

# Πληροφοριακό σύστημα για online βιβλιοθήκη

Αναγνωστόπουλος Βασίλης - Θάνος (ΜΠΠΛ 13002)

Βιδάλης Γιάννης (ΜΠΠΛ 13\_\_\_)

Λιόλης Γιώργος (ΜΠΠΛ 13049)

Χρόνη Ειρήνη (ΜΠΠΛ 13\_\_\_)

Αθήνα, 2015

# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>1</b>
1.1	Περιβάλλον του έργου . . . . .	1
1.2	Περιγραφή του προβλήματος και των εναλλακτικών λύσεων . . .	1
1.3	Σκοπός και στόχος του Π.Σ. . . . .	2
1.4	Υφιστάμενη κατάσταση . . . . .	3
1.5	Βασικές οντότητες και εμπλεκόμενοι στην υλοποίηση του έργου	3
1.6	Μοντέλο διαδικασία υλοποίησης και ανάπτυξης του λογισμικού .	3
<b>2</b>	<b>Φάση: Έναρξη</b>	<b>4</b>
2.1	Ανάλυση Απαιτήσεων Πληροφοριακού Συστήματος . . . . .	5
2.2	Αρχιτεκτονική . . . . .	5
2.3	Χρήστες . . . . .	6
2.4	Λειτουργικές απαιτήσεις . . . . .	7
2.4.1	Λειτουργικότητα και διαγράμματα περίπτωσης χρήσης . .	7
2.4.2	Διάγραμμα βασικών κλάσεων . . . . .	10
2.4.3	Περιορισμοί σχεδιασμού . . . . .	10
2.5	Μη λειτουργικές απαιτήσεις . . . . .	11
<b>3</b>	<b>Φάση: Εκπόνηση Μελέτης</b>	<b>12</b>
3.1	Διαγράμματα περιπτώσεων χρήσης . . . . .	12
3.2	Διάγραμμα δραστηριότητας . . . . .	14
3.3	Διαγράμματα καταστάσεων . . . . .	14
<b>4</b>	<b>Φάση: Κατασκευή</b>	<b>14</b>
4.1	Διάγραμμα δραστηριότητας . . . . .	14
<b>5</b>	<b>Υλοποίηση της Βάσης Δεδομένων</b>	<b>14</b>
5.1	Εννοιολογικό μοντέλο . . . . .	14
5.2	Λογικός σχεδιασμός . . . . .	20
5.3	Μετατροπή μοντέλου οντοτήτων - συσχετίσεων σε σχεσιακό σχήμα	20
5.4	Δημιουργία ειδικού χρήστη . . . . .	22
5.5	Δημιουργία πινάκων . . . . .	22
<b>6</b>	<b>Ανάλυση και Σχεδίαση</b>	<b>26</b>
6.1	Περιπτώσεις χρήσης και διαγράμματα περιπτώσεων χρήσης . . .	26
6.2	Διαγράμματα σειράς . . . . .	26
6.3	Διάγραμμα Βασικών κλάσεων πεδίου Εφαρμογής . . . . .	28
6.4	Αναλυτικά διαγράμματα κλάσεων συνοδευμένα από OCL περιορισμούς . . . . .	28
6.5	Διαγράμματα Επικοινωνίας και Αλληλεπίδρασης . . . . .	28
	<b>Βιβλιογραφία</b>	<b>28</b>

## Κατάλογος πινάκων

2.1	Τα εικονίδια του διαγράμματος περιπτώσεων - χρήσης. . . . .	8
5.1	Τα εικονίδια του διαγράμματος οντοτήτων συσχετίσεων. . . . .	18
6.1	Τα εικονίδια του διαγράμματος σειράς. . . . .	27

## Κατάλογος σχημάτων

1.1	Οι βασικές οντότητες του Πληροφορικού Συστήματος . . . . .	3
1.2	Κύκλος ζωής ανάπτυξης λογισμικού με την RUP [2] . . . . .	4
2.1	Γενικό διάγραμμα περιπτώσεων - χρήσης . . . . .	9
2.2	Το βασικό διάγραμμα κλάσεων . . . . .	10
3.1	Το διάγραμμα περιπτώσεων χρήσης των χρηστών . . . . .	13
3.2	Το βασικό διάγραμμα δραστηριότητας για την ενοικίαση ενός βιβλίου . . . . .	15
3.3	Το βασικό διάγραμμα δραστηριότητας για την επιστροφή ενός βιβλίου . . . . .	16
3.4	Διάγραμμα καταστάσεων για ένα βιβλίο . . . . .	16
4.1	Το διάγραμμα δραστηριότητας για την ενοικίαση ενός βιβλίου . . . . .	17
5.1	Διάγραμμα οντοτήτων - συσχετίσεων . . . . .	19
5.2	Σχεσιακό σχήμα . . . . .	21

## Κατάλογος πηγαίου κώδικα

1	Δημιουργία της βάσης δεδομένων . . . . .	22
2	Δημιουργία του πίνακα users . . . . .	22
3	Δημιουργία του πίνακα books . . . . .	23
4	Δημιουργία του πίνακα categories . . . . .	23
5	Δημιουργία του πίνακα authors . . . . .	24
6	Δημιουργία του πίνακα writes . . . . .	24
7	Δημιουργία του πίνακα publishers . . . . .	24
8	Δημιουργία του πίνακα editions . . . . .	25
9	Δημιουργία του πίνακα paperbooks . . . . .	25
10	Δημιουργία του πίνακα borrows . . . . .	26

# 1 Εισαγωγή

Η εργασία αυτή έχει ως σκοπό την σχεδίαση και ανάλυση ενός πληροφοριακού συστήματος. Συγκεκριμένα η παρούσα αναφορά περιγράφει την βασική λειτουργικότητα και τις σχεδιαστικές αποφάσεις που αφορούν την υλοποίηση του Πληροφοριακού Συστήματος (Π.Σ.) για την δημιουργία μίας ηλεκτρονικής βιβλιοθήκης. Θα παρουσιαστούν οι λειτουργικές και οι μη λειτουργικές απαιτήσεις του συστήματος, το περιβάλλον το οποίο θα χρησιμοποιείται καθώς και οι χρήστες του. Στην συνέχεια θα γίνει μοντελοποίηση του συστήματος με την βοήθεια διαγραμμάτων των διαγραμμάτων της uml (π.χ. διαγραμμάτων κλάσεων, διαγραμμάτων ροής και περιπτώσεων χρήσης, κ.λ.π.) και έπειτα θα ακολουθήσει η υλοποίηση του συστήματος.

## 1.1 Περιβάλλον του έργου

Η παρούσα μελέτη είναι μελέτη σκοπιμότητας για την οργάνωση και τη λειτουργία της δημοτικής βιβλιοθήκης του Καλλικρατικού Δήμου Διονύσου, η οποία από εδώ και πέρα απλώς θα καλείται απλώς ως "βιβλιοθήκη".

Η βιβλιοθήκη θα οργανωθεί σύμφωνα με τα σύγχρονα πρότυπα έτσι ώστε να ικανοποιεί πλήρως τις ανάγκες των χρηστών της και την οργάνωση της συλλογής της βιβλιοθήκης σύμφωνα με τους Διεθνείς κανόνες καταλογογράφησης και ευρετηρίασης, τη δημιουργία ηλεκτρονικού Δημοσίου Καταλόγου Ανοικτής Πρόσβασης και την παροχή σύγχρονων υπηρεσιών πληροφόρησης στους χρήστες της, την ορθολογική οργάνωση του διαθέσιμου χώρου λειτουργίας της και γενικότερα τη διαχείριση των διαθέσιμων πόρων για την βελτιστοποίηση των στόχων της.

Ο σχεδιασμός της βιβλιοθήκης θα πρέπει να είναι ευέλικτος και θα πρέπει

- να επιτρέπει την πλήρη αξιοποίηση των προτύπων οργάνωσης και λειτουργίας, καθώς και των προϊόντων τεχνολογίας και πληροφόρησης,
- υιοθέτηση νέων προτύπων στις μεθόδους επεξεργασίας, οργάνωσης, αποθήκευσης και διάδοσης των πληροφοριών,
- χρησιμοποίηση νέων προϊόντων τεχνολογίας και
- παροχή νέων υπηρεσιών πληροφόρησης

## 1.2 Περιγραφή του προβλήματος και των εναλλακτικών λύσεων

Ο δήμος Διονύσου για την επιμόρφωση των κατοίκων του Δήμου προωθεί την δημιουργία μίας νέας δημοτικής βιβλιοθήκης. Μία βιβλιοθήκη είναι ένας οργανισμός, ο οποίος λειτουργεί ως κρίκος σύνδεσης των μελών μίας κοινότητας με τις γνώσεις και τις πληροφορίες που χρειάζονται. Τα μέλη της συγκεκριμένης βιβλιοθήκης είναι όλοι οι κάτοικοι του Δήμου και σκοπός της είναι:

1. να βοηθήσει στην επιμόρφωση των κατοίκων της περιοχής

2. να συμβάλει στην παροχή πληροφοριών που διαφορετικά θα ήταν δυσπρόσιτες στους κατοίκους

Οι εναλλακτικές λύσεις είναι:

1. να μην γίνει καμία ενέργεια, πράγμα που θα οδηγούσε στην στασιμότητα της επιμόρφωσης των κατοίκων.
2. να συνεχιστεί η κατασκευή της βιβλιοθήκης, πράγμα που θα οδηγήσει στην περαιτέρω επιμόρφωση των κατοίκων

Ακόμα, με την διείσδυση των νέων τεχνολογιών στην καθημερινότητα, οι άνθρωποι χρησιμοποιούν όλο και περισσότερο το διαδίκτυο για την πραγματοποίηση απλών καθημερινών διαδικασιών. Παρά το γεγονός ότι η χρήση του internet παραμένει χαμηλή στη Ελλάδα συγκριτικά με την Ευρώπη, σχεδόν ένας στους πέντε Έλληνες (ποσοστό 20,08%) χρησιμοποιεί πια το διαδίκτυο, ενώ το 17,9% του πληθυσμού το χρησιμοποιεί τακτικά τουλάχιστον μια φορά την εβδομάδα. Οι νεαρότερες ηλικιακές ομάδες (16-24 ετών: 42%, 25-34 ετών: 30%) και οι κάτοικοι των αστικών πόλεων με ανώτερη μόρφωση, αποτελούν με σημαντική διαφορά τις ομάδες πληθυσμού με την υψηλότερη πρόσβαση [8].

Βασισμένοι στα παραπάνω, θεωρείται σημαντικό για την προώθηση της νέας δημοτικής βιβλιοθήκης της να αναπτυχθεί ένα σύστημα λογισμικού για την ενοικίαση βιβλίων μέσω διαδικτύου.

### **1.3 Σκοπός και στόχος του Π.Σ.**

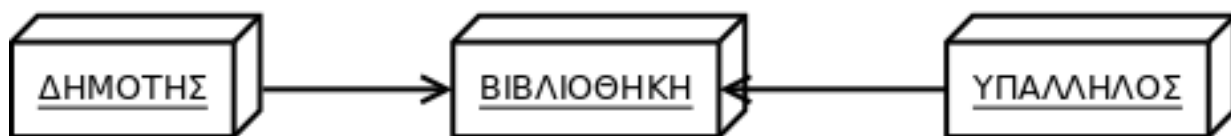
Σκοπός της συγκεκριμένης μελέτης είναι η υλοποίηση ενός Π.Σ. που αποσκοπεί στην κατασκευή μίας βιβλιοθήκης που η διαχείριση της θα γίνεται με ηλεκτρονικούς τρόπους. Ταυτόχρονα το νέο Π.Σ. θα βοηθήσει στην οργάνωση της βιβλιοθήκης. Το Π.Σ. αναμένεται να αποφέρει οφέλη στους τομείς:

- της διαφήμισης, μίας και η ιστοσελίδα θα βοηθήσει στην προώθηση της δημοτικής βιβλιοθήκης
- της εξυπηρέτησης των χρηστών, μίας και οι δημότες δεν θα πρέπει να περιμένουν στην σειρά για την απόκτηση θέσης για την κράτηση ενός βιβλίου.
- της οργάνωσης της βιβλιοθήκης, μίας και θα δημιουργηθεί ένα αυτόματο σύστημα επεξεργασίας της διαθεσιμότητας των βιβλίων
- της μείωση του κόστους, μίας και θα μειωθούν οι εργαζόμενοι οι οποίοι θα πρέπει να απασχολούνται στην δημοτική βιβλιοθήκη.

Η επιτυχία του έργου θα κριθεί κυρίως από το εύρος χρήσης του και από την αξιοποίηση των εξειδικευμένων δυνατοτήτων του, που αποσκοπούν κύρια στην αυτοματοποίηση του συστήματος.

Μετά την ολοκλήρωση του έργου του Π.Σ. θα ωφεληθούν άμεσα:

- οι εργαζόμενοι της εταιρείας μίας και θα απλοποιηθεί η διαδικασία για την συντήρηση της βιβλιοθήκης
- οι κάτοικοι που χρησιμοποιούν την βιβλιοθήκη, μίας και η βιβλιοθήκη θα είναι πιο εύκολα προσβάσιμη.



Σχήμα 1.1: Οι βασικές οντότητες του Πληροφορικού Συστήματος

## 1.4 Υφιστάμενη κατάσταση

????

## 1.5 Βασικές οντότητες και εμπλεκόμενοι στην υλοποίηση του έργου

Οι βασικοί εμπλεκόμενοι στην υλοποίηση του έργου είναι οι χρήστες και οι υπάλληλοι της βιβλιοθήκης. Το άμεσο περιβάλλον του έργου και το επιχειρηματικό μοντέλο παριστάνεται στο σχήμα 1.1. Οι οντότητες αυτές αναλύονται παρακάτω.

Ως χρήστες ορίζονται όλοι όσοι επιθυμούν να δανειστούν κάποιο βιβλίο. Ως υπάλληλοι ορίζονται όλοι οι υπάλληλοι της βιβλιοθήκης, είτε μόνιμοι είτε συμβασιούχοι. Οι υπάλληλοι αυτοί έχουν αναλάβει την συντήρηση της βιβλιοθήκης και θα πρέπει να εξυπηρετούν του χρήστες της, να εξοπλίζουν την βιβλιοθήκη με νέα βιβλία καθώς και να προωθούν την βιβλιοθήκη.

Η βασική απαίτηση του συστήματος (δηλαδή η επιχειρηματική απαίτηση του Π.Σ.) είναι η γεφύρωση του χάσματος μεταξύ των χρηστών και των υπαλλήλων με σκοπό την αύξηση της χρηστικότητας της βιβλιοθήκης.

Στις παρακάτω ενότητες θα περιγραφούν οι προδιαγραφές και οι περιορισμοί στους οποίους θα πρέπει να συμμορφώνεται το υπό μελέτη πληροφοριακό σύστημα. Θα πρέπει να διασφαλίζει ότι θα ικανοποιούνται οι ανάγκες των ενδιαφερόμενων και για να γίνει αυτό θα πρέπει να οριστούν με ακρίβεια οι λειτουργικές και οι μη λειτουργικές απαιτήσεις.

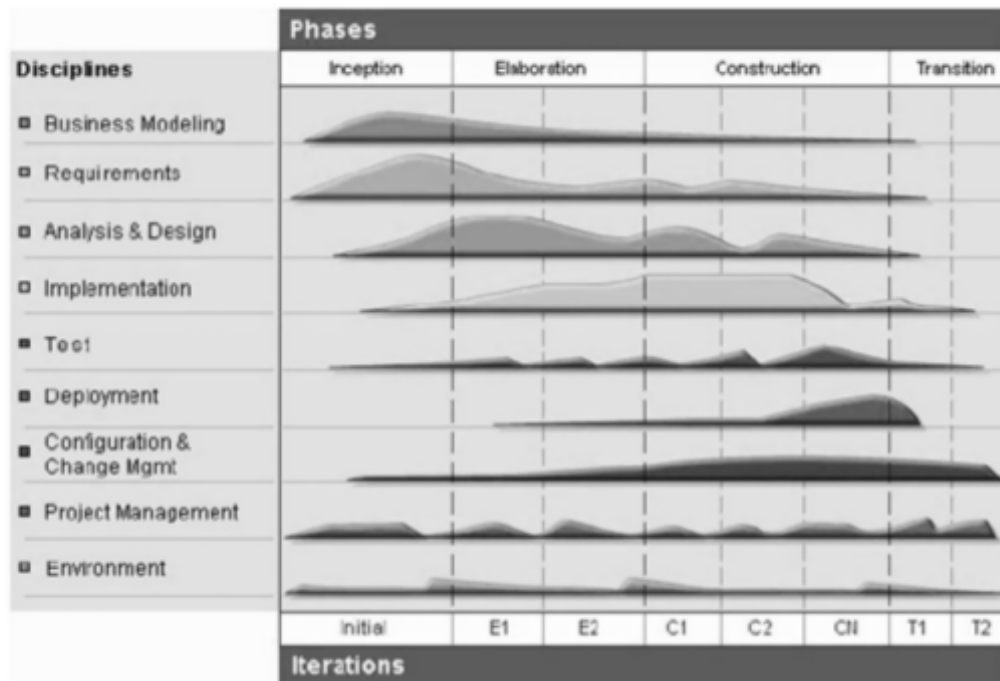
Θα παρουσιαστούν οι απαιτήσεις που πρέπει να ικανοποιεί το Πληροφοριακό Σύστημα, οι βασικές λειτουργίες που πρέπει να επιτελεί, οι πληροφορίες που πρέπει να αποθηκεύει και οι κανόνες που επιβάλλονται από την λειτουργία του συστήματος.

## 1.6 Μοντέλο διαδικασία υλοποίησης και ανάπτυξης του λογισμικού

Ο κύκλος ζωής του λογισμικού προτείνεται να είναι επαναληπτικός [4] και γι' αυτό το λόγο προτείνεται να χρησιμοποιηθεί η διαδικασία Unified της Rational (αγγλ. Rational Unified Process - RUP).

Η διαδικασία Rational Unified Process (RUP) αποτελείται από ένα σύνολο οδηγιών σχετικά με τις τεχνικές και οργανωτικές απόψεις της ανάπτυξης λογισμικού, οι οποίες συνοψίζονται παρακάτω [2]:

**Καθοδήγηση από τις περιπτώσεις χρήσης:** η ανάπτυξη σχεδιάζεται και οργανώνεται χρησιμοποιώντας έναν κατάλογο από περιπτώσεις χρήσεις.



Σχήμα 1.2: Κύκλος ζωής ανάπτυξης λογισμικού με την RUP [2]

**Καθοδήγηση με βάση την αρχιτεκτονική:** Η διαδικασία ανάπτυξης οδηγεί στην κατασκευή μία αρχιτεκτονικής συστήματος που επιτρέπει την εφαρμογή των απαιτήσεων. Αυτή η αρχιτεκτονική βασίζεται στον προσδιορισμό μίας επαναληπτικής δομής ή οποία βασίζεται στο εννοιολογικό μοντέλο του συστήματος.

**Επαναληπτική:** Η ανάπτυξη χωρίζεται σε επαναλήψεις ή κύκλους ανάπτυξης. Σε κάθε επανάληψη, νέα χαρακτηριστικά προστίθενται στο σύστημα ή διορθώνονται ήδη υλοποιημένα, με αποτέλεσμα το σύστημα να γίνεται πιο πλήρες και πιο κοντά στο τελικό επιθυμητό σύστημα.

**Αποφυγή του κινδύνου:** Τα στοιχεία που ενέχουν τον μεγαλύτερο κίνδυνο για το έργο απευθύνονται πιο νωρίς.

Ο κύκλος ζωής λογισμικού όπως προτείνεται από την RUP φαίνεται στο σχήμα 1.2.

## 2 Φάση: Έναρξη

Η φάση της έναρξης (αγγλ. inception) είναι η πρώτη φάση της Unified Process (UP), στην οποία οι κύριες απαιτήσεις του έργου ανακαλύπτονται και έκταση του συστήματος γίνεται κατανοητή. Το προϊόν της φάσης αυτής είναι ένα προκαταρκτικό εννοιολογικό μοντέλο, ανάλυση απαιτήσεων του προκαταρκτικού εννοιολογικού μοντέλου, συνήθως με την μορφή ενός καταλόγου των περιπτώσεων υψηλού επιπέδου χρήσης και ένα χρονοδιάγραμμα (που σε αυτή την εργασία κρίνεται σκόπιμο να μην υλοποιηθεί)[2]. Ακόμα θα δοθεί ένα πρώτο

διάγραμμα τάξεων το οποίο θα δείχνει την υλοποίηση των περιπτώσεων χρήσης.

## **2.1 Ανάλυση Απαιτήσεων Πληροφοριακού Συστήματος**

Η ανάλυση απαιτήσεων περιλαμβάνει τις εργασίες για τον καθορισμό των αναγκών ή των προϋποθέσεων που χρειάζονται για την ολοκλήρωση ενός προϊόντος (στην συγκεκριμένη περίπτωση του πληροφοριακού συστήματος). Στην ανάλυση απαιτήσεων λαμβάνονται υπόψη οι ενδεχόμενες αντικρουόμενες απαιτήσεις των διαφόρων μερών ενώ ταυτόχρονα αναλύονται και τεκμηριώνονται οι τυχόν απαιτήσεις του προϊόντος [3]. Για να είναι επιτυχές ένα πληροφοριακό σύστημα θα πρέπει να είναι προσαρμοσμένο στις ανάγκες, απαιτήσεις, αλλά και προσδοκίες του τελικού χρήστη. Αυτό σημαίνει ότι το ζητούμενο είναι, τί πραγματικά επιθυμεί ο χρήστης, τί ακριβώς περιμένει από το σύστημα και πόσο φιλικό είναι αυτό σε αυτόν και κατά πόσο ικανοποιεί τους σκοπούς για τους οποίους υλοποιήθηκε.

Οι απαιτήσεις λογισμικού περιλαμβάνουν 3 διαφορετικά επίπεδα [6]:

- Επιχειρηματικές απαιτήσεις
- Απαιτήσεις χρηστών
- Λειτουργικές απαιτήσεις

Οι επιχειρηματικές απαιτήσεις αντιπροσωπεύουν τους υψηλού επιπέδου στόχους του οργανισμού ή των πελατών που ζητούν το σύστημα. Ορίζουν τον σκοπό και το πεδίο εφαρμογής του νέου συστήματος λογισμικού και περιγράφουν γιατί ο οργανισμός θέλει να εφαρμόσει το σύστημα. [6].

Οι απαιτήσεις των χρηστών περιγράφουν τους στόχους των χρηστών ή τα καθήκοντα που θα έχουν οι χρήστες στο προϊόν. Οι ανάγκες των χρηστών περιγράφουν τί θα κάνουν οι χρήστες μέσα στο σύστημα. Θα πρέπει να ευθυγραμμίζονται με τις επιχειρηματικές απαιτήσεις. [6]

Τέλος οι λειτουργικές απαιτήσεις καθορίζουν την λειτουργικότητα του λογισμικού που πρέπει να φτιάξουν τα μέλη της ομάδας ανάπτυξης έτσι ώστε το προϊόν να επιτρέπει στους χρήστες να εκπληρώνουν τα καθήκοντα τους καλύπτοντας έτσι τις επιχειρησιακές απαιτήσεις [6].

Η ανάλυση απαιτήσεων συντελεί στην καλή οργάνωση και εκτέλεση του έργου, που με τη σειρά τους εξασφαλίζουν τη λειτουργικότητά του για όλες τις εμπλεκόμενες πλευρές. Στο τέλος, τα οφέλη αυτά έχουν άμεσο αντίκρισμα στη μείωση του κόστους, τόσο για την επιχείρηση που υλοποιεί το έργο όσο και για τον πελάτη που θα το χρησιμοποιήσει [7].

Οι λειτουργικές απαιτήσεις μαζί με τα χαρακτηριστικά ποιότητας και άλλες μη λειτουργικές απαιτήσεις δημιουργούν την προδιαγραφή των απαιτήσεων λογισμικού [6].

Η παρούσα αναφορά περιγράφει την βασική λειτουργικότητα και τις σχεδιαστικές αποφάσεις που αφορούν την υλοποίηση του Π.Σ. της δημοτικής βιβλιοθήκης.

## **2.2 Αρχιτεκτονική**

Οι γενικές αρχές, σε λειτουργικό και τεχνολογικό επίπεδο, που θα διέπουν το Π.Σ. που θα αναπτυχθεί είναι:



1. Συστήματα "ανοικτής" αρχιτεκτονικής (αγγλ. open architecture). Είναι δηλαδή υποχρεωτική η χρήση ανοικτών προτύπων που θα διασφαλίζουν ανεξαρτησία από συγκεκριμένο προμηθευτή και:
  - ομαλή συνεργία και λειτουργία μεταξύ των επιμέρους Υποσυστημάτων του πληροφοριακού συστήματος,
  - δικτυακή συνεργασία μεταξύ εφαρμογών ή/και συστημάτων τα οποία βρίσκονται σε διαφορετικά υπολογιστικά συστήματα,
  - επεκτασιμότητα των υποσυστημάτων, χωρίς αλλαγές στη δομή και αρχιτεκτονική τους, για την αντιμετώπιση των μεταβαλλόμενων / αυξανόμενων αναγκών
  - εύκολη επέμβαση στη λειτουργικότητα των υποσυστημάτων (συντηρισιμότητα - maintainability)
  - ύψιστη διασφάλιση των δεδομένων.
2. Αρθρωτή αρχιτεκτονική του συστήματος, ώστε να επιτρέπονται μελλοντικές επεκτάσεις και αντικαταστάσεις, ενσωματώσεις, αναβαθμίσεις ή αλλαγές διακριτών τμημάτων λογισμικού ή εξοπλισμού.
3. Εξασφάλιση πλήρους λειτουργικότητας μέσω του εσωτερικού δικτύου (αγγλ. intranet) και του διαδικτύου (αγγλ. internet) όπου αυτό απαιτείται.
4. Χρήση γραφικού περιβάλλοντος λειτουργίας (αγγλ. GUI) του χρήστη για την αποδοτική χρήση του Π.Σ. και την ευκολία εκμάθησής τους.
5. Ενσωμάτωση στο Π.Σ. άμεσης υποστήριξης βοήθειας (αγγλ. online help) και οδηγιών στην ελληνική γλώσσα, προς τους χρήστες ανά διαδικασία ή/και οθόνη.
6. Μηνύματα λαθών (αγγλ. error messages) στην ελληνική γλώσσα και ειδοποίηση των χρηστών με όρους οικείους προς αυτούς.
7. Τήρηση από το Π.Σ. στοιχείων auditing για ιχνηλάτηση ενεργειών χρηστών.
8. Διασφάλιση της πληρότητας, ακεραιότητας, εμπιστευτικότητας και ασφάλειας των δεδομένων των Υποσυστημάτων κατά τη χρήση και τη δικτυακή διακίνησή τους.
9. Τεκμηρίωση του Π.Σ. μέσω της αναλυτικής περιγραφής της βάσης δεδομένων. Σύνταξη τεχνικών εγχειριδίων του συστήματος και των εργαλείων διαχείρισης (αγγλ. system manuals), καθώς και λεπτομερή εγχειρίδια λειτουργίας του συστήματος (αγγλ. operation manuals ) και υποστήριξης των χρηστών (αγγλ. user manuals).

## 2.3 Χρήστες

Το πληροφορικό σύστημα θα έχει ως χρήστες τους υπαλλήλους της δημοτικής βιβλιοθήκης αλλά και τους κατοίκους του δήμου. Βασική απαίτηση του συστήματος είναι οργάνωση της βιβλιοθήκης έτσι ώστε να μπορεί να διαχειρίζεται τα βιβλία που θα έχει και να τα δανείζει στους κατοίκους του δήμου χωρίς να υπάρχει κίνδυνος απώλειας τους.

## 2.4 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις είναι οι κύριες δυνατότητες του συστήματος. Αναπαριστούν το "τί" θα κάνει το σύστημα που θα αναπτυχθεί, χωρίς να αναφέρονται στον τρόπο με τον οποίο ("πώς") το σύστημα θα το κάνει [6].

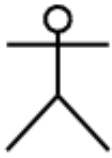


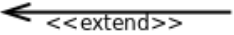
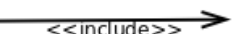


### 2.4.1 Λειτουργικότητα και διαγράμματα περίπτωσης χρήσης

Η λειτουργικότητα ενός συστήματος μετράται από το πόσο καλά ικανοποιεί τις λειτουργικές απαιτήσεις των ενδιαφερόμενων. Το Π.Σ. για την κράτηση των θέσεων υλοποιεί τη απαιτούμενη μηχανογράφηση για την βιβλιοθήκη. Βασική απαίτηση από το Π.Σ. είναι η αποθήκευση των απαιτούμενων πληροφοριών για την ενοικίαση των βιβλίων. Τέλος μέσω του Π.Σ. θα πρέπει να μπορούν να γίνονται τα εξής:

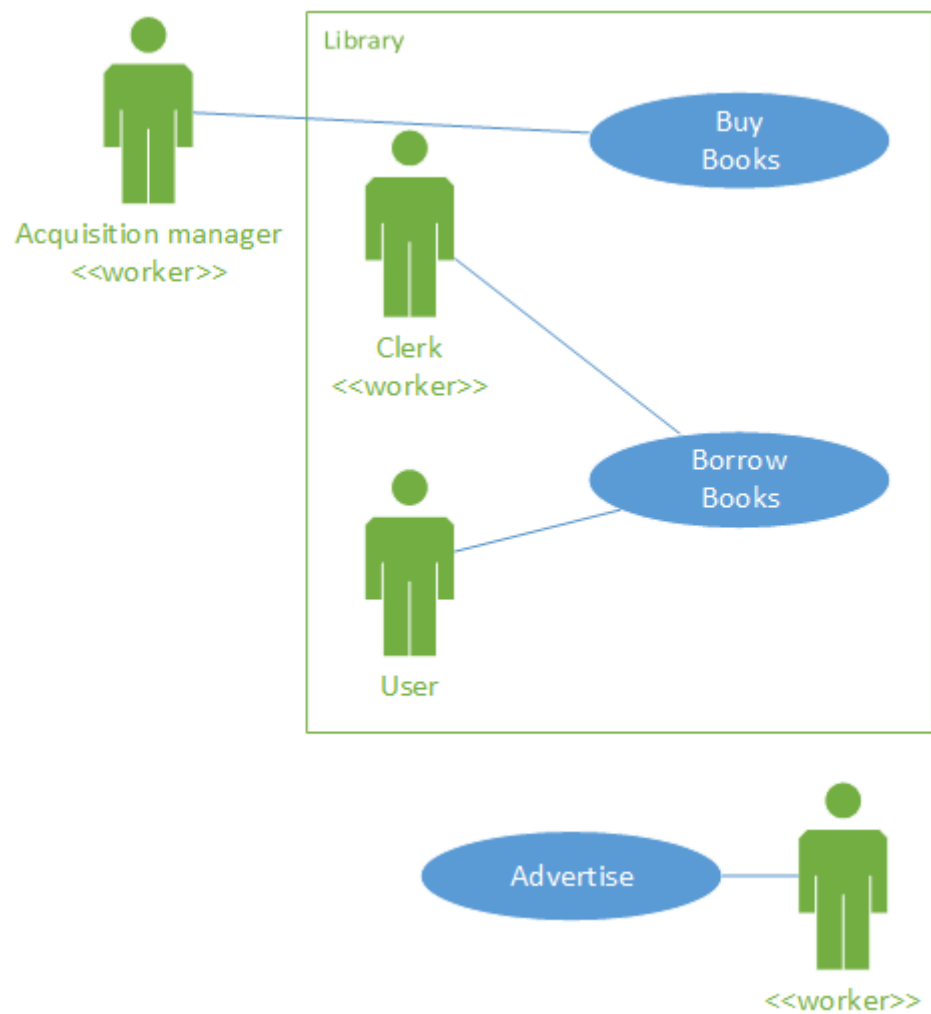
- Αυτόματη ενημέρωση της κεντρικής ιστοσελίδας του Π.Σ. ανάλογα με τις πληροφορίες που εισάγουν οι διαχειριστές του Π.Σ. .
- Αυτόματη δημιουργία λογαριασμών, μέσω των οποίων οι δημότες θα μπορούν να βλέπουν ποια βιβλία είναι διαθέσιμα.
- Το σύστημα θα πρέπει να επιτρέπει στους χρήστες να ενημερώνουν τις προσωπικές τους πληροφορίες.
- Το σύστημα θα πρέπει να μπορεί να παρέχει πληροφορίες για τα βιβλία.
- Θα πρέπει να επιτρέπει την επικοινωνία μεταξύ των δημοτών και των υπαλλήλων της βιβλιοθήκης.
- Το σύστημα θα πρέπει να κρατά το ιστορικό των ενοικιάσεων ενός δημότη.
- Το σύστημα θα πρέπει να επιτρέπει την δημιουργία στατιστικών στοιχείων για τα βιβλία.
- Το σύστημα θα πρέπει να βοηθάει στην προώθηση της δημοτικής βιβλιοθήκης μέσω της κεντρικής ιστοσελίδας.

Όλες οι απαιτούμενες λειτουργικότητες υψηλού επιπέδου από το Π.Σ. φαίνονται στο UML διάγραμμα χρήσης (αγγλ. use case diagram) (βλ. σχήμα 2.1). Στην ενότητα 3 θα αναπτυχθούν σε uml διάγραμμα και οι υπόλοιπες λειτουργικότητες που περιγράφονται σε αυτή την ενότητα

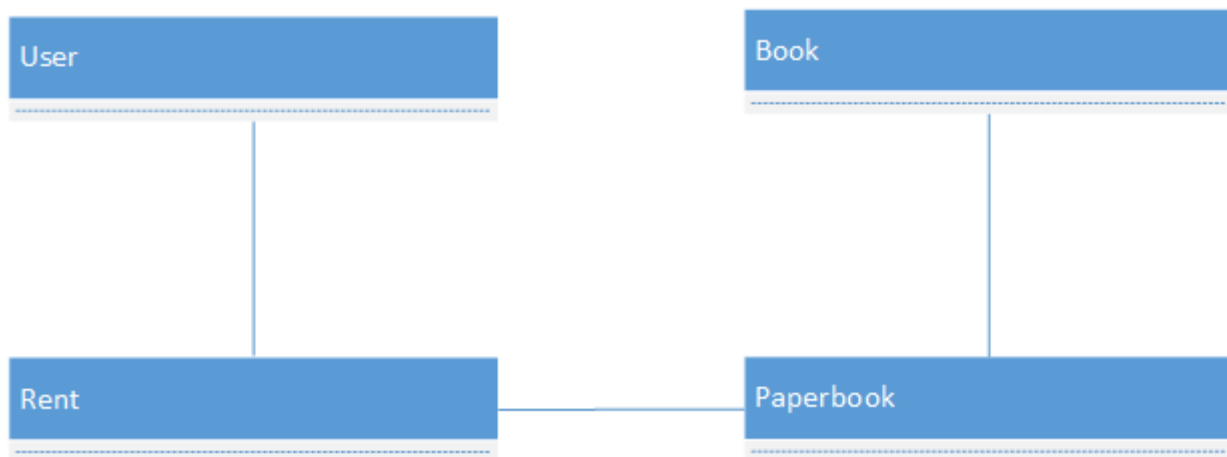
Τα διαγράμματα περιπτώσεων - χρήσης (αγγλ. Use Case Diagrams) περιγράφουν τη συμπεριφορά ενός συστήματος από την οπτική γωνία των χρηστών. Επιτρέπουν τον ορισμό των ορίων του συστήματος και του περιβάλλοντος [4]. Οι συμβολισμοί που χρησιμοποιούνται στα διαγράμματα περιπτώσεων - χρήσης φαίνονται στον πίνακα 2.1. Ακόμα έχει χρησιμοποιηθεί και ένας μη στάνταρ συμβολισμός για να ξεχωρίζει τους υπαλλήλους από τους υπόλοιπους χρήστες. Για την ακρίβεια κάτω από τους ενεργοποιούς που υποδηλώνουν τους υπάλληλους της βιβλιοθήκης έχει προστεθεί ως κείμενο το: <<worker>>.

 Χρήστης	Ο ενεργοποιός του συστήματος. Ο ενεργοποιός αναπαριστά ένα ρόλο που παίζεται από ένα άτομο ή πράγμα που αλληλεπιδρά με το σύστημα [4].
 χρήση	Περίπτωση χρήσης. Περιγράφει τις δυνατές αλληλεπιδράσεις με το σύστημα [4].
	Η σχέση <<communicates>>. Η σχέση αυτή ορίζεται μεταξύ περιπτώσεων χρήσης και σημαίνει ότι ένα στιγμιότυπο της πηγής (περίπτωσης χρήσης) συμπεριλαμβάνει τη συμπεριφορά του στόχου (περίπτωση χρήσης) [4].
	Η σχέση <<extend>>. Δείχνει προαιρετική συμπεριφορά μίας περίπτωση χρήσης [4].
	Η σχέση <<include>>. Χρησιμοποιείται για να δείξει λειτουργικότητα που τη μοιράζονται πολλές περιπτώσεις χρήσης [4].
	Η γενίκευση περιπτώσεων χρήσης. Ο ειδικός ενεργοποιός κληρονομεί τις περιπτώσεις χρήσης του γενικού ενεργοποιού. Το βέλος πρέπει να δείχνει το γενικότερο ενεργοποιό. [4].
	Τα όρια του συστήματος.

**Πίνακας 2.1: Τα εικονίδια του διαγράμματος περιπτώσεων - χρήσης.**



**Σχήμα 2.1: Γενικό διάγραμμα περιπτώσεων - χρήσης**



**Σχήμα 2.2: Το βασικό διάγραμμα κλάσεων**

### 2.4.2 Διάγραμμα βασικών κλάσεων

Ένα διάγραμμα κλάσεων δείχνει την στατική δομή των κλάσεων του συστήματος και των σχέσεων μεταξύ τους. Ένα διάγραμμα κλάσεων συνήθως αποτελείται από:

- Κλάσεις (αγγλ. Classes)
- Διαπροσωπίες (αγγλ. Interfaces)
- Συνεργασίες (αγγλ. Collaborations)
- Συσχετίσεις (αγγλ. Relationships)

Το μοντέλο των κλάσεων χρησιμοποιείται κατά την διάρκεια της ανάλυσης για να περιγράψει τις λειτουργικές απαιτήσεις. Επίσης χρησιμοποιείται κατά την διάρκεια του σχεδιασμού για να περιγράψει το λεξιλόγιο του συστήματος, τις συνεργασίες και το λογικό σχήμα της βάσης δεδομένων [1].

Στο σχήμα 2.2 φαίνονται οι βασικές κλάσεις καθώς και οι συσχετίσεις μεταξύ τους. Οι βασικές κλάσεις προέκυψαν από το σχήμα 2.1 καθώς κάθε μία οντότητα του σχήματος θα έπρεπε να αντιστοιχηθεί σε μία κλάση.

Συγκεκριμένα έχουμε την κλάση User που αναπαριστά τους ενεργοποιούς του συστήματος. Την κλάση Book που αναπαριστά τα βιβλία και τέλος έχουμε την κλάση Paperbook η αναπαριστά τα "φυσικά" έντυπα των βιβλίων τα οποία ενοικιάζονται στους χρήστες και τέλος έχουμε την κλάση "Rent" η οποία υποδηλώνει την ενοικίαση ενός βιβλίου από έναν χρήστη. Από το σχήμα 2.1 φαίνονται και οι μεταξύ συνδέσεις μεταξύ των κλάσεων.

Στην ενότητα 3 το διάγραμμα κλάσεων θα αναπτυχθεί περισσότερο και σε κάθε κλάση θα προστεθούν και οι μεταβλητές και οι συναρτήσεις τους.

### 2.4.3 Περιορισμοί σχεδιασμού

Οι σχεδιαστές πρέπει να δημιουργήσουν ένα σύστημα το οποίο θα προσαρμόζεται σε κάποιους περιορισμούς. Αν αυτοί οι περιορισμοί δεν συνυπολογιστούν τότε το σύστημα δεν θα λειτουργεί σωστά και θα καταλήξει σε αποτυχία.

Το Π.Σ. για την βιβλιοθήκη θα είναι υπεύθυνο για την ενοικίαση των βιβλίων. Επομένως πριν την ενοικίαση ενός βιβλίου θα πρέπει να πραγματοποιείται ένας έλεγχος για την αποφυγή διπλοκρατήσεων.

## 2.5 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις είναι οι περιορισμοί που τίθενται στις λειτουργικές απαιτήσεις, ή στις απαιτήσεις ποιότητας. Αυτές περιλαμβάνουν πληθώρα ιδιοτήτων συμπεριλαμβάνοντας την επίδοση, τους περιορισμούς πολιτικής, την ασφάλεια, την προστασία προσωπικών δεδομένων, την αξιοπιστία. Καθορίζονται γενικά ως ένα βαθμό μετά την μοντελοποίηση των επιχειρηματικών διαδικασιών. Η μοντελοποίηση των μη λειτουργικών χαρακτηριστικών της επιχείρησης θεωρείται ως ένα δύσκολο πρόβλημα, καθώς η μοντελοποίηση επικεντρώνεται στην λειτουργική συμπεριφορά [6].

Για το Π.Σ. για την βιβλιοθήκη οι μη λειτουργικές απαιτήσεις οι οποίες πρέπει να τηρούνται είναι οι παρακάτω:

**Επίδοση:** Η επίδοση έχει να κάνει με περιορισμούς της ταχύτητας που θα πρέπει να εκτελούνται οι διεργασίες, την ποσότητα των δεδομένων που θα αποθηκεύονται και τους χρόνους απόκρισης του συστήματος [6]. Παρακάτω υπάρχουν κάποιοι τέτοιοι περιορισμοί:

**Απόκριση:** Οι λειτουργίες του εσωτερικού δικτυακού κόμβου πρέπει να έχουν χρόνο απόκρισης εντός ολίγων δευτερολέπτων

**Εκτέλεση απλών ερωτημάτων:** Το 90% των συναλλαγών θα πρέπει να ολοκληρώνεται σε χρόνο μικρότερο των 2 sec

**Εκτέλεση σύνθετων ερωτημάτων:** Το 90% των συναλλαγών θα πρέπει να ολοκληρώνεται σε χρόνο μικρότερο των 4 sec

**Ασφάλεια και προστασία προσωπικών πληροφοριών:** Η ασφάλεια είναι ένας κρίσιμος παράγοντας για όλες τις εφαρμογές. Κατά το σχεδιασμό του Π.Σ. θα πρέπει να ληφθεί ειδική μέριμνα και να δρομολογηθούν οι κατάλληλες δράσεις για:

- την προστασία της ακεραιότητας και της διαθεσιμότητας των πληροφοριών
- την προστασία των προς επεξεργασία και αποθηκευμένων προσωπικών δεδομένων αναζητώντας και εντοπίζοντας με μεθοδικό τρόπο τα τεχνικά μέτρα και τις οργανωτικο-διοικητικές διαδικασίες

**Χρηστικότητα - Προσβασιμότητα:** Το σχεδιασμένο σύστημα θα πρέπει να διακρίνεται από υψηλό επίπεδο χρηστικότητας στην οργάνωση και παρουσίαση των ψηφιακών υπηρεσιών που θα παρέχει. Θα πρέπει να λαμβάνονται υπόψιν οι διαφορετικές ομάδες χρηστών κι επομένως τους διαφορετικούς τρόπους εκπλήρωσης της παρεχόμενης λειτουργικότητας χωρίς να μειώνεται η χρηστικότητα των εφαρμογών. Ο σχεδιασμός των εφαρμογών με βασική αρχή την επίτευξη υψηλής χρηστικότητας και εργονομίας είναι κρίσιμος παράγοντας επιτυχίας για το παρόν έργο.

Οι κυριότερες αρχές προς την κατεύθυνση της χρηστικότητας περιλαμβάνουν:

**Συμβατότητα:** Οι web-εφαρμογές που θα υλοποιηθούν θα πρέπει να είναι προσβάσιμες με τρεις (3) τουλάχιστον, από τους πιο διαδεδομένους φυλλομετρητές (web browsers).

**Συνέπεια:** Οι εφαρμογές θα πρέπει να έχουν ομοιόμορφη εμφάνιση (κατά το δυνατόν) και να τηρείται συνέπεια στη χρήση των λεκτικών και των συμβόλων. Αντίστοιχη συνέπεια πρέπει να επιδεικνύουν οι οποιεσδήποτε γραφικές απεικονίσεις και οι τοποθετήσεις αντικειμένων. Στο επίπεδο των εφαρμογών και διαδραστικών λειτουργιών, παρόμοιες λεκτικές και λειτουργικές απεικονίσεις πρέπει να αντιστοιχούν σε ανάλογα αποτελέσματα.

**Αξιοπιστία:** Ο χρήστης πρέπει να έχει σαφείς διαβεβαιώσεις δια μέσου της εμφάνισης και συμπεριφοράς του συστήματος ότι οι συναλλαγές του θα διεκπαιρώνονται με ασφάλεια.

**Διαφάνεια:** Ο χρήστης θα πρέπει να "συναλλάσσεται" με το Σύστημα χωρίς να αντιλαμβάνεται τεχνικές λεπτομέρειες ή εσωτερικές διεργασίες διεκπεραίωσης των συναλλαγών.

**Συντηρησιμότητα:** Το Π.Σ. θα πρέπει να συντηρείται εύκολα και να επιτρέπονται μελλοντικές επεκτάσεις και αντικαταστάσεις, αναβαθμίσεις ή αλλαγές του Π.Σ. .

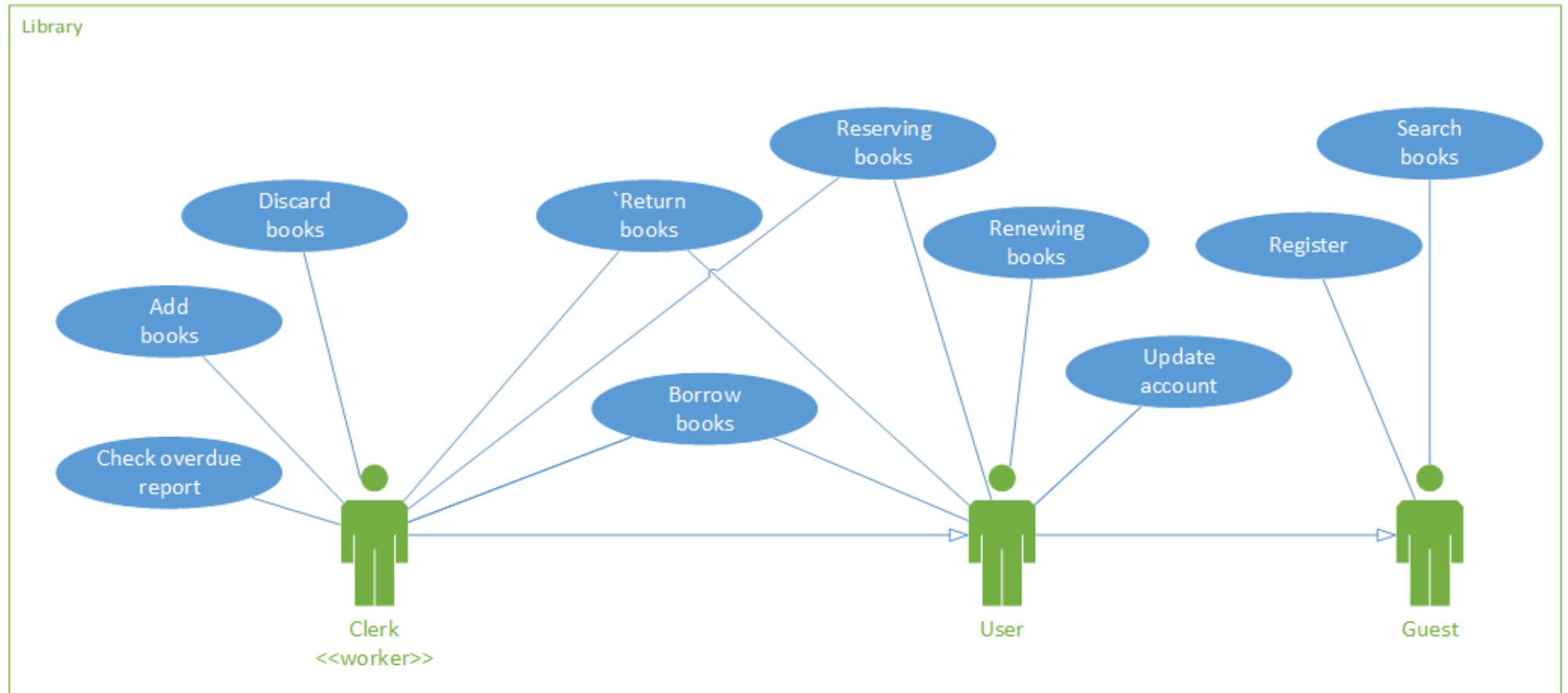
### 3 Φάση: Εκπόνηση Μελέτης

Η φάση της εκπόνησης της μελέτης (αγγλ. elaboration phase) περιλαμβάνει μία πιο λεπτομερή ανάλυση απαιτήσεων, η οποία πραγματοποιείται με την επέκταση των περιπτώσεων χρήσης, δηλαδή, γράφοντας την αλληλουχία των βημάτων που χαρακτηρίζει κάθε μία από τις πιθανές περιπτώσεις τους. Επομένως το εννοιολογικό μοντέλο εξευγενισμέται μετά την ανάλυση των περιπτώσεων χρήσης [2].

Η φάση της εκπόνησης της μελέτης είναι επαναληπτική. Σε κάθε επανάληψη αναπτύσσονται μία ή περισσότερες περιπτώσεις χρήσης και σε κάθε επανάληψη, οι πληροφορίες που εξάγονται από τις περιπτώσεις χρήσης ενσωματώνονται στο τελικό προϊόν. Στην φάση εκπόνησης αντιμετωπίζονται οι πιο πολύπλοκες περιπτώσεις χρήσεις [2].

#### 3.1 Διαγράμματα περιπτώσεων χρήσης

Σε συνέχεια από την ενότητα 2 θα συνεχιστεί η ανάλυση απαιτήσεων με εμβάθυνση στις περιπτώσεις χρήσης. Συγκεκριμένα θα αναλυθούν οι ενέργειες που θα πρέπει να εκτελούνται από τους ενεργοποιούς του πληροφοριακού συστήματος



Σχήμα 3.1: Το διάγραμμα περιπτώσεων χρήσης των χρηστών



### **3.2 Διάγραμμα δραστηριότητας**

Ένα διάγραμμα δραστηριότητας αναπαριστά την κατάσταση εκτέλεσης ενός μηχανισμού σαν μία σειρά βημάτων που ομαδοποιούνται σειριακά σαν παράλληλες διακλαδώσεις ροής ελέγχου [4]. Βοηθούν στην αποτύπωση λεπτομερειών που δεν φαίνονται στα διαγράμματα των περιπτώσεων χρήσης.

Στο σχήμα 3.2

### **3.3 Διαγράμματα καταστάσεων**

Τα διαγράμματα καταστάσεων αναπαριστούν μηχανές καταστάσεων από την άποψη των καταστάσεων και των μεταβάσεων. Όπως και τα διαγράμματα δραστηριότητας, είναι διαγράμματα συμπεριφοράς, μόνο που αντί να μοντελοποιούν δραστηριότητες, μοντελοποιούν τις καταστάσεις του συστήματος, ενός ενεργοποιού ή μίας οντότητας σε μία συγκεκριμένη στιγμή [4, 2].

## **4 Φάση: Κατασκευή**

### **4.1 Διάγραμμα δραστηριότητας**

Σε συνέχεια από την ενότητα 3 έχουμε:

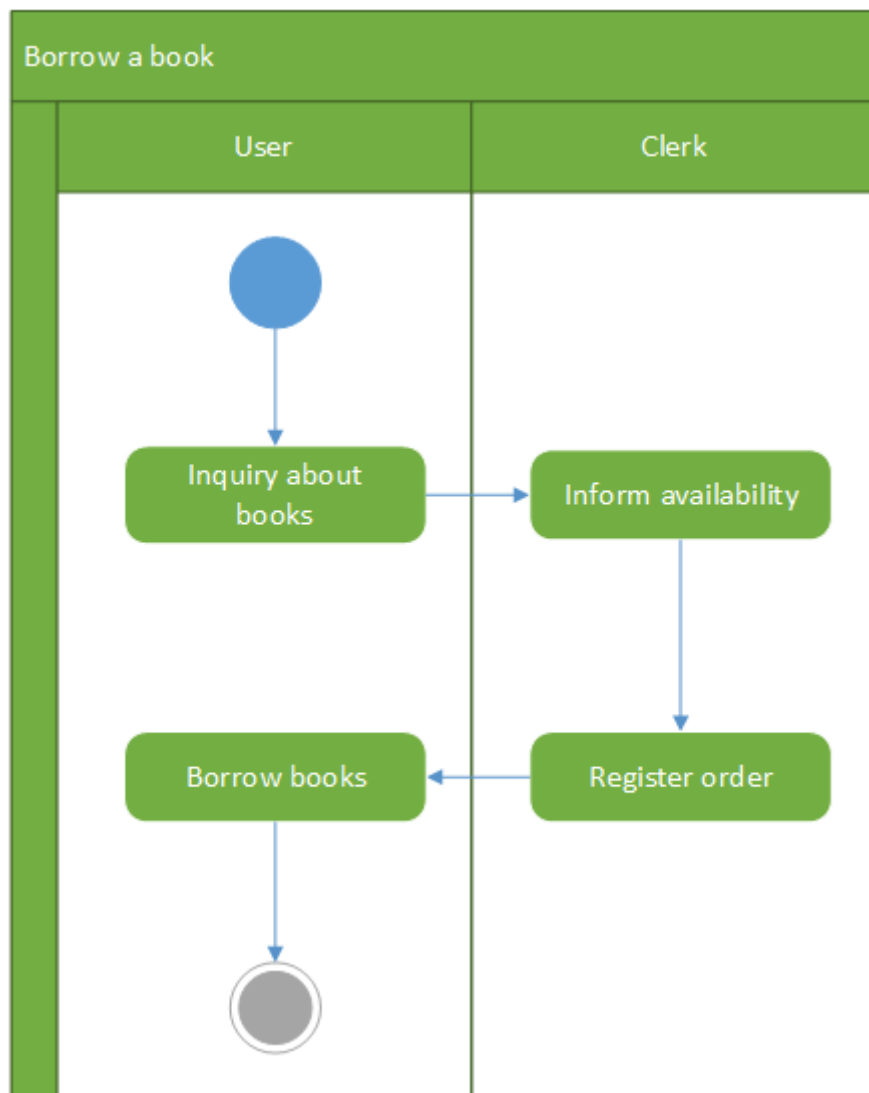
Στο σχήμα 3.2 φαίνεται το εκτενές σχήμα του διαγράμματος δραστηριότητας της ενοικίασης ενός βιβλίου

## **5 Υλοποίηση της Βάσης Δεδομένων**

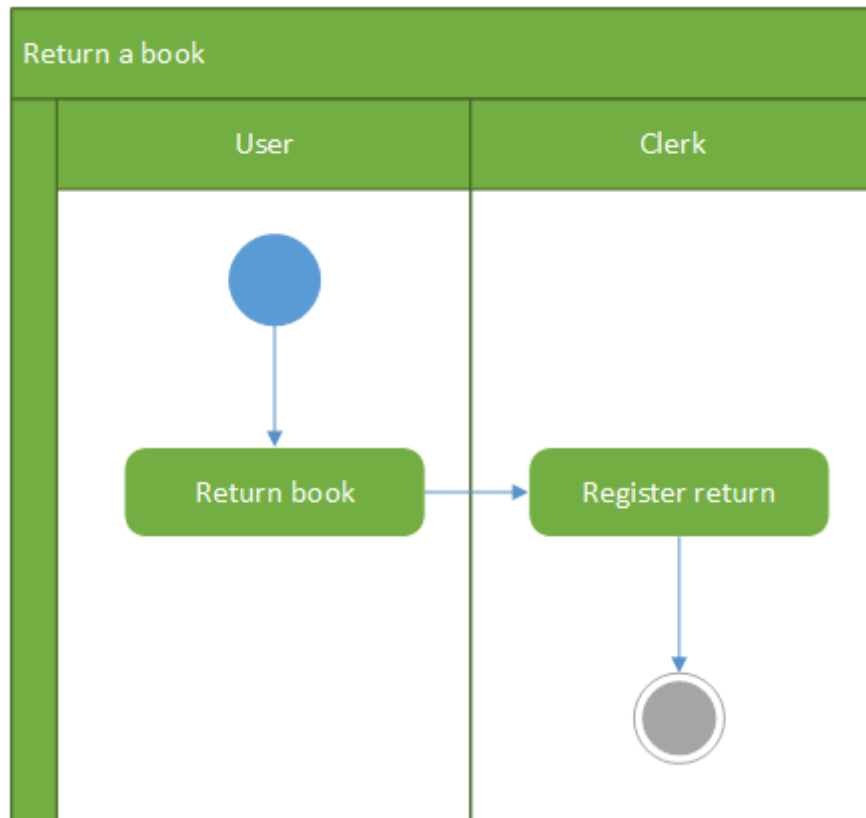
Σε αυτή την ενότητα περιγράφεται η υλοποίηση της βάσης δεδομένων στο ΣΔΒΔ της MariaDB η οποία είναι ένα fork της MySQL.

### **5.1 Εννοιολογικό μοντέλο**

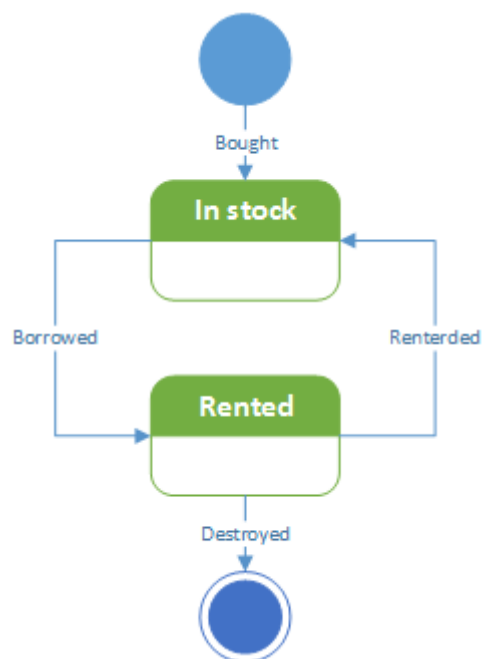
Στο διάγραμμα 5.1 φαίνονται μόνο τα πιο βασικά πεδία. Όλα τα πεδία φαίνονται αναλυτικά στο σχήμα 5.2.



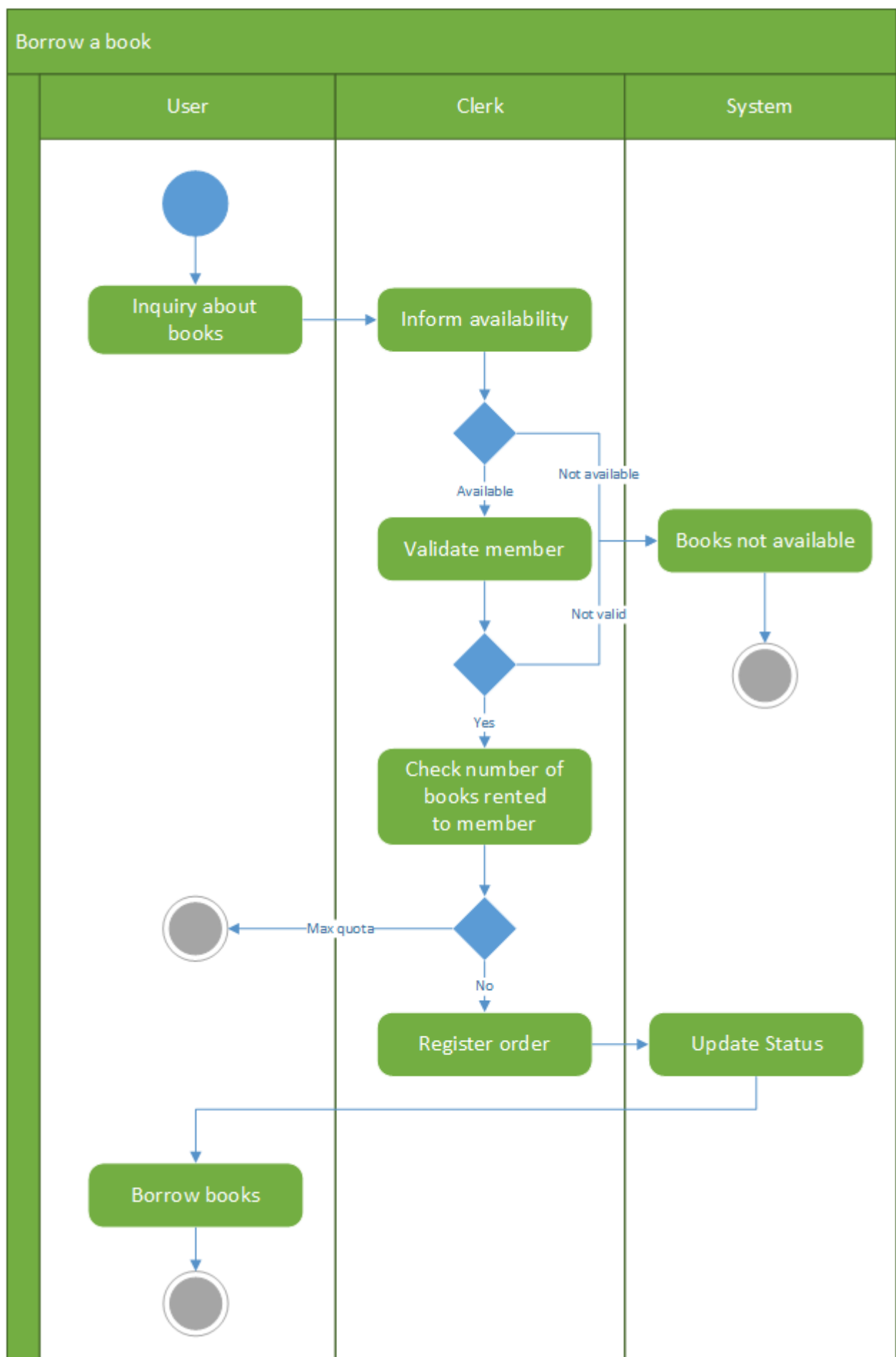
**Σχήμα 3.2:** Το βασικό διάγραμμα δραστηριότητας για την ενοικίαση ενός βιβλίου





**Σχήμα 3.3:** Το βασικό διάγραμμα δραστηριότητας για την επιστροφή ενός βιβλίου



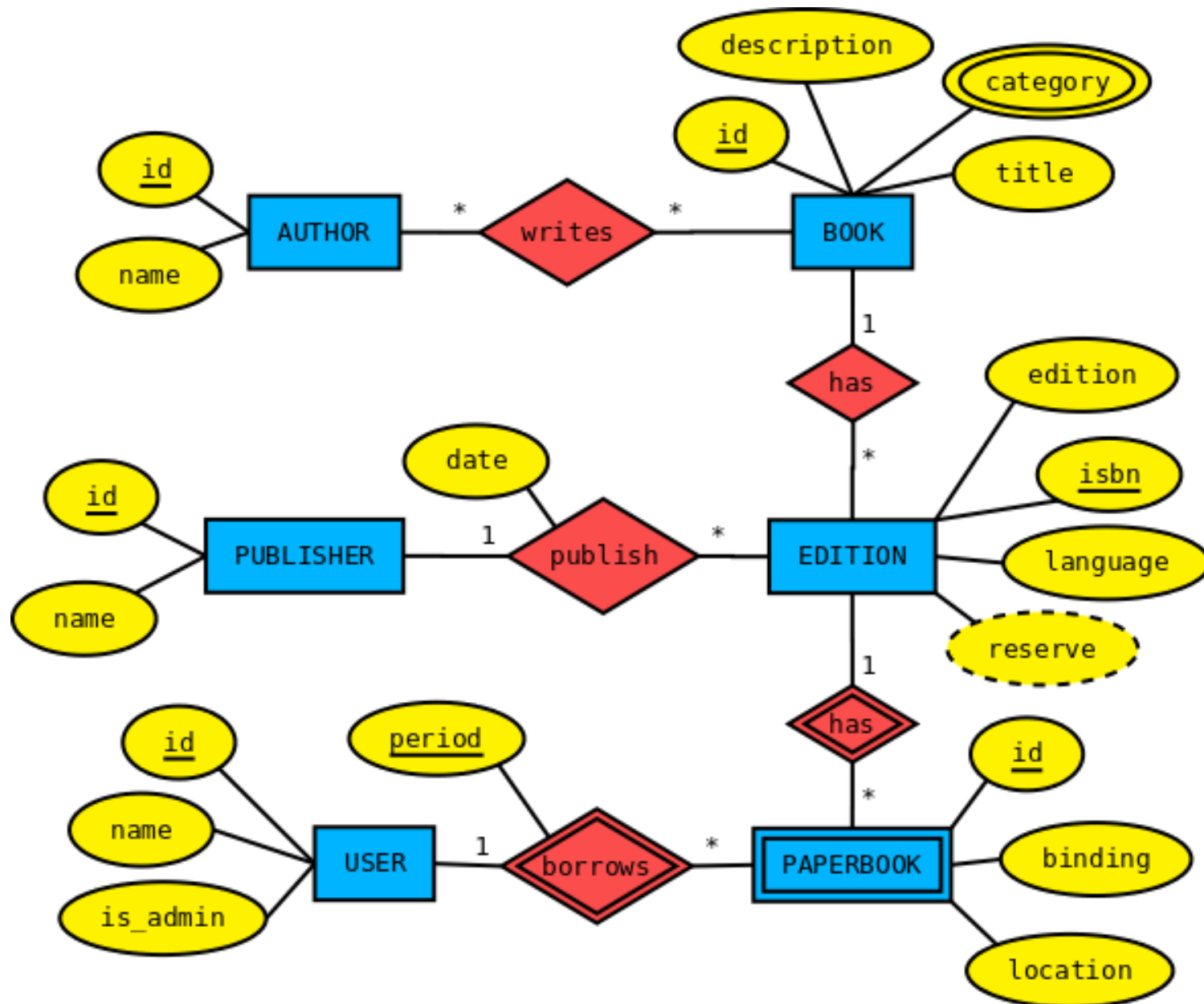
**Σχήμα 3.4:** Διάγραμμα καταστάσεων για ένα βιβλίο



**Σχήμα 4.1: Το διάγραμμα δραστηριότητας για την ενοικίαση ενός βιβλίου**

	Ορθογώνια: οντότητες.
	Ρόμβοι: συσχετίσεις.
	Γραμμές: συνδέουν χαρακτηριστικά με οντότητες, οντότητες με συσχετίσεις.
	Ελλείψεις: χαρακτηριστικά.
	Διπλές ελλείψεις: πλειότιμα χαρακτηριστικά.
	Υπογραμμίσεις: πρωτεύοντα κλειδιά.
	Διπλό ορθογώνιο: Αδύναμο σύνολο οντοτήτων.
	Τρίγωνο: Εξειδίκευση.

**Πίνακας 5.1: Τα εικονίδια του διαγράμματος οντοτήτων συσχετίσεων.**



Σχήμα 5.1: Διάγραμμα οντοτήτων - συσχετίσεων

## 5.2 Λογικός σχεδιασμός

Ο λογικός σχεδιασμός είναι η διαδικασία μετατροπής ενός εννοιολογικού μοντέλου (διαισθητικής περιγραφής) σε τυπικά σχήματα εκφρασμένα στο επιλεγμένο (υποστηριζόμενο από το ΣΔΒΔ) μοντέλο δεδομένων (π.χ. Σχεσιακό μοντέλο) [5].

## 5.3 Μετατροπή μοντέλου οντοτήτων - συσχετίσεων σε σχεσιακό σχήμα

Οι κανόνες που χρησιμοποιούνται για την μετατροπή του μοντέλου οντοτήτων - συσχετίσεων σε σχεσιακό σχήμα φαίνονται παρακάτω [5]:

**Οντότητες:** Ένα ισχυρό σύνολο οντοτήτων μετατρέπεται σε πίνακα (με τα ίδια χαρακτηριστικά). Ένα αδύναμο σύνολο οντοτήτων γίνεται πίνακας που περιλαμβάνει μια στήλη για το πρωτεύον κλειδί του ισχυρότερου συνόλου οντοτήτων που το ταυτοποιεί.

**Συσχετίσεις 1:1 :** Οι συσχετίσεις 1:1 μπορούν να αναπαρασταθούν είτε προσθέτοντας το πρωτεύον κλειδί της μία πλευράς ως επιπλέον χαρακτηριστικό στον πίνακα της άλλη πλευράς είτε δημιουργώντας ένα νέο πίνακα που έχει τα κλειδιά των δύο πινάκων.

**Συσχετίσεις 1:N :** Οι συσχετίσεις 1:N μπορούν να αναπαρασταθούν απλά με την προσθήκη ενός επιπλέον χαρακτηριστικού στην πλευρά N (το πρωτεύον κλειδί τις πλευράς 1). Εάν η συμμετοχή στην πλευρά N είναι μερική, μπορεί να προκύψει μία στήλη που να έχει πολλές κενές τιμές. Οπότε η δημιουργία ενός νέου πίνακα μπορεί να συμφέρει.

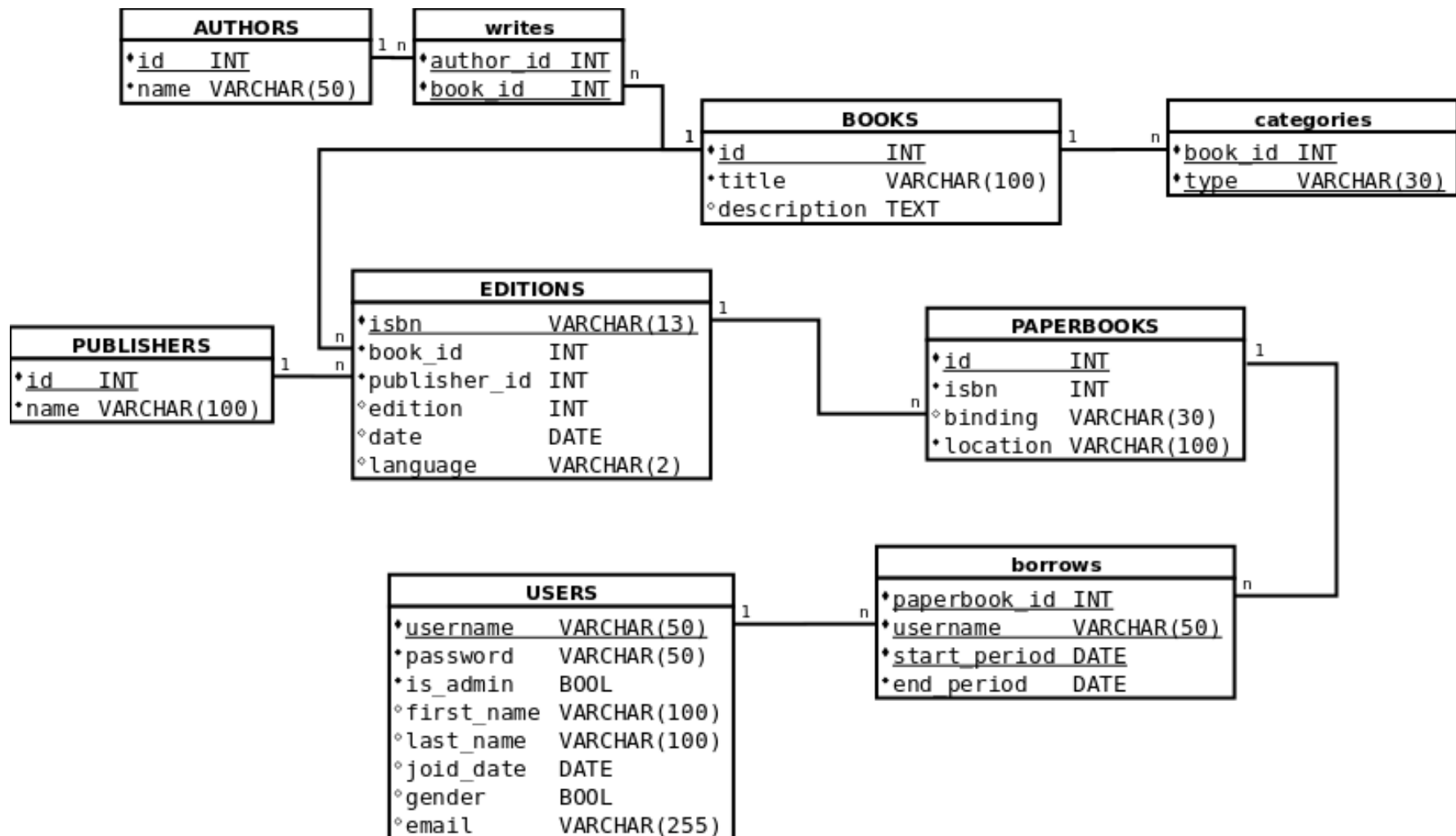
**Συσχετίσεις M:N :** Ένα σύνολο συσχετίσεων M:N αναπαριστάται ως πίνακας με στήλες για τα πρωτεύοντα κλειδιά των οντοτήτων που συμμετέχουν, και επιπλέον όλα τα χαρακτηριστικά του συνόλου συσχετίσεων.

**Σύνθετα χαρακτηριστικά :** Τα σύνθετα χαρακτηριστικά μετατρέπονται σε ένα σύνολο απλών.

**Πλειότιμα χαρακτηριστικά :** Από τα πλειότιμα χαρακτηριστικά ενός συνόλου οντοτήτων προκύπτει νέος πίνακας. Ο πίνακας αυτός έχει ως στήλες το πρωτεύον κλειδί του συνόλου οντοτήτων και μία ακόμα που αντιστοιχεί στο πλειότιμο χαρακτηριστικό.

**Εξειδίκευση :** Όταν έχουμε εξειδίκευση, τότε προκύπτει ένας πίνακας για κάθε εμπλεκόμενο σύνολο οντοτήτων, όπου καθένας από τους πίνακες εξειδίκευσης συμπεριλαμβάνει ως στήλη το πρωτεύον κλειδί του πίνακα γενίκευσης.

Ακολουθώντας τους παραπάνω κανόνες προκύπτει το σχήμα 5.2.



Σχήμα 5.2: Σχεσιακό σχήμα



## 5.4 Δημιουργία ειδικού χρήστη

Σαν πρώτο βήμα είναι η δημιουργία ειδικού χρήστη για την βάση έτσι ώστε να μην χρησιμοποιούμε συνέχεια τον υπερχρήστη. Για την επίτευξη αυτού του σκοπού πληκτρολογούμε στο terminal τα παρακάτω:

```
$ mysql --user=root --password
Enter password:
```

Και αφού έχουμε μπει στην βάση δημιουργούμε ένα καινούργιο χρήστη και μία νέα βάση και δίνουμε στον χρήστη την δυνατότητα να την τροποποιεί. Όλα αυτά επιτυγχάνονται με τα παρακάτω:

### Πηγαίος κώδικας 1: Δημιουργία της βάσης δεδομένων

---

```
1 MariaDB [(none)]> CREATE USER 'tme119'@'localhost' IDENTIFIED BY
   ↳ 'tme119_password' ;
2 MariaDB [(none)]> GRANT USAGE ON *.* TO 'tme119'@'localhost' IDENTIFIED BY
   ↳ 'tme119_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0
   ↳ MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;
3 MariaDB [(none)]> CREATE DATABASE `tme119` CHARACTER SET utf8 COLLATE
   ↳ utf8_general_ci;
4 MariaDB [(none)]> GRANT ALL PRIVILEGES ON `tme119`.* TO 'tme119'@'localhost';
```

---

Ο λόγος για τον οποίο δημιουργούμε έναν καινούργιο χρήστη είναι για λόγους ασφαλείας έτσι ώστε να μην χρησιμοποιούμε συνέχεια τον υπερχρήστη. Ακόμα στην νέα βάση προτιμήθηκε να χρησιμοποιηθεί το utf8 έτσι ώστε η βάση μας να υποστηρίζει και τα ελληνικά. Μετά την δημιουργία του καινούργιου χρήστη ξαναμπάνουμε στο ΣΔΒΔ χρησιμοποιώντας τον καινούργιο χρήστη:

```
$ mysql --user=tme119 --password
Enter password:
```

Παρατηρούμε ότι στις μόνες βάσεις που έχει πρόσβαση ο συγκεκριμένος χρήστης είναι η βάση που δημιουργήσαμε πριν και η βάση που είναι δημιουργημένη από την MariaDB και περιέχει πληροφορίες για την βάση.

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| tme119 |
+-----+
2 rows in set (0.00 sec)
```

## 5.5 Δημιουργία πινάκων

### Πηγαίος κώδικας 2: Δημιουργία του πίνακα users

```

1 MariaDB [book]> CREATE TABLE users (uid INT PRIMARY KEY AUTO_INCREMENT,
  ↳ username VARCHAR(30) UNIQUE NOT NULL, password VARCHAR(50) NOT NULL,
  ↳ is_admin BOOL NOT NULL, first_name VARCHAR(100), last_name VARCHAR(100),
  ↳ join_date DATE, gender BOOL, email VARCHAR(255));
2
3 Query OK, 0 rows affected (0.06 sec)
4
5 MariaDB [tmell19]> describe users;
6 +-----+-----+-----+-----+-----+-----+
7 | Field      | Type           | Null | Key | Default | Extra          |
8 +-----+-----+-----+-----+-----+-----+
9 | uid        | int(11)        | NO   | PRI | NULL    | auto_increment |
10 | username   | varchar(30)    | NO   | UNI | NULL    |                |
11 | password   | varchar(50)    | NO   |     | NULL    |                |
12 | is_admin   | tinyint(1)     | NO   |     | NULL    |                |
13 | first_name | varchar(100)   | YES  |     | NULL    |                |
14 | last_name  | varchar(100)   | YES  |     | NULL    |                |
15 | join_date  | date           | YES  |     | NULL    |                |
16 | gender     | tinyint(1)     | YES  |     | NULL    |                |
17 | email      | varchar(255)   | YES  |     | NULL    |                |
18 +-----+-----+-----+-----+-----+-----+
19 9 rows in set (0.00 sec)
20
21 MariaDB [tmell19]> INSERT INTO
  ↳ users(username,password,is_admin,first_name,last_name,join_date,gender,email)
  ↳ VALUES('anagno',MD5('anagno'),1,'Vasilis','Anagnostopoulos','2015-02-
  ↳ 15',1,'anagnwstopoulos@hotmail.com');
22 Query OK, 1 row affected (0.02 sec);

```

---

### Πηγαίος κώδικας 3: Δημιουργία του πίνακα books

```

1 MariaDB [book]> CREATE TABLE books (id int UNSIGNED NOT NULL AUTO_INCREMENT,
  ↳ title VARCHAR(100) NOT NULL, description TEXT NOT NULL, PRIMARY KEY (id) );
2
3 Query OK, 0 rows affected (0.12 sec)
4
5 MariaDB [tmell19]> describe books;
6 +-----+-----+-----+-----+-----+-----+
7 | Field      | Type           | Null | Key | Default | Extra          |
8 +-----+-----+-----+-----+-----+-----+
9 | id        | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
10 | title     | varchar(100)    | NO   |     | NULL    |                |
11 | description | text           | NO   |     | NULL    |                |
12 +-----+-----+-----+-----+-----+-----+
13 3 rows in set (0.01 sec)

```

---

### Πηγαίος κώδικας 4: Δημιουργία του πίνακα categories

```

1 MariaDB [book]> CREATE TABLE categories (book_id INT UNSIGNED NOT NULL, type
  ↳ VARCHAR(30), PRIMARY KEY (book_id,type), FOREIGN KEY (book_id) REFERENCES
  ↳ books(id) );
2 Query OK, 0 rows affected (0.08 sec)
3
4 MariaDB [tmell9]> describe categories;
5 +-----+-----+-----+-----+-----+-----+
6 | Field      | Type                | Null | Key | Default | Extra |
7 +-----+-----+-----+-----+-----+-----+
8 | book_id    | int(10) unsigned    | NO   | PRI | NULL    |       |
9 | type       | varchar(30)         | NO   | PRI |         |       |
10 +-----+-----+-----+-----+-----+-----+
11 2 rows in set (0.00 sec)

```

---

### Πηγαίος κώδικας 5: Δημιουργία του πίνακα authors

```

1 MariaDB [book]> CREATE TABLE authors (id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  ↳ name VARCHAR(50) NOT NULL, PRIMARY KEY (id));
2 Query OK, 0 rows affected (0.08 sec)
3
4 MariaDB [tmell9]> describe authors;
5 +-----+-----+-----+-----+-----+-----+
6 | Field | Type                | Null | Key | Default | Extra |
7 +-----+-----+-----+-----+-----+-----+
8 | id    | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
9 | name  | varchar(50)         | NO   |     | NULL    |               |
10 +-----+-----+-----+-----+-----+-----+
11 2 rows in set (0.00 sec)

```

---

### Πηγαίος κώδικας 6: Δημιουργία του πίνακα writes

```

1 MariaDB [book]> CREATE TABLE writes (author_id INT UNSIGNED NOT NULL, book_id
  ↳ INT UNSIGNED NOT NULL, PRIMARY KEY (author_id, book_id), FOREIGN KEY
  ↳ (book_id) REFERENCES books(id), FOREIGN KEY (author_id) REFERENCES
  ↳ authors(id) );
2 Query OK, 0 rows affected (0.06 sec)
3
4 MariaDB [tmell9]> describe writes;
5 +-----+-----+-----+-----+-----+-----+
6 | Field      | Type                | Null | Key | Default | Extra |
7 +-----+-----+-----+-----+-----+-----+
8 | author_id  | int(10) unsigned    | NO   | PRI | NULL    |       |
9 | book_id    | int(10) unsigned    | NO   | PRI | NULL    |       |
10 +-----+-----+-----+-----+-----+-----+
11 2 rows in set (0.01 sec)

```

---

### Πηγαίος κώδικας 7: Δημιουργία του πίνακα publishers

---

```

1 MariaDB [book]> CREATE TABLE publishers ( id INT UNSIGNED NOT NULL
  ↳ AUTO_INCREMENT, name VARCHAR(50) NOT NULL, PRIMARY KEY (id) );
2 Query OK, 0 rows affected (0.03 sec)
3
4 MariaDB [tmel19]> describe publishers;
5 +-----+-----+-----+-----+-----+-----+
6 | Field | Type                | Null | Key | Default | Extra          |
7 +-----+-----+-----+-----+-----+-----+
8 | id    | int(10) unsigned    | NO   | PRI | NULL    | auto_increment |
9 | name  | varchar(50)         | NO   |     | NULL    |                |
10 +-----+-----+-----+-----+-----+-----+
11 2 rows in set (0.00 sec)

```

---

### Πηγαίος κώδικας 8: Δημιουργία του πίνακα editions

---

```

1 MariaDB [book]> CREATE TABLE editions ( isbn VARCHAR(50) UNIQUE NOT NULL,
  ↳ book_id INT UNSIGNED NOT NULL, publisher_id INT UNSIGNED NOT NULL,
  ↳ edition INT UNSIGNED, date DATE, language VARCHAR(2), PRIMARY KEY
  ↳ (isbn), FOREIGN KEY (book_id) REFERENCES books(id), FOREIGN KEY
  ↳ (publisher_id) REFERENCES publishers(id) );
2 Query OK, 0 rows affected (0.09 sec)
3
4 MariaDB [tmel19]> describe editions;
5 +-----+-----+-----+-----+-----+-----+
6 | Field          | Type                | Null | Key | Default | Extra          |
7 +-----+-----+-----+-----+-----+-----+
8 | isbn           | varchar(50)         | NO   | PRI | NULL    |                |
9 | book_id        | int(10) unsigned    | NO   | MUL | NULL    |                |
10 | publisher_id   | int(10) unsigned    | NO   | MUL | NULL    |                |
11 | edition        | int(10) unsigned    | YES  |     | NULL    |                |
12 | date           | date                | YES  |     | NULL    |                |
13 | language       | varchar(2)          | YES  |     | NULL    |                |
14 +-----+-----+-----+-----+-----+-----+
15 6 rows in set (0.00 sec)

```

---

### Πηγαίος κώδικας 9: Δημιουργία του πίνακα paperbooks

---

```

1 MariaDB [book]> CREATE TABLE paperbooks (id INT UNSIGNED NOT NULL
  ↳ AUTO_INCREMENT, isbn VARCHAR(50) NOT NULL, binding VARCHAR(30), location
  ↳ VARCHAR(100), PRIMARY KEY (id), FOREIGN KEY (isbn) REFERENCES
  ↳ editions(isbn));
2 Query OK, 0 rows affected (0.04 sec)
3
4 MariaDB [tmel19]> describe paperbooks;
5 +-----+-----+-----+-----+-----+-----+
6 | Field          | Type                | Null | Key | Default | Extra          |
7 +-----+-----+-----+-----+-----+-----+

```

```

8 | id          | int(10) unsigned | NO | PRI | NULL | auto_increment |
9 | isbn        | varchar(50)     | NO | MUL | NULL |                  |
10 | binding     | varchar(30)     | YES |    | NULL |                  |
11 | location    | varchar(100)    | YES |    | NULL |                  |
12 +-----+-----+-----+-----+-----+-----+
13 4 rows in set (0.00 sec)

```

### Πηγαίος κώδικας 10: Δημιουργία του πίνακα borrows

```

1 MariaDB [book]> CREATE TABLE borrows (paperbook_id INT UNSIGNED NOT
  ↳ NULL,username VARCHAR(50) NOT NULL, start_period DATE NOT NULL, end_period
  ↳ DATE, PRIMARY KEY (paperbook_id, username, start_period), FOREIGN KEY
  ↳ (paperbook_id) REFERENCES paperbooks(id), FOREIGN KEY (username)
  ↳ REFERENCES users(username));
2 Query OK, 0 rows affected (0.04 sec)
3
4 MariaDB [tmel19]> describe borrows;
5 +-----+-----+-----+-----+-----+-----+
6 | Field          | Type                | Null | Key | Default | Extra |
7 +-----+-----+-----+-----+-----+-----+
8 | paperbook_id   | int(10) unsigned   | NO   | PRI | NULL    |       |
9 | username       | varchar(50)        | NO   | PRI | NULL    |       |
10 | start_period   | date               | NO   | PRI | NULL    |       |
11 | end_period     | date               | YES  |    | NULL    |       |
12 +-----+-----+-----+-----+-----+-----+
13 4 rows in set (0.00 sec)

```

## 6 Ανάλυση και Σχεδίαση


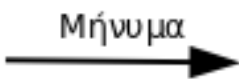
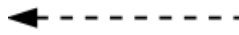
### 6.1 Περιπτώσεις χρήσης και διαγράμματα περιπτώσεων χρήσης

Όπως αναφέρθηκε και στην ενότητα 2.4.1 τα διαγράμματα περιπτώσεων - χρήσης (αγγλ. Use Case Diagrams) περιγράφουν τη συμπεριφορά ενός συστήματος από την οπτική γωνία ενός χρήστη.

### 6.2 Διαγράμματα σειράς

Τα διαγράμματα σειράς αναπαριστούν αλληλεπιδράσεις ανάμεσα στα αντικείμενα από μία χρονική άποψη. Η αναπαράσταση επικεντρώνεται στην έκφραση των αλληλεπιδράσεων. Οι συμβολισμοί που χρησιμοποιούνται στα διαγράμματα σειράς φαίνονται στον πίνακα 6.1.

Οι πιο σημαντική αλληλεπίδραση του πληροφοριακού μας συστήματος είναι η αγορά των εισιτηρίων και γι' αυτό το λόγο αναπαριστάται και σε διάγραμμα σειράς. Όπως φαίνεται και στο διάγραμμα σειράς στο σχήμα ?? ο πελάτης για να κλείσει το εισιτήριο του θα πρέπει πρώτα να συνδεθεί με το πληροφοριακό σύστημα. Έπειτα επιλέγει την ταινία που θέλει να δει και τον κινηματογράφο

<div data-bbox="304 647 544 736" data-label="Text"> <p><u>Αντικείμενο</u></p> </div>	<p>Αντικείμενο. Ένα αντικείμενο αναπαριστάται με ένα ορθογώνιο και μία κάθετη γραμμή, που καλείται γραμμή ζωής του αντικειμένου [4].</p>
	<p>Ενεργοποίηση αντικειμένου. Μία ενεργοποίηση ανταποκρίνεται στο χρόνο κατά την διάρκεια του οποίου ένα αντικείμενο εκτελεί μία ενέργεια, είτε απευθείας ή μέσω άλλου αντικειμένου, που το χρησιμοποιεί σαν ημισυμβαλλόμενο. Οι ενεργοποιήσεις αναπαριστώνται με ορθογώνιες ράβδους, που τοποθετούνται κατά μήκος των γραμμών ζωής. Η αρχή και το τέλος μίας ράβδου ανταποκρίνεται αντίστοιχα στην αρχή και το τέλος μίας ενεργοποίησης [4].</p>
	<p>Συγχρονισμένο Μηνύματα. Τα αντικείμενα επικοινωνούν ανταλλάσσοντας μηνύματα, τα οποία αναπαριστώνται με οριζόντια βέλη σχεδιασμένα από τον αποστολέα του μηνύματος προς τον παραλήπτη του μηνύματος [4]. Επειδή είναι συγχρονισμένα, θα πρέπει να περιμένει να τελειώσει η διαδικασία πριν προχωρήσει στην επόμενη</p>
	<p>Επιστροφή. Βέλος επιστροφής μηνύματος.</p>

**Πίνακας 6.1: Τα εικονίδια του διαγράμματος σειράς.**

στον οποίο θέλει να την δει. Αφού ολοκληρωθούν με επιτυχία τα παραπάνω επιλέγει τις θέσεις, αφού πρώτα κάνει έλεγχο για την διαθεσιμότητα τους. Ολοκληρώνοντας τα παραπάνω, όλα τα στοιχεία περνάνε και στην φόρμα παραγγελίας

### **6.3 Διάγραμμα Βασικών κλάσεων πεδίου Εφαρμογής**

### **6.4 Αναλυτικά διαγράμματα κλάσεων συνοδευμένα από OCL περιορισμούς**

### **6.5 Διαγράμματα Επικοινωνίας και Αλληλεπίδρασης**

## **Αναφορές**

- [1] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, The (2Nd Edition)*. Pearson Higher Education, 2004.
- [2] R.S. Wazlawick. *Object-Oriented Analysis and Design for Information Systems: Modeling with UML, OCL, and IFML*. Elsevier Science, 2014.
- [3] Wikipedia. Requirements analysis --- Wikipedia, the free encyclopedia. [http://en.wikipedia.org/wiki/Requirements\\_analysis](http://en.wikipedia.org/wiki/Requirements_analysis). [Πρόσβαση στις 8 Ιουλίου 2014].
- [4] Μαρία Βίρβου. Σημειώσεις διδασκαλίας για το μάθημα "Τεχνολογία Λογισμικού" - Η Γλώσσα μοντελοποίησης uml και μία διαδικασία εφαρμογής. Σημειώσεις μαθήματος, 2014.
- [5] Γιάννης Θεοδωρίδης. Σημειώσεις διδασκαλίας για το μάθημα "Βάσεις Δεδομένων". Σημειώσεις μαθήματος, 2013.
- [6] Τριάντης Α. Κωνσταντίνος. Ανάλυση απαιτήσεων για την ανάπτυξη πληροφοριακών συστημάτων. Μεθοδολογίες ανάλυσης απαιτήσεων στο πλαίσιο εναλλακτικών κύκλων ζωής έργων πληροφοριακών συστημάτων. Διενέργεια σχετικής μελέτης περίπτωσης. Master's thesis, Πανεπιστήμιο Πειραιώς, 2013.
- [7] Κέντρο Παραγωγικότητας Κύπρου. Requirements analysis - -- Wikipedia, the free encyclopedia. <http://www.kepa.gov.cy/diktiothite/Portal/PortalDocuments.aspx?DocumentId=a735c138-a74e-483b-8720-31bc9aca7169>. [Πρόσβαση στις 29 Δεκεμβρίου 2014].
- [8] Ειδική Υπηρεσία Διαχείρισης Επιχειρησιακού Προγράμματος. Έρευνα για τη διεύθυνση του Ίντερνετ στην Ελλάδα. [http://www.infosoc.gr/infosoc/el-GR/grafeiotypou/news/opis\\_news/general/e\\_europe.htm](http://www.infosoc.gr/infosoc/el-GR/grafeiotypou/news/opis_news/general/e_europe.htm). [Πρόσβαση στις 7 Ιανουαρίου 2015].