση Συστήματος
<input type="checkbox"/>	Διόρθωση Σφαλμάτων Συστήματος
Βαθμός Αμεσότητας Λήψης Μέτρων	
<input checked="" type="checkbox"/>	Υψηλός (Άμεση αντιμετώπιση του προβλήματος)
<input type="checkbox"/>	Μέτριος (Υπάρχουν προβλήματα αλλά μπορούν να αντιμετωπιστούν)
<input type="checkbox"/>	Χαμηλός (Αρκετή ανοχή της υπάρχουσας κατάστασης)
Περιγραφή Προβληματικής Κατάστασης	
Οι σχολικές απαιτήσεις έχουν αυξηθεί σημαντικά με αποτέλεσμα πολλοί μαθητές να διαβάζουν αρκετές ώρες προκειμένου να ανταπεξέλθουν στα καθήκοντά τους. Η πίεση αυτή σε συνδυασμό με άλλες αντικειμενικές δυσκολίες, όπως η δυσκολία κατανόησης της	

σχολικής ύλης, η πιθανή έλλειψη μεταδοτικότητας εκπαιδευτικών, η δυσκολία στην παρακολούθηση και αφομοίωση της διδασκαλίας, το διαφορετικό μορφωτικό και διανοητικό επίπεδο καθώς και η έλλειψη διαθέσιμου χρόνου, εμποδίζουν τον μαθητή να επικεντρωθεί σε αυτό που θέλει, σε αυτό που συμπληρώνει και εμπλουτίζει τη γνώση του. Τα υπάρχοντα συστήματα δεν ανταποκρίνονται σε αυτές τις νέες συνθήκες που έχουν διαμορφωθεί. Η ύλη για κάθε μάθημα δεν έχει οργανωθεί με τρόπο ώστε να εξυπηρετεί την καθημερινή μελέτη του μαθητή. Ταυτόχρονα ο μαθητής αποσπάται από τον όγκο των μαθημάτων που έχει και χάνεται στην προσπάθεια εντοπισμού, ανάλυσης και μεθόδευσης της διάχυτης πληροφορίας.

Αρχικές Προτάσεις Αντιμετώπισης

Επιζητείται μια πλήρης ανάλυση της υπάρχουσας κατάστασης με σκοπό τη σχεδίαση και ανάπτυξη μιας εφαρμογής. Η εφαρμογή θα πρέπει να παρέχει γρήγορη πρόσβαση στα σχολικά μαθήματα με απλό και κατανοητό τρόπο. Επίσης θα πρέπει να δίνει τη δυνατότητα στοχευμένης αναζήτησης.

Να Συμπληρωθεί από τη Διοίκηση

- Η Αίτηση Εγκρίνεται
- Προτείνεται Αναθεώρηση
- Απορρίπτεται για τους παρακάτω λόγους:

6.2.2. Ανάθεση Ρόλων

Βάσει του σεναρίου που έχει δημιουργηθεί τον ρόλο του Scrum Master αναλαμβάνει ο νεοπροσληφθείς Κωνσταντίνος, ως ειδικός στις ευέλικτες μεθοδολογίες και ιδιαίτερα στη Scrum.

Για τον ρόλο του Product Owner επιλέγεται η Αθηνά, ένα ιδιαίτερα επικοινωνιακό άτομο, με εκτεταμένη εμπειρία στη συλλογή και την ανάλυση των απαιτήσεων πληροφοριακών συστημάτων.

Η Ομάδα Ανάπτυξης αποτελείται από τους προγραμματιστές Βαγγέλη και Κατερίνα οι οποίοι έχουν αρκετή εμπειρία σε αντικειμενοστρεφείς γλώσσες προγραμματισμού, με ιδιαίτερη προτίμηση τη Ruby.

Στην πραγματικότητα όλους τους παραπάνω ρόλους ανέλαβε ένα και μόνο άτομο, ο συγγραφέας της παρούσας μελέτης.

6.2.3. Προσδιορισμός Κινδύνων

Μία εξίσου σημαντική διαδικασία πριν την εκκίνηση του έργου είναι ο προσδιορισμός και η αξιολόγηση τυχόν κινδύνων. Στόχος της δραστηριότητας αυτής είναι να εντοπισθούν οι πηγές των κινδύνων του έργου και να υπολογισθούν οι επιπτώσεις τους. Οι κίνδυνοι μπορούν να προκύψουν από:

- ενδεχόμενη αντίσταση των χρηστών σε κάποια αλλαγή όπως η υιοθέτηση νέας τεχνολογίας ή του τρόπου διεκπεραίωσης των εργασιών τους
- μη διαθεσιμότητα κρίσιμων πόρων
- νομικά θέματα
- ανταγωνιστικές αντιδράσεις

6.2.4. Προϋπολογισμός Έργου

Επιπλέον, η εκτίμηση του κόστους του έργου, έχει σημαίνουσα αξία για την επιχείρηση. Πριν την εκκίνηση του έργου, είναι απαραίτητο να δημιουργηθεί ένα προσχέδιο το οποίο να περιγράφει τις προγραμματισμένες δαπάνες και τα έσοδα που σχετίζονται με το έργο καθώς και ανάλυση σχετικά με τις πιθανές χρηματοροές και την απόδοση της επένδυσης. Η οικονομική ανάλυση που αφορά την ανάπτυξη της υπό εξέταση εφαρμογής δεν αποτελεί μέρος της διπλωματικής εργασίας.

6.3. Στάδιο 1^ο – Όραμα

Στο στάδιο αυτό προετοιμάζεται η δήλωση οράματος από τον Product Owner και παρουσιάζεται στον πελάτη προς έγκριση. Σε περίπτωση που χρειαστεί γίνονται τροποποιήσεις.

Πίνακας 3: Δήλωση Οράματος για το έργο BookYourWay

ΔΗΛΩΣΗ ΟΡΑΜΑΤΟΣ ΓΙΑ ΤΟ ΕΡΓΟ BookYourWay	
Όνομα Έργου:	BookYourWAY
Περιγραφή Προβλήματος - Ευκαιρίας	<p>Οι σχολικές απαιτήσεις έχουν αυξηθεί σημαντικά. Αρκετές φορές οι μαθητές δυσκολεύονται να παρακολουθήσουν, να κατανοήσουν και να αφομοιώσουν τη διδασκαλία. Καθώς οι προσλαμβάνουσες του εκάστοτε μαθητή ποικίλουν, είναι συχνά δύσκολο να διεξαχθεί από τους εκπαιδευτικούς μια επιτυχημένη διδασκαλία η οποία θα βασίζεται στη δυνατότητά τους να μεταδώσουν τη γνώση και τη νεοδιδασχθείσα ύλη λαμβάνοντας υπόψιν πρωτίστως τη διαφορετικότητα του κάθε μαθητή τόσο σε διανοητικό όσο και σε μορφωτικό επίπεδο. Επιζητείται επομένως η ανάπτυξη μιας εφαρμογής η οποία θα βοηθά τους μαθητές στις καθημερινές τους σχολικές απαιτήσεις και θα τους φέρνει ένα βήμα πιο κοντά προς την κατάκτηση της γνώσης καθώς και στην ορθότερη διαχείρισή της.</p>
Στόχοι Έργου:	<p>Η προτεινόμενη εφαρμογή θα παρέχει:</p> <ul style="list-style-type: none">• Γρήγορη πρόσβαση και εύκολη πλοήγηση στα ηλεκτρονικά σχολικά βιβλία• Αποτελεσματική αναζήτηση συγκεκριμένου βιβλίου, κεφαλαίου, ενότητας, παραγράφου, ορισμού, ερώτησης/άσκησης ή και μεμονωμένης λέξης• Δυνατότητα δημιουργίας μιας βάσης δεδομένων σχολικών συγγραμμάτων• Παροχή βοήθειας στους μαθητές στην εκπλήρωση των μαθητικών τους υποχρεώσεων. <p>Οι παρεχόμενες υπηρεσίες της εφαρμογής απευθύνονται σε μαθητές</p>

	της πρωτοβάθμιας ή δευτεροβάθμιας εκπαίδευσης.
Περιγραφή Έργου:	Θα αναπτυχθεί μια νέα εφαρμογή διαχείρισης εκπαιδευτικών συγγραμμάτων η οποία θα παρέχει πρόσβαση στα ειδικά διαμορφωμένα ηλεκτρονικά σχολικά βιβλία αφενός και αφετέρου τη δυνατότητα αναζήτησης εννοιών, ορισμών, λέξεων κλειδιών, ερωτήσεων/ασκήσεων ή και ολόκληρου κειμένου στα ηλεκτρονικά βιβλία.
Οφέλη Έργου:	Τα οφέλη της εφαρμογής θα αφορούν: <ul style="list-style-type: none"> • Τη δυνατότητα άμεσης και ολοκληρωμένης γνώσης για συγκεκριμένο θέμα, • Τη γρήγορη επίλυση αποριών και προβλημάτων, • Την καλή οργάνωση του χρόνου μελέτης, • Την κατανόηση, αφομοίωση και συστηματοποίηση της σχολικής ύλης, • Τη μελέτη σε οποιοδήποτε χώρο και οποιαδήποτε χρονική στιγμή.

6.4. Στάδιο 2^ο – Οδικός Χάρτης Έργου

Ο οδικός χάρτης αποτυπώνει τις αρχικές απαιτήσεις του έργου αναλύοντας τις ανάγκες και τις προσδοκίες του πελάτη. Αποτελεί ένα είδος «χαρτογράφησης» του πεδίου που αφορά στον σχεδιασμό και την εκτέλεση της ανάπτυξης της εφαρμογής. Είναι σημαντικό να παρατηρηθεί ότι οι απαιτήσεις αλλάζουν και εξελίσσονται κατά τη διάρκεια υλοποίησης της εφαρμογής με την εμπλοκή πάντα του πελάτη. Ο οδικός χάρτης συντάσσεται από τον Product Owner.

Παρακάτω καθορίζονται οι αρχικές απαιτήσεις, οι οποίες στη συνέχεια ομαδοποιούνται και δημιουργείται μια λίστα με τα επιθυμητά χαρακτηριστικά (features).

6.4.1. Αρχικές Απαιτήσεις Πελάτη

Οι αρχικές απαιτήσεις του πελάτη που αφορούν την εφαρμογή BookYourWay έχουν ως εξής:

1. Ο χρήστης πληκτρολογώντας την ηλεκτρονική διεύθυνση της εφαρμογής πρέπει να οδηγείται στην αρχική σελίδα.
2. Ο χρήστης θα μπορεί να δημιουργήσει λογαριασμό, έτσι ώστε η εφαρμογή να θυμάται τα στοιχεία του κατά την είσοδό του.
 - i. Η εφαρμογή θα πρέπει να διατηρεί μια λίστα των λογαριασμών των χρηστών στη βάση δεδομένων της,
3. Η εφαρμογή θα μπορεί να προβάλλει βιβλία τα οποία θα είναι καταχωρημένα στη βάση δεδομένων της,
4. Ο χρήστης - μαθητής θα μπορεί να επιλέξει το βιβλίο που επιθυμεί να διαβάσει,
5. Ο χρήστης - μαθητής θα μπορεί να διατηρεί λίστα με αγαπημένα,
6. Ο χρήστης - μαθητής θα μπορεί να κάνει αναζήτηση βιβλίου με διάφορες μεθόδους όπως: τίτλο, όνομα συγγραφέα, κατηγορία, φράση ή λέξη κλειδί,
 - i. Η εφαρμογή θα πρέπει να προβάλλει τα βιβλία και λεπτομέρειες ως αποτέλεσμα της αναζήτησης,
6. Ο χρήστης - διαχειριστής αφού εισέλθει στο σύστημα, μέσω του λογαριασμού του θα μπορεί να εκτελεί διαχειριστικές εργασίες,
 - i. Ο χρήστης - διαχειριστής θα μπορεί να καταχωρεί/δημιουργεί βιβλία
 - ii. Ο χρήστης - διαχειριστής θα μπορεί να διαγράφει βιβλία
 - iii. Ο χρήστης - διαχειριστής θα μπορεί να ανανεώνει βιβλία.

6.4.2. Επιθυμητά Χαρακτηριστικά

Από τις ανωτέρω απαιτήσεις δημιουργήθηκαν επιθυμητά χαρακτηριστικά (features) τα οποία αποτυπώνονται στον πίνακα που ακολουθεί:

Πίνακας 4: Επιθυμητά Χαρακτηριστικά

Features					
Τίτλος	Κατάταξη	Περιγραφή	Τύπος	Συσχετιζόμενες Ιστορίες από Product Backlog)	Ολοκληρωμένες Ιστορίες από Product Backlog
Ταυτοποίηση χρηστών	4	Ανάπτυξη λειτουργικότητας ταυτοποίησης χρηστών και διαχείρισης λογαριασμών	Απαίτηση	7, 8	7, 8
Αρχική σελίδα	1	Ο χρήστης θα μπορεί να οδηγείται στην αρχική σελίδα της εφαρμογής όποτε το επιθυμεί	Απαίτηση	1	1
Προβολή Βιβλίων	3	Ο χρήστης - μαθητής θα μπορεί να επιλέξει ένα βιβλίο και να έχει πρόσβαση στο περιεχόμενό του.	Απαίτηση	6, 7	6, 7
Διαχείριση Βιβλίων	2	Ο χρήστης - διαχειριστής θα μπορεί να δημιουργεί, να επεξεργάζεται και να διαγράφει βιβλία	Απαίτηση	2, 3, 4, 5	2, 3, 4, 5

Τα επιθυμητά χαρακτηριστικά, με βάση τον παραπάνω πίνακα, είναι σύνθετες απαιτήσεις οι οποίες περιλαμβάνουν πολλές εργασίες. Απαιτείται λοιπόν η περαιτέρω αποσύνθεσή τους σε ιστορίες χρηστών έτσι ώστε να περιγράφεται μία συγκεκριμένη εργασία που πρέπει να εκτελεστεί. Επομένως ένα επιθυμητό χαρακτηριστικό μπορεί να αποτελείται από πολλές ιστορίες χρηστών οι οποίες αντίστοιχα περιλαμβάνουν μία και μόνο εργασία.

6.4.3. Βασικό Σχέδιο Έργου

Το στάδιο αυτό ολοκληρώνεται με τη σύνταξη ενός εγγράφου που ονομάζεται **Βασικό Σχέδιο Έργου** (Baseline Project Plan) και κοινοποιείται, βάσει του σεναρίου, στον ιδιοκτήτη της ALTA Designs. Η διαδικασία αυτή δεν αποτελεί μέρος της μεθοδολογίας Scrum, ωστόσο δίνεται η δυνατότητα σε μέλη της διοίκησης, τα οποία δε συμμετέχουν στο έργο, να γνωρίζουν τι αναμένεται από την επιτυχή ολοκλήρωση του έργου.

Το έγγραφο χωρίζεται σε τέσσερις ενότητες:

- *Ενότητα 1^η – Εισαγωγή:* Η εισαγωγή περιγράφει συνοπτικά το περιεχόμενο του Βασικού Σχεδίου Έργου. Χωρίζεται σε δύο μέρη: α) την επισκόπηση του έργου και β) την πρόταση
- *Ενότητα 2^η – Περιγραφή Συστήματος:* Στην ενότητα αυτή περιγράφονται οι κύριες λειτουργίες του συστήματος.
- *Ενότητα 3^η – Μελέτη Σκοπιμότητας:* Στη μελέτη σκοπιμότητας γίνεται αναφορά στα οικονομικά οφέλη του έργου, τις τεχνικές δυσκολίες, τις νομικές διευθετήσεις, τις λειτουργικές ρυθμίσεις καθώς και στο χρονοδιάγραμμα του έργου.
- *Ενότητα 4^η – Διαχειριστικά Θέματα:* Στην ενότητα αυτή περιγράφονται ζητήματα που αφορούν στη διαχείριση αλλά και τη μεθοδολογία που θα ακολουθηθεί κατά τη διάρκεια του έργου.

Παρακάτω παρουσιάζεται το Βασικό Σχέδιο Έργου για τους σκοπούς της διπλωματικής εργασίας:

Πίνακας 5: Βασικό Σχέδιο Έργου

Βασικό Σχέδιο Έργου	
Όνομα Έργου:	BookYourWay
Εισαγωγή:	
<u>Επισκόπηση Έργου</u>	
	Με το παρόν έργο θα αναπτυχθεί μία νέα διαδικτυακή εφαρμογή διαχείρισης εκπαιδευτικών συγγραμμάτων. Οι παρεχόμενες υπηρεσίες της εφαρμογής απευθύνονται σε μαθητές της

πρωτοβάθμιας ή δευτεροβάθμιας εκπαίδευσης. Η εφαρμογή στοχεύει στην επίλυση των κάτωθι θεμάτων των χρηστών/μαθητών:

- Αυξημένες σχολικές απαιτήσεις
- Δυσκολία κατανόησης της σχολικής ύλης
- Αδυναμία παρακολούθησης και αφομοίωσης της διδασκαλίας
- Πιθανή έλλειψη μεταδοτικότητας εκπαιδευτικών
- Διαφορετικό μορφωτικό και διανοητικό επίπεδο
- Έλλειψη διαθέσιμου χρόνου

Προτείνεται

Η εφαρμογή προτείνεται να παρέχει στους χρήστες τα κάτωθι:

- Τη δυνατότητα πρόσβασης στα ειδικά διαμορφωμένα ηλεκτρονικά σχολικά βιβλία,
- Τη δυνατότητα αναζήτησης εννοιών, ορισμών, λέξεων κλειδιών, ερωτήσεων/ασκήσεων ή και ολόκληρου κειμένου από μια βάση δεδομένων σχολικών συγγραμμάτων,
- Τη δυνατότητα δημιουργίας/καταχώρησης, ανάγνωσης, ενημέρωσης, και καταστροφής ολόκληρων βιβλίων ή μέρους αυτών,
- Τη δυνατότητα παρουσίασης της πληροφορίας με μοναδικό και χρηστικό τρόπο.

Περιγραφή Συστήματος:

Η εφαρμογή διαχείρισης εκπαιδευτικών συγγραμμάτων θα αναπτυχθεί χρησιμοποιώντας την αντικειμενοστρεφή γλώσσα προγραμματισμού Ruby. Συγκεκριμένα θα χρησιμοποιηθεί το Ruby on Rails, ένα πλαίσιο ανάπτυξης λογισμικού Ιστού ανοιχτού κώδικα για τη γλώσσα προγραμματισμού Ruby. Η εφαρμογή χρειάζεται να επικοινωνεί με μια βάση δεδομένων. Η βάση δεδομένων που θα χρησιμοποιηθεί ανήκει στις «μη σχεσιακές» βάσεις και ονομάζεται MongoDB. Ταυτόχρονα η εφαρμογή προκειμένου να μπορεί να κάνει αναζήτηση ολόκληρου κειμένου θα πρέπει να συνδέεται και να συνεργάζεται με ένα σύστημα αναζήτησης. Το σύστημα αναζήτησης που χρησιμοποιείται ονομάζεται Elasticsearch.

Οποιοσδήποτε χρήστης είναι συνδεδεμένος στο διαδίκτυο θα μπορεί να προσπελάσει την ιστοσελίδα της εφαρμογής. Η αρχική σελίδα θα αναφέρει γενικές πληροφορίες σχετικά με την εφαρμογή και τους στόχους της. Με σχετικό σύνδεσμο ο χρήστης θα μπορεί να μαθαίνει περισσότερα στοιχεία. Επίσης από την αρχική σελίδα θα δίνεται η δυνατότητα εγγραφής χρήστη αλλά και εισόδου στην εφαρμογή. Για την εγγραφή του χρήστη θα συμπληρώνεται ειδική φόρμα με απαραίτητα και προαιρετικά πεδία. Για την είσοδο στην εφαρμογή επίσης

θα πρέπει να συμπληρώνεται ειδική φόρμα με δύο πεδία: α) το όνομα και β) το αναγνωριστικό χρήστη. Επισημαίνεται ότι για να χρησιμοποιηθεί η εφαρμογή ο χρήστης θα πρέπει να είναι εγγεγραμμένος.

Κατά την είσοδό του στην εφαρμογή ο εγγεγραμμένος χρήστης θα μεταφέρεται στον προσωπικό του χώρο όπου θα είναι ειδικά διαμορφωμένος ανάλογα με τα στοιχεία που έχει δηλώσει κατά το στάδιο εγγραφής του. Εκεί θα του δίνεται η δυνατότητα προβολής των σχολικών συγγραμμάτων με τρόπο απλό και εύχρηστο. Συγκεκριμένα όταν θα επιλέγει για προβολή ένα βιβλίο θα παρουσιάζεται απευθείας το πρώτο κεφάλαιο του βιβλίου. Σε ειδικά διαμορφωμένο πλαίσιο δεξιά του κειμένου θα υπάρχουν σύνδεσμοι με τους τίτλους των κεφαλαίων και ενοτήτων του βιβλίου έτσι ώστε ο χρήστης να μπορεί να πλοηγηθεί εύκολα και γρήγορα στην ενότητα που επιθυμεί.

Ταυτόχρονα ο εγγεγραμμένος χρήστης θα μπορεί να αναζητήσει μέσω μιας φόρμας αναζήτησης οποιοδήποτε ηλεκτρονικό βιβλίο είναι καταχωρημένο στη βάση δεδομένων της εφαρμογής. Έτσι συμπληρώνοντας το αντίστοιχο πεδίο με τον τίτλο, το όνομα, τη λέξη, τη φράση ή και συνδυασμό αυτών, θα πραγματοποιείται αναζήτηση από την εφαρμογή. Αφού επεξεργαστεί τα στοιχεία που έχουν ζητηθεί απαντά στον χρήστη επιστρέφοντας συνδέσμους με τα βιβλία που ικανοποιούν τα κριτήρια αναζήτησης. Η αναζήτηση μπορεί να είναι καθολική, σε ολόκληρο το πλήθος των συλλογών ή και περιορισμένη, για παράδειγμα σε συγκεκριμένη σχολική χρονιά ή κατηγορία μαθημάτων.

Ο διαχειριστής κατά την είσοδό του στην εφαρμογή θα μεταφέρεται σε ειδικά διαμορφωμένη σελίδα διαχείρισης (Dashboard). Εκεί θα του δίνεται η δυνατότητα να εκτελεί διαχειριστικές εργασίες, δηλαδή θα μπορεί να καταχωρεί/δημιουργεί, να διαγράφει και να ανανεώνει βιβλία. Επίσης θα μπορεί να διαχειρίζεται τους λογαριασμούς των χρηστών.

Άλλα χαρακτηριστικά της εφαρμογής είναι τα κάτωθι:

- Οι οθόνες για όλους τους χρήστες και όλες τις λειτουργίες είναι απόλυτα ομοιογενείς και μοναδικές
- Κάθε διαδικασία εισαγωγής δεδομένων υποστηρίζεται από τυποποιημένες φόρμες
- Η εφαρμογή χρησιμοποιεί γραφικό περιβάλλον αλληλεπίδρασης, δηλαδή διάφορα GUI objects που έχουν καθιερωθεί διεθνώς.

Μελέτη Σκοπιμότητας:

A. Οικονομική Ανάλυση

Τα οικονομικά οφέλη της εφαρμογής είναι κυρίως άυλα όπως:

- Εξοικονόμηση χρόνου για τους χρήστες
- Αύξηση της ικανοποίησης τους από τη χρήση της εφαρμογής

Β. Τεχνική Ανάλυση

Η ομάδα υλοποίησης, διαθέτει τις απαραίτητες γνώσεις σε θέματα διαχείρισης έργου, ανάλυσης συστημάτων, αντικειμενοστρεφούς προγραμματισμού και σχεδιασμού ιστοσελίδων. Οι στόχοι της εφαρμογής μπορούν να επιτευχθούν, επιστρατεύοντας αυτές τις ικανότητες.

Γ. Ανάλυση Λειτουργικότητας

Η προτεινόμενη εφαρμογή θα παρέχει:

- Γρήγορη πρόσβαση και εύκολη πλοήγηση στα ηλεκτρονικά σχολικά βιβλία,
- Αποτελεσματική αναζήτηση συγκεκριμένου βιβλίου, κεφαλαίου, ενότητας, παραγράφου, ορισμού, ερώτησης/άσκησης ή και μεμονωμένης λέξης,
- Δυνατότητα δημιουργίας μιας βάσης δεδομένων σχολικών συγγραμμάτων,
- Παροχή βοήθειας στους μαθητές στην εκπλήρωση των μαθητικών τους υποχρεώσεων.

Δ. Έλεγχος Νομιμότητας

Η εφαρμογή δεν περιέχει υλικό το οποίο χρειάζεται ειδική άδεια προτού χρησιμοποιηθεί. Το λογισμικό είναι ανοιχτού κώδικα και τα σχολικά συγγράμματα διακινούνται ελεύθερα στο Διαδίκτυο, κυρίως σε μορφή pdf.

Ε. Χρονοδιάγραμμα έργου

Ο συνολικός χρόνος που απαιτείται για την υλοποίηση του έργου είναι 6 μήνες.

Διαχειριστικά Θέματα:

Η μεθοδολογία που θα χρησιμοποιηθεί για την υλοποίηση του έργου ονομάζεται Scrum. Κάθε δραστηριότητα πρέπει να υλοποιηθεί σε καθορισμένο χρόνο έτσι ώστε η παρούσα διπλωματική να παραδοθεί στην ώρα της. Ο Scrum Master είναι υπεύθυνος για την τήρηση των διαδικασιών και την ανάπτυξη της εφαρμογής.

6.5. Στάδιο 3^ο – Προγραμματισμός Εκδόσεων

Το επόμενο βήμα, μετά την καταγραφή των επιθυμητών χαρακτηριστικών είναι η δημιουργία του Product Backlog από τον Product Owner.

6.5.1. Product Backlog

Το Product Backlog αποτελείται από ιστορίες χρηστών οι οποίες ιεραρχούνται σε μια λίστα ανεκτέλεστων εργασιών. Η κατάταξή τους γίνεται ανάλογα με την προτεραιότητά τους. Η λίστα αυτή ενημερώνεται και αναθεωρείται κατά τη διάρκεια ανάπτυξης της εφαρμογής, εφόσον υπάρχουν νέα πιο λεπτομερή στοιχεία.

Ο παρακάτω πίνακας παρουσιάζει το Product Backlog της εφαρμογής:

Πίνακας 6: Product Backlog

Product Backlog						
A/A	Τίτλος Ιστορίας	Κατάταξη	Περιγραφή	Έναρξη	Λήξη	Feature
1	Αρχική Σελίδα	1	Ως: Χρήστης Επιθυμώ να: Βρεθώ στην αρχική σελίδα Έτσι ώστε: Να γνωρίζω ότι βρίσκομαι στο σωστό ιστότοπο Όταν: Πληκτρολογώ http://www.bookyourway.gr	1/7/2015	2/7/2015	Αρχική σελίδα
2	Καταχώρηση Βιβλίων	2	Ως: Διαχειριστής Επιθυμώ να: Καταχωρώ βιβλία Έτσι ώστε: Να καταχωρούνται στη βάση δεδομένων Όταν: Επιλέγω το σύνδεσμο "Καταχώρηση"	3/7/2015	7/7/2015	Διαχείριση Βιβλίων

3	Προβολή όλων των Βιβλίων	3	Ως: Διαχειριστής Επιθυμώ να: Προβάλλονται όλα τα βιβλία σε πίνακα Έτσι ώστε: Να εκτελώ διαχειριστικές εργασίες Όταν: Επιλέγω το εκάστοτε βιβλίο	8/7/2015	8/7/2015	Διαχείριση Βιβλίων
4	Επεξεργασία Βιβλίων	4	Ως: Διαχειριστής Επιθυμώ να: Επεξεργάζομαι τα βιβλία Έτσι ώστε: Να μπορώ να διορθώσω ή να αλλάξω το υλικό Όταν: Επιλέγω το σύνδεσμο "Επεξεργασία"	9/7/2015	9/7/2015	Διαχείριση Βιβλίων
5	Διαγραφή Βιβλίων	5	Ως: Διαχειριστής Επιθυμώ να: Διαγράψω βιβλία Έτσι ώστε: Να διαγράφονται από τη βάση δεδομένων Όταν: Επιλέγω το σύνδεσμο "Διαγραφή"	10/7/2015	10/7/2015	Διαχείριση Βιβλίων
6	Κατηγοριοποίηση Βιβλίων ανά Σχολική Βαθμίδα και Τάξη	6	Ως: Μαθητής Επιθυμώ να: Κατηγοριοποιούνται τα βιβλία ανά σχολική βαθμίδα και τάξη Έτσι ώστε: Να βρίσκω εύκολα αυτό που με αφορά Όταν: Εισέρχομαι στο λογαριασμό μου	11/7/2015	12/7/2015	Προβολή Βιβλίων
7	Προβολή Περιεχομένου Βιβλίου	7	Ως: Μαθητής Επιθυμώ να: Έχω πρόσβαση στο περιεχόμενο ενός βιβλίου Έτσι ώστε: Να μπορώ να διαβάσω το κεφάλαιο που με ενδιαφέρει Όταν: Επιλέγω το σύνδεσμο "Προβολή"	13/07/2015	13/07/2015	Προβολή Βιβλίων

8	Είσοδος Μαθητή στην Εφαρμογή	8	Ως: Μαθητής Επιθυμώ να: Εισέλθω στην εφαρμογή Έτσι ώστε: Να έχω πρόσβαση στο υλικό της εφαρμογής Όταν: Επιλέγω "Log in"	14/07/201 5	15/07/201 5	Ταυτοποίηση χρηστών
9	Είσοδος Διαχειριστή στην Εφαρμογή	9	Ως: Διαχειριστής Επιθυμώ να: Έχω λογαριασμό διαχειριστή Έτσι ώστε: Να εκτελώ διαχειριστικές εργασίες Όταν: Επιλέγω "Log in"	16/07/201 5	16/07/201 5	Ταυτοποίηση χρηστών

6.5.2. Σχεδιασμός Εκδόσεων

Στη συνέχεια η ομάδα έργου αναλαμβάνει να προγραμματίσει την υλοποίηση της κάθε ιστορίας που περιέχει το Product Backlog σχεδιάζοντας τις εκδόσεις. Οι προγραμματισμένες εκδόσεις για την εφαρμογή BookYourWay είναι οι εξής:

Πίνακας 7: Εκδόσεις

Εκδόσεις			
Έκδοση	Συσχετιζόμενη Ιστορία	Στόχος	Ημ/νία Δημοσίευσης
1.0	Αρχική Σελίδα	Κατασκευή της αρχικής σελίδας της εφαρμογής	2/7/2015
1.1	Καταχώρηση Βιβλίων	Ανάπτυξη λειτουργικότητας για την καταχώρηση βιβλίων	7/7/2015
1.2	Προβολή όλων των Βιβλίων	Δημιουργία όψεως για την την εμφάνιση όλων των βιβλίων σε λίστα	8/7/2015

1.3	Επεξεργασία Βιβλίων	Ανάπτυξη λειτουργικότητας για την επεξεργασία βιβλίων	9/7/2015
1.4	Διαγραφή Βιβλίων	Ανάπτυξη λειτουργικότητας για τη διαγραφή βιβλίων	10/7/2015
1.5	Κατηγοριοποίηση Βιβλίων ανά Σχολική Βαθμίδα και Τάξη	Δημιουργία όψεων για την κατηγοριοποίηση των βιβλίων ανά σχολική βαθμίδα και τάξη	12/7/2015
1.6	Προβολή Περιεχομένου Βιβλίου	Δημιουργία όψεως για την εμφάνιση του περιεχομένου κάθε βιβλίου	13/7/2015
1.7	Είσοδος Μαθητή στην Εφαρμογή	Ανάπτυξη λειτουργικότητας ταυτοποίησης χρήστη-μαθητή και δημιουργία κατάλληλων όψεων	15/07/2015
1.8	Είσοδος Διαχειριστή στην Εφαρμογή	Ανάπτυξη λειτουργικότητας ταυτοποίησης χρήστη-διαχειριστή και δημιουργία κατάλληλων όψεων	16/07/2015

6.6. Στάδιο 4^ο – Σχεδιασμός Sprint

Κάθε έκδοση για να υλοποιηθεί χωρίζεται σε επιμέρους εργασίες, δηλαδή σε Sprints. Τα Sprints που πρέπει να εκτελεστούν για κάθε προγραμματισμένη έκδοση καταγράφονται στο Sprint Backlog. Ο παρακάτω πίνακας αποτυπώνει το Sprint Backlog της εφαρμογής:

Πίνακας 8: Sprint Backlog

Sprint Backlog					
Sprint	Συσχετιζόμενη Ιστορία	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.0.1	Αρχική Σελίδα	Σκελετός Εφαρμογής	Δημιουργία απαραίτητων αρχείων, πάνω στα οποία στηρίζεται η εφαρμογή	Ολοκληρώθηκε	1
1.0.2	Αρχική Σελίδα	Σύνδεση με Βάση Δεδομένων	Σύνδεση της εφαρμογής με τη Βάση δεδομένων MongoDB, εγκατάσταση του πακέτου Mongoid	Ολοκληρώθηκε	1
1.0.3	Αρχική Σελίδα	Εγκατάσταση πακέτου bootstrap-sass	Παροχή front-end πλαισίου ανάπτυξης ιστοσελίδων	Ολοκληρώθηκε	1
1.0.4	Αρχική Σελίδα	Εγκατάσταση πακέτου font-awesome-rails	Παροχή εικονογραφικής γραμματοσειράς font awesome	Ολοκληρώθηκε	1
1.0.5	Αρχική Σελίδα	Εγκατάσταση πακέτου Haml	Παρέχει τη δυνατότητα γραφής κώδικα HTML με πιο γρήγορο τρόπο	Ολοκληρώθηκε	1
1.0.6	Αρχική Σελίδα	Δημιουργία controller και view για τις στατικές σελίδες	Δημιουργία του controller Pages και της μεθόδου home	Ολοκληρώθηκε	1
1.0.7	Αρχική Σελίδα	Συγγραφή κώδικα αρχικής σελίδας	Συγγραφή του κώδικα σε haml βάσει του πλαισίου Bootstrap	Ολοκληρώθηκε	2
1.1.1	Καταχώρηση Βιβλίων	Καθορισμός οντοτήτων	Κατηγοριοποίηση των αντικειμένων ενδιαφέροντος και των σχέσεων μεταξύ τους	Ολοκληρώθηκε	8
1.1.2	Καταχώρηση Βιβλίων	Δημιουργία του πόρου Book	Δημιουργία του μοντέλου δεδομένων Book και της διεπαφής για τη διαχείρισή του	Ολοκληρώθηκε	2

1.1.3	Καταχώρηση Βιβλίων	Δημιουργία μοντέλου Publisher	Δημιουργία του μοντέλου δεδομένων Publisher	Ολοκληρώθηκε	1
1.1.4	Καταχώρηση Βιβλίων	Δημιουργία μοντέλου Author	Δημιουργία του μοντέλου δεδομένων Author	Ολοκληρώθηκε	1
1.1.5	Καταχώρηση Βιβλίων	Δημιουργία μοντέλου Preface	Δημιουργία του μοντέλου δεδομένων Preface	Ολοκληρώθηκε	1
1.1.6	Καταχώρηση Βιβλίων	Δημιουργία μοντέλου Preface Paragraph	Δημιουργία του μοντέλου δεδομένων Preface Paragraph	Ολοκληρώθηκε	1
1.1.7	Καταχώρηση Βιβλίων	Δημιουργία μοντέλου School Grade	Δημιουργία του μοντέλου δεδομένων School Grade	Ολοκληρώθηκε	1
1.1.8	Καταχώρηση Βιβλίων	Δημιουργία μοντέλου Chapter	Δημιουργία του μοντέλου δεδομένων Chapter	Ολοκληρώθηκε	1
1.1.9	Καταχώρηση Βιβλίων	Δημιουργία μοντέλου Section	Δημιουργία του μοντέλου δεδομένων Section	Ολοκληρώθηκε	1
1.1.10	Καταχώρηση Βιβλίων	Δημιουργία μοντέλου Paragraph	Δημιουργία του μοντέλου δεδομένων Paragraph	Ολοκληρώθηκε	1
1.1.11	Καταχώρηση Βιβλίων	Προσθήκη Admin namespace	Δημιουργία ξεχωριστού route για την διαχείριση των βιβλίων	Ολοκληρώθηκε	1
1.1.12	Καταχώρηση Βιβλίων	Παραμετροποίηση του books_controller.rb	Προσθήκη κώδικα στη μέθοδο new και στη μέθοδο book_params	Ολοκληρώθηκε	2
1.1.13	Καταχώρηση Βιβλίων	Διαμόρφωση της όψεως books/new.html.haml	Συγγραφή κώδικα της σελίδας	Ολοκληρώθηκε	4
1.2.1	Προβολή όλων των Βιβλίων	Διαμόρφωση της όψεως books/index.html.haml	Συγγραφή κώδικα σελίδας	Ολοκληρώθηκε	2
1.3.1	Επεξεργασία Βιβλίων	Διαμόρφωση όψεως books/edit.html.haml	Συγγραφή κώδικα σελίδας	Ολοκληρώθηκε	2
1.3.2	Επεξεργασία Βιβλίων	Παραμετροποίηση όψεως books/index.html.haml	Δημιουργία συνδέσμου για την επεξεργασία του εκάστοτε βιβλίου	Ολοκληρώθηκε	1

1.4.1	Διαγραφή Βιβλίων	Παραμετροποίηση όψεως books/index.html.haml	Δημιουργία συνδέσμου για διαγραφή του εκάστοτε βιβλίου	Ολοκληρώθηκε	1
1.5.1	Κατηγοριοποίηση Βιβλίων	Δημιουργία Ελεγκτή student/books_controller.rb	Δημιουργία Ελεγκτή με δύο μεθόδους: index και show	Ολοκληρώθηκε	1
1.5.2	Κατηγοριοποίηση Βιβλίων	Προσθήκη Student namespace στο routes.rb	Δημιουργία ξεχωριστού route για την κατηγοριοποίηση και προβολή των βιβλίων	Ολοκληρώθηκε	1
1.5.3	Κατηγοριοποίηση Βιβλίων	Παραμετροποίηση Ελεγκτή student/books_controller.rb	Δημιουργία του module Student	Ολοκληρώθηκε	1
1.5.4	Κατηγοριοποίηση Βιβλίων	Διαμόρφωση Όψεως student/books/index.html.haml	Συγγραφή κώδικα της σελίδας	Ολοκληρώθηκε	4
1.5.5	Κατηγοριοποίηση Βιβλίων	Παραμετροποίηση Ελεγκτή student/books_controller.rb	Τροποποίηση της μεθόδου index	Ολοκληρώθηκε	2
1.6.1	Προβολή Περιεχομένου Βιβλίου	Δημιουργία Όψεως student/books/show.html.haml	Προβολή του περιεχομένου κάθε βιβλίου	Ολοκληρώθηκε	3
1.7.1	Είσοδος Μαθητή στην Εφαρμογή	Εγκατάσταση πακέτου 'devise'	Παροχή λειτουργίας ταυτοποίησης των χρηστών	Ολοκληρώθηκε	1
1.7.2	Είσοδος Μαθητή στην Εφαρμογή	Δημιουργία μοντέλου User	Δημιουργία του μοντέλου δεδομένων User, υπεύθυνο για την αποθήκευση δεδομένων που αφορούν στον λογαριασμό του χρήστη μαθητή, στη βάση	Ολοκληρώθηκε	1
1.7.3	Είσοδος Μαθητή στην Εφαρμογή	Δημιουργία Student Controller	Ορισμός προσβάσεων μαθητή	Ολοκληρώθηκε	1

1.7.4	Είσοδος Μαθητή στην Εφαρμογή	Παραμετροποίηση Ελεγκτή controllers/student	Ορισμός προσβάσεων μαθητή	Ολοκληρώθηκε	1
1.7.5	Είσοδος Μαθητή στην Εφαρμογή	Διαμόρφωση Όψεως devise /registrations /new.html.haml	Διαμόρφωση της όψεως που αφορά στη φόρμα εγγραφής του χρήστη στην εφαρμογή	Ολοκληρώθηκε	2
1.7.6	Είσοδος Μαθητή στην Εφαρμογή	Διαμόρφωση Όψεως devise/sessions/new.html.haml	Διαμόρφωση της όψεως που αφορά στη φόρμα εισόδου του χρήστη στην εφαρμογή	Ολοκληρώθηκε	2
1.7.7	Είσοδος Μαθητή στην Εφαρμογή	Διαμόρφωση Όψεως devise/passwords/new.html.haml	Διαμόρφωση της όψεως που αφορά στη φόρμα επανέκδοσης κωδικού πρόσβασης των χρηστών	Ολοκληρώθηκε	1
1.8.1	Είσοδος Διαχειριστή στην Εφαρμογή	Δημιουργία μοντέλου Administrator	Δημιουργία του μοντέλου δεδομένων Administrator, υπεύθυνο για την αποθήκευση δεδομένων που αφορούν στον λογαριασμό του διαχειριστή, στη βάση	Ολοκληρώθηκε	1
1.8.2	Είσοδος Διαχειριστή στην Εφαρμογή	Δημιουργία Admin Controller	Ορισμός προσβάσεων διαχειριστή	Ολοκληρώθηκε	1
1.8.3	Είσοδος Διαχειριστή στην Εφαρμογή	Παραμετροποίηση Ελεγκτή controllers/admin/books_controller.rb	Ορισμός προσβάσεων διαχειριστή	Ολοκληρώθηκε	1

6.7. Στάδιο 5^ο – Καθημερινές Συναντήσεις

Τα μέλη της ομάδας, βάσει του σεναρίου, συναντιούνται καθημερινά για να παρακολουθήσουν την εξέλιξη των εργασιών. Το αποτέλεσμα των συναντήσεων αυτών είναι η σταδιακή υλοποίηση της εφαρμογής, εκτελώντας τις εργασίες που έχουν οριστεί για κάθε έκδοση και η αντιμετώπιση των εμποδίων που προκύπτουν.

6.8. Υλοποίηση Εκδόσεων

Σε αυτή την ενότητα υλοποιούνται οι προγραμματισμένες εκδόσεις και εκτελούνται οι εργασίες τους με στόχο να παραδοθούν σταδιακά τα λειτουργικά κομμάτια της εφαρμογής.

6.8.1. Έκδοση 1.0 – Αρχική Σελίδα

Από το Product Backlog (πίνακας 6) της εφαρμογής BookYourWay, η πρώτη ιστορία χρήστη που πρέπει να υλοποιηθεί, βάσει της σειράς κατάταξης είναι η «Αρχική Σελίδα». Τα sprints που πρέπει να εκτελεστούν είναι τα ακόλουθα:

Πίνακας 9: Έκδοση 1.0 – Αρχική Σελίδα

Έκδοση 1.0				
Ιστορία: Αρχική Σελίδα				
Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.0.1	Σκελετός Εφαρμογής	Δημιουργία απαραίτητων αρχείων, πάνω στα οποία στηρίζεται η εφαρμογή	Ολοκληρώθηκε	1
1.0.2	Σύνδεση με Βάση Δεδομένων	Σύνδεση της εφαρμογής με τη Βάση δεδομένων MongoDB, εγκατάσταση του πακέτου Mongoid	Ολοκληρώθηκε	1
1.0.3	Εγκατάσταση πακέτου bootstrap-sass	Παροχή front-end πλαισίου ανάπτυξης ιστοσελίδων	Ολοκληρώθηκε	1
1.0.4	Εγκατάσταση πακέτου font-awesome-rails	Παροχή εικονογραφικής γραμματοσειράς font awesome	Ολοκληρώθηκε	1

1.0.5	Εγκατάσταση πακέτου Haml	Παρέχει τη δυνατότητα γραφής κώδικα HTML με πιο γρήγορο τρόπο	Ολοκληρώθηκε	1
1.0.6	Δημιουργία controller και view για τις στατικές σελίδες	Δημιουργία του controller Pages και της μεθόδου home	Ολοκληρώθηκε	1
1.0.7	Συγγραφή κώδικα αρχικής σελίδας	Συγγραφή του κώδικα σε haml βάσει του πλαισίου Bootstrap	Ολοκληρώθηκε	2

6.8.1.1. Sprint 1.0.1 – Σκελετός Εφαρμογής

Η ανάπτυξη της εφαρμογής γίνεται σε περιβάλλον Linux, συγκεκριμένα χρησιμοποιείται μία από τις πιο δημοφιλείς διανομές η Ubuntu 14.04. Θεωρείται δεδομένο ότι για την ανάπτυξη της εφαρμογής έχουν εγκατασταθεί διάφορα προαπαιτούμενα πακέτα και βιβλιοθήκες όπως:

- η Ruby έκδοσης 2.2.2,
- το Ruby on Rails έκδοσης 4.2.3,
- η MongoDB έκδοσης 3.0.5
- η Elasticsearch έκδοσης,

για τα οποία δε θα γίνει ανάλυση της διαδικασίας εγκατάστασης και των παραμετροποιήσεων που ενδεχομένως να χρειάζονται για την ομαλή και απρόσκοπτη λειτουργία τους.

Για τη δημιουργία του σκελετού της εφαρμογής BookYourWay, εκτελείται η εντολή:

```
~$ rails new bookyourway --skip-active-record
```

Η εντολή rails new δημιουργεί ένα νέο φάκελο με το όνομα bookyourway και μέσα σε αυτό το φάκελο διαμορφώνονται τα αρχεία που δομούν την ομώνυμη εφαρμογή. Το --skip-active-record είναι μια σημαντική παράμετρος, διότι παραβλέπει τον σχηματισμό του πακέτου ActiveRecord το οποίο χρησιμοποιείται για την υλοποίηση μιας Αντικειμενοστρεφούς Σχεσιακής Αντιστοίχισης⁷ (ORM – Object Relational

⁷ Αντικειμενοστρεφή Σχεσιακή Αντιστοίχιση (ORM – Object Relational Mapping) είναι μια τεχνική προγραμματισμού για τη μετατροπή δεδομένων μεταξύ ασύμβατων συστημάτων και πιο

Mapping). Εφόσον η εφαρμογή θα χρησιμοποιήσει μη σχεσιακή βάση δεδομένων το συγκεκριμένο πακέτο πρέπει να παραληφθεί από την αρχική εγκατάσταση.

Οι κατάλογοι και τα αρχεία που δημιουργήθηκαν κατά την εκτέλεση της ανωτέρω εντολής είναι τα εξής:

bookyourway	
app	
assets	# Περιέχει αρχεία CSS, Javascript και εικόνες που χρησιμοποιεί η εφαρμογή
controllers	# Περιέχει αρχεία που αφορούν τις όψεις
helpers	# Περιέχει βοηθητικά αρχεία που αφορούν τις όψεις και τους ελεγκτές
mailers	# Περιέχει αρχεία που αφορούν τη διαχείριση των emails
models	# Περιέχει αρχεία που αφορούν τα μοντέλα
views	# Περιέχει αρχεία που αφορούν τις όψεις
bin	# Περιέχει δυαδικά εκτελέσιμα αρχεία
config	# Περιέχει αρχεία ρυθμίσεων της εφαρμογής
config.ru	# Αρχείο ρυθμίσεων του Rack middleware
db	# Περιέχει αρχεία διαχείρισης της βάσης δεδομένων
Gemfile	# Αρχείο διαχείρισης διαφόρων πακέτων λογισμικού
Gemfile.lock	# Λίστα με τα πακέτα λογισμικού που χρησιμοποιούνται
lib	# Περιέχει τις βιβλιοθήκες της εφαρμογής
log	# Περιέχει αρχεία με καταγεγραμμένα λάθη της εφαρμογής
public	# Περιέχει αρχεία προσβάσιμα στο ευρύ κοινό, όπως οι σελίδες λάθους
Rakefile	# Αρχείο που βοηθάει στην ανάπτυξη, τη δημιουργία πακέτων και στις δοκιμές του Ruby on Rails
README.rdoc	# Αρχείο το οποίο περιέχει πληροφορίες για την εφαρμογή

συγκεκριμένα μεταξύ μιας σχεσιακής βάσης δεδομένων και ενός συστήματος που στηρίζεται σε αντικειμενοστρεφή γλώσσα προγραμματισμού.

test	# Περιέχει αρχεία που εξυπηρετούν τις δοκιμές και τους ελέγχους της εφαρμογής
tmp	# Περιέχει προσωρινά αρχεία
vendor	# Περιέχει βιβλιοθήκες που χρειάζονται για τρίτες εφαρμογές

6.8.1.2. Sprint 1.0.2 – Σύνδεση με Βάση Δεδομένων

Η βάση δεδομένων που χρησιμοποιεί η εφαρμογή για τη συλλογή, οργάνωση και μοντελοποίηση των δραστηριοτήτων της είναι η MongoDB.

Για να μπορέσει η MongoDB να συνδεθεί με μια RoR εφαρμογή, όπως είναι η BookYourWay, θα πρέπει να εγκατασταθεί ένα πακέτο λογισμικού (gem) που ονομάζεται Mongoid. Το Mongoid είναι ένας Απεικονιστής Αντικειμένου – Εγγράφου (Object Document Mapper), δηλαδή ρυθμίζει την αντιστοίχιση ενός αντικειμένου σε ένα έγγραφο της MongoDB. Ουσιαστικά αντικαθιστά τις λειτουργίες του ActiveRecord.

Για την εγκατάσταση του Mongoid προστίθεται στο αρχείο Gemfile της εφαρμογής η παρακάτω γραμμή:

```
gem 'mongoid', '~> 4.0.0'
```

Στη συνέχεια, προκειμένου να γίνει η εγκατάσταση, εκτελείται η εντολή:

```
~$ bundle install
```

Οποιαδήποτε ρύθμιση του Mongoid γίνεται από το αρχείο mongoid.yml που βρίσκεται στο φάκελο config. Για να δημιουργηθεί αυτό το αρχείο εκτελείται η εντολή:

```
~$ rails g mongoid:config
```

6.8.1.3. Sprint 1.0.3 – Εγκατάσταση πακέτου bootstrap-sass

Το Bootstrap είναι ένα πλαίσιο ανοιχτού κώδικα για τη γρήγορη ανάπτυξη ιστοσελίδων. Παρέχει ένα σύνολο CSS στυλ για όλα τα βασικά στοιχεία HTML έτσι ώστε να επιτυγχάνεται μια ενιαία και σύγχρονη εμφάνιση. Ταυτόχρονα διαθέτει JavaScript στοιχεία παρέχοντας πρόσθετη διεπαφή χρήστη με παράθυρα διαλόγου, επεξηγήσεις και καρουσέλ. Επίσης περιέχει διάφορα στοιχεία περιβάλλοντος όπως κουμπιά ετικέτες, εικονίδια, προειδοποιητικά μηνύματα κ.α.

Αρχικά για την εγκατάσταση του bootstrap-sass προστίθεται στο αρχείο Gemfile της εφαρμογής η παρακάτω γραμμή:

```
gem 'bootstrap-sass'
```

Στη συνέχεια, εκτελείται η εντολή:

```
~$ bundle install
```

Για να χρησιμοποιηθούν τα στοιχεία του πακέτου Bootstrap θα πρέπει να τροποποιηθεί το αρχείο app/assets/stylesheets/application.css. Έτσι, πρώτα μετονομάζεται το αρχείο από application.css σε application.scss με την παρακάτω εντολή:

```
~$ mv app/assets/stylesheets/application.css  
app/assets/stylesheets/application.scss
```

Στη συνέχεια αντικαθίσταται το περιεχόμενο του αρχείου με τις παρακάτω γραμμές κώδικα:

```
@import "bootstrap-sprockets";  
@import "bootstrap";
```

Επίσης τροποποιείται το αρχείο app/assets/javascripts/application.js προσθέτοντας τη γραμμή:

```
//= require bootstrap
```

6.8.1.4. Sprint 1.0.4 – Εγκατάσταση πακέτου font-awesome-rails

Το πακέτο font-awesome-rails παρέχει τη δυνατότητα στην εφαρμογή να χρησιμοποιήσει την εικονογραφική γραμματοσειρά font awesome η οποία περιέχει 585 εικονίδια/χαρακτήρες. Τα εικονίδια αυτά δεν απαιτούν JavaScript και μπορούν εύκολα να προσαρμόσουν το μέγεθος, το χρώμα και τη σκίαση, εφόσον χρειαστεί, μέσω της CSS.

Για την εγκατάσταση του πακέτου font-awesome-rails προστίθεται στο Gemfile η γραμμή:

```
gem 'font-awesome-rails'
```

Στη συνέχεια, εκτελείται η εντολή:

```
~$ bundle install
```

Επίσης τροποποιείται το αρχείο `app/assets/stylesheets/application.css` και προστίθεται η γραμμή:

```
@import "font-awesome";
```

6.8.1.5. Sprint 1.0.5 – Εγκατάσταση πακέτου Haml

Η Haml (HTML Abstraction Markup Language) είναι μια ελαφριά γλώσσα σήμανσης η οποία παρέχει τη δυνατότητα γραφής κώδικα HTML με πιο γρήγορο τρόπο.

Η διαδικασία εγκατάστασης της Haml είναι όμοια με τις παραπάνω. Επομένως, όπως ισχύει για κάθε gem, προστίθεται στο Gemfile η γραμμή:

```
gem 'haml'
```

Στη συνέχεια, εκτελείται η εντολή:

```
~$ bundle install
```

6.8.1.6. Sprint 1.0.6 – Δημιουργία Controller & View για Στατικές Σελίδες

Για να δημιουργηθεί η αρχική σελίδα της εφαρμογής, θα χρειαστεί ένας ελεγκτής (controller), ο οποίος θα είναι υπεύθυνος για όλες τις στατικές σελίδες και μια όψη (view) που καθορίζει το περιεχόμενο της σελίδας και αντιστοιχεί σε συγκεκριμένη ενέργεια (action) του ελεγκτή. Με μία εντολή μπορούν να δημιουργηθούν και τα δύο.

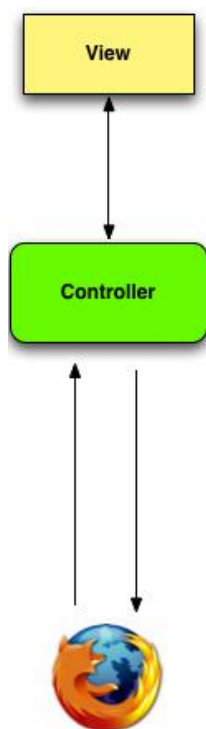
```
~$ rails g Pages home
```

Έτσι στον φάκελο `controllers` υπάρχει πλέον ένα αρχείο που ονομάζεται `pages_controller.rb` το οποίο περιέχει μια κλάση της Ruby και μια μέθοδο που ονομάζεται `home`. Όταν μια μέθοδος ορίζεται μέσα σε έναν ελεγκτή τότε η μέθοδος αυτή αποκαλείται ενέργεια (action), είναι δηλαδή η ενέργεια που ζητείται από τον ελεγκτή να εκτελέσει.

Ταυτόχρονα στον φάκελο `views` έχει δημιουργηθεί φάκελος `pages` μέσα στον οποίο υπάρχει το αρχείο `home.html`. Στο αρχείο αυτό θα γραφεί ο κώδικας που αφορά την παρουσίαση της αρχικής σελίδας σε haml. Επομένως θα πρέπει να μετονομαστεί το αρχείο σε `home.html.haml` εκτελώντας την εντολή:

Έχοντας κατά νου την αρχιτεκτονική MVC, όταν ο χρήστης πληκτρολογεί το URL της εφαρμογής, ο browser στέλνει αίτημα στον controller, μέσω του εξυπηρετητή, για την εμφάνιση της αρχικής σελίδας. Ο controller επεξεργάζεται τις πληροφορίες και στέλνει τα δεδομένα του στο view, το οποίο διαμορφώνει την παρουσίαση. Το view στέλνει την παρουσίαση πίσω στον controller με τη μορφή HTML. Τέλος ο controller διαβιβάζει την HTML σελίδα στον browser για να την προβάλει. Το διάγραμμα που ακολουθεί αποτυπώνει τη διαδικασία αυτή:

Διάγραμμα 8: Αναπαράσταση επικοινωνίας των browser, controller και view.



6.8.2. Έκδοση 1.1 – Καταχώρηση Βιβλίων

Η έκδοση 1.1 της εφαρμογής αφορά την υλοποίηση εκείνης της λειτουργικότητας που επιτρέπει στον διαχειριστή την καταχώρηση βιβλίων. Για την επίτευξη αυτού του σκοπού εκτελούνται τα παρακάτω sprints.

Πίνακας 10: Έκδοση 1.1 – Καταχώρηση Βιβλίων

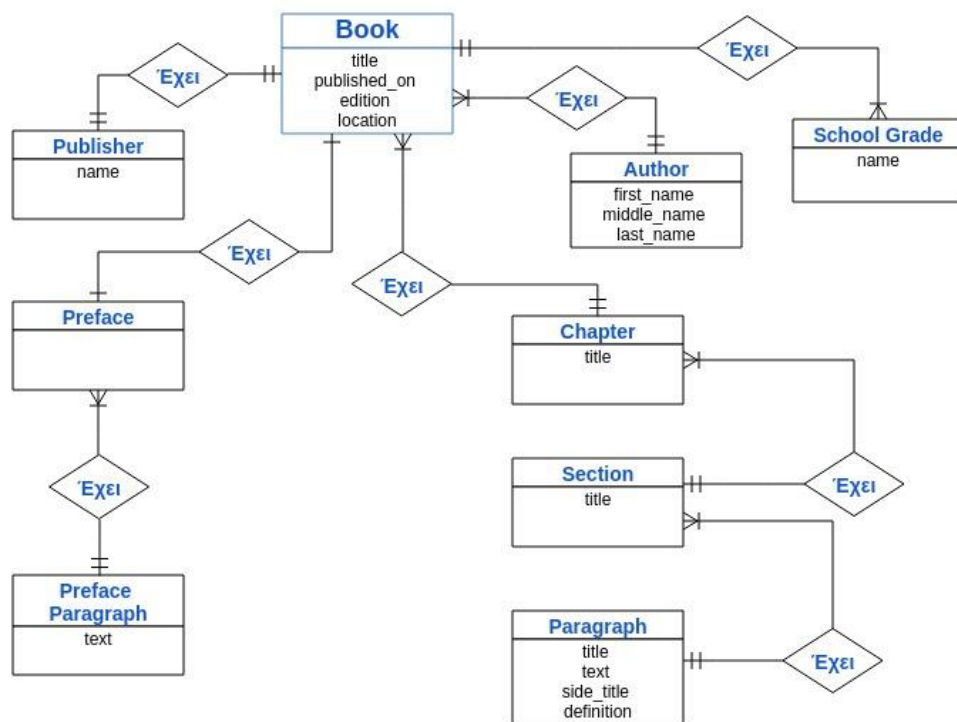
Έκδοση 1.1				
Ιστορία: Καταχώρηση Βιβλίων				
Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.1.1	Καθορισμός οντοτήτων	Κατηγοριοποίηση των αντικειμένων ενδιαφέροντος και των σχέσεων μεταξύ τους	Ολοκληρώθηκε	8
1.1.2	Δημιουργία του πόρου Book	Δημιουργία του μοντέλου δεδομένων Book και της διεπαφής για τη διαχείρισή του	Ολοκληρώθηκε	2
1.1.3	Δημιουργία μοντέλου Publisher	Δημιουργία του μοντέλου δεδομένων Publisher	Ολοκληρώθηκε	1
1.1.4	Δημιουργία μοντέλου Author	Δημιουργία του μοντέλου δεδομένων Author	Ολοκληρώθηκε	1
1.1.5	Δημιουργία μοντέλου Preface	Δημιουργία του μοντέλου δεδομένων Preface	Ολοκληρώθηκε	1
1.1.6	Δημιουργία μοντέλου Preface Paragraph	Δημιουργία του μοντέλου δεδομένων Preface Paragraph	Ολοκληρώθηκε	1
1.1.7	Δημιουργία μοντέλου School Grade	Δημιουργία του μοντέλου δεδομένων School Grade	Ολοκληρώθηκε	1
1.1.8	Δημιουργία μοντέλου Chapter	Δημιουργία του μοντέλου δεδομένων Chapter	Ολοκληρώθηκε	1
1.1.9	Δημιουργία μοντέλου Section	Δημιουργία του μοντέλου δεδομένων Section	Ολοκληρώθηκε	1

1.1.10	Δημιουργία μοντέλου Paragraph	Δημιουργία του μοντέλου δεδομένων Paragraph	Ολοκληρώθηκε	1
1.1.11	Προσθήκη Admin namespace	Δημιουργία ξεχωριστού route για την διαχείριση των βιβλίων	Ολοκληρώθηκε	1
1.1.12	Παραμετροποίηση του books_controller.rb	Προσθήκη κώδικα στη μέθοδο new και στη μέθοδο book_params	Ολοκληρώθηκε	2
1.1.13	Διαμόρφωση της όψεως books/new.html.haml	Συγγραφή κώδικα της σελίδας	Ολοκληρώθηκε	4

6.8.2.1. Sprint 1.1.1 – Καθορισμός Οντοτήτων

Ένα από τα βασικότερα βήματα είναι ο καθορισμός του μοντέλου δεδομένων έτσι ώστε να απεικονιστούν τα αντικείμενα ενδιαφέροντος και οι μεταξύ τους σχέσεις. Στο παρακάτω διάγραμμα οντοτήτων – συσχετίσεων (Ο/Σ) παριστάνονται οι διάφοροι τύποι οντοτήτων της εφαρμογής, οι συνδέσεις και τα χαρακτηριστικά τους.

Διάγραμμα 9: Οντότητες – Συσχετίσεις (Ο/Σ) της Εφαρμογής



6.8.2.2. Sprint 1.1.2 – Δημιουργία Πόρου Book

Ο πόρος Book αποτελείται από το μοντέλο δεδομένων Book και τη διεπαφή που χειρίζεται το μοντέλο αυτό. Για τη δημιουργία του πόρου Book εκτελείται η εντολή:

```
~$ rails g scaffold Book title:string published_on:date edition:string location:string
```

Τα νέα αρχεία που σχηματίζονται είναι το μοντέλο book.rb το οποίο περιέχει την κλάση Book καθώς και τα χαρακτηριστικά του μοντέλου, title, published_on, edition και location, που ορίστηκαν κατά την εκτέλεση της εντολής scaffold.

```
class Book
  include Mongoid::Document

  field :title, type: String
  field :published_on, type: String
  field :edition, type: String
  field :location, type: String

  has_one :publisher, dependent: :destroy
  has_many :authors, dependent: :destroy
  has_one :preface, dependent: :destroy
  has_many :chapters, dependent: :destroy
  belongs_to :school_grade

  accepts_nested_attributes_for :publisher, reject_if: lambda {|attributes|
    attributes['name'].blank?}, allow_destroy: true
  accepts_nested_attributes_for :authors, allow_destroy: true
  accepts_nested_attributes_for :preface, allow_destroy: true
  accepts_nested_attributes_for :chapters, reject_if: lambda {|attributes|
    attributes['title'].blank?}, allow_destroy: true
end
```

Ταυτόχρονα η κλάση book περιέχει τις συσχετίσεις του μοντέλου book με άλλα μοντέλα, όπως αυτές καθορίστηκαν στο διάγραμμα Οντοτήτων – Συσχετίσεων (*Διάγραμμα 9*). Επομένως ένα βιβλίο έχει έναν εκδότη, πολλούς συγγραφείς, έναν πρόλογο, ανήκει σε μια σχολική βαθμίδα και αποτελείται από πολλά κεφάλαια. Η μέθοδος accepts_nested_attributes_for χρησιμοποιείται προκειμένου να επιτρέψει στο

μοντέλο book να αποθηκεύει εγγραφές οι οποίες περιέχουν χαρακτηριστικά των μοντέλων publisher, author, preface και chapter.

Επίσης σχηματίζεται ο ελεγκτής books_controller.rb που περιέχει την κλάση BooksController και τις μεθόδους index, show, new, edit, create, update και destroy.

```
class BooksController < ApplicationController
  before_action :set_book, only: [:show, :edit, :update, :destroy]
  def index
    @books = Book.all
  end
  def show
  end
  def new
    @book = Book.new
  end
  def edit
  end
  def create
    @book = Book.new(book_params)
    respond_to do |format|
      if @book.save
        format.html { redirect_to book_path(@book), notice: 'Book was
          successfully created.' }
        format.json { render :show, status: :created, location: @book }
      else
        format.html { render :new }
        format.json { render json: @book.errors, status:
          :unprocessable_entity }
      end
    end
  end
  def update
    respond_to do |format|
      if @book.update(book_params)
```

```

    format.html { redirect_to book_path(@book), notice: 'Book was
    successfully updated.' }
    format.json { render :show, status: :ok, location: @book }
  else
    format.html { render :edit }
    format.json { render json: @book.errors, status:
      :unprocessable_entity }
  end
end
end
def destroy
  @book.destroy
  respond_to do |format|
    format.html { redirect_to books_url, notice: 'Book was
    successfully destroyed.' }
    format.json { head :no_content }
  end
end
private
def set_book
  @book = Book.find(params[:id])
end
def book_params
  params.require(:book).permit(:title, :published_on, :edition,
    :location)
end
end
end
end

```

Επιπλέον σχηματίζεται ο φάκελος books μέσα στον φάκελο των όψεων ο οποίος περιέχει τα αρχεία edit.html.erb, index.html.erb, new.html.erb και show.html.erb. Τα αρχεία αυτά αντιστοιχούν σε HTML σελίδες που διαμορφώνουν την διεπαφή της εφαρμογής για τη δημιουργία, ανάγνωση, ανανέωση και διαγραφή των αντικειμένων/βιβλίων.

6.8.2.3. Sprint 1.1.3 – Δημιουργία μοντέλου Publisher

Για τη δημιουργία του μοντέλου δεδομένων Publisher εκτελείται η εντολή:

```
~$ rails g model Publisher name:string
```

Έτσι σχηματίζεται το αρχείο publisher.rb μέσα στο φάκελο models το οποίο περιέχει την εξής κλάση:

```
class Publisher
  include Mongoid::Document
  field :name, type: String
  belongs_to :book
end
```

Η κλάση Publisher έχει το χαρακτηριστικό name και ανήκει (belongs_to) στο book. Με τον τρόπο αυτό ολοκληρώνεται η ένα-προς-ένα (1-1) συσχέτιση μεταξύ του μοντέλου Publisher και Book. Η σχέση μεταξύ δύο οντοτήτων είναι αμφίδρομη, δηλαδή, όπως περιεγράφηκε στην ενότητα 6.8.2.2, το βιβλίο έχει ένα εκδότη αλλά και ο εκδότης ανήκει σε ένα βιβλίο.

6.8.2.4. Sprint 1.1.4 – Δημιουργία μοντέλου Author

Με την ίδια διαδικασία δημιουργείται το μοντέλο Author, εκτελώντας δηλαδή την εντολή:

```
~$ rails g model Author first_name:string middle_name:string last_name:string
```

Το αρχείο author.rb περιέχει την κλάση:

```
class Author
  include Mongoid::Document
  field :first_name, type: String
  field :middle_name, type: String
  field :last_name, type: String

  belongs_to :book
end
```

Τα χαρακτηριστικά της κλάσης Author είναι τα first_name, middle name και last_name. Αυτά αποτελούν τα επιθυμητά προς αποθήκευση στοιχεία, για κάθε συγγραφέα, στην βάση δεδομένων, δηλαδή το όνομα, το μεσαίο όνομα (εφόσον υπάρχει) και το επώνυμο. Ταυτόχρονα το μοντέλο Author ανήκει στο μοντέλο Book, επομένως κάθε συγγραφέας ανήκει σε ένα βιβλίο.

6.8.2.5. Sprint 1.1.5 – Δημιουργία μοντέλου Preface

Το μοντέλο Preface δημιουργείται εκτελώντας την εντολή:

```
~$ rails g model Preface
```

Η κλάση που σχηματίζεται μέσα στο αρχείο preface.rb είναι:

```
class Preface
  include Mongoid::Document

  embeds_many :preface_paragraphs
  belongs_to :book
  accepts_nested_attributes_for :preface_paragraphs, allow_destroy: true
end
```

Στη κλάση αυτή υπάρχει μία διαφοροποίηση σε σχέση με τις προηγούμενες. Περιέχει τη δήλωση embeds_many η οποία χρησιμοποιείται για το σχηματισμό συσχετίσεων μεταξύ εγγράφων που ενσωματώνονται και αποθηκεύονται μέσα σε άλλα έγγραφα της βάσης δεδομένων και ανήκουν στην ίδια συλλογή, σε αντίθεση με τη δήλωση has_many η οποία χρησιμοποιείται για το σχηματισμό συσχετίσεων μεταξύ εγγράφων που ανήκουν σε διαφορετική συλλογή. Έτσι λοιπόν μεταξύ των κλάσεων Preface και PrefaceParagraphs υπάρχει η σχέση «περιλαμβάνει» και «περιλαμβάνεται», δηλαδή ο πρόλογος περιλαμβάνει πολλές παραγράφους και αντίστροφα κάθε παράγραφος του πρόλογου περιλαμβάνεται στον πρόλογο.

6.8.2.6. Sprint 1.1.6 – Δημιουργία μοντέλου Preface Paragraph

Πρώτα εκτελείται η εντολή:

```
~$ rails g model PrefaceParagraph text:string
```

Στη συνέχεια σχηματίζεται το αρχείο preface_paragraph.rb που περιέχει την κλάση:

```
class PrefaceParagraph
  include Mongoid::Document
  field :text, type: String
  embedded_in :preface
end
```

Η δήλωση `embedded_in` περιγράφει τη σχέση «περιλαμβάνεται» που αναλύθηκε στην προηγούμενη ενότητα.

6.8.2.7. Sprint 1.1.7 – Δημιουργία μοντέλου School Grade

Για το σχηματισμό του μοντέλου εκτελείται η εντολή:

```
~$ rails g model SchoolGrade name:string
```

Έτσι δημιουργείται το αρχείο `school_grade.rb` που περιέχει τη κλάση `SchoolGrade`.

```
class SchoolGrade
  include Mongoid::Document
  field :name, type: String
  has_many :books
end
```

Στο παραπάνω αρχείο εκφράζεται η συσχέτιση «ένα-προς-πολλά» (1-N) ανάμεσα στις κλάσεις `SchoolGrade` και `Books`, δηλαδή κάθε σχολική τάξη έχει πολλά βιβλία ενώ κάθε βιβλίο ανήκει σε μία μόνο σχολική τάξη.

6.8.2.8. Sprint 1.1.8 – Δημιουργία μοντέλου Chapter

Αρχικά εκτελείται η εντολή:

```
~$ rails g model Chapter title:string
```

Το μοντέλο που δημιουργείται περιέχει την κλάση:

```
class Chapter
  include Mongoid::Document
  field :title, type: String
  belongs_to :book
  has_many :sections, dependent: :destroy
end
```

```
accepts_nested_attributes_for :sections, allow_destroy: true
end
```

Οι σχέσεις των κλάσεων περιγράφονται με συσχετίσεις (1-N). Έτσι κάθε κεφάλαιο ανήκει σε ένα μόνο βιβλίο και κάθε κεφάλαιο έχει πολλές ενότητες.

6.8.2.9. Sprint 1.1.9 – Δημιουργία μοντέλου Section

Για τη δημιουργία του μοντέλου section εκτελείται η εντολή:

```
~$ rails g model Section title:string
```

Το μοντέλο που παράγεται αυτόματα από το πλαίσιο Ruby on Rails είναι το section.rb και περιέχει την κλάση:

```
class Section
  include Mongoid::Document
  field :title, type: String
  belongs_to :chapter
  has_many :paragraphs, dependent: :destroy
  accepts_nested_attributes_for :paragraphs, allow_destroy: true
end
```

Η σχέση που υφίστανται ανάμεσα στις κλάσεις Section – Chapter και Section – Paragraphs εκφράζεται ομοίως με τη συσχέτιση 1-N δηλαδή κάθε ενότητα ανήκει σε ένα κεφάλαιο και κάθε ενότητα έχει πολλές παραγράφους.

6.8.2.10. Sprint 1.1.10 – Δημιουργία μοντέλου Paragraph

Το μοντέλο Paragraph σχηματίζεται εκτελώντας στο terminal την εντολή:

```
~$ rails g model Paragraph title:string text:string side_title:string
definition:boolean
```

Το αρχείο που δημιουργείται περιέχει την κλάση Paragraph.

```
class Paragraph
  include Mongoid::Document
  field :title, type: String
  field :text, type: String
```

```
field :side_title, type: String
field :definition, type: Boolean
belongs_to :section
end
```

6.8.2.11. Sprint 1.1.11 – Προσθήκη Admin namespace στο routes.rb

Σκοπός αυτού του sprint είναι η δημιουργία ξεχωριστού route για τη διαχείριση των βιβλίων προκειμένου να διαφοροποιηθούν οι εργασίες του διαχειριστή από εκείνες του απλού χρήστη, αλλά και να προετοιμαστεί το έδαφος για τον καθορισμό των προσβάσεων σε μεταγενέστερο στάδιο.

Αρχικά τροποποιείται το αρχείο routes.rb έτσι ώστε η γραμμή resources :books να συμπεριληφθεί στο κάτωθι block:

```
namespace :admin do
  resources :books
end
```

Επίσης δημιουργείται υπό-φάκελος admin στο φάκελο controller που περιέχει το αρχείο books_controller.rb καθώς και υπό-φάκελος admin στο φάκελο views που περιέχει τον φάκελο books με όλα τα html αρχεία του.

Ταυτόχρονα γίνονται τροποποιήσεις στα html αρχεία του φακέλου books, συγκεκριμένα στις γραμμές εκείνες που αφορούν του συνδέσμους show, edit, destroy, new και back έτσι ώστε να οδηγούν στο σωστό path μετά την αλλαγή που έγινε στο route.

Τέλος μετασχηματίζεται το αρχείο books_controller.rb προκειμένου η κλάση BooksController να συμπεριληφθεί σε ένα module που ονομάζεται Admin, δηλαδή:

```
module Admin
  class BooksController < ApplicationController
    .
    .
    .
  end
end
```


6.8.2.12. Sprint 1.1.12 – Παραμετροποίηση του books_controller.rb

Σε αυτό το sprint γίνεται παραμετροποίηση του ελεγκτή books_controller. Συγκεκριμένα προστίθεται ο παρακάτω κώδικας στη μέθοδο new:

```
def new
  @book = Book.new
  @publisher = @book.build_publisher
  @author = @book.authors.build
  @chapter = @book.chapters.build
  @section = @chapter.sections.build
  @paragraph = @section.paragraphs.build
  render layout: 'admin'
end
```

Με τον τρόπο αυτό δηλώνεται ότι κάθε φορά που καλείται η μέθοδος new για τη δημιουργία ενός στιγμιότυπου του αντικειμένου book, για παράδειγμα κατά την καταχώρηση των στοιχείων ενός βιβλίου σε μια φόρμα, αναμένεται από το σύστημα να δημιουργήσει στιγμιότυπα για τα αντικείμενα publisher, author, chapter, section και paragraph έτσι ώστε να μπορούν να προβληθούν στην ίδια φόρμα με το βιβλίο ο εκδότης, οι συγγραφείς, τα κεφάλαια, οι ενότητες και οι παράγραφοι.

Η πρόταση render layout: 'admin' δηλώνει ότι, όταν καλείται η μέθοδος new θα πρέπει να χρησιμοποιείται η διάταξη 'admin'. Η διάταξη ουσιαστικά αποτελείται από ένα head tag της HTML και καθορίζει ποια στυλ CSS και JavaScript θα χρησιμοποιηθούν για τη προβολή της σελίδας books/new.html.haml.

Επιπλέον προκειμένου το σύστημα, κατά τη δημιουργία στιγμιότυπου ενός βιβλίου, να μπορέσει να αποθηκεύσει τα χαρακτηριστικά που αφορούν τον εκδότη, τους συγγραφείς, τα κεφάλαια, τις ενότητες και τις παραγράφους θα πρέπει να τροποποιηθεί η private μέθοδος book_params ως εξής:

```
def book_params
  params.require(:book).permit(:title, :published_on, :edition,
    :location, :school_grade_id, publisher_attributes: [:id, :name,
    :_destroy], authors_attributes: [:id, :first_name, :last_name,
    :_destroy], chapters_attributes: [:id, :title, :_destroy,
```

```

sections_attributes: [:id, :title, :_destroy, paragraphs_attributes:
[:id, :title, :text, :_destroy]])
end

```

6.8.2.13. Sprint 1.1.13 – Διαμόρφωση της όψεως books/new.html.haml

Η όψη books/new.html.haml διαμορφώνεται προσθέτοντας τον παρακάτω κώδικα.

```

= render 'partials/side_navbar'
.col-sm-9.col-sm-offset-3.col-md-10.col-md-offset-2.main
%h2.page-header Καταχώρηση Βιβλίου
.well
  = simple_nested_form_for ['admin', @book], html: {class: 'form-
horizontal'}, wrapper: :horizontal_form do |f|
    .panel.panel-info
      .panel-heading
        %h4 Γενικά Στοιχεία
      .panel-body
        %fieldset{class: 'well'}
          = f.input :title, label: 'Τίτλος Βιβλίου'
          = f.input :edition, label: 'Έκδοση'
          = f.input :location, label: 'Τόπος Έκδοσης'
          = f.input :published_on, label: 'Έτος Έκδοσης'
          = f.input :school_grade_id, collection: SchoolGrade.all,
            value_method: :id, label_method: :name, label: 'Σχολική Τάξη'
        .panel-heading
          %h4 Στοιχεία Εκδότη
        .panel-body
          = f.fields_for :publisher, html: {class: 'form-horizontal'},
            wrapper: :horizontal_form do |pub|
            %fieldset{class: 'well'}
              = pub.input :name, label: 'Όνομα'
        .panel-heading
          %h4 Συγγραφείς
        .panel-body
          = f.fields_for :authors, html: {class: 'form-horizontal'},

```

```

wrapper: :horizontal_form do |auth|
  %fieldset{class: 'well'}
    = auth.input :first_name, label: 'Όνομα Συγγραφέα'
    = auth.input :last_name, label: 'Επώνυμο Συγγραφέα'
    = auth.link_to_remove '<i class= "fa fa-minus">&nbsp;
      Συγγραφέα</i>'.html_safe, class: 'btn btn-danger btn-xs
      add_rm_btn'
= f.link_to_add '<i class= "fa fa-plus">&nbsp;
  Συγγραφέα</i>'.html_safe, :authors, class: 'btn btn-success
  btn-xs add_rm_btn'
.panel-heading
  %h4 Περιεχόμενο
.panel-body
= f.fields_for :chapters, html: {class: 'form-horizontal'},
  wrapper: :horizontal_form do |chap|
  %fieldset{class: 'well'}
    = chap.input :title, label: 'Τίτλος Κεφαλαίου'
    = chap.link_to_remove '<i class= "fa fa-minus">&nbsp;
      Κεφάλαιο</i>'.html_safe, class: 'btn btn-danger btn-xs
      add_rm_btn'
    = chap.fields_for :sections, html: {class: 'form-
      horizontal'}, wrapper: :horizontal_form do |sec|
      %fieldset{class: 'well'}
        = sec.input :title, label: 'Τίτλος Ενότητας'
        = sec.link_to_remove '<i class= "fa fa-minus">&nbsp;
          Ενότητα</i>'.html_safe, class: 'btn btn-danger btn-xs
          add_rm_btn'
        = sec.fields_for :paragraphs, html: {class: 'form-
          horizontal'}, wrapper: :horizontal_form do |par|
            %fieldset{class: 'well'}
              = par.input :title, label: 'Τίτλος Παραγράφου'
              = par.input :text, as: :text, label: 'Περιεχόμενο
              Παραγράφου'
              = par.link_to_remove '<i class= "fa fa-minus">&nbsp;
                Παράγραφος</i>'.html_safe, class: 'btn btn-danger

```

```

      btn-xs add_rm_btn'
    = sec.link_to_add '<i class= "fa fa-plus">&nbsp;
      Παράγραφος</i>'.html_safe, :paragraphs, class: 'btn
      btn-success btn-xs add_rm_btn'
    = chap.link_to_add '<i class= "fa fa-plus">&nbsp;
      Ενότητα</i>'.html_safe, :sections, class: 'btn btn-success
      btn-xs add_rm_btn'
    = f.link_to_add '<i class= "fa fa-plus">&nbsp;
      Κεφάλαιο</i>'.html_safe, :chapters, class: 'btn btn-success
      btn-xs add_rm_btn'
  .row
  .col-xs-offset-4.col-xs-4
    = f.submit 'Καταχώρηση Βιβλίου', class: 'btn btn-primary btn-
      block btn-lg'

```

Ο χρήστης μπορεί να καταχωρεί τα στοιχεία του βιβλίου χρησιμοποιώντας μία φόρμα η οποία περιέχει όλα τα απαραίτητα πεδία. Στην ενότητα 6.9.2 παρατίθενται εικόνες που επιδεικνύουν τη συγκεκριμένη λειτουργία.

Με την πρόταση `= render 'partials/side_navbar'`, δηλώνεται στο σύστημα ότι θα πρέπει να προβάλλει και να χρησιμοποιήσει των κώδικα που βρίσκεται στο αρχείο `_side_navbar.html.haml` μέσα στον φάκελο `partials`. Το αρχείο αυτό είναι υπεύθυνο για την εμφάνιση της πλαϊνής μπάρας πλοήγησης.

```

.col-sm-3.col-md-2.sidebar
  %ul.nav.nav-sidebar
    %li
      = link_to '<i class= "fa fa-list-alt">&nbsp; Όλα τα
      Βιβλία</i>'.html_safe, admin_books_path
    %ul.nav.nav-sidebar
      %li.dropdown
        %a.dropdown-toggle{"aria-expanded" => "false", "aria-haspopup" =>
        "true", "data-toggle" => "dropdown", :href => "#"}
        %i.fa.fa-book &nbsp; Διαχείριση Βιβλίων
        %span.caret
        %ul.dropdown-menu
          %li

```

```

= link_to '<i class= "fa fa-plus">&nbsp;   Καταχώρηση
  Βιβλίου</i>'.html_safe, new_admin_book_path
%li
= link_to '<i class= "fa fa-trash-o">&nbsp;   Διαγραφή
  Βιβλίου</i>'.html_safe, admin_books_path

```

Παρατηρείται λοιπόν ότι το Ruby on Rails παρέχει τη δυνατότητα διαχωρισμού ενός λειτουργικού μέρους της εφαρμογής σε ξεχωριστό αρχείο έτσι ώστε ο κώδικας να είναι επαναχρησιμοποιήσιμος, αναγνώσιμος και ευκολονόητος.

6.8.3. Έκδοση 1.2 – Προβολή όλων των Βιβλίων

Σε αυτήν την έκδοση δημιουργείται η όψη που καθορίζει τον τρόπο προβολής όλων των βιβλίων με τη μορφή πίνακα. Έτσι ο διαχειριστής μπορεί να επιλέξει ένα συγκεκριμένο βιβλίο, από μια λίστα βιβλίων και να εκτελέσει σε αυτό διαχειριστικές εργασίες όπως προβολή, επεξεργασία και διαγραφή. Το sprint που πρέπει να ολοκληρωθεί είναι το ακόλουθο:

Πίνακας 11: Έκδοση 1.2 – Προβολή όλων των Βιβλίων

Έκδοση 1.2				
Ιστορία: Προβολή όλων των Βιβλίων				
Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.2.1	Διαμόρφωση της όψεως books/index.html.haml	Συγγραφή κώδικα σελίδας	Ολοκληρώθηκε	2

6.8.3.1. Sprint 1.2.1 – Διαμόρφωση της Όψεως books/index.html.haml

Με τη συγγραφή του κατάλληλου κώδικα δημιουργείται ένας πίνακας του οποίου οι σειρές αποτελούν στιγμιότυπα της κλάσεις book, δηλαδή κάθε σειρά είναι ένα ξεχωριστό βιβλίο. Ο πίνακας απαριθμεί όλα τα βιβλία που είναι αποθηκευμένα στη βάση δεδομένων. Παρακάτω ακολουθεί ο κώδικας της όψης books/index.html.haml.

```

=render 'partials/side_navbar'
.col-sm-9.col-sm-offset-3.col-md-10.col-md-offset-2.main
  %p#notice= notice
  %h2 Όλα τα Βιβλία
  .table-responsive
    %table.table.table-hover
      %thead
        %tr
          %th Τίτλος
          %th Εκδότης
          %th Συγγραφείς
      %tbody
        - @books.each do |book|
          %tr
            %td= book.title
            %td= book.publisher[:name]
            %td= book.authors.collect(&:last_name).to_sentence
            %td= link_to '<i class= "fa fa-eye" data-toggle="tooltip" data-
              placement="top" title="Προβολή"></i>'.html_safe,
              admin_book_path(book)
          %br
            = link_to '<i class= "fa fa-plus" data-toggle="tooltip" data-
              placement="top" title="Καταχώρηση"></i>'.html_safe,
              new_admin_book_path

```

6.8.4. Έκδοση 1.3 – Επεξεργασία Βιβλίων

Μια από τις πλέον απαραίτητες διαχειριστικές λειτουργίες είναι η δυνατότητα επεξεργασίας του εκάστοτε βιβλίου, έτσι ώστε ο διαχειριστής να μπορεί να διορθώσει ή να αλλάξει το περιεχόμενο του. Τα sprints που εκτελούνται είναι τα ακόλουθα:

Πίνακας 12: Έκδοση 1.3 – Επεξεργασία Βιβλίων

Έκδοση 1.3

Ιστορία: Επεξεργασία Βιβλίων				
Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.3.1	Διαμόρφωση όψεως books/edit.html.haml	Συγγραφή κώδικα σελίδας	Ολοκληρώθηκε	2
1.3.2	Παραμετροποίηση όψεως books/index.html.haml	Δημιουργία συνδέσμου για την επεξεργασία του εκάστοτε βιβλίου	Ολοκληρώθηκε	1

6.8.4.1. Sprint 1.3.1 – Διαμόρφωση Όψεως books/edit.html.haml

Προκειμένου να είναι εφικτή η επεξεργασία ενός βιβλίου, η όψη books/edit.html.haml διαμορφώνεται ανάλογα χρησιμοποιώντας τον παρακάτω κώδικα:

```
= render 'partials/side_navbar'
.col-sm-9.col-sm-offset-3.col-md-10.col-md-offset-2.main
%h2.page-header Καταχώρηση Βιβλίου
.well
  = simple_nested_form_for ['admin', @book], html: {class: 'form-
horizontal'}, wrapper: :horizontal_form do |f|
    .panel.panel-info
      .panel-heading
        %h4 Γενικά Στοιχεία
      .panel-body
        %fieldset{class: 'well'}
          = f.input :title, label: 'Τίτλος Βιβλίου'
          = f.input :edition, label: 'Έκδοση'
          = f.input :location, label: 'Τόπος Έκδοσης'
          = f.input :published_on, label: 'Έτος Έκδοσης'
          = f.input :school_grade_id, collection: SchoolGrade.all,
            value_method: :id, label_method: :name, label: 'Σχολική Τάξη'
      .panel-heading
        %h4 Στοιχεία Εκδότη
      .panel-body
```

```

= f.fields_for :publisher, html: {class: 'form-horizontal'},
  wrapper: :horizontal_form do |pub|
  %fieldset{class: 'well'}
    = pub.input :name, label: 'Όνομα'
.panel-heading
  %h4 Συγγραφείς
.panel-body
= f.fields_for :authors, html: {class: 'form-horizontal'},
  wrapper: :horizontal_form do |auth|
  %fieldset{class: 'well'}
    = auth.input :first_name, label: 'Όνομα Συγγραφέα'
    = auth.input :last_name, label: 'Επώνυμο Συγγραφέα'
    = auth.link_to_remove '<i class= "fa fa-minus">&nbsp;
      Συγγραφέα</i>'.html_safe, class: 'btn btn-danger btn-xs
      add_rm_btn'
    = f.link_to_add '<i class= "fa fa-plus">&nbsp;
      Συγγραφέα</i>'.html_safe, :authors, class: 'btn btn-success
      btn-xs add_rm_btn'
.panel-heading
  %h4 Περιεχόμενα
.panel-body
= f.fields_for :chapters, html: {class: 'form-horizontal'},
  wrapper: :horizontal_form do |chap|
  %fieldset{class: 'well'}
    = chap.input :title, label: 'Τίτλος Κεφαλαίου'
    = chap.link_to_remove '<i class= "fa fa-minus">&nbsp;
      Κεφάλαιο</i>'.html_safe, class: 'btn btn-danger btn-xs
      add_rm_btn'
    = chap.fields_for :sections, html: {class: 'form-
      horizontal'}, wrapper: :horizontal_form do |sec|
      %fieldset{class: 'well'}
        = sec.input :title, label: 'Τίτλος Ενότητας'
        = sec.link_to_remove '<i class= "fa fa-minus">&nbsp;
          Ενότητα</i>'.html_safe, class: 'btn btn-danger btn-xs
          add_rm_btn'

```



```

= sec.fields_for :paragraphs, html: {class: 'form-
horizontal'}, wrapper: :horizontal_form do |par|
%fieldset{class: 'well'}
= par.input :title, label: 'Τίτλος Παραγράφου'
= par.input :text, as: :text, label: 'Περιεχόμενο
Παραγράφου'
= par.link_to_remove '<i class= "fa fa-minus">&nbsp;
Παράγραφος</i>'.html_safe, class: 'btn btn-danger
btn-xs add_rm_btn'
= sec.link_to_add '<i class= "fa fa-plus">&nbsp;
Παράγραφος</i>'.html_safe, :paragraphs, class: 'btn
btn-success btn-xs add_rm_btn'
= chap.link_to_add '<i class= "fa fa-plus">&nbsp;
Ενότητα</i>'.html_safe, :sections, class: 'btn btn-success
btn-xs add_rm_btn'
= f.link_to_add '<i class= "fa fa-plus">&nbsp;
Κεφάλαιο</i>'.html_safe, :chapters, class: 'btn btn-success
btn-xs add_rm_btn'
.row
.col-xs-offset-4.col-xs-4
= f.submit 'Ανανέωση Βιβλίου', class: 'btn btn-primary btn-
block btn-lg'

```

6.8.4.2. Sprint 1.3.2 – Παραμετροποίηση Όψεως books/index.html.haml

Στον συγκεντρωτικό πίνακα βιβλίων που δημιουργήθηκε στην έκδοση 1.2 προστίθεται σύνδεσμος ο οποίος επιτρέπει την επεξεργασία συγκεκριμένου βιβλίου. Για την επίτευξη αυτού τροποποιείται η όψη books/index.html.haml ως εξής:

```

=render 'partials/side_navbar'
.col-sm-9.col-sm-offset-3.col-md-10.col-md-offset-2.main
%p#notice= notice
%h2 Όλα τα Βιβλία
.table-responsive
%table.table.table-hover
%thead

```

```
| Τίτλος | Εκδότης | Συγγραφείς | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - |                   |                              |                                                          |                                                                                                                                                      |                                                                                                                                                                  | |-------------------|------------------------------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------| | <i>book.title</i> | <i>book.publisher[:name]</i> | <i>book.authors.collect(&amp;:last_name).to_sentence</i> | <i>link_to '&lt;i class= "fa fa-eye" data-toggle="tooltip" data-placement="top" title="Προβολή"&gt;&lt;/i&gt;' .html_safe, admin_book_path(book)</i> | <i>link_to '&lt;i class= "fa fa-pencil" data-toggle="tooltip" data-placement="top" title="Επεξεργασία"&gt;&lt;/i&gt;' .html_safe, edit_admin_book_path(book)</i> | |-------------------|------------------------------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------| | | |
| - |                                                                                                                                                        | |--------------------------------------------------------------------------------------------------------------------------------------------------------| | <i>link_to '&lt;i class= "fa fa-plus" data-toggle="tooltip" data-placement="top" title="Καταχώρηση"&gt;&lt;/i&gt;' .html_safe, new_admin_book_path</i> | |--------------------------------------------------------------------------------------------------------------------------------------------------------| | | |

```

6.8.5. Έκδοση 1.4 – Διαγραφή Βιβλίων

Οι κύριες διαχειριστικές εργασίες της εφαρμογής ολοκληρώνονται με την ανάπτυξη της λειτουργικότητας που αφορά στη διαγραφή του εκάστοτε βιβλίου. Το sprint που υλοποιείται αναφέρεται στον πίνακα που ακολουθεί:

Πίνακας 13: Έκδοση 1.4 – Διαγραφή Βιβλίων

Έκδοση 1.4				
Ιστορία: Διαγραφή Βιβλίων				
Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)

1.4.1	Παραμετροποίηση όψεως books/index.html.haml	Δημιουργία συνδέσμου για διαγραφή του εκάστοτε βιβλίου	Ολοκληρώθηκε	1
-------	---	---	--------------	---

6.8.5.1. Sprint 1.4.1 – Παραμετροποίηση Όψεως books/index.html.haml

Όπως και στην έκδοση 1.3, τροποποιείται το αρχείο books/index.html.haml προκειμένου να προστεθεί ο κατάλληλος σύνδεσμος για την διαγραφή του εκάστοτε βιβλίου. Έτσι ο κώδικας του αρχείου διαμορφώνεται ως κάτωθι:

```
=render 'partials/side_navbar'
.col-sm-9.col-sm-offset-3.col-md-10.col-md-offset-2.main
%p#notice= notice
%h2 Όλα τα Βιβλία
.table-responsive
%table.table.table-hover
%thead
%tr
%th Τίτλος
%th Εκδότης
%th Συγγραφείς
%tbody
- @books.each do |book|
%tr
%td= book.title
%td= book.publisher[:name]
%td= book.authors.collect(&:last_name).to_sentence
%td= link_to '<i class= "fa fa-eye" data-toggle="tooltip" data-
placement="top" title="Προβολή"></i>'.html_safe,
admin_book_path(book)
%td= link_to '<i class= "fa fa-pencil" data-toggle="tooltip"
data-placement="top" title="Επεξεργασία"></i>'.html_safe,
edit_admin_book_path(book)
%td= link_to '<i class= "fa fa-trash-o" data-toggle="tooltip"
data-placement="top" title="Διαγραφή"></i>'.html_safe,
```

```

admin_book_path(book), method: :delete, data: { confirm:
  'Are you sure?' }

%br
= link_to '<i class= "fa fa-plus" data-toggle="tooltip" data-
placement="top" title="Καταχώρηση"></i>'.html_safe,
new_admin_book_path

```

6.8.6. Έκδοση 1.5 – Κατηγοριοποίηση Βιβλίων

Σε αυτή την έκδοση δημιουργείται η ιστοσελίδα που μεταβαίνει ο χρήστης-μαθητής κατά την είσοδό του στην εφαρμογή. Επίσης υλοποιείται εκείνη η λειτουργικότητα που επιτρέπει την κατηγοριοποίηση των βιβλίων ανά σχολική βαθμίδα και τάξη. Ο παρακάτω πίνακας αναφέρει συνοπτικά τα sprints που εκτελούνται.

Πίνακας 14: Έκδοση 1.5 – Κατηγοριοποίηση Βιβλίων

Έκδοση 1.5				
Ιστορία: Κατηγοριοποίηση Βιβλίων				
Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.5.1	Δημιουργία Ελεγκτή student/books_controller.rb	Δημιουργία Ελεγκτή με δύο μεθόδους: index και show	Ολοκληρώθηκε	1
1.5.2	Προσθήκη Student namespace στο routes.rb	Δημιουργία ξεχωριστού route για την κατηγοριοποίηση και προβολή των βιβλίων	Ολοκληρώθηκε	1
1.5.3	Παραμετροποίηση Ελεγκτή student/ books_controller.rb	Δημιουργία του module Student	Ολοκληρώθηκε	1

1.5.4	Διαμόρφωση Όψεως student/books/index.html.haml	Συγγραφή κώδικα της σελίδας	Ολοκληρώθηκε	4
1.5.5	Παραμετροποίηση Ελεγκτή student/books_controller.rb	Τροποποίηση της μεθόδου index	Ολοκληρώθηκε	2

6.8.6.1. Sprint 1.5.1 – Δημιουργία Ελεγκτή student/books_controller.rb

Για τη δημιουργία του ελεγκτή books εκτελείται η εντολή:

```
~$ rails g Books index show
```

Έτσι σχηματίζεται το αρχείο books_controller.rb στον φάκελο controllers το οποίο περιέχει την παρακάτω κλάση:

```
class BooksController < ApplicationController
  def index
    @books = Book.all
  end
  def show
    render layout: 'show'
  end
end
```

Επίσης δημιουργείται υπό-φάκελος student στον φάκελο controller που περιέχει το αρχείο books_controller.rb καθώς και υπό-φάκελος student στον φάκελο views που περιέχει τον φάκελο books με όλα τα html αρχεία του (index και show).

6.8.6.2. Sprint 1.5.2 – Προσθήκη Student namespace στο routes.rb

Εφόσον ο ελεγκτής που δημιουργήθηκε στο sprint 1.5.1, ασχολείται αποκλειστικά με τις εργασίες που αφορούν στον χρήστη-μαθητή, τροποποιείται το αρχείο routes.rb ώστε η νέα γραμμή resources :books να συμπεριληφθεί στο κάτωθι block:

```
namespace :student do
  resources :books
end
```

6.8.6.3. Sprint 1.5.3 – Παραμετροποίηση Ελεγκτή student/books_controller.rb

Το αρχείο student/books_controller.rb μετασχηματίζεται έτσι ώστε η κλάση BooksController να συμπεριληφθεί στο module που ονομάζεται Student, δηλαδή:

```
module Student
  class BooksController < ApplicationController
    .
    .
    .
  end
end
```

6.8.6.4. Sprint 1.5.4 – Διαμόρφωση Όψεως student/books/index.html.haml

Η όψη student/books/index.html.haml περιέχει τον κώδικα που ρυθμίζει τον τρόπο με τον οποίο προβάλλεται η ιστοσελίδα κατά την είσοδο του χρήστη-μαθητή στο σύστημα. Ο κώδικας αυτής της όψης διαμορφώνεται ως εξής:

```
%p#notice= notice
.row
  .col-sm-12
    %h2 Όλο το Υλικό
  .row &nbsp;
  .row
    .col-sm-4
      %a#nav_text_decor{'tabindex' => '0', 'role' => 'button', 'data-toggle' => 'popover', 'data-trigger' => 'focus', 'data-container' => 'body', 'data-placement' => 'bottom', 'data-html' => 'true', 'data-title' => "<h4 class='text-center'>Υλικό ανά τάξη</h4>", 'data-content' => (link_to "<p class='text-center'>Α' Δημοτικού</p>".html_safe, root_path) + (link_to "<p class='text-center'>Β' Δημοτικού</p>".html_safe) + (link_to "<p class='text-center'>Γ' Δημοτικού</p>".html_safe) + (link_to "<p class='text-center'>Δ' Δημοτικού</p>".html_safe) + (link_to "<p class='text-center'>Ε' Δημοτικού</p>".html_safe) + (link_to "<p class='text-center'>ΣΤ' Δημοτικού</p>".html_safe)}
```

```

%nav#nav1_color.nav_no_margin.navbar.navbar-default
  %h3.nav_text.text-center Δημοτικό
  .well.well-lg.well_color
    = image_tag '036.svg', size: '90x90', class: 'img-responsive
      center_image'
.col-sm-4
  %a#nav_text_decor{'tabindex' => '0', 'role' => 'button', 'data-toggle'
=> 'popover', 'data-trigger' => 'focus', 'data-container' => 'body',
'data-placement' => 'bottom', 'data-html' => 'true', 'data-title' =>
"<h4 class='text-center'>Υλικά ανά τάξη</h4>", 'data-content' =>
(link_to "<p class='text-center'>A' Γυμνασίου</p>".html_safe,
student_books_path(school_grade: "A' Γυμνασίου")) + (link_to "<p
class='text-center'>B' Γυμνασίου</p>".html_safe,
student_books_path(school_grade: "B' Γυμνασίου")) + (link_to "<p
class='text-center'>Γ' Γυμνασίου</p>".html_safe,
student_books_path(school_grade: "Γ' Γυμνασίου"))}
%nav#nav2_color.nav_no_margin.navbar.navbar-inverse
  %h3.nav_text.text-center Γυμνάσιο
  .well.well-lg.well_color
    = image_tag '026.svg', size: '90x90', class: 'img-responsive
      center_image'
.col-sm-4
  %a#nav_text_decor{'tabindex' => '0', 'role' => 'button', 'data-toggle'
=> 'popover', 'data-trigger' => 'focus', 'data-container' => 'body',
'data-placement' => 'bottom', 'data-html' => 'true', 'data-title' =>
"<h4 class='text-center'>Υλικά ανά τάξη</h4>", 'data-content' =>
(link_to "<p class='text-center'>A' Λυκείου</p>".html_safe,
root_path) + (link_to "<p class='text-center'>B'
Λυκείου</p>".html_safe) + (link_to "<p class='text-center'>Γ'
Λυκείου</p>".html_safe)}
%nav#nav3_color.nav_no_margin.navbar.navbar-inverse
  %h3.nav_text.text-center Λύκειο
  .well.well-lg.well_color
    = image_tag '127.svg', size: '90x90', class: 'img-responsive
      center_image'

```

```

.table-responsive
  %table.table.table-hover
    %thead
      %tr
        %th Τίτλος
        %th Εκδότης
    %tbody
      - @books.each do |book|
        %tr
          %td= book.title
          %td= book.publisher[:name]
          %td= link_to '<i class= "fa fa-eye" data-toggle="tooltip" data-
            placement="top" title="Προβολή"></i>'.html_safe,
            student_book_path(book)

```

6.8.6.5. Sprint 1.5.5 - Παραμετροποίηση Ελεγκτή student/books_controller.rb

Σε αυτό το sprint τροποποιείται η μέθοδος index της κλάσης BooksController έτσι ώστε να είναι εφικτή η κατηγοριοποίηση των βιβλίων ανάλογα με τη σχολική βαθμίδα και τάξη. Για παράδειγμα όταν ο χρήστης θα επιλέγει το υλικό της Α' Γυμνασίου, θα εμφανίζονται μόνο τα βιβλία της Α' Γυμνασίου. Στη μέθοδο index λοιπόν προστίθεται ο παρακάτω κώδικάς:

```

class BooksController < ApplicationController
  def index
    if params[:school_grade].blank?
      @books = Book.all
    else
      @school_grade_id = SchoolGrade.find_by(name:
        params[:school_grade]).id
      @books = Book.where(school_grade_id:
        @school_grade_id).order('created_at DESC')
    end
  end
end

```



```

def show
  render layout: 'show'
end
end

```

6.8.7. Έκδοση 1.6 – Προβολή Περιεχομένου Βιβλίου

Ο μαθητής επιλέγοντας το βιβλίο που τον ενδιαφέρει έχει πρόσβαση στο περιεχόμενο του. Για την υλοποίηση αυτής της έκδοσης εκτελείται το παρακάτω sprint:

Πίνακας 15: Έκδοση 1.6 – Προβολή Περιεχομένου Βιβλίου

Έκδοση 1.6				
Ιστορία: Προβολή Περιεχομένου Βιβλίου				
Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.6.1	Δημιουργία Όψεως student/books/show.html.haml	Προβολή του περιεχομένου κάθε βιβλίου	Ολοκληρώθηκε	3

6.8.7.1. Sprint 1.6.1 – Δημιουργία Όψεως student/books/show.html.haml

Στο φάκελο student/books δημιουργείται η όψη show.html.haml, η οποία ρυθμίζει τον τρόπο παρουσίασης των βιβλίων και περιέχει τον παρακάτω κώδικα:

```

= render 'partials/top_navbar'
.row
  .col-xs-9
    .well.well-lg.well_color_title_math
      %h1.h1_color.text-center= @book.title
      - @book.chapters.each do |chap|
        .well.well_color
          %section
            %h2.anchor{:id => chap.title}= chap.title
            - chap.sections.each do |sec|

```

```

.well.well_color.well_color_sec_math
  %section
    %h2.h3_color.anchor{:id => sec.title}= sec.title
  - sec.paragraphs.each do |p|
    .well.well_color
      %h3.anchor{:id => p.title}= p.title
      - if p.definition == true
        %p
          %strong= p.text
      - else
        %p= p.text
%nav.col-xs-3.bs-docs-sidebar
  %ul#sidebar.nav.nav-stacked.affix
    - @book.chapters.each do |chap|
      %li
        %a{:href => '#' + chap.title}= chap.title
      - chap.sections.each do |sec|
        %li
          %a{:href => '#' + sec.title}= sec.title
        - sec.paragraphs.each do |par|
          %li
            - unless par.title.empty?
              %a{:href => '#' + par.title}= par.title

```

6.8.8. Έκδοση 1.7 – Είσοδος Μαθητή στην Εφαρμογή

Στόχος της έκδοσης αυτής είναι η ανάπτυξη της λειτουργίας της εφαρμογής που αφορά στην ταυτοποίηση του χρήστη-μαθητή. Τα sprints που εκτελούνται εξετάζονται παρακάτω.

Πίνακας 16: Έκδοση 1.7 – Είσοδος Μαθητή στην Εφαρμογή

Έκδοση 1.7

Ιστορία: Είσοδος Μαθητή στην Εφαρμογή

Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.7.1	Εγκατάσταση πακέτου 'devise'	Παροχή λειτουργίας ταυτοποίησης των χρηστών	Ολοκληρώθηκε	1
1.7.2	Δημιουργία μοντέλου User	Δημιουργία του μοντέλου δεδομένων User, υπεύθυνο για την αποθήκευση δεδομένων που αφορούν στον λογαριασμό του χρήστη μαθητή, στη βάση	Ολοκληρώθηκε	1
1.7.3	Δημιουργία Student Controller	Ορισμός προσβάσεων μαθητή	Ολοκληρώθηκε	1
1.7.4	Παραμετροποίηση Ελεγκτή controllers/student	Ορισμός προσβάσεων μαθητή	Ολοκληρώθηκε	1
1.7.5	Διαμόρφωση Όψεως devise /registrations /new .html.haml	Διαμόρφωση της όψεως που αφορά στη φόρμα εγγραφής του χρήστη στην εφαρμογή	Ολοκληρώθηκε	2
1.7.6	Διαμόρφωση Όψεως devise/sessions/new.html.haml	Διαμόρφωση της όψεως που αφορά στη φόρμα εισόδου του χρήστη στην εφαρμογή	Ολοκληρώθηκε	2
1.7.7	Διαμόρφωση Όψεως devise/passwords/new.html.haml	Διαμόρφωση της όψεως που αφορά στη φόρμα επανέκδοσης κωδικού πρόσβασης των χρηστών	Ολοκληρώθηκε	1

6.8.8.1. Sprint 1.7.1. – Εγκατάσταση πακέτου 'devise'

Το 'devise' είναι ένα πακέτο λογισμικού το οποίο παρέχει έτοιμη λειτουργική λύση στο θέμα της ταυτοποίησης μιας εφαρμογής. Κρίνεται ωστόσο, απαραίτητη η παραμετροποίηση και η προσαρμογή του στις ανάγκες και απαιτήσεις της υπό ανάπτυξη εφαρμογής.

Για την εγκατάσταση του πακέτου 'devise' προστίθεται στο αρχείο Gemfile η γραμμή:

```
gem 'devise'
```

Στη συνέχεια εκτελείται η εντολή:

```
~$ bundle install
```

Προκειμένου να δημιουργηθούν τα απαραίτητα αρχεία ρυθμίσεων του 'devise', χρησιμοποιείται ο γεννήτορας του Ruby on Rails ως κάτωθι:

```
~$ rails generate devise:install
```

Ταυτόχρονα γίνονται παραμετροποιήσεις στα αρχεία config/environments/development.rb και config/environments/production.rb προκειμένου να είναι εφικτή η αποστολή email για την επιβεβαίωση του κωδικού πρόσβασης (password) ή την ανάκτησή του σε περίπτωση που χρειαστεί. Έτσι προστίθενται η παρακάτω γραμμή στο κάθε αρχείο αντίστοιχα.

```
config.action_mailer.default_url_options = { host: 'localhost', port: 3000 }
```

Το 'devise', επίσης, παρέχει τις κατάλληλες html όψεις για όλες τις λειτουργίες του. Για να δημιουργηθούν αυτές εκτελείται η εντολή:

```
~$ rails generate devise:views
```

6.8.8.2. Sprint 1.7.2 – Δημιουργία μοντέλου User

Οι χρήστες της εφαρμογής είναι οι μαθητές και ο διαχειριστής. Με τη βοήθεια του 'devise' παράγεται το μοντέλο User, το οποίο είναι υπεύθυνο για τη διαχείριση των λογαριασμών των μαθητών, με την παρακάτω εντολή:

```
~$ rails generate devise User name
```

Το αρχείο model/user.rb που δημιουργείται περιέχει την κλάση User με τις κάτωθι ιδιότητες:

```
class User
  include Mongoid::Document
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :trackable, :validatable

  ## Database authenticatable
```

```

field :name,          type: String
field :email,         type: String, default: ""
field :encrypted_password, type: String, default: ""

## Recoverable
field :reset_password_token,  type: String
field :reset_password_sent_at, type: Time

## Rememberable
field :remember_created_at, type: Time

## Trackable
field :sign_in_count,      type: Integer, default: 0
field :current_sign_in_at, type: Time
field :last_sign_in_at,   type: Time
field :current_sign_in_ip, type: String
field :last_sign_in_ip,   type: String
end

```

6.8.8.3. Sprint 1.7.3 – Δημιουργία Student Controller

Στη φάση αυτή δημιουργείται το αρχείο `controllers/student_controller.rb` για να οριστούν οι προσβάσεις των μαθητών. Το αρχείο αυτό περιέχει την παρακάτω κλάση:

```

class StudentController < ApplicationController
  before_filter :authenticate_user!
end

```

6.8.8.4. Sprint 1.7.4 – Παραμετροποίηση Ελεγκτή `controllers/student`

Σε αυτό το Sprint γίνεται παραμετροποίηση του ελεγκτή `books_controller.rb` που βρίσκεται μέσα στον φάκελο `controllers/student` έτσι ώστε να κληρονομεί ιδιότητες από τον ελεγκτή `controllers/student_controller.rb` που δημιουργήθηκε κατά την υλοποίηση του sprint 1.7.3. Επομένως η γραμμή που ορίζει την κλάση στον ελεγκτή `books` τροποποιείται ως εξής:

```

Class BooksController < StudentController

```

Με τον τρόπο αυτό, για να έχουν οι μαθητές πρόσβαση στο υλικό, απαιτείται η δημιουργία λογαριασμού.

6.8.8.5. Sprint 1.7.5 – Διαμόρφωση Όψεως devise /registrations /new .html.haml

Η όψη devise/registrations/new.html.haml είναι υπεύθυνη για την παρουσίαση της φόρμας εγγραφής χρήστη στην εφαρμογή και διαμορφώνεται ως εξής:

```
.row
  %h2.col-sm-offset-4.col-sm-4.text-center Δημιουργία λογαριασμού
.row &nbsp;
.row &nbsp;
.row
  .form.col-sm-offset-4.col-sm-4
    = form_for(resource, as: resource_name, url:
      registration_path(resource_name)) do |f|
    = devise_error_messages!
    .form-group
      = f.label :name
      = f.text_field :name, class: 'form-control input-lg', autofocus: true
    .form-group
      = f.label :email
      = f.email_field :email, class: 'form-control input-lg'
    .form-group
      = f.label :password
      - if @minimum_password_length
        %em (#{@minimum_password_length} characters minimum)
      = f.password_field :password, class: 'form-control input-lg',
        autocomplete: "off"
    .form-group
      = f.label :password_confirmation
      = f.password_field :password_confirmation, class: 'form-control
        input-lg', autocomplete: "off"
    .row &nbsp;
    .row &nbsp;
    .row
```

```

.col-xs-offset-3.col-xs-6
  .form-group.text-center
    = f.submit "Sign up", class: 'btn btn-primary btn-block'
= render "devise/shared/links"

```

6.8.8.6. Sprint 1.7.6 – Διαμόρφωση Όψευς devise/sessions/new.html.haml

Η όψη devise/sessions/new.html.haml τροποποιείται με τον παρακάτω κώδικα:

```

.row
  %h2.col-sm-offset-4.col-sm-4.text-center Sign in
.row &nbsp;
.row &nbsp;
.row
  .form.col-sm-offset-4.col-sm-4
    = form_for(resource, as: resource_name, url:
      session_path(resource_name)) do |f|
      .form-group
        = f.label :email
        = f.email_field :email, class: 'form-control input-lg', autofocus:
          true
      .form-group
        = f.label :password
        = f.password_field :password, class: 'form-control input-lg',
          autocomplete: 'off'
      .form-group
      - if devise_mapping.rememberable?
        = f.check_box :remember_me
        = f.label :remember_me
      .row &nbsp;
      .row &nbsp;
      .row
        .col-xs-offset-3.col-xs-6
          .form-group.text-center
            = f.submit 'Sign in', class: 'btn btn-primary btn-block'
    = render 'devise/shared/links'

```

Με τον τρόπο αυτό επιτρέπεται η διαμόρφωση της φόρμας εισόδου του χρήστη στην εφαρμογή.

6.8.8.7. Sprint 1.7.7 – Διαμόρφωση Όψεως devise/passwords/new.html.haml

Η όψη devise/passwords/new.html.haml ευθύνεται για τον τρόπο παρουσίασης της φόρμας καταχώρησης στοιχείων για την επανέκδοση κωδικού πρόσβασης των χρηστών. Ο κώδικας αυτής της όψης τροποποιείται ως κάτωθι:

```
row
  %h2.col-sm-offset-4.col-sm-4.text-center Ξεχάσατε τον κωδικό σας;
.row &nbsp;
.row &nbsp;
.row
  .form.col-sm-offset-4.col-sm-4
    = form_for(resource, as: resource_name, url:
      password_path(resource_name), html: { method: :post }) do |f|
    = devise_error_messages!
    .form-group
      = f.label :email
      = f.email_field :email, class: 'form-control input-lg', autofocus:
        true
    .row &nbsp;
    .row &nbsp;
    .row
      .col-xs-offset-3.col-xs-6
        .form-group.text-center
          = f.submit 'Επανεκδοση Κωδικού', class: 'btn btn-primary btn-
            block'
    = render 'devise/shared/links'
```


6.8.9. Έκδοση 1.8 – Είσοδος Διαχειριστή στην Εφαρμογή

Αυτή η έκδοση επικεντρώνεται στην ανάπτυξη της λειτουργίας της εφαρμογής που αφορά στην ταυτοποίηση του διαχειριστή. Τα sprints που εκτελούνται είναι τα κάτωθι:

Πίνακας 17: Έκδοση 1.8 – Είσοδος Διαχειριστή στην Εφαρμογή

Έκδοση 1.8				
Ιστορία: Είσοδος Διαχειριστή στην Εφαρμογή				
Sprint	Εργασία (Task)	Στόχος	Κατάσταση	Χρόνος Υλοποίησης (ώρες)
1.8.1	Δημιουργία μοντέλου Administrator	Δημιουργία του μοντέλου δεδομένων Administrator, υπεύθυνο για την αποθήκευση δεδομένων που αφορούν στον λογαριασμό του διαχειριστή, στη βάση	Ολοκληρώθηκε	1
1.8.2	Δημιουργία Admin Controller	Ορισμός προσβάσεων διαχειριστή	Ολοκληρώθηκε	1
1.8.3	Παραμετροποίηση Ελεγκτή controllers/admin/bo oks_controller.rb	Ορισμός προσβάσεων διαχειριστή	Ολοκληρώθηκε	1

6.8.9.1. Sprint 1.8.1 – Δημιουργία μοντέλου Administrator

Για τη διαχείριση των λογαριασμών των διαχειριστών υπεύθυνο είναι το μοντέλο administrator, το οποίο παράγεται, με τη βοήθεια του devise, εκτελώντας την εντολή:

```
~$ rails generate devise Administrator name
```

6.8.9.2. Sprint 1.8.2 – Δημιουργία Admin Controller

Στη φάση αυτή δημιουργείται το αρχείο controllers/admin_controller.rb για να οριστούν οι προσβάσεις του διαχειριστή. Το αρχείο αυτό περιέχει την παρακάτω κλάση:

```
class AdminController < ApplicationController
```

```
before_filter :authenticate_administrator!  
end
```

6.8.9.3. Sprint 1.8.3 – Παραμετροποίηση Ελεγκτή `controllers/admin/books_controller.rb`

Σε αυτό το Sprint γίνεται παραμετροποίηση του ελεγκτή `books_controller.rb` που βρίσκεται μέσα στον φάκελο `controllers/admin` έτσι ώστε να κληρονομεί ιδιότητες από τον ελεγκτή `controllers/admin_controller.rb` που δημιουργήθηκε κατά την υλοποίηση του sprint 1.8.2. Επομένως η γραμμή που ορίζει την κλάση στον ελεγκτή `books`, τροποποιείται ως εξής:

```
class BooksController < AdminController
```

Με τον τρόπο αυτό για όλες οι διαχειριστικές εργασίες απαιτείται πλέον εξουσιοδότηση.

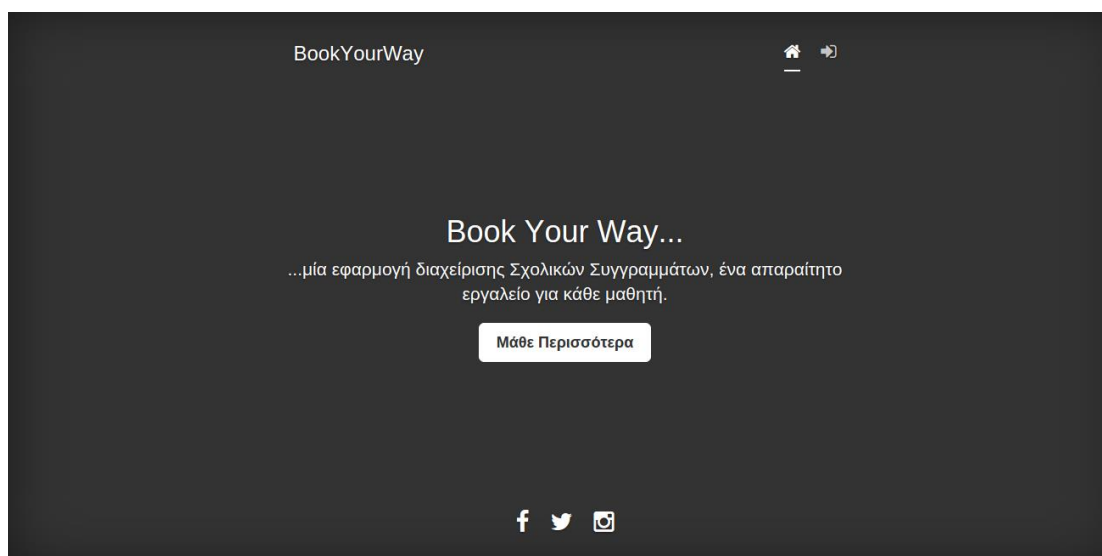
6.9. Στάδιο 6^ο – Ανασκόπηση Scrum

Μετά την υλοποίηση όλων των sprints κάθε έκδοσης, πραγματοποιείται συνάντηση μεταξύ του Product Owner, της Ομάδας ανάπτυξης και του πελάτη προκειμένου να παρουσιαστεί η νέα λειτουργικότητα της εφαρμογής. Στις παρακάτω ενότητες η παρουσίαση πραγματοποιείται με την παράθεση στιγμιότυπων (screenshots).

6.9.1. Ανασκόπηση Έκδοσης 1.0

Σκοπός της έκδοσης 1.0 ήταν η δημιουργία της αρχικής σελίδας της εφαρμογής. Για την επίτευξη αυτού του σκοπού ολοκληρώθηκαν συνολικά επτά sprints. Η παρακάτω εικόνα επιδεικνύει το τελικό αποτέλεσμα:

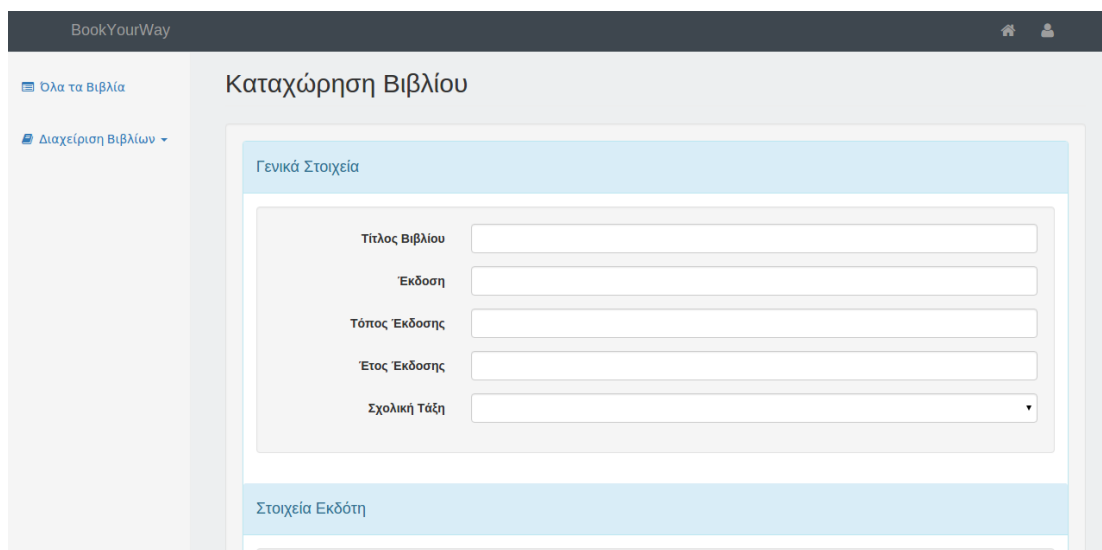
Εικόνα 3: Αρχική Σελίδα



6.9.2. Ανασκόπηση Έκδοσης 1.1

Η έκδοση 1.1 είχε ως στόχο τη δημιουργία εκείνης της λειτουργικότητας της εφαρμογής που θα επιτρέπει στο διαχειριστή του συστήματος τη καταχώρηση βιβλίων. Παράλληλα δημιουργήθηκαν η απαραίτητες όψεις για την υποστήριξη αυτής της λειτουργικότητας. Παρακάτω παρατίθενται εικόνες της εφαρμογής.

Εικόνα 4: Πρώτη οθόνη καταχώρησης βιβλίων



Εικόνα 5: Δεύτερη οθόνη καταχώρησης βιβλίων

BookYourWay

Όλα τα βιβλία

Διαχείριση βιβλίων ▾

Στοιχεία Εκδότη

Όνομα

Συγγραφείς

Όνομα Συγγραφέα

Επώνυμο Συγγραφέα

- Συγγραφέα

+ Συγγραφέα

Περιεχόμενο

Εικόνα 6: Τρίτη οθόνη καταχώρησης βιβλίων

BookYourWay

Όλα τα βιβλία

Διαχείριση βιβλίων ▾

Περιεχόμενο

Τίτλος Κεφαλαίου

- Κεφάλαιο

Τίτλος Ενότητας

- Ενότητα

Τίτλος Παραγράφου

Περιεχόμενο Παραγράφου

- Παράγραφος

+ Παράγραφος

Εικόνα 7: Τέταρτη οθόνη καταχώρησης βιβλίων

BookYourWay

Όλα τα Βιβλία

Διαχείριση Βιβλίων ▾

Τίτλος Παραγράφου

Περιεχόμενο Παραγράφου

- Παράγραφος

+ Παράγραφος

+ Ενότητα

+ Κεφάλαιο

Καταχώρηση Βιβλίου

6.9.3. Ανασκόπηση Έκδοσης 1.2

Με την ολοκλήρωση της έκδοσης 1.2 δημιουργήθηκε η όψη που προβάλλει όλα τα καταχωρημένα βιβλία σε ένα πίνακα. Έτσι ο διαχειριστής έχει την δυνατότητα να επιλέξει προβολή, επεξεργασία ή και διαγραφή ενός συγκεκριμένου βιβλίου. Η παρακάτω εικόνα επιδεικνύει το αποτέλεσμα αυτής της έκδοσης.

Εικόνα 8: Προβολή όλων των Βιβλίων

BookYourWay

Όλα τα Βιβλία

Διαχείριση Βιβλίων ▾

Όλα τα Βιβλία

Τίτλος	Εκδότης	Συγγραφείς			
Θρησκευτικά Α' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων	Γριζοπούλου	👁	✎	🗑
Μαθηματικά Α' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων	Βανδουλάκης	👁	✎	🗑
Πληροφορική Β' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων	Αράπογλου	👁	✎	🗑
Οικιακή Οικονομία Β' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων	Αμπελιώτης	👁	✎	🗑
Βιολογία Γ' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων	Μαυρικάκη	👁	✎	🗑
Νεότερη και Σύγχρονη Ιστορία Γ' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων	Λούβη	👁	✎	🗑

+

6.9.4. Ανασκόπηση Έκδοσης 1.3

Σκοπός της έκδοσης 1.3 ήταν η ανάπτυξη της λειτουργικότητας της εφαρμογής που αφορά στην επεξεργασία του εκάστοτε βιβλίου από τον διαχειριστή. Οι οθόνες της εφαρμογής κατά την επεξεργασία είναι όμοιες με τις οθόνες κατά τη διαδικασία της καταχώρησης εφόσον και στις δύο περιπτώσεις χρησιμοποιείται η ίδια φόρμα συμπλήρωσης στοιχείων. Παρακάτω παρατίθενται εικόνες της εφαρμογής.

Εικόνα 9: Πρώτη Οθόνη Επεξεργασίας Βιβλίου

BookYourWay

Όλα τα βιβλία

Διαχείριση Βιβλίων

Επεξεργασία Βιβλίου

Γενικά Στοιχεία

Τίτλος Βιβλίου	Θρησκευτικά Α' Γυμνασίου
Έκδοση	Επανέκδοση
Τόπος Έκδοσης	Αθήνα
Έτος Έκδοσης	2014
Σχολική Τάξη	Α' Γυμνασίου

Στοιχεία Εκδότη

Εικόνα 10: Δεύτερη Οθόνη Επεξεργασίας Βιβλίου

BookYourWay

Όλα τα βιβλία

Διαχείριση Βιβλίων

Επεξεργασία Βιβλίου

Στοιχεία Εκδότη

Όνομα	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
-------	--

Συγγραφείς

Όνομα Συγγραφέα	Όλγα
Επώνυμο Συγγραφέα	Γριζοπούλου

- Συγγραφέα

+ Συγγραφέα

Περιεχόμενο

Εικόνα 11: Τρίτη Οθόνη Επεξεργασίας Βιβλίου

BookYourWay

Όλα τα βιβλία

Διαχείριση Βιβλίων

Περιεχόμενο

Τίτλος Κεφαλαίου: Εισαγωγή: Κρατώντας στα χέρια μας το πρώτο τμήμα της Αγίας Γραφής, την Παλαιά Διαθήκη

Κεφάλαιο

Τίτλος Ενότητας: 1. Γιατί μελετούμε την Αγία Γραφή

Ενότητα

Τίτλος Παραγράφου: Ποιοι διαβάζουν την Αγία Γραφή;

Περιεχόμενο Παραγράφου: ένα σπουδαίο βιβλίο που το σέβονται και το μελετούν. Ποια είναι η θέση της Βίβλου στον παγκόσμιο πολιτισμό;

Παράγραφος

Τίτλος Παραγράφου: Ποια είναι η θέση της Βίβλου στον παγκόσμιο πολιτισμό;

Εικόνα 12: Τέταρτη Οθόνη Επεξεργασίας Βιβλίου

BookYourWay

Όλα τα βιβλία

Διαχείριση Βιβλίων

Τίτλος Παραγράφου: Ποια είναι η θέση της Αγίας Γραφής στη ζωή των χριστιανών;

Περιεχόμενο Παραγράφου: Για τους χριστιανούς όλου του κόσμου η Βίβλος είναι το ιερό τους βιβλίο. Ας δούμε γιατί: Μια έκφραση που συχνά χρησιμοποιούν γι' αυτήν είναι: ο λόγος του Θεού. Τι σημαίνει

Παράγραφος

+ Παράγραφος

+ Ενότητα

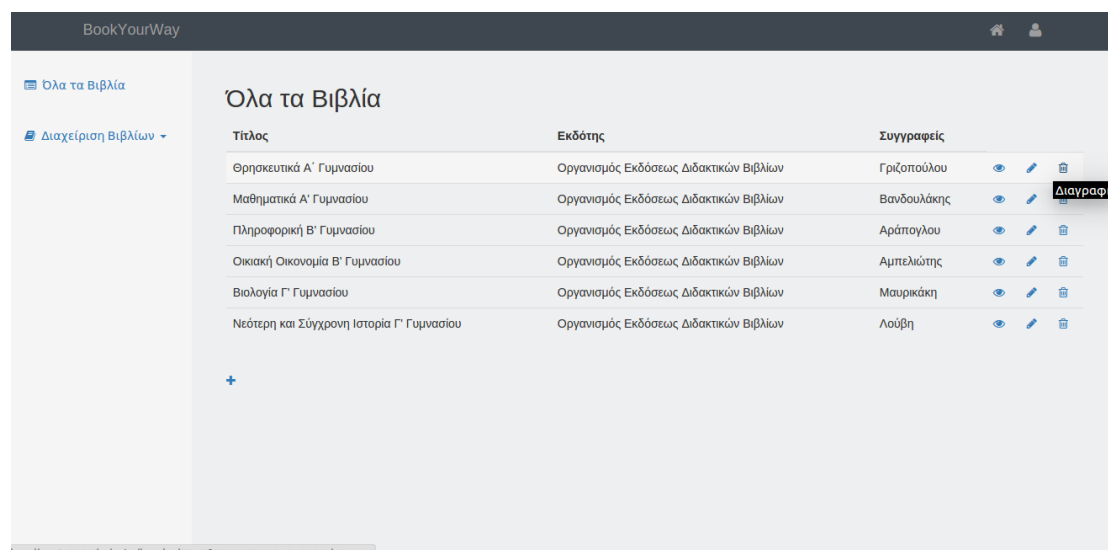
+ Κεφάλαιο

Ανανέωση Βιβλίου

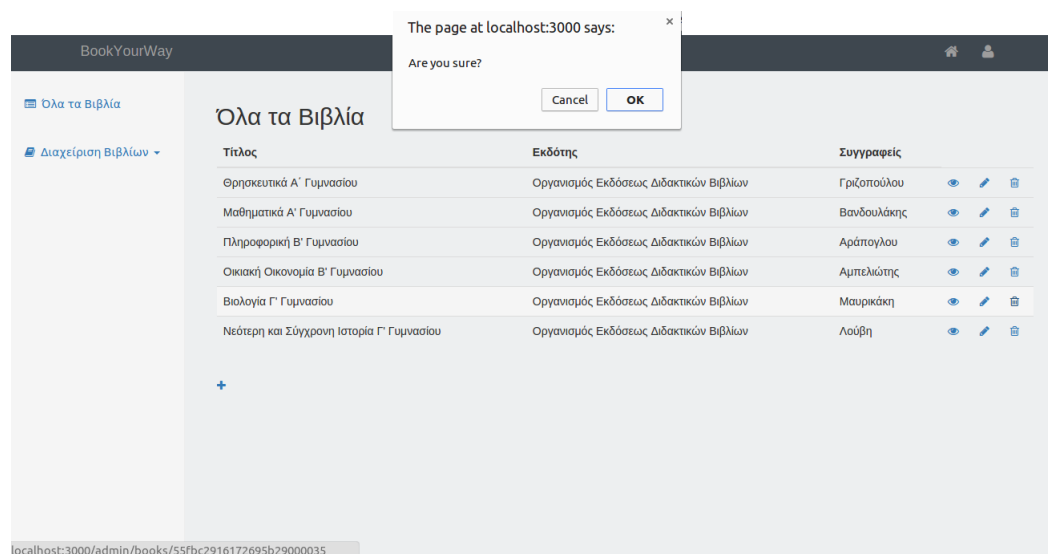
6.9.5. Ανασκόπηση Έκδοσης 1.4

Σε αυτή την έκδοση αναπτύχθηκε η λειτουργικότητα της εφαρμογής που αφορά στη διαγραφή των βιβλίων. Με τον τρόπο αυτό ο διαχειριστής πατώντας τον αντίστοιχο σύνδεσμο μπορεί να διαγράψει ένα βιβλίο μαζί με όλο το περιεχόμενό του. Οι κάτωθι οθόνες απεικονίζουν αυτήν τη λειτουργικότητα.

Εικόνα 13: Πρώτη Οθόνη Διαγραφής Βιβλίων



Εικόνα 14: Δεύτερη Οθόνη Διαγραφής Βιβλίων



6.9.6. Ανασκόπηση Έκδοσης 1.5

Η έκδοση 1.5 είχε ως στόχο την δημιουργία της λειτουργικότητας της εφαρμογής βάσει της οποίας τα βιβλία θα κατηγοριοποιούνται ανάλογα με τη σχολική βαθμίδα και την τάξη. Παρακάτω παρατίθενται η οθόνη όπου ο εγγεγραμμένος μαθητής θα βλέπει κατά την είσοδό του στο σύστημα. Στην οθόνη αυτή μπορεί να επιλέξει το βιβλίο που επιθυμεί ανά σχολική βαθμίδα και τάξη. Οι παρακάτω εικόνες επιδεικνύουν το τελικό αποτέλεσμα.

Εικόνα 15: Πρώτη Οθόνη Κατηγοριοποίησης Βιβλίων

BookYourWay

Όλο το Υλικό

Δημοτικό Γυμνάσιο Λύκειο

Τίτλος	Εκδότης
Θρησκευτικά Α' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Μαθηματικά Α' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Πληροφορική Β' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Οικιακή Οικονομία Β' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Βιολογία Γ' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Νεότερη και Σύγχρονη Ιστορία Γ' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων

Εικόνα 16: Δεύτερη Οθόνη Κατηγοριοποίησης Βιβλίων

BookYourWay

Όλο το Υλικό

Δημοτικό Γυμνάσιο Λύκειο

Υλικό ανά τάξη

- Α' Γυμνασίου
- Β' Γυμνασίου
- Γ' Γυμνασίου

Τίτλος	Εκδότης
Θρησκευτικά Α' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Μαθηματικά Α' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Πληροφορική Β' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Οικιακή Οικονομία Β' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Βιολογία Γ' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων
Νεότερη και Σύγχρονη Ιστορία Γ' Γυμνασίου	Οργανισμός Εκδόσεως Διδακτικών Βιβλίων

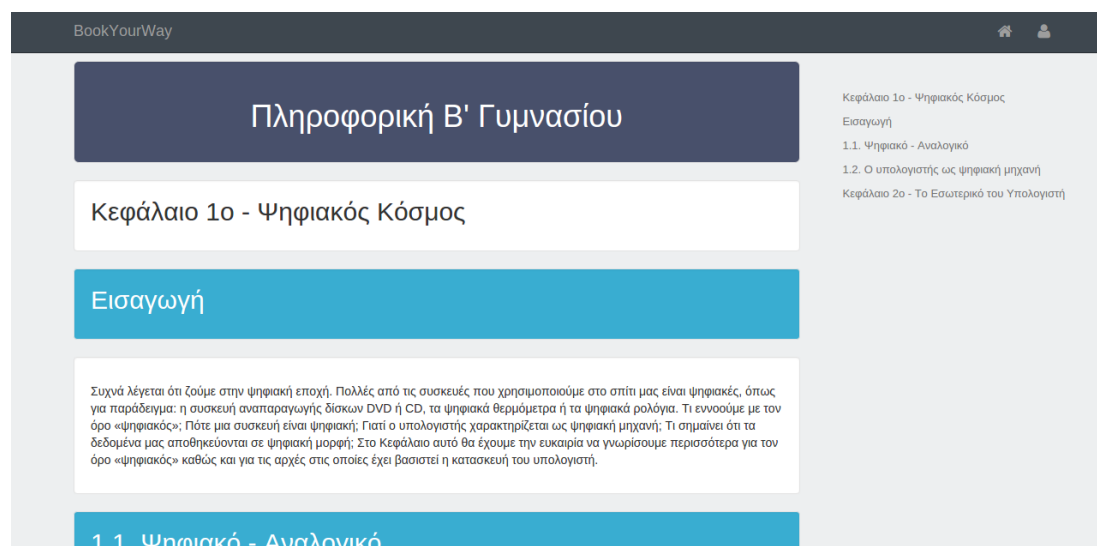
Εικόνα 17: Τρίτη Οθόνη Κατηγοριοποίησης Βιβλίων



6.9.7. Ανασκόπηση Έκδοσης 1.6

Με την έκδοση 1.6 αναπτύχθηκε η οθόνη προβολής του περιεχομένου κάθε βιβλίου. Στην εικόνα που ακολουθεί παρουσιάζεται το τελικό αποτέλεσμα.

Εικόνα 18: Προβολή Περιεχομένου Βιβλίου

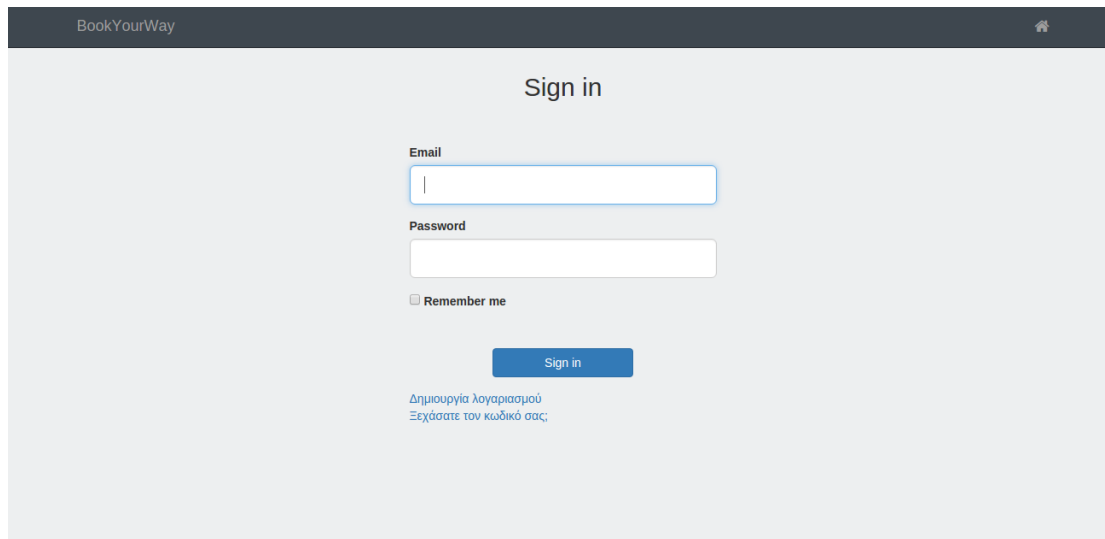


6.9.8. Ανασκόπηση Εκδόσεων 1.7 & 1.8

Σε αυτές τις εκδόσεις αναπτύχθηκε η λειτουργικότητα της εφαρμογής που αφορά στην ταυτοποίηση του χρήστη-μαθητή και του χρήστη-διαχειριστή. Δημιουργήθηκαν

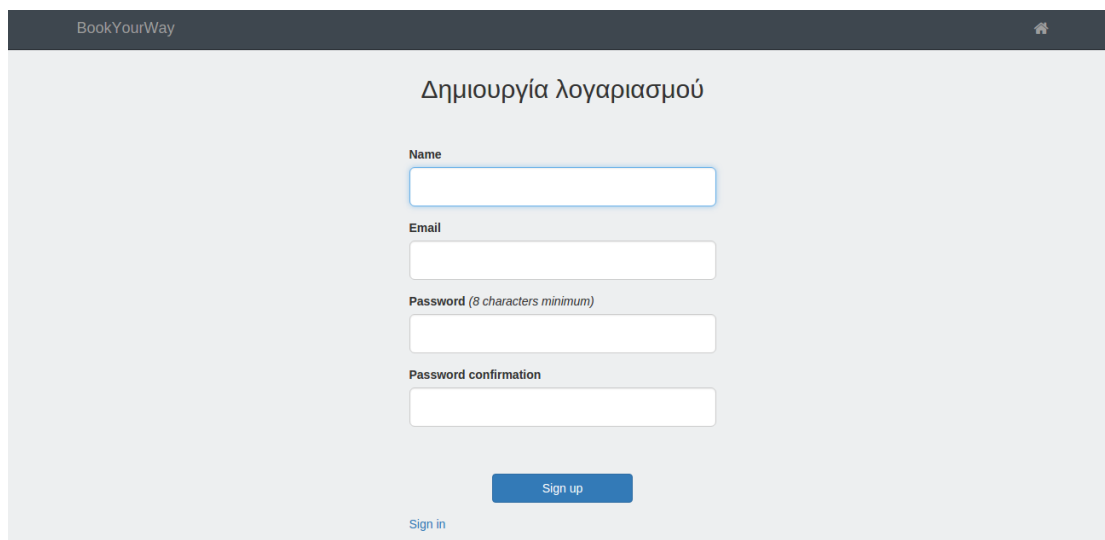
οθόνες για την είσοδο, την δημιουργία λογαριασμού και την επανέκδοση κωδικών κοινές για όλους τους χρήστες (μαθητή και διαχειριστή). Παρακάτω παρατίθενται ανάλογες εικόνες.

Εικόνα 19: Sign In



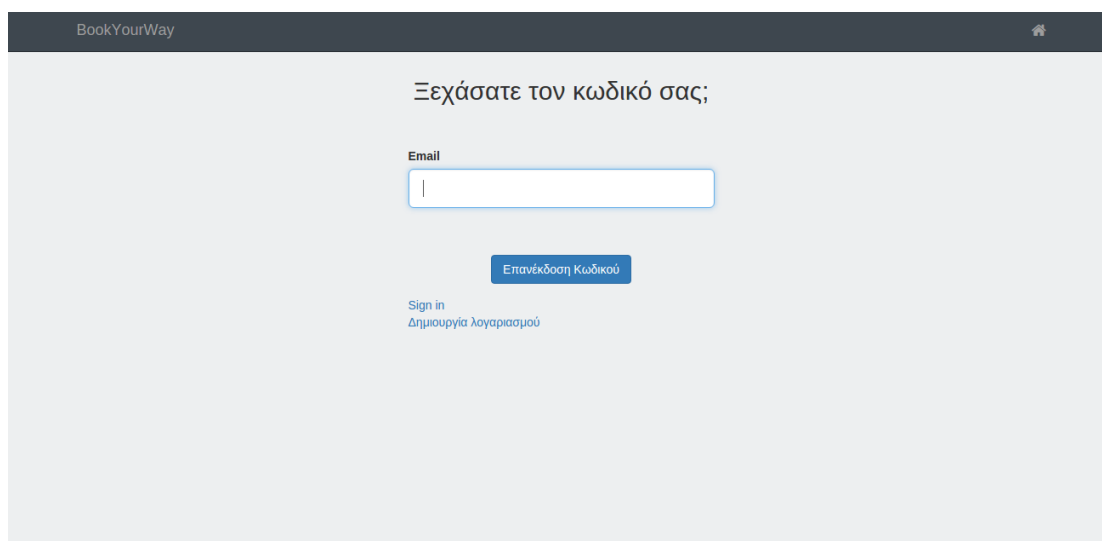
The screenshot shows the 'Sign in' page of the BookYourWay application. At the top, there is a dark header with the text 'BookYourWay' on the left and a home icon on the right. The main content area has a light gray background. The title 'Sign in' is centered at the top of the form. Below the title, there are two input fields: 'Email' and 'Password'. The 'Email' field contains a vertical cursor. Below the 'Password' field, there is a checkbox labeled 'Remember me'. A blue button with the text 'Sign in' is positioned below the checkbox. At the bottom of the form, there are two lines of text: 'Δημιουργία λογαριασμού' and 'Ξεχάσατε τον κωδικό σας;'. The entire form is centered on the page.

Εικόνα 20: Sign Up



The screenshot shows the 'Δημιουργία λογαριασμού' (Sign Up) page of the BookYourWay application. At the top, there is a dark header with the text 'BookYourWay' on the left and a home icon on the right. The main content area has a light gray background. The title 'Δημιουργία λογαριασμού' is centered at the top of the form. Below the title, there are four input fields: 'Name', 'Email', 'Password (8 characters minimum)', and 'Password confirmation'. A blue button with the text 'Sign up' is positioned below the 'Password confirmation' field. At the bottom left of the form, there is a link labeled 'Sign in'. The entire form is centered on the page.

Εικόνα 21: Επανεκδοση Κωδικού Πρόσβασης



BookYourWay

Ξεχάσατε τον κωδικό σας;

Email

Επανεκδοση Κωδικού

Sign in
Δημιουργία λογαριασμού

6.10. Στάδιο 7^ο – Ανατροφοδότηση Scrum

Στο τέλος κάθε sprint ο Scrum Master, ο Product Owner και η Ομάδα Ανάπτυξης ανταλλάσσουν απόψεις σχετικά με την πορεία του έργου, τα εργαλεία και τις βιβλιοθήκες λογισμικού που χρησιμοποιούνται καθώς και πιθανές βελτιώσεις στον τρόπο λειτουργίας της εφαρμογής.

Βάσει του σεναρίου αυτής της εργασίας η Ομάδα Ανάπτυξης, ο Scrum Master και ο Product Owner κατέληξαν ομόφωνα ότι κατά την υλοποίηση της δεύτερης έκδοσης της εφαρμογής θα εξερευνηθούν τρόποι βελτίωσης του front-end της εφαρμογής χρησιμοποιώντας JavaScript βιβλιοθήκες όπως το AngularJS για τον επαναπροσδιορισμό και την ανασχεδίαση των διεπαφών της εφαρμογής.

Επιπλέον θα ενσωματωθεί πλήρως στην εφαρμογή η μηχανή αναζήτησης ElasticSearch προκειμένου να είναι εφικτή η αναζήτηση κειμενικών δεδομένων σε πραγματικό χρόνο και να μπορούν οι χρήστες, εύκολα, γρήγορα και αποτελεσματικά, να εντοπίζουν λέξεις, φράσεις η και ολόκληρες προτάσεις μέσα από εκατοντάδες βιβλία.

ΚΕΦΑΛΑΙΟ 7: ΕΠΙΣΚΟΠΗΣΗ ΠΑΡΟΜΟΙΩΝ ΕΦΑΡΜΟΓΩΝ

7.1. Εισαγωγή

Η εφαρμογή που πραγματεύεται η εργασία έχει χαρακτηριστικά ενός Συστήματος Διαχείρισης Περιεχομένου. Στο παρόν κεφάλαιο γίνεται επισκόπηση παρόμοιων εφαρμογών.

7.2. Συστήματα Διαχείρισης Περιεχομένου

Το Σύστημα Διαχείρισης Περιεχομένου (ΣΔΠ, Content management System, CMS) είναι μια εφαρμογή λογισμικού που χρησιμοποιείται για τη συλλογή, οργάνωση και παρουσίαση περιεχομένου μέσα από ένα σύνολο διαδικασιών. Σκοπός του ΣΔΠ είναι η αυτοματοποίηση των επαναλαμβανόμενων διαδικασιών καθώς και η εύκολη υλοποίησή τους χωρίς την απαίτηση τεχνικών γνώσεων από τους χρήστες.

Σύμφωνα με τον Boiko (2005), ένα ΣΔΠ αποτελείται από τρία διαφορετικά αλληλεξαρτώμενα συστήματα. Αυτά είναι το σύστημα συλλογής, το σύστημα διαχείρισης και το σύστημα δημοσίευσης. Το σύστημα συλλογής είναι υπεύθυνο για τις διαδικασίες που πραγματοποιούνται έτσι ώστε να μετατραπεί μια ακατέργαστη πληροφορία σε κατάλληλο προς δημοσίευση περιεχόμενο. Οι διαδικασίες αυτές αφορούν:

- Στη συγγραφή του περιεχομένου.
- Στη συγκέντρωση του περιεχομένου από υπάρχουσα πηγή.
- Στην τροποποίηση και επεξεργασία του περιεχομένου αποβάλλοντας περιττές πληροφορίες και αναδιαμορφώνοντας τη δομή του.
- Στην αλλαγή του τρόπου παρουσίασης

Το σύστημα διαχείρισης είναι υπεύθυνο για την αποθήκευση του περιεχομένου καθώς και για κάθε είδους αρχείο που χρησιμοποιείται. Συγκεκριμένα περιλαμβάνει διαδικασίες που αφορούν:

- Στην αποθήκευση στη βάση δεδομένων.

- Στις ρυθμίσεις για την παραμετροποίηση και τη διαχείριση του ΣΔΠ.
- Στη ροή εργασιών, δηλαδή τα στάδια που θα πρέπει να πραγματοποιηθούν έτσι ώστε το περιεχόμενο να καταστεί έτοιμο προς δημοσίευση.

Το σύστημα δημοσίευσης είναι υπεύθυνο για την εξαγωγή περιεχομένου από την αποθήκη δεδομένων και την μετέπειτα δημιουργία δημοσιεύσεων. Το σύστημα αυτό παρέχει:

- Εργαλεία που ελέγχουν τι δημοσιεύεται και πώς.
- Ειδικά πρότυπα που δημιουργούν αυτόματες δημοσιεύσεις
- Υπηρεσίες για τη δημιουργία δυναμικών ιστοσελίδων

Μολονότι δεν είναι δυνατή η απόλυτη σύγκριση της εφαρμογής που πραγματεύεται η παρούσα εργασία με τα Συστήματα Διαχείρισης Περιεχομένου, διότι το πεδίο εφαρμογής τους είναι σαφώς ευρύτερο, κοινό γνώρισμα είναι οι διαδικασίες που συντελούνται για την ολοκληρωμένη δημιουργία, αποθήκευση και παρουσίαση του περιεχομένου.

7.3. Παρόμοιες εφαρμογές

Εστιάζοντας αποκλειστικά σε εφαρμογές οι οποίες υποστηρίζουν τη δημιουργία, αποθήκευση και παρουσίαση ηλεκτρονικών βιβλίων, μπορούν καταρχήν να ταξινομηθούν σε αυτόνομες και διαδικτυακές.

Αυτόνομες εφαρμογές είναι εκείνες που εγκαθίσταται σε τοπικούς ηλεκτρονικούς υπολογιστές ή κινητές συσκευές και δεν χρειάζονται κάποιο άλλο λογισμικό, υλικό ή δίκτυο για να μπορέσουν να εκτελεστούν, καθώς επίσης δεν απαιτείται η ύπαρξη κάποιας βάσης δεδομένων.

Οι διαδικτυακές εφαρμογές είναι προσβάσιμες μέσω ενός δικτύου, κωδικοποιούνται σε ένα πρόγραμμα περιήγησης και δεν απαιτούν πολύπλοκες διαδικασίες για να εκτελεστούν. Επίσης δεν χρειάζονται χώρο στον τοπικό δίσκο του πελάτη και είναι συμβατές με τα περισσότερα λειτουργικά συστήματα εφόσον τρέχουν σε ένα κοινό φυλλομετρητή. Οι εφαρμογές αυτές πολλές φορές ονομάζονται και υπηρεσίες.

Πέρα από τη διάκριση σε αυτόνομες ή διαδικτυακές, οι παρακάτω εφαρμογές εξετάζονται και ως προς τις λειτουργίες που περιλαμβάνουν, την ευκολία στη χρήση, τη δυνατότητα μορφοποίησης καθώς και τους τύπους των αρχείων που υποστηρίζουν.

7.3.1. Creatavist

Το Creatavist αποτελεί μια διαδικτυακή εφαρμογή δημιουργίας η-βιβλίων. Διακρίνεται κυρίως για τους πολλαπλούς τρόπους μορφοποίησης των ψηφιακών βιβλίων περιοδικών και photobooks και συγκαταλέγεται στις κορυφαίες εφαρμογές του είδους του.

Χαρακτηριστικά

- Δυνατότητα επιβολής περιορισμών δοκιμαστικής περιόδου για ό,τι δημοσιεύεται
- Δημιουργία δημόσιας σελίδας για την εμφάνιση των η-βιβλίων σε υποψήφιους αγοραστές
- Χρήση λογότυπου στα ψηφιακά έργα (απαιτείται μηνιαία συνδρομή) και δυνατότητα προστασίας τους με τη χρήση κωδικού πρόσβασης
- Μετάφραση των η-βιβλίων σε άλλες γλώσσες όπως γαλλικά, ισπανικά και γερμανικά
- Χρήση επεξεργαστή φωτογραφιών, για την εύκολη αλλαγή μεγέθους και κατάλληλης τοποθέτησής τους βάσει των προτιμήσεων του χρήστη
- Δημιουργία κάθε τύπου η-βιβλίου και ψηφιακών εγγράφων όπως ταξιδιωτικούς οδηγούς, περιοδικά, φωτογραφικά άλμπουμ, εκπαιδευτικά βιβλία, οδηγοί μελέτης, εικονογραφημένα βιβλία κ.α. με απλό και εύκολο τρόπο (drag & drop)
- Σύνδεση με υπηρεσίες διάθεσης και δημοσίευσης η-βιβλίων όπως η Amazon και η Barnes & Noble
- Υποστήριξη πολλών διαφορετικών αρχείων και ενσωμάτωσή τους στα η-βιβλία όπως: φωτογραφίες, video, αρχεία ήχου, χάρτες, PDF, GIF, slideshow, charts καθώς και κώδικα HTML και JavaScript
- Υποστήριξη για την εκμάθηση των λειτουργιών της εφαρμογής μέσα από ειδικά σεμινάρια, tutorials και εκτενείς οδηγούς
- Εξαγωγή του τελικού η-βιβλίου σε μορφές αρχείων που υποστηρίζονται από iBooks, Kindle, Kobo, Nook, ePub και web.

7.3.2. eBook Maestro Pro

Το eBook Maestro PRO είναι μία απλή και αποτελεσματική αυτόνομη εφαρμογή η οποία παρέχει λειτουργίες και δυνατότητες μορφοποίησης που βοηθούν στη δημιουργία υψηλής ποιότητας ηλεκτρονικών η-βιβλίων. Παρόλο που η σχεδίαση της εφαρμογής είναι ξεπερασμένη, παρέχει αρκετές λειτουργίες που την καθιστούν ξεχωριστή.

Χαρακτηριστικά

- Δημιουργία ηλεκτρονικών βιβλίων, περιοδικών, οδηγών, εκθέσεων, τεκμηριώσεων, επιμορφωτικών μαθημάτων, εκπαιδευτικού υλικού κ.α.
- Επιλογή έτοιμων προτύπων για τη γρήγορη δημιουργία η-βιβλίων
- Δυνατότητα ενσωμάτωσης πολλών αρχείων σε ένα η-βιβλίο όπως: HTML σελίδες, graphic files, Java Scripts, VB Scripts, CSS, Flash files, Shockwave files, Sound files, Video files, Java Applets κ.α.
- Υποστήριξη text-to-speech
- Δυνατότητα ανεύρεσης λέξεων-κλειδιών με αυτόματο μαρκάρισμα
- Δημιουργία ανεξάρτητου εκτελέσιμου EXE αρχείου του τελικού η-βιβλίου.

7.3.3. iBooks Author

Το iBooks Author είναι μια αυτόνομη εφαρμογή δημιουργίας και έκδοσης η-βιβλίων προσφέροντας ένα ευρύ φάσμα επιλογών οι οποίες ενισχύουν τη δημιουργικότητα.

Χαρακτηριστικά

- Δυνατότητα χρήσης λογότυπου
- Μετάφραση των η-βιβλίων σε άλλες γλώσσες
- Ευκολία στη χρήση
- Δημιουργία διαφορετικών τύπων η-βιβλίων όπως μυθιστορήματα, εικονογραφημένα βιβλία, άλμπουμ φωτογραφιών, κουίζ, εκπαιδευτικά εγχειρίδια κ.α.

- Ενσωμάτωση πολλαπλών τύπων αρχείων όπως: εικόνες, ήχος, video, αρχεία keynote, αρχεία 3D (.dae), HTML καθώς και widgets όπως αναδυόμενα παράθυρα, γκαλερί και διαδραστικές εικόνες
- Εξαγωγή του η-βιβλίου ως αρχείο PDF ή σε μορφή TXT.

7.3.4. Lucidpress

Το Lucidpress είναι μια διαδικτυακή εφαρμογή (υπηρεσία) δημιουργίας η-βιβλίων.

Χαρακτηριστικά

- Δεν απαιτείται εγκατάσταση, η εφαρμογή εκτελείται στον browser
- Δυνατότητα χρήσης πολλών διαφορετικών αρχείων για τη δημιουργία των η-βιβλίων
- Έτοιμα πρότυπα για τη σχεδίαση διαφημιστικών εντύπων, φυλλαδίων, περιοδικών, αφισών, ετήσιων εκθέσεων εταιρειών καθώς και ειδικά πρότυπα για εκτυπωτικές εργασίες
- Ευκολία στη σχεδίαση με λειτουργία drag and drop.
- Συνεργασία σε πραγματικό χρόνο για τη δημιουργία περιεχομένου, με δυνατότητα διαβαθμισμένης πρόσβασης, μεταξύ των μελών μιας ομάδας
- Εξαγωγή του τελικού η-βιβλίου σε μορφή αρχείου PDF, JPG ή PNG
- Δυνατότητα διαμοιρασμού σε Facebook, LinkedIn, Google+, καθώς και ενσωμάτωσης σε ιστοσελίδα χρησιμοποιώντας embedded code.

7.3.5. Pressbooks

Το Pressbooks ανήκει στις διαδικτυακές εφαρμογές δημιουργίας η-βιβλίων παρέχοντας πολλαπλές λειτουργίες παραμετροποίησης κυρίως σε συγγραφείς μυθιστορημάτων και εγχειριδίων.

Χαρακτηριστικά

- Επικεντρώνεται κυρίως στη δημιουργία μυθιστορημάτων, απομνημονευμάτων, ακαδημαϊκών κειμένων, λογοτεχνικών περιοδικών και σχολικών βιβλίων

- Δυνατότητα χρήσης αρχείων HTML, MOBI και PDF καθώς και ευχέρεια χρήσης αρχείων εικόνων. Ωστόσο δεν επιτρέπεται η εισαγωγή αρχείων ήχου και video
- Επιλογή διαφορετικών προτύπων για επαγγελματικό σχεδιασμό η-βιβλίων
- Δυνατότητα μετάφρασης σε διάφορες γλώσσες
- Δημοσίευση βιβλίων σε μορφή αρχείου PDF, EPUB, MOBI, ICML (InDesign), XHTML, Pressbooks XML και WordPress XML.

7.6. Συμπεράσματα

Ο πίνακας που ακολουθεί παρέχει συγκεντρωτικά στοιχεία των κύριων χαρακτηριστικών των εφαρμογών που εξετάστηκαν.

Πίνακας 18: Σύγκριση Παρόμοιων Εφαρμογών

	Creatavist	eBook Maestro Pro	iBooks Author	Lucidpress	Pressbooks
Γενικές Λειτουργίες					
Δοκιμαστική Περίοδος	✓	✓			✓
Προστασία Περιεχομένου	✓	✓	✓	✓	✓
Εισαγωγή Λογότυπου	✓	✓	✓	✓	✓
Δυνατότητα Συγγραφής σε άλλες Γλώσσες (εκτός αγγλικής)	✓	✓	✓	✓	✓
Δυνατότητα Πώλησης των η-βιβλίων	✓	✓	✓	✓	✓
Επεξεργασία Φωτογραφιών	✓	✓	✓	✓	✓
Μορφές η-Βιβλίων					
Οδηγοί	✓	✓	✓	✓	✓

Περιοδικά	✓	✓	✓	✓	
Διδακτικό Υλικό	✓	✓	✓	✓	
Tests	✓	✓	✓	✓	
Φωτογραφικά Άλμπουμ	✓	✓	✓	✓	
Εκπαιδευτικά Εγχειρίδια	✓	✓	✓	✓	✓
Υποστηριζόμενα Αρχεία					
HTML	✓	✓	✓	✓	✓
Φωτογραφίες	✓	✓	✓	✓	✓
Ήχος	✓	✓	✓	✓	
Video	✓	✓	✓	✓	
MOBI	✓	✓	✓	✓	✓
EPUB	✓	✓	✓	✓	✓
PDB	✓		✓	✓	✓
LIT	✓		✓	✓	
PDF	✓	✓	✓	✓	✓
Βοηθήματα & Υποστήριξη					
Tutorials	✓	✓	✓	✓	✓
Online Υποστήριξη	✓	✓	✓	✓	✓
Εγχειρίδιο Οδηγιών	✓	✓	✓		

Από την παραπάνω επισκόπηση των εφαρμογών είναι έκδηλο ότι η διαχείριση περιεχομένου αφορά τέσσερις δραστηριότητες:

1. Δημιουργία
2. Επεξεργασία
3. Διαγραφή

4. Προβολή

Αυτό που αλλάζει σε κάθε εφαρμογή είναι ο τρόπος με τον οποίο υλοποιούνται οι ανωτέρω δραστηριότητες.

ΚΕΦΑΛΑΙΟ 8: ΣΥΜΠΕΡΑΣΜΑΤΑ

8.1. Εισαγωγή

Η παραπάνω διπλωματική εργασία αφορά στη διαχείριση ενός έργου πληροφορικής καθώς και στην ανάπτυξη μιας διαδικτυακής εφαρμογής διαχείρισης εκπαιδευτικών συγγραμμάτων.

Κυριότερα προτείνεται μια πρωτότυπη εκπαιδευτική εφαρμογή με συγκεκριμένες λειτουργίες, οι οποίες συμβάλλουν στην καλύτερη διαχείριση των εκπαιδευτικών συγγραμμάτων. Διευκολύνει την καταχώρηση, ανανέωση και παρουσίαση βιβλίων στους μαθητές. Στόχος αυτής της εφαρμογής είναι η ελκυστικότερη και ευκολότερη πρόσβαση από τους μαθητές στη σχολική ύλη, η οποία επιτυγχάνεται μέσω των διαφορετικών προσεγγίσεων στην προβολή των βιβλίων. Η εφαρμογή αυτή παύει τη στατικότητα των απλών βιβλίων και αναπτύσσει τη δια-δραστικότητα η οποία αφυπνίζει το ενδιαφέρον του μαθητή και προσδίδει ιδιαίτερη ευελιξία στην πλοήγησή του.

8.2. Λειτουργικότητα & Ευχρηστία

Η εφαρμογή έχει τα κάτωθι χαρακτηριστικά:

- Είναι διαδικτυακή, δηλαδή εκτελείται στον browser πολλών διαφορετικών συσκευών και λειτουργικών συστημάτων. Έτσι ελαχιστοποιείται η ανάγκη δημιουργίας ξεχωριστού κώδικα, διευκολύνονται οι προγραμματιστικές εργασίες, μεγιστοποιείται η προσβασιμότητα των χρηστών και τα η-βιβλία εμφανίζονται το ίδιο ελκυστικά σε υπολογιστές desktop, κινητά και tablets
- Δύναται να χρησιμοποιηθεί για τη δημιουργία σχολικών βιβλίων, επιμορφωτικών μαθημάτων, εκπαιδευτικού υλικού και εγχειριδίων
- Η εμφάνιση των βιβλίων γίνεται με τέτοιο τρόπο ώστε να παρέχεται ευκολία στην ανάγνωση και στην αναζήτηση
- Η δημιουργία των βιβλίων γίνεται με τρόπο απλό και κατανοητό ακόμα και από χρήστες που δεν έχουν ιδιαίτερη εξοικείωση με ανάλογες εφαρμογές.

Η ευχρηστία της εφαρμογής αφορά στην επίτευξη των παρακάτω στόχων:

- **Αποδοτικότητα.** Η εφαρμογή επιτρέπει στους χρήστες να κάνουν τις εργασίες τους αποδοτικά, επιτρέποντάς τους να καταχωρήσουν τις πληροφορίες που χρειάζονται για τη δημιουργία των η-βιβλίων
- **Αποτελεσματικότητα.** Οι χρήστες, εφόσον μάθουν να χρησιμοποιούν την εφαρμογή για να επιτελούν τις εργασίες τους, μπορούν να διατηρήσουν ένα υψηλό επίπεδο παραγωγικότητας
- **Ασφάλεια.** Για την αποφυγή σφαλμάτων εμφανίζονται ειδικά μηνύματα στους χρήστες, όταν για παράδειγμα δεν έχουν συμπληρώσει ένα απαραίτητο πεδίο ή όταν επιθυμούν τη διαγραφή ενός καταχωρημένου βιβλίου
- **Ωφέλεια.** Η εφαρμογή παρέχει ένα κατάλληλο σύνολο λειτουργιών, οι οποίες επιτρέπουν στους χρήστες να εκτελούν τις εργασίες τους με έναν καθορισμένο τρόπο έτσι ώστε να επιτρέπεται η δημιουργία του η-βιβλίου
- **Δυνατότητα εκμάθησης.** Οι χρήστες μπορούν να μάθουν να χρησιμοποιούν την εφαρμογή χωρίς ιδιαίτερη προσπάθεια, μελετώντας τη διεπαφή και δοκιμάζοντας ορισμένες ενέργειες. Έτσι μπορούν να αρχίσουν αμέσως να κάνουν τις εργασίες τους χωρίς να ξοδεύουν χρόνο για την εκμάθησή του
- **Απομνημόνευση.** Οι λειτουργίες της εφαρμογής είναι σαφείς, λογικές και σε αλληλουχία. Έτσι οι χρήστες βοηθούνται να θυμούνται τον τρόπο με τον οποίο χρησιμοποιείται η εφαρμογή έστω και αν δεν τη χειρίζονται συχνά.

8.3. Μελλοντικές Επεκτάσεις

Η παρούσα πρόταση αναπτύσσεται σε ένα πρώτο επίπεδο και ως εκ τούτου είναι λογικό και αναμενόμενο ότι επιδέχεται βελτιώσεις ή τυχόν αλλαγές. Οι μη υποστηριζόμενες λειτουργίες της εφαρμογής αποτελούν ταυτόχρονα και το έναυσμα για τις μελλοντικές τροποποιήσεις και επεκτάσεις. Αυτές απαριθμούνται παρακάτω:

1. Επεξεργασία κειμένου και μορφοποίηση χρησιμοποιώντας WYSIWYG editor
2. Εισαγωγή αρχείων εικόνων, ήχου, video, PDF, .docx, MOBI και ePub
3. Εξαγωγή αρχείων σε μορφή PDF ή ePub3 και αποστολή τους μέσω e-mail
4. Δυναμική αναζήτηση σε όλα τα βιβλία ή σε κάθε βιβλίο ξεχωριστά
5. Δυνατότητα text to speech
6. Δημιουργία αντιγράφων σε τοπικό δίσκο, Google Drive, Dropbox κ.α.

Τέλος κρίνεται σκόπιμο σε μεταγενέστερο στάδιο η επανασχεδίαση της εφαρμογής χρησιμοποιώντας το πλαίσιο AngularJS ως front-end και το Ruby on Rails ως back-end (rails API), έτσι ώστε να γίνει πλήρης διαχωρισμός των διαδραστικών στοιχείων και των διεπαφών της εφαρμογής από τα δεδομένα καθώς επίσης να είναι εφικτή η επαναχρησιμοποίηση των καταχωρημένων η-βιβλίων σε άλλες εφαρμογές.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Ελληνική

Bell, J. (2007). *Πώς να Συντάξετε μια Επιστημονική Εργασία: Οδηγός Ερευνητικής Μεθοδολογίας*. Απόδοση: Πανάγου Ελεάννα, 1^η Έκδοση, Αθήνα: Μεταίχμιο.

Rogers, Y., Sharp, H. & Preece, J. (2011). *Σχεδίαση Διαδραστικότητας: Επεκτείνοντας την Αλληλεπίδραση Ανθρώπου – Υπολογιστή*. Απόδοση: Σαμαράς Γιάννης, 3^η Έκδοση, Αθήνα: Μ. Γκιούρδας.

Ξένη

Athayde, J. & Williams, B. (2012). *The Rails View: Creating a Beautiful and Maintainable User Experience*. North Carolina: The Pragmatic Programmers.

Cohn, M. (2012). *Succeeding with Agile – Software Development using Scrum*. Michigan: Addison-Wesley.

DeCarlo, D. (2004). *eXtreme Project Management: Using Leadership, Principles and Tools to Deliver Value in the Face of Volatility*. Jossey-Bass.

Gamble, A., Carneiro, C. & Barazi, R. (2013). *Beginning Rails 4*. 3rd Edition, New York: Apress Media.

Hows, D., Membrey, P. & Plugge, E. (2014). *MongoDB Basics*. New York: Apress Media.

Koch, H. & Eisenbarth, T. (2014). *Growing Rails Applications in Practice*. Augsburg: Makandra, Leanpub.

Layton, M. C. (2012). *Agile Project Management For Dummies*. New Jersey: John Wiley & Sons Inc.

Paro, A. (2015). *ElasticSearch Cookbook*. 2nd Edition, Birmingham: Packt Publishing.

Pham, A. & Pham, P. (2012). *Scrum in Action: Agile Software Project Management and Development*. Boston: Course Technology.

Project Management Institute (PMI). (2013). *A Guide to the Project Management Body of Knowledge*. 5th Edition, Newtown Square, PA: Project Management Institute.

Rege, G. (2012). *Ruby and MongoDB Web Development Beginner's Guide*. Birmingham: Packt Publishing.

Richardson, L. & Amundsen, M. (2013). *RESTful Web APIs*. Sebastopol: O'Reilly.

Robertson, S. & Robertson, J. (2006). *Mastering the Requirements Process*. 2nd Edition, Boston: Addison Wesley.

Ruby, S., Thomas, D. & Hansson, D. (2012). *Agile Web Development with Rails 4*. North Carolina: Pragmatic Programmers.

Schaffer, E. & Lahiri, A. (2014). *Institutionalization of UX. A Step-by-Step Guide to a User Experience Practice*. 2nd Edition, Boston: Addison Wesley.

Smart, J. (2015). *BDD in Action: Behavior-Driven Development for the whole software lifecycle*. Shelter Island: Manning Publications Co.

Valim, J. (2013). *Crafting Rails 4 Applications*. North Carolina: Pragmatic Programmers.

Wysocki, R. K. (2014). *Effective Project Management: Traditional, Agile, Extreme*. 7th Edition, Indianapolis: Wiley.

Διαδικτυακοί Τόποι

Beck, Kent et al. (2001). *Manifesto for Agile Software Development*. Agile Alliance. Ανακτήθηκε 4 Μαΐου, 2015, από <http://agilemanifesto.org/iso/el/>

Βικιπαίδεια. (8 Φεβρουαρίου, 2015). *Αντικειμενοστρεφής Προγραμματισμός*. Ανακτήθηκε 2 Ιουνίου, 2015, από http://el.wikipedia.org/wiki/Αντικειμενοστρεφής_προγραμματισμός

Βικιπαίδεια. (17 Ιουλίου, 2013). *Δομή Δεδομένων*. Ανακτήθηκε 2 Ιουνίου, 2015, από http://el.wikipedia.org/wiki/Δομή_δεδομένων

Βικιπαίδεια. (28 Σεπτεμβρίου, 2014). *Model-View-Controller*. Ανακτήθηκε 5 Ιουνίου, 2015, από <https://el.wikipedia.org/wiki/Model-view-controller>

- Βικιπαίδεια. (25 Απριλίου, 2015). *Object Relational Mapping*. Ανακτήθηκε 30 Απριλίου, 2015, από https://en.wikipedia.org/wiki/Object-relational_mapping
- Βικιπαίδεια. (19 Ιουλίου, 2015). *Representational State Transfer*. Ανακτήθηκε 21 Ιουλίου, 2015, από https://en.wikipedia.org/wiki/Representational_state_transfer
- Βικιπαίδεια. (9 Σεπτεμβρίου, 2014). *Σχεδιαστικά Πρότυπα*. Ανακτήθηκε 31 Μαρτίου, 2015, από http://el.wikipedia.org/wiki/Σχεδιαστικά_πρότυπα
- Fernandez, D. J. & Fernandez, J. D. (2009). *Agile Project Management – Agilism Versus Traditional Approaches*. Ανακτήθηκε 27 Μαΐου, 2015, από http://iacis.org/jcis/articles/Fernandez_Fernandez_200809_49_2.pdf
- Gormley, C. & Tong, Z. (2014). *ElasticSearch: The Definitive Guide*. Ανακτήθηκε 11 Ιουνίου, 2015, από <https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html>
- Hartl, M. (2015). *The Ruby on Rails Tutorial: Learn Web Development With Rails*. Ανακτήθηκε 23 Απριλίου, 2015, από <https://www.railstutorial.org/book>
- Hass, K. (2007). *The Blending of Traditional and Agile Project Management*. Ανακτήθηκε 25 Μαΐου, 2015, από <https://www.projectsmart.co.uk/the-blending-of-traditional-and-agile-project-management.php>
- Κέντρο Πληροφορίας & Νέων Τεχνολογιών Ν. Φλώρινας. *Θεωρία Βάσεων Δεδομένων*. Ανακτήθηκε 8 Ιουνίου, 2015, από <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-DataBasesTheory.html>
- Kohn, M. (2009). *Introduction to Scrum Methodology*. Ανακτήθηκε 6 Μαΐου, 2015, από <https://www.scrumalliance.org/resources/47>
- Merriam Webster Dictionary. *Definition of Database*. Ανακτήθηκε 8 Ιουνίου, 2015, από <http://www.merriam-webster.com/dictionary/database>

Schwaber, K. & Sutherland, J. (Ιούλιος, 2013). *The Scrum Guide*. Ανακτήθηκε 12 Ιουνίου, 2015, από <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>

Venners, B. (29 Σεπτεμβρίου, 2003). *The Philosophy of Ruby: A conversation with Yukihiro Matsumoto, Part I*. Ανακτήθηκε 2 Ιουνίου, 2015, από <http://www.artima.com/intv/ruby4.html>

Wake, B. (17 Αυγούστου, 2003). *Invest in Good Stories, and Smart Tasks*. Ανακτήθηκε 13 Ιουνίου, 2015, από <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>