

## Εξαμηνιαία εργασία στις βάσεις δεδομένων

Γαλανόπουλος Σπυρίδων (AM 03120093),  
Ντόκας Ευθύμιος (AM 03120631),  
Πλάτανος Χαράλαμπος (AM 03120108)

03/06/2023

# Contents

<b>1</b>	<b>Αναφορά</b>	<b>2</b>
1.1	ER Diagram . . . . .	2
1.2	Relational Diagram . . . . .	3
1.3	DDL scripts . . . . .	4
1.3.1	Δημιουργία tables . . . . .	4
1.3.2	Constraints . . . . .	8
1.3.3	Indexes . . . . .	9
1.3.4	Triggers . . . . .	11
1.3.5	Events . . . . .	11
1.3.6	Views . . . . .	11
1.4	DML script/Insertions . . . . .	12
1.5	Queries . . . . .	13
<b>2</b>	<b>Οδηγός εγκατάστασης</b>	<b>14</b>
2.1	Βήμα 1 . . . . .	14
2.2	Βήμα 2 . . . . .	14
2.3	Βήμα 3 . . . . .	14
2.4	Βήμα 4 . . . . .	14
<b>3</b>	<b>User manual</b>	<b>15</b>
3.1	Login/Registration . . . . .	15
3.2	Admin . . . . .	16
3.2.1	Profile . . . . .	16
3.2.2	Add School Library . . . . .	17
3.2.3	Manager Requests . . . . .	17
3.2.4	Database Stats . . . . .	17
3.2.5	BackUp/Restore . . . . .	19
3.2.6	Schools . . . . .	19
3.2.7	Manager Profiles . . . . .	20
3.3	Manager . . . . .	21
3.3.1	Book Browser . . . . .	22
3.3.2	Review Acceptance . . . . .	24
3.3.3	Book Reviews . . . . .	24
3.3.4	Borrows . . . . .	25
3.3.5	Reservations . . . . .	25
3.3.6	Member Requests . . . . .	26
3.3.7	Profile . . . . .	26
3.3.8	Member Profiles . . . . .	27
3.4	Member-Teacher/Student . . . . .	28
3.4.1	Book Browser . . . . .	28
3.4.2	Borrows . . . . .	29
3.4.3	Borrowed Books . . . . .	30
3.4.4	Reservations . . . . .	30
3.4.5	Profile . . . . .	30

## Μέρος 1

### Αναφορά

#### Link για το Git repo

Ο πηγαίος κώδικας της εφαρμογής μας βρίσκεται [εδώ](#).

#### 1.1 ER Diagram

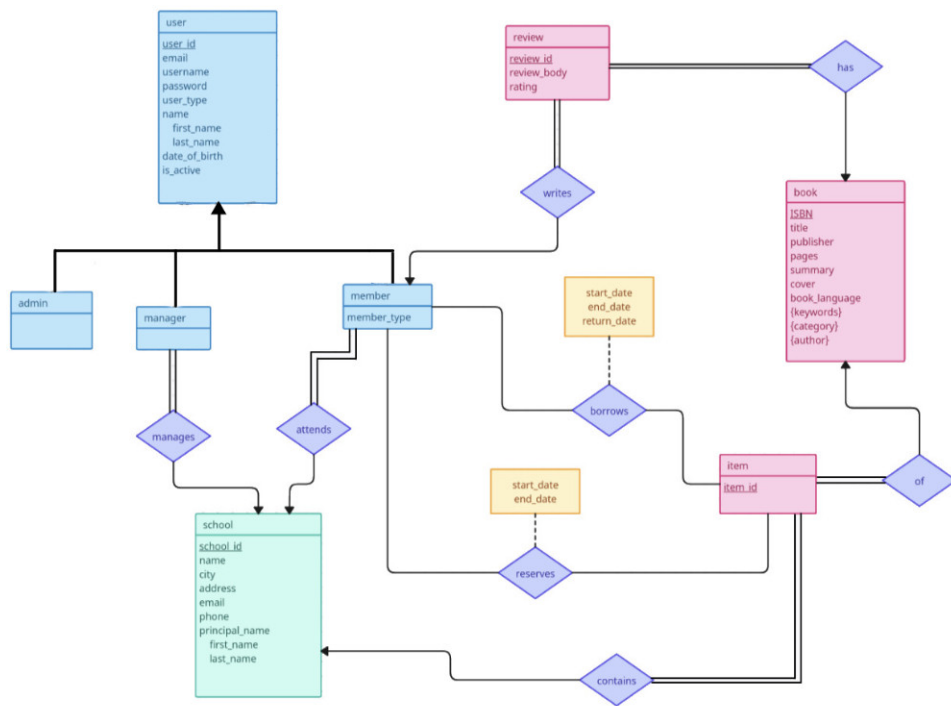


Figure 1.1: Διάγραμμα ER της βάσης δεδομένων

Στο Σχήμα 1.1 απεικονίζεται το ER διάγραμμα της βάσης δεδομένων. Οι βασικές οντότητες της βάσης είναι οι χρήστες (users), τα βιβλία (books), τα σχολεία (schools), τα αντικείμενα (items) και οι αξιολογήσεις (reviews).

- Η οντότητα school αναπαριστά κάθε μεμονωμένη σχολική βιβλιοθήκη, και περιέχει ως εκ τούτου τις πληροφορίες που είναι απαραίτητες για κάθε σχολείο.
- Οι users διακρίνονται περαιτέρω σε τρεις τύπους: διαχειριστές συστήματος (admins), χειριστές βιβλιοθήκης (managers) και μέλη βιβλιοθήκης (members). Οι τελευταίοι δύο συνδέονται με ένα μοναδικό σχολείο, ενώ ο admin δεν εξαρτάται από κάποιο συγκεκριμένο. Αξίζει να προστεθεί ότι, αναφορικά με τον τρόπο οργάνωσης της εφαρμογής μας, τα members διακρίνονται επιπλέον σε εκπαιδευτικούς (teachers) και μαθητές (students). Εντούτοις, και οι δύο τύποι έχουν τα ίδια πεδία και συμμετέχουν στις ίδιες σχέσεις, οπότε δεν κρίναμε σκόπιμο να αναπαραστήσουμε τη διάκριση αυτή στο ER διάγραμμα (οι μόνες διαφορές έγκεινται στο όριο του αριθμού βιβλίων που κάποιος από αυτούς μπορεί να δανειστεί σε μία εβδομάδα, καθώς και στο γεγονός ότι οι teachers δε χρειάζονται έγκριση για τη δημοσιοποίηση αξιολογήσεων).

- Η οντότητα book αναπαριστά τους διαφορετικούς τίτλους βιβλίων που εμφανίζονται στη βάση. Περιέχει γενικές πληροφορίες για ένα βιβλίο, και είναι η οντότητα για την οποία μπορούν να κάνουν κράτηση τα μέλη μιας βιβλιοθήκης.
- Η οντότητα item αναπαριστά ένα μεμονωμένο αντίτυπο ενός συγκεκριμένου βιβλίου. Είναι η οντότητα την οποία μπορούν να δανειστούν τα μέλη μιας βιβλιοθήκης. Βέβαια, δεδομένης της δομής της εφαρμογής σύμφωνα με την εκφώνηση, δεν είναι απαραίτητη η συμπερίληψη αυτής της οντότητας. Θα μπορούσαμε εναλλακτικά να την αντικαταστήσουμε με μία relationship "school has book", που θα είχε ως attribute τον αριθμό των αντιτύπων στα αποθέματα ενός σχολείου. Κρίναμε όμως ότι είναι σημαντικό να υπάρχει η δυνατότητα διάκρισης των αντιτύπων μεταξύ τους, ώστε οι διαχειριστές της βιβλιοθήκης να έχουν πληρέστερη εποπτεία της κατάστασης των αποθεμάτων της.
- Η οντότητα review αναπαριστά την αξιολόγηση ενός χρήστη για ένα βιβλίο. Προφανώς η αξιολόγηση δεν αφορά το συγκεκριμένο αντικείμενο (item), αλλά τον τίτλο του βιβλίου (book).

## 1.2 Relational Diagram

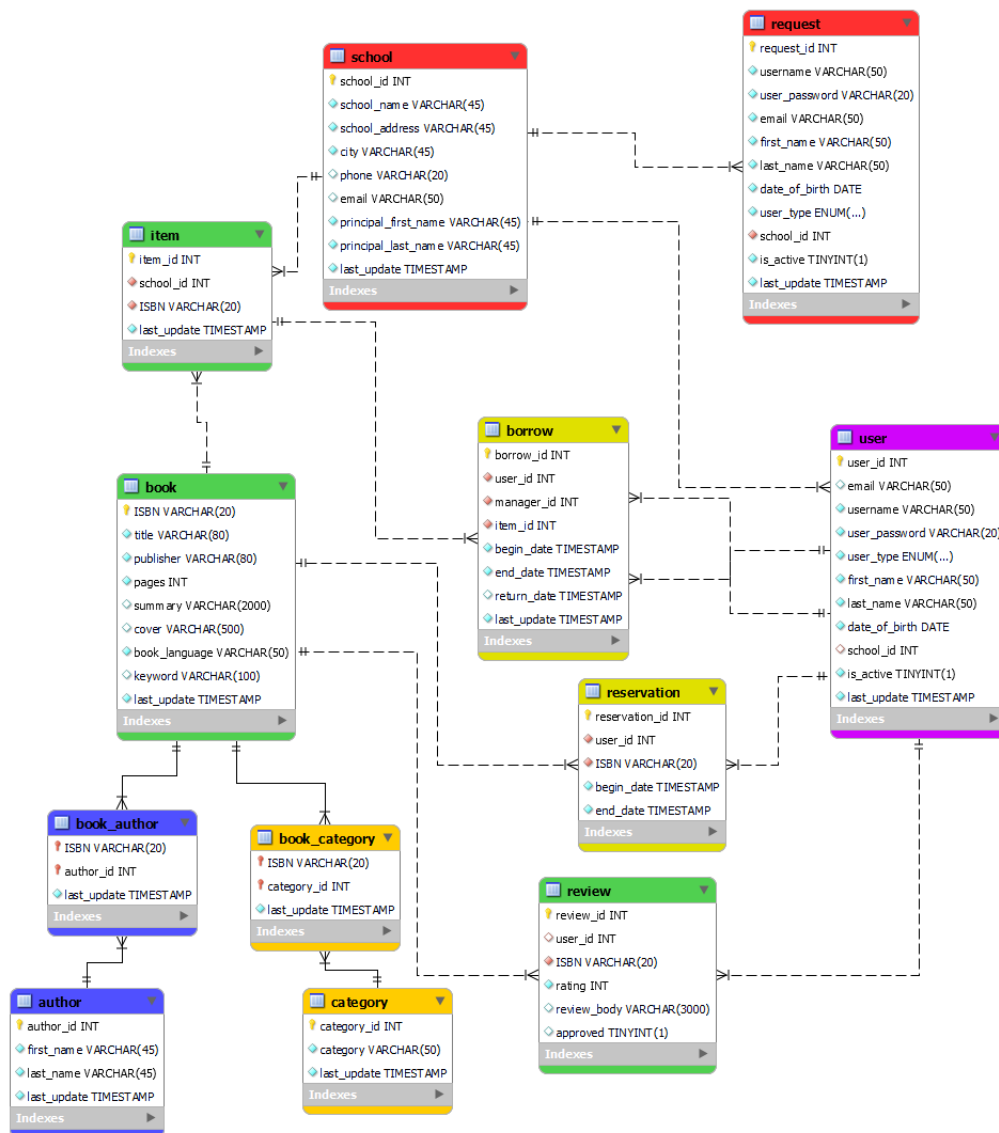


Figure 1.2: Σχεσιακό διάγραμμα της βάσης δεδομένων

Στο Σχήμα 1.2 παρατίθεται το σχεσιακό διάγραμμα της βάσης δεδομένων. Ως γνωστόν, το σχεσιακό διάγραμμα αποτελεί μια λεπτομερή αναπαράσταση του τρόπου με τον οποίο το μοντέλο ER υλοποιείται στην πραγματικότητα στη βάση δεδομένων. Παρακάτω παραθέτουμε επεξηγήσεις για τις σημαντικότερες σχεδιαστικές επιλογές που έγιναν κατά τη μετάβαση από το ER στο σχεσιακό διάγραμμα

1. Στο ER διάγραμμα εμφανίζονται αρκετές one-to-many σχέσεις με total participation από τη μεριά του "many" (για παράδειγμα manager manages school, book has review κτλ). Αυτές τις αναπαράστησάμε στη βάση με χρήση foreign

keys στη μεριά του "many", που δείχνουν στο αντίστοιχο entry του table στη μεριά του "one". Έτσι, για παράδειγμα ο manager διαθέτει ως foreign key το *school\_id* του σχολείου στο οποίο εργάζεται. Η χρήση foreign keys καθιστά εφικτή την αναζήτηση στα πλαίσια διαφόρων queries (π.χ. εύρεση των items με συγκεκριμένο ISBN που υπάρχουν σε ένα σχολείο).

2. Για τις many-to-many σχέσεις από την άλλη, χρησιμοποιήσαμε ξεχωριστά tables, ώστε να μπορούμε να αποθηκεύουμε εύκολα όλες τις πληροφορίες που αφορούν μια συγκεκριμένη έκφανση μιας σχέσης και να καθίσταται εφικτή η υλοποίηση περίπλοκων queries.
3. Στο ER διάγραμμα είχαμε αναπαράσθησει τους συγγραφείς, τις κατηγορίες και τις λέξεις-κλειδιά ενός βιβλίου με multivalued attributes. Αυτό όμως δεν είναι πρακτικά υλοποιήσιμο στο relational diagram, οπότε είχαμε 2 εναλλακτικές επιλογές: Είτε θα τα αναπαριστούσαμε με strings, όπου με κόμματα θα χωρίζαμε π.χ. τον έναν συγγραφέα από τον άλλο είτε θα δημιουργούσαμε 2 ξεχωριστά tables: ένα table που να περιέχει το όνομα π.χ. ενός συγγραφέα και ένα δεύτερο table που να περιέχει ζεύγη (βιβλίο-συγγραφέας) που να δείχνει ότι ένα συγκεκριμένο βιβλίο είναι γραμμένο από ένα συγκεκριμένο συγγραφέα. Εν τέλει αποφασίσαμε να υλοποιήσουμε τους συγγραφείς και τις κατηγορίες με τη δεύτερη μέθοδο και τα keywords με την πρώτη. Αποφασίσαμε να αναπαράσθουμε τα keywords με τον τρόπο αυτό αφού δε χρησιμοποιούνται σε κανένα query, υπάρχουν απλά για να συμπληρώνουν την περιγραφή του βιβλίου, οπότε είναι λογικό να τα κωδικοποιήσουμε στη βάση με τον απλούστερο και φθηνότερο σε μνήμη τρόπο. Αντίθετα, οι συγγραφείς και οι κατηγορίες συμμετέχουν σε αρκετά queries (π.χ. αναζήτηση βιβλίων που έχουν γραφεί από συγκεκριμένους συγγραφείς ή που ανήκουν σε συγκεκριμένες κατηγορίες), οπότε έπρεπε να αποθηκευτούν με μία πιο ευέλικτη δομή.
4. Όσον αφορά τους users, σε αντίθεση με το ER, όπου υπάρχουν οντότητες που είναι specializations της οντότητας user, στη βάση δεδομένων αποφασίσαμε να αποθηκεύσουμε όλους τους users στο ίδιο table (και να διακρίνουμε τον έναν τύπο από τον άλλο με ένα enum - το *user\_type*). Λάβαμε αυτή την απόφαση δεδομένων των μικρών διαφοροποιήσεων μεταξύ των διαφορετικών τύπων χρηστών. Ενδεικτικά αναφέρουμε ότι η μόνη διαφορά μεταξύ διαφορετικών τύπων χρηστών είναι ότι οι admins δεν έχουν *school\_id*. Ήταν λοιπόν σαφώς ευκολότερο απλώς να επιτρέψουμε στο *school\_id* να είναι κενό από ό,τι να δημιουργήσουμε ξεχωριστά tables. Με τον τρόπο αυτό μάλιστα επιταχύνονται σημαντικά και τα queries, αφού δε χρειάζεται να κάνουμε inner joins για να μεταβαίνουμε από το manager table στο user table για να αποσπάσουμε μια πληροφορία από ένα attribute που είναι κοινό για όλους.

Όσον αφορά το enum, αξίζει να αναφερθεί ότι οι τιμές που παίρνει βρίσκονται στο σύνολο { 'ADMIN', 'MANAGER', 'TEACHER', 'STUDENT' }. Εδώ δηλαδή αποφασίσαμε να αποθηκεύουμε τον τύπο του κάθε member, που άλλωστε είναι απαραίτητος για τον καθορισμό των δικαιωμάτων των members μέσα στην εφαρμογή.

5. Με βάση την εκφώνηση προκύπτει ότι τα υποψήφια νέα members και οι υποψήφιοι νέοι managers πρέπει πρώτα να υποβάλουν αίτηση, την οποία θα δεχτεί ή θα απορρίψει κάποιος manager ή admin αντίστοιχα. Χρειαζόταν λοιπόν να συμπεριλάβουμε στη βάση μας μία δομή που να αποθηκεύει τους υποψήφιους νέους χρήστες. Για τον σκοπό αυτό δημιουργήσαμε τον πίνακα requests. Δεδομένου ότι οι managers και τα members έχουν τα ίδια ακριβώς στοιχεία, δεν υπήρχε λόγος να διακρίνουμε διαφορετικούς τύπους request, απλά συμπεριελάβαμε ένα enum attribute με τον τύπο του υποψήφιου χρήστη (που παίρνει τις τιμές 'MANAGER', 'TEACHER', 'STUDENT').

Σημειωτέον ότι στο ER διάγραμμα αποφασίσαμε να μη συμπεριλάβουμε καμία αναφορά στις αιτήσεις χρηστών, δεδομένου ότι αυτές δεν αφορούν τόσο τη δομή της βάσης, όσο τον τρόπο με τον οποίο μεταβάλλονται οι πληροφορίες σε αυτή.

## 1.3 DDL scripts

Στις παρακάτω ενότητες παρατίθενται τα διάφορα DDL scripts που χρησιμοποιήσαμε για τη δημιουργία της βάσης μας.

### 1.3.1 Δημιουργία tables

Παρακάτω παρατίθενται οι εντολές που περιέχει το DDL script για τη δημιουργία των διαφόρων tables της βάσης (*ddl.sql*), μαζί με συνοπτικές επεξηγήσεις για τα κυριότερα attributes κάθε table.

Πρώτα απ' όλα όμως, ας επισημάνουμε το γεγονός ότι σε όλους τους πίνακες της βάσης συμπεριλάβαμε το attribute *last\_update*, που σε ρεαλιστικές συνθήκες θα καθιστούσε ευκολότερη τη διαχείριση της βάσης και τον εντοπισμό και διόρθωση λαθών.

Επίσης, πριν τη δημιουργία των διαφόρων tables, συμπεριλάβαμε στο DDL τις ακόλουθες εντολές, ώστε να διαγράφεται οποιαδήποτε έκδοση της βάσης υπήρχε ήδη αποθηκευμένη.

```
drop database if exists myLibs;
create database myLibs;
USE myLibs;
```

## Σχολείο

```
CREATE TABLE school (
  school_id INT NOT NULL AUTO_INCREMENT,
  school_name VARCHAR(45) NOT NULL,
  school_address VARCHAR(45) NOT NULL,
  city VARCHAR(45) NOT NULL,
  phone VARCHAR(20),
  email VARCHAR(50),
  principal_first_name VARCHAR(45) NOT NULL,
  principal_last_name VARCHAR(45) NOT NULL,
  last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (school_id)
);
```

## Βιβλίο

```
CREATE TABLE book (
  ISBN VARCHAR(20) NOT NULL,
  title VARCHAR(80) NOT NULL,
  publisher VARCHAR(80) NOT NULL,
  pages INT NOT NULL,
  summary VARCHAR(2000),
  cover VARCHAR(500),
  book_language VARCHAR(50) NOT NULL,
  keyword VARCHAR (100),
  last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (ISBN)
);
```

Αξίζει να σημειωθεί ότι, δεδομένου ότι το ISBN ενός βιβλίου είναι μοναδικό, δε χρειαζόταν να δημιουργήσουμε ξεχωριστό πεδίο "book\_id" για primary key, αλλά μπορούσαμε να χρησιμοποιήσουμε απευθείας το ISBN.

Επίσης επισημαίνουμε ότι, αναφορικά με το cover, επιλέξαμε να αποθηκεύουμε στη βάση ένα url, το οποίο προβάλλουμε στη σελίδα HTML. Στην ουσία γίνεται GET request στο συγκεκριμένο url, και στο response πρέπει να περιέχεται η αντίστοιχη εικόνα. Όταν ένας χρήστης προσθέτει ή τροποποιεί ένα βιβλίο στη βάση και ανεβάζει ένα αρχείο για το εξώφυλλο, αποθηκεύουμε στο τοπικό file system του server την εικόνα του βιβλίου, και στη βάση αποθηκεύουμε το route '/images/ < ISBN >'. Τα βασικά πλεονεκτήματα αυτής της μεθόδου είναι ότι 1. μειώνει το μέγεθος της βάσης, αφού αποθηκεύουμε μόνο ένα string και όχι ολόκληρη την εικόνα και 2. επιτρέπει να προβάλλονται εικόνες από εξωτερικούς συνδέσμους, που δεν είναι αποθηκευμένες στο file system του server.

## Κατηγορία

```
CREATE TABLE category (
  category_id INT NOT NULL AUTO_INCREMENT,
```

```

category VARCHAR(50) NOT NULL,
last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY(category_id)
);

CREATE TABLE book_category(
ISBN VARCHAR(20) NOT NULL,
category_id INT NOT NULL,
last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY (ISBN, category_id),
FOREIGN KEY (ISBN) REFERENCES book(ISBN) ON DELETE RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (category_id) REFERENCES category(category_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

Για την υλοποίηση περίπλοκων queries που αφορούν κατηγορίες, αποφασίσαμε να τις αναπαραστήσουμε με ξεχωριστό relation. Αυτό προϋπέθετε όμως να κωδικοποιήσουμε και τη many-to-many σχέση μεταξύ των βιβλίων και των κατηγοριών (από το ER). Για τον σκοπό αυτό δημιουργήσαμε το relation *book\_category*, το οποίο έχει ως foreign keys το id της κατηγορίας και το ISBN του βιβλίου, αναπαριστώντας τη σχέση "το βιβλίο με το συγκεκριμένο ISBN ανήκει στη συγκεκριμένη κατηγορία".

## Συγγραφέας

```

CREATE TABLE author (
author_id INT NOT NULL AUTO_INCREMENT,
first_name VARCHAR(45) NOT NULL,
last_name VARCHAR(45) NOT NULL,
last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY(author_id)
);

CREATE TABLE book_author(
ISBN VARCHAR(20) NOT NULL,
author_id INT NOT NULL,
PRIMARY KEY (ISBN, author_id),
last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (ISBN) REFERENCES book(ISBN) ON DELETE RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (author_id) REFERENCES author(author_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

Για την αποθήκευση των συγγραφέων στη βάση χρησιμοποιήσαμε το ίδιο ακριβώς μοτίβο με τις κατηγορίες, δημιουργώντας ένα νέο relation για τους ίδιους τους συγγραφείς και ένα επιπλέον για τη σχέση "συγγραφέας έχει συγγράψει βιβλίο".

## Αντικείμενο

```

CREATE TABLE item (
item_id INT NOT NULL AUTO_INCREMENT,
school_id INT NOT NULL,
ISBN VARCHAR(20) NOT NULL,
last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY (item_id),
FOREIGN KEY (school_id) REFERENCES school(school_id) ON DELETE RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (ISBN) REFERENCES book(ISBN) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

## Χρήστης

```

CREATE TABLE user (
user_id INT NOT NULL AUTO_INCREMENT,

```

```

email VARCHAR(50) NULL,
username VARCHAR(50) NOT NULL,
user_password VARCHAR (20) NOT NULL,
user_type ENUM('ADMIN', 'MANAGER', 'TEACHER', 'STUDENT') NOT NULL,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50) NOT NULL,
date_of_birth DATE NOT NULL,
school_id INT NULL,
is_active TINYINT(1) DEFAULT 1 NOT NULL,
last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY (user_id),
FOREIGN KEY (school_id) REFERENCES school(school_id) ON DELETE RESTRICT ON UPDATE CASCADE
);

```

Τα στοιχεία είναι όλα υποχρεωτικά για όλους τους χρήστες για λογιστικούς λόγους. Μοναδική εξαίρεση αποτελεί το *school\_id*, το οποίο δε θα μπορούσε να είναι υποχρεωτικό για τους διαχειριστές (admins) που δεν υπάρχουν σε κάποιο συγκεκριμένο σχολείο.

Προκειμένου να υπάρχει δυνατότητα απενεργοποίησης των λογαριασμών κάποιων χρηστών, προσθέσαμε στον πίνακα user το boolean πεδίο *is\_active*. Αξίζει να σημειωθεί ότι δεν κάνουμε διάκριση μεταξύ διαγεγραμμένων και απενεργοποιημένων χρηστών, ώστε να διατηρούνται πάντα οι πληροφορίες στη βάση για χρήστες με συμπεριφορά που παραβιάζει τους κανόνες μιας βιβλιοθήκης.

## Αίτηση

```

CREATE TABLE request (
request_id INT NOT NULL AUTO_INCREMENT,
username VARCHAR(50) NOT NULL,
user_password VARCHAR (20) NOT NULL,
email VARCHAR(50) NOT NULL,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50) NOT NULL,
date_of_birth DATE NOT NULL,
user_type ENUM('MANAGER', 'TEACHER', 'STUDENT') NOT NULL,
school_id INT NOT NULL,
is_active TINYINT(1) DEFAULT 1 NOT NULL,
last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY(school_id) REFERENCES school(school_id) ON DELETE RESTRICT ON UPDATE CASCADE,
PRIMARY KEY (request_id)
);

```

Όπως και για την περίπτωση των χρηστών, οι απορριφθείσες αιτήσεις δε διαγράφονται, αλλά απενεργοποιούνται (μέσω του πεδίου *is\_active*) προς αποφυγή λαθών.

## Δανεισμός

```

CREATE TABLE borrow (
borrow_id INT NOT NULL AUTO_INCREMENT,
user_id INT NOT NULL,
manager_id INT NOT NULL,
item_id INT NOT NULL,
begin_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
end_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
return_date TIMESTAMP NULL,
last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
PRIMARY KEY (borrow_id),
FOREIGN KEY (manager_id) REFERENCES user(user_id) ON DELETE RESTRICT ON UPDATE CASCADE,
FOREIGN KEY (user_id) REFERENCES user(user_id) ON DELETE RESTRICT ON UPDATE CASCADE,
);

```



```
FOREIGN KEY (item_id) REFERENCES item(item_id) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

Ο πίνακας `borrow` περιέχει πρώτα απ' όλα πληροφορίες για τους μετέχοντες σε έναν δανεισμό, και συγκεκριμένα τον χρήστη που δανείζεται, το αντικείμενο (αντίτυπο βιβλίου) που του παραχωρείται και τον διαχειριστή που του δανείζει το βιβλίο. Οι πληροφορίες αυτές κωδικοποιούνται ως foreign keys που συνδέονται με primary keys των αντίστοιχων πεδίων. Με τον τρόπο αυτό υλοποιείται ουσιαστικά η many-to-many σχέση του δανεισμού από το ER διάγραμμα.

Έπειτα περιέχονται τρεις ημερομηνίες, η ημερομηνία έναρξης δανεισμού, η προθεσμία επιστροφής του βιβλίου και η ημερομηνία επιστροφής του βιβλίου. Παρότι, με βάση τις προδιαγραφές της εκφώνησης, είναι περιττό να διατηρούνται και η ημερομηνία δανεισμού και η προθεσμία επιστροφής (δεδομένου ότι τα βιβλία παραχωρούνται για μία εβδομάδα), αποφασίσαμε να συμπεριλάβουμε την πληροφορία και για τα δύο, ώστε να είναι εύκολη η τροποποίηση της εφαρμογής σε περίπτωση που ο συγκεκριμένος κανόνας (σχετικά με τη διάρκεια δανεισμού) αλλάξει. Το πεδίο `return_date` παρέχει διττή πληροφορία: αν το βιβλίο έχει επιστραφεί, τότε το `return_date` περιέχει την ημερομηνία επιστροφής του. Αν το βιβλίο δεν έχει ακόμα επιστραφεί, τότε είναι κενό.

## Κράτηση

```
CREATE TABLE reservation (
    reservation_id INT NOT NULL AUTO_INCREMENT,
    user_id INT NOT NULL,
    ISBN VARCHAR(20) NOT NULL,
    begin_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    end_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (reservation_id),
    FOREIGN KEY (user_id) REFERENCES user(user_id) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (ISBN) REFERENCES book(ISBN) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

Η αποθήκευση των κρατήσεων στη βάση ακολουθεί το ίδιο μοτίβο με τους δανεισμούς, με τη βασική διαφορά ότι σε αυτή την περίπτωση δε γίνεται κράτηση σε `item`, αλλά σε `book`. Άλλωστε, ένας χρήστης που κάνει κράτηση για ένα βιβλίο δεν ενδιαφέρεται για συγκεκριμένο αντίτυπο, απλά θέλει ένα από τα διαθέσιμα αντίτυπα της βιβλιοθήκης του σχολείου του.

## Αξιολόγηση

```
CREATE TABLE review (
    review_id INT NOT NULL AUTO_INCREMENT,
    user_id INT,
    ISBN VARCHAR(20) NOT NULL,
    rating INT NOT NULL,
    review_body VARCHAR(3000),
    approved TINYINT(1) DEFAULT 0,
    PRIMARY KEY (review_id),
    FOREIGN KEY (user_id) REFERENCES user(user_id) ON DELETE SET NULL ON UPDATE CASCADE,
    FOREIGN KEY (ISBN) REFERENCES book(ISBN) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

Αξιοσημείωτο για τις αξιολογήσεις είναι ότι για τις αξιολογήσεις των μαθητών απαιτείται πρώτα έγκριση από `manager`. Για τον σκοπό αυτό υπάρχει το πεδίο `approved`, που είναι 1 για τα reviews των μαθητών που έχουν εγκριθεί (καθώς και όλα τα reviews εκπαιδευτικών), και 0 για reviews μαθητών που αναμένουν έγκριση. Αξίζει να σημειωθεί ότι, σε αντίθεση με την περίπτωση των `users` και των `requests`, εδώ οι απορριφθείσες αξιολογήσεις διαγράφονται από τη βάση, αφού δεν υπάρχει λόγος να διατηρείται η πληροφορία που περιέχουν.

### 1.3.2 Constraints

Παρακάτω παρατίθενται οι περιορισμοί που θέσαμε για τα δεδομένα (στο αρχείο [constraints.sql](#)), ώστε να διασφαλίζεται ότι τα δεδομένα που υπάρχουν εντός της βάσης είναι έγκυρα. Αξίζει βέβαια να αναφερθεί ότι ένας μεγάλος αριθμός περιορισμών

έχουν ήδη οριστεί στα πλαίσια των εντολών CREATE TABLE, και συγκεκριμένα οι περιορισμοί για μη κενά πεδία και foreign keys.

```
ALTER TABLE school ADD CONSTRAINT unique_school_name UNIQUE (school_name);
ALTER TABLE school ADD CONSTRAINT unique_phone UNIQUE (phone);

ALTER TABLE book ADD CONSTRAINT pages_positive CHECK(pages > 0);
ALTER TABLE book ADD CONSTRAINT chk_ISBN CHECK (LENGTH(ISBN) = 13 AND ISBN REGEXP '^[0-9]+$');

ALTER TABLE category ADD CONSTRAINT unique_category UNIQUE (category);

ALTER TABLE author ADD CONSTRAINT unique_author UNIQUE (first_name, last_name);

ALTER TABLE borrow ADD CONSTRAINT check_borrow_date CHECK(begin_date < end_date);

ALTER TABLE reservation ADD CONSTRAINT check_reservation_date CHECK(begin_date < end_date);

ALTER TABLE user ADD CONSTRAINT user_unique_email UNIQUE (email);
ALTER TABLE user ADD CONSTRAINT user_unique_username UNIQUE (username);
ALTER TABLE user ADD CONSTRAINT user_email_not_null CHECK(user_type = 'ADMIN' OR email IS NOT NULL);

ALTER TABLE request ADD CONSTRAINT request_unique_email UNIQUE (email);

ALTER TABLE review ADD CONSTRAINT check_rating CHECK(rating >= 1 AND rating <=5);
```

Αξίζει να σημειωθεί ότι, παρότι στους users, στα categories και στους συγγραφείς θα μπορούσαμε ήδη να χρησιμοποιήσουμε κάποια από τα attributes τους ως primary keys (π.χ. το username για τους χρήστες, το category για την κατηγορία και το ζεύγος (*first\_name*, *last\_name*) για τους συγγραφείς), επιλέξαμε να δημιουργήσουμε ξεχωριστά πεδία για ids, αφού η σύγκριση μεταξύ αριθμών είναι γρηγορότερη από τη σύγκριση μεταξύ strings, επιταχύνοντας έτσι τα queries μας.

### 1.3.3 Indexes

Παρακάτω παρατίθενται τα ευρετήρια που προσθέσαμε στη βάση (από το αρχείο [indexes.sql](#)), με αιτιολογήσεις για τις επιλογές μας.

Αξίζει να σημειωθεί ότι παντού παρακάτω λάβαμε υπόψιν ότι η βάση δημιουργεί ούτως ή άλλως αυτόματα indexes για τα primary keys κάθε πεδίου.

#### Σχολείο

Δεδομένου ότι αφενός όλες οι αναζητήσεις στον πίνακα school γίνονται μέσω του *school\_id*, αφετέρου ο πίνακας school πιθανότατα δε θα περιέχει πολλά rows (τα σχολεία της χώρας θα είναι σίγουρα τάξεις μεγέθους λιγότερα από τους χρήστες της εφαρμογής), αποφασίσαμε να μη δημιουργήσουμε indexes για τον πίνακα school.

#### Βιβλίο

```
CREATE INDEX idx_title ON book (title);
```

Οι μόνες περιπτώσεις που επιλέγουμε rows από τον πίνακα book είναι στα πλαίσια των joins με τον πίνακα αυτό και στα πλαίσια της αναζήτησης βιβλίων με φίλτρο. Στα joins, η επιλογή των books γίνεται με κριτήριο το ISBN, που είναι ούτως ή άλλως το primary key του βιβλίου. Στην αναζήτηση με φίλτρο το μόνο άλλο attribute του βιβλίου με κριτήριο το οποίο γίνεται αναζήτηση στον πίνακα book είναι ο title. Για αυτό και τον προσθέσαμε ως index.

#### Κατηγορία

```
CREATE INDEX idx_category ON category (category);
```

Στα πλαίσια της αναζήτησης βιβλίων με φίλτρα ο χρήστης επιλέγει τις κατηγορίες που τον ενδιαφέρουν με βάση το όνομά τους. Έπειτα στο backend χρησιμοποιούμε τα ονόματα των κατηγοριών προκειμένου να βρούμε τα αντίστοιχα *category\_id*

για την υλοποίηση των joins με τον πίνακα *book\_category*. Βοηθάει λοιπόν να υπάρχει ευρετήριο για το όνομα της κατηγορίας, ώστε αυτή η αναζήτηση να γίνεται γρηγορότερα.

## Συγγραφέας

```
CREATE INDEX idx_author_name ON author (first_name, last_name);
```

Προσθέσαμε τον παραπάνω index με το ίδιο σκεπτικό με την προηγούμενη παράγραφο (για το *category*), αφού χρησιμοποιείται στα πλαίσια της αναζήτησης βιβλίων με φίλτρο το όνομα του συγγραφέα.

## Αντικείμενο

```
CREATE INDEX idx_item_school_id_ISBN ON item (ISBN, school_id);
```

Σε πολλά σημεία της βάσης αναζητώνται αντικείμενα με βάση το ISBN τους, επειδή χρησιμοποιείται στα inner joins με τον πίνακα book. Επίσης, στα πλαίσια της αναζήτησης του αριθμού των διαθέσιμων αντιτύπων σε ένα σχολείο (π.χ. στο *school\_book* view που περιγράφουμε παρακάτω) αναζητούμε αντικείμενα με κριτήριο το ζεύγος (ISBN, *school\_id*). Κρίναμε λοιπόν σκόπιμο να προσθέσουμε τον παραπάνω index, ο οποίος όχι μόνο επιταχύνει τα queries στο */books* και στο */book/ < isbn >*, αλλά και σε όλα τα υπόλοιπα queries με joins με τον πίνακα book, αφού δημιουργεί index και για το attribute ISBN. Φυσικά, δεδομένου ότι πουθενά δεν κάνουμε queries όπου αναζητούμε items με κριτήριο το *school\_id*, δεν κρίναμε σκόπιμο να συμπεριλάβουμε τέτοιο index.

## Χρήστης

```
CREATE INDEX idx_username ON user(username, user_password);  
CREATE INDEX idx_user_name ON user(last_name, first_name);  
CREATE INDEX idx_user_school ON user(school_id);
```

Τα περισσότερα queries στον πίνακα user γίνονται είτε με το *user\_id* (στα πλαίσια των joins) είτε με το username. Επίσης, στο login αναζητούμε έναν χρήστη στη βάση με κριτήριο το ζεύγος (username, *user\_password*). Για αυτούς τους λόγους κρίναμε σκόπιμο να δημιουργήσουμε τον index *idx\_username*.

Πέραν αυτού, οι managers μπορούν να αναζητήσουν δανεισμούς, κρατήσεις και προφίλ χρηστών με κριτήριο το ονοματεπώνυμο, για αυτό και δημιουργήσαμε το *idx\_user\_name*.

Τέλος, δεδομένου ότι σε ορισμένες περιπτώσεις αναζητούμε χρήστες εντός ενός σχολείου, κρίναμε σκόπιμο να δημιουργήσουμε και τον *idx\_user\_school* για την επιτάχυνση των σχετικών queries.

## Αίτηση

```
CREATE INDEX idx_request_school_id ON request(is_active, school_id);  
CREATE INDEX idx_request_email ON request(email);
```

Τις περισσότερες φορές που αναζητούμε requests ζητούμε να επιστραφούν όλα τα active requests που βρίσκονται εντός ενός συγκεκριμένου σχολείου (για τους managers). Κρίναμε λοιπόν σκόπιμο να υπάρχει index βασισμένο σε αυτά τα δύο κριτήρια. Επίσης, κατά την έγκριση ενός request και τη δημιουργία ενός νέου χρήστη, πρώτα αναζητείται το αντίστοιχο row στο request. Για αυτό και δημιουργήσαμε τον index *idx\_request\_email*.

## Δανεισμός

```
CREATE INDEX idx_borrow ON borrow(user_id, item_id);
```

Στα περισσότερα queries με borrow αναζητούμε με κριτήριο το ζεύγος (*user\_id, item\_id*), για αυτό και δημιουργήσαμε το παραπάνω index.

## Κράτηση

```
CREATE INDEX idx_reservation ON reservation(user_id, ISBN);
```

Τα reservations σε γενικές γραμμές έχουν παρόμοια queries με τα borrows, για αυτό και δημιουργήσαμε πρακτικά τον ίδιο index για αυτά.

## Αξιολόγηση

```
CREATE INDEX idx_review ON review(ISBN, user_id);
```

Τα συνηθέστερα queries στα reviews αφορούν γίνονται είτε με κριτήριο το ISBN (π.χ. στο */book*) είτε με κριτήριο το ζεύγος (ISBN, *user\_id*). Για αυτό και δημιουργήσαμε τον παραπάνω index.

### 1.3.4 Triggers

Παρακάτω παρατίθεται το περιεχόμενο του αρχείου *triggers.sql*.

```
DELIMITER $$
CREATE TRIGGER delete_reserved_book AFTER INSERT ON borrow
FOR EACH ROW
BEGIN
    DELETE FROM reservation
    WHERE user_id = NEW.user_id AND ISBN = (SELECT ISBN FROM item WHERE item_id = NEW.item_id);
END $$
DELIMITER ;
```

Γενικά είναι καλή πρακτική τα triggers να μην είναι πολλά σε αριθμό, καθώς εκτελούνται αρκετά συχνά, με αποτέλεσμα να καθυστερούν την εκτέλεση των queries. Για τον σκοπό αυτό δημιουργήσαμε μόνο ένα trigger, το οποίο, όπως φαίνεται και παραπάνω, διαγράφει οποιεσδήποτε κρατήσεις έχουν γίνει για ένα βιβλίο από ένα member το οποίο δανείζεται το βιβλίο αυτό.

### 1.3.5 Events

Παρακάτω παρατίθεται το περιεχόμενο του αρχείου *events.sql*.

```
DELIMITER $$

CREATE EVENT delete_old_reservations
ON SCHEDULE EVERY 1 DAY
DO
    DELETE FROM reservation
    WHERE end_date < NOW()

$$

DELIMITER ;
```

Το μοναδικό event που προσθέσαμε εξασφαλίζει ότι οι κρατήσεις που έχουν λήξει διαγράφονται.

### 1.3.6 Views

Παρακάτω παρατίθεται το περιεχόμενο του αρχείου *views.sql*.

```
CREATE VIEW school_book AS
SELECT b.ISBN, s.school_id, COUNT(i.ISBN) as total_copies, COUNT(i.ISBN)-COUNT(br.borrow_id) as
    available_copies
FROM (SELECT ISBN FROM book) b
CROSS JOIN (SELECT school_id FROM school) s
LEFT JOIN item i ON i.ISBN = b.ISBN AND s.school_id = i.school_id
LEFT JOIN (SELECT * FROM borrow WHERE return_date IS NULL) br ON br.item_id = i.item_id
GROUP BY b.ISBN, s.school_id;
```

Όπως αναφέραμε και στον σχολιασμό του ER, η εναλλακτική στη δημιουργία του table item ήταν να υπάρχει μία σχέση *school\_book* στην οποία να φαίνεται πόσα αντίτυπα ενός βιβλίου υπάρχουν σε ένα σχολείο. Δεδομένου όμως ότι στη σελίδα προβολής των βιβλίων θέλαμε να υπάρχει δυνατότητα οι managers να μπορούν να δουν τον αριθμό αντιτύπων στο σχολείο τους (καθώς και τον αριθμό των αντιτύπων που είναι διαθέσιμα και δεν έχουν δανειστεί σε members), κρίναμε σκόπιμο να προσθέσουμε τη σχέση *school\_book*, ώστε να απλοποιήσουμε τα σχετικά queries (τα οποία απαιτούσαν περίπλοκα joins μεταξύ tables).

## 1.4 DML script/Insertions

Παρακάτω παρατίθεται το DML script (*insertions.sql*), το οποίο χρησιμοποιήσαμε για να γεμίσουμε με δεδομένα τη βάση μας. Αξίζει να σημειωθεί ότι, δεδομένου του μεγάλου αριθμού από entries που προσθέσαμε, παρακάτω παραθέτουμε ενδεικτικά μόνο τις πρώτες 3 rows από κάθε insertion. Άλλωστε, όλες οι υπόλοιπες έχουν την ίδια δομή.

```
INSERT INTO school(school_name, school_address, city, phone, email, principal_first_name,
principal_last_name) VALUES
('Model Lyceum Anavryta', 'Kifisias Av. 184', 'Athens', '2108012944', ' mail@lyk-peir-anavr.att.sch.gr',
'Theodoros', 'Otzakoglou'),
('1st Primary School of Gerakas', 'Gerakas AV. 28', 'Athens', '2106616000', '
mail@1-dim-gerakas.att.sch.gr', 'Elias', 'Dimos'),
('Athens College', 'Stefanou Delta 15', 'Psychiko', '2106798100', 'generalcoordinator@athenscollege.edu.gr',
'Kostas', 'Synolakis');
```

```
INSERT INTO book(ISBN, title, publisher, pages, summary, cover, book_language, keyword) VALUES

('9781472950249', 'Harry Potter and the Philosophers Stone', 'Bloomsbury Publishing', 223, 'The first book
in the Harry Potter series', 'https://m.media-amazon.com/images/I/91eopoUCjLL._AC_UL400_.jpg\n',
'English', 'Harry Potter, wizards, magic'),
('9780747574484', 'Harry Potter and the Chamber of Secrets', 'Bloomsbury Publishing', 251, 'The second book
in the Harry Potter series', 'https://m.media-amazon.com/images/I/81dDwGo2jmL._AC_UY218_.jpg\n',
'English', 'Harry Potter, wizards, magic'),
('9780747581086', 'Harry Potter and the Prisoner of Azkaban', 'Bloomsbury Publishing', 317, 'The third book
in the Harry Potter series', 'https://m.media-amazon.com/images/I/91II+F1H7LL._AC_UY218_.jpg\n',
'English', 'Harry Potter, wizards, magic');
```

```
INSERT INTO item(school_id, ISBN) VALUES
(2, '9780131988422'),
(2, '9780131988422'),
(2, '9780131988422');
```

```
INSERT INTO author (first_name, last_name) VALUES
('J.K.', 'Rowling'),
('William', 'Shakespeare'),
('Jane', 'Austen');
```

```
INSERT INTO book_author(ISBN, author_id) VALUES
('9780571241705', 13),
('9780571283590', 13),
('9780571299461', 13);
```

```
INSERT INTO category(category) VALUES
('Fantasy'),
('Adventure'),
('Romance');
```

```
INSERT INTO book_category(ISBN, category_id) VALUES
('9780486295557', 11),
('9789601603133', 5),
('9780141439518', 17);
```

```

INSERT INTO user(email, username, user_password, user_type, first_name, last_name, date_of_birth, school_id)
VALUES
('spyrosgal@hotmail.gr', 'spyrosgal', 'asdf', 'ADMIN', 'Spiros', 'Galanopoulos', '2002-10-24', NULL),
('k.stanley@anatolia.edu.gr', 'k.stanley', 'password123', 'MANAGER', 'Karen', 'Stanley', '1980-07-15', 5),
('p.gibson@anatolia.edu.gr', 'p.gibson', 'password123', 'MANAGER', 'Patrick', 'Gibson', '1985-02-23', 5);

INSERT INTO borrow (user_id, manager_id, item_id, begin_date, end_date, return_date) VALUES
(6, 2, 43, '2023-01-01', '2023-01-08', '2023-01-05'),
(8, 2, 128, '2023-02-01', '2023-02-08', '2023-02-05'),
(7, 2, 198, '2023-03-01', '2023-03-08', '2023-03-05');

INSERT INTO reservation (user_id, ISBN, begin_date, end_date) VALUES
(6, '9782253098604', '2023-5-29', '2023-6-6'),
(7, '9780262033848', '2023-5-28', '2023-6-5'),
(8, '9780131873742', '2023-5-27', '2023-6-4');

INSERT INTO review (user_id, ISBN, rating, review_body, approved) VALUES
(6, '9780961408828', 2, 'Not my cup of tea', 1),
(7, '9780262033848', 3, 'Was fine', 1),
(8, '9780156947695', 4, 'Was great', 1);

```

## 1.5 Queries

Στην εκφώνηση μάς ζητήθηκε συνολικά η υλοποίηση 12 queries. Αυτά έχουν υλοποιηθεί σε συγκεκριμένα σημεία του αρχείου [app.py](#), ώστε να μπορούν οι χρήστες να δουν τα αποτελέσματά τους μέσω της εφαρμογής. Δεδομένης της πολύπλοκης φύσης ορισμένων από αυτά (π.χ. σε κάποια χρησιμοποιούνται διαφορετικά φίλτρα αναζήτησης ανάλογα με τις επιλογές του χρήστη), αποφύγαμε να τα παραθέσουμε εδώ. Εντούτοις, μπορούν να βρεθούν εύκολα στον κώδικα του [app.py](#), αφού πριν από καθένα προηγείται σχόλιο με τον αριθμό του (π.χ. 3.1\_1, 3.2\_2 κ.ο.κ).

## Μέρος 2

### Οδηγός εγκατάστασης

#### 2.1 Βήμα 1

Αρχικά, κάνουμε clone το [git repo](https://github.com/harris28/myLibs) της εφαρμογής μας σε directory της επιλογής μας. Μέσω terminal μπορούμε απλά να γράψουμε: `git clone https://github.com/harris28/myLibs.git`

#### 2.2 Βήμα 2

Στη συνέχεια, θα πρέπει ο υπολογιστής μας να έχει έναν SQL server. Συγκεκριμένα εμείς χρησιμοποιούμε το software stack LAMP (**L**inux, **A**pache, **M**ySQL, **P**HP/**P**ython) και MariaDB ([εδώ](#) μπορείτε να βρείτε έναν πολύ καλό οδηγό για την εγκατάσταση του LAMP) ενώ για Windows υπάρχει το [XAMP](#). Αφού ολοκληρώσουμε την εγκατάσταση του, συνδεόμαστε σε αυτόν π.χ. με την εντολή: `mysql -u root -p`, πληκτρολογούμε τον κωδικό μας και ύστερα τρέχοντας την εντολή: `source ./db/db-create.sql`; στήνουμε την βάση μας (schema και data).

#### 2.3 Βήμα 3

Στο σημείο αυτό, αφού κάνουμε `cd` στο directory όπου κατεβάσαμε το [git repo](#), πρέπει να κατεβάσουμε τις απαραίτητες βιβλιοθήκες της εφαρμογής μας (προτείνεται αυτό να γίνει σε ένα virtual enviroment για να αποφευχθούν πιθανά conflicts με τις ήδη υπάρχουσες βιβλιοθήκες του συστήματός μας) γράφοντας στο command line της MariaDB: `pip install -r requirements.txt`

#### 2.4 Βήμα 4

Τέλος, είμαστε έτοιμοι να τρέξουμε την εφαρμογή μας μέσω του αρχείου `run.py` πληκτρολογώντας την εντολή: `python3 run.py` στο terminal. Αν χρησιμοποιούμε Unix-like OS μπορούμε να χρησιμοποιήσουμε και την εντολή `./run.sh` (πιθανότατα θα χρειαστεί προγουμένως να δώσουμε execution permission στο script με την εντολή `chmod +x ./run.sh`). Ανοίγουμε ύστερα έναν browser, περιηγούμεστε στην διεύθυνση [127.0.0.1:5000](http://127.0.0.1:5000) όπου το login page θα πρέπει να μας εμφανίζεται. Η εφαρμογή μας είναι έτοιμη για χρήση!

## Μέρος 3

### User manual

#### 3.1 Login/Registration

Ο χρήστης με την είσοδο του στην εφαρμογή (<http://127.0.0.1:5000/login>) συναντά την παρακάτω φόρμα εισόδου:

#### Welcome to myLibs

Please enter your credentials

Username

Password

Log in

Don't have an account? Register [here](#)

Figure 3.1: Login Form

Σε περίπτωση που είναι εγγεγραμμένος στο σύστημα myLibs μπορεί να εισάγει το Username, Password και να κάνει login . Διαφορετικά πατώντας το "here" μπορεί να οδηγηθεί στην σελίδα εγγραφής (<http://127.0.0.1:5000/register>) όπου συναντά την παρακάτω φόρμα.

#### Welcome to myLibs

Register Below

Username

Password

Email

First Name

Last Name

Date of Birth



School

Usertype

Register

Figure 3.2: Registration Form

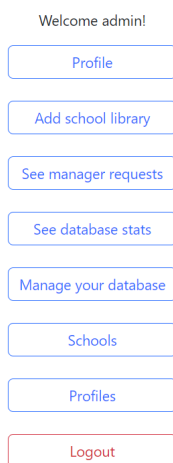
Ο υποψήφιος χρήστης καλείται να εισαγάγει μη χρησιμοποιούμενο Username, Password, έγκυρο και μη χρησιμοποιούμενο



email, μικρό όνομα, επίθετο, ημερομηνία γέννησης στα αντίστοιχα πεδία. Παράλληλα, καλείται να επιλέξει το σχολείο στο οποίο δραστηριοποιείται και το οποίο οφείλει να είναι ενεργό στο σύστημα myLibs, καθώς και τον τύπο του χρήστη που του αντιστοιχεί (Μαθητής-Student, Καθηγητής-Teacher, Διαχειριστής-Manager). Έπειτα από την υποβολή των παραπάνω με το κουμπί Register, τα στοιχεία του αποστέλλονται στον αντίστοιχο Admin (αν πρόκειται για υποψήφιο μέλος με ιδιότητα Manager) ή Manager (αν πρόκειται για υποψήφιο μέλος Καθηγητή/Μαθητή) για έγκριση και αυτοματοποιημένη ενεργοποίηση του λογαριασμού του (δείτε ενότητες Manager Requests, Member Requests για την διαδικασία έγκρισης). Ο ίδιος αν και εφόσον εγκριθεί θα ενημερωθεί με email.

## 3.2 Admin

Ο admin κάνοντας login οδηγείται στην αρχική σελίδα <http://127.0.0.1:5000/index>. Μπορεί να κάνει logout πατώντας στο "logout".



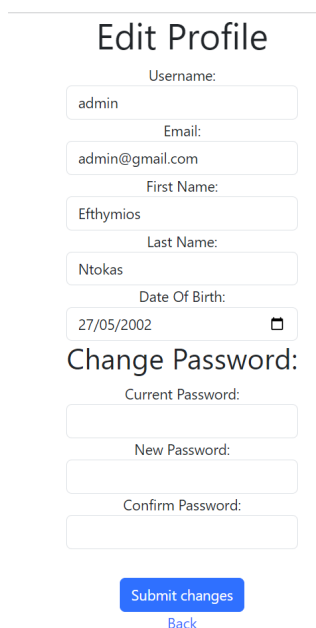
Welcome admin!

- Profile
- Add school library
- See manager requests
- See database stats
- Manage your database
- Schools
- Profiles
- Logout

Figure 3.3: Admin Start Page

### 3.2.1 Profile

Ο admin μεταβαίνοντας στο <http://127.0.0.1:5000/profile> με κλικ στο "Profile" της αρχικής βλέπει και επεξεργάζεται τα στοιχεία του στο σύστημα myLibs σύμφωνα με τις προδιαγραφές της βάσης. Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.



### Edit Profile

Username:

Email:

First Name:

Last Name:

Date Of Birth:

### Change Password:

Current Password:

New Password:

Confirm Password:

[Back](#)

Figure 3.4: Admin Profile

### 3.2.2 Add School Library

Ο admin μεταβαίνοντας στο [http://127.0.0.1:5000/add\\_school\\_library](http://127.0.0.1:5000/add_school_library) με κλικ στο "Add school library" καλείται να εισαγάγει τα στοιχεία του σχολείου που θέλει να προσθέσει στο σύστημα και να πατήσει "Register". Μπορεί να επιστρέψει στην αρχική σελίδα με το "Back to index".

### Add a School Library

School Name

School Address

City

Phone

Email

Principal First Name

Principal Last Name

Register

[Back](#)

Figure 3.5: Add School Library

### 3.2.3 Manager Requests

Ο admin μεταβαίνοντας στο <http://127.0.0.1:5000/requests> με κλικ στο "See manager requests" της αρχικής βλέπει τις διάφορες αιτήσεις εγγραφής από διαχειριστές σχολικών βιβλιοθηκών (managers). Για να εγκρίνει ή απορρίψει κάποια από αυτές πατάει το αντίστοιχο κουμπί Accept/Reject. Έπειτα από αποδοχή αποστέλλεται αυτόματα email ενεργοποίησης στον αιτούντα. Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.

See pending manager requests					
Username	Email	Name	Date of Birth	School Name	Manage
Rosalia	rosa@gmail.com	Rosalia Papadakis	1984-06-07	Athens College	<div>AcceptReject</div>

[Back](#)

Figure 3.6: Manager Requests

### 3.2.4 Database Stats

Ο admin μεταβαίνοντας στο <http://127.0.0.1:5000/stats> με κλικ στο "See Database Stats" έχει την δυνατότητα να συλλέξει διάφορα στατιστικά για την τωρινή κατάσταση της βάσης του. Μπορεί να επιστρέψει στην αρχική σελίδα με το "Back".

## Database Stats

Total borrows per school and per month/year

Book category authors and teacher borrows

Managers with same number of borrows

Miscellaneous stats

Back

Figure 3.7: Database Stats

### Borrows for each school per month or year

Όταν ο admin κάνει κλικ στο "Total borrows per school and per month/year" οδηγείται στο [http://127.0.0.1:5000/query\\_3.1.1](http://127.0.0.1:5000/query_3.1.1). Εκεί μπορεί να επιλέξει έτος και μήνα/ολόκληρο το έτος και να πατήσει "search" για να δει τα εν λόγω στατιστικά. Με το "back" επιστρέφει πίσω.

Total borrows per school and per month/year

Select Year:

2020

Select Month:

Whole Year

Search

School	Total Borrowed
--------	----------------

[Back](#)

Figure 3.8: Borrows for each school

### Book category authors and teacher borrows for that category in the last year

Όταν ο admin κάνει κλικ στο "Book category author and teacher borrows" οδηγείται στο [http://127.0.0.1:5000/query\\_3.1.2](http://127.0.0.1:5000/query_3.1.2). Εκεί μπορεί να επιλέξει κατηγορία και να πατήσει "search" για να δει του συγγραφείς της κατηγορίας καθώς και ποιοι δάσκαλοι δανείστηκαν βιβλία αυτής της κατηγορίας το τελευταίο έτος. Με το "back" επιστρέφει πίσω.

### Book category authors and teacher borrows for that category in the last year

Select Category:

Adventure

Search

Author
--------

Teacher
---------

[Back](#)

Figure 3.9: Book category authors - teacher borrows

### Manager with same number of borrows in a year (with more than 20 borrows)

Όταν ο admin κάνει κλικ στο "Managers with same number of borrows" οδηγείται στο [http://127.0.0.1:5000/query\\_3.1.5](http://127.0.0.1:5000/query_3.1.5).

Εκεί μπορεί να επιλέξει έτος και μήνα/ολόκληρο το έτος και να πατήσει "search" για να δει τα εν λόγω στατιστικά. Με το "back" επιστρέφει πίσω.

Manager with same number of borrows in a year (with more than 20 borrows)

Select Year:

2023

Search

Manager Names	Total Borrows
---------------	---------------

[Back](#)

Figure 3.10: Book category authors - teacher borrows

### Miscellaneous stats

Όταν ο admin κάνει κλικ στο "Miscellaneous stats" οδηγείται στο [http://127.0.0.1:5000/button\\_queries](http://127.0.0.1:5000/button_queries). Εκεί πατώντας στο επιθυμητό κύκλο και πατώντας "Show Results" μπορεί να δει αποτελέσματα για το επιλεγμένο query. Με το "back" επιστρέφει πίσω.

Miscellaneous stats

Find young teachers (age < 40 years) who have borrowed the most books and the number of books ☐

Find authors whose books have not been borrowed ☐

Find top 3 category pairs in borrowed books ☐

Find all authors who have written at least 5 books less than the author with the most books ☐

Show results

No query selected

[Back](#)

Figure 3.11: Miscellaneous Stats

### 3.2.5 BackUp/Restore

Ο admin μεταβαίνοντας στο [http://127.0.0.1:5000/manage\\_db](http://127.0.0.1:5000/manage_db) με κλικ στο "Manage Db" έχει την δυνατότητα να κατεβάσει αρχείο .sql ως backup της βάσης πατώντας στο "Backup database". Συγχρόνως, μπορεί να επαναφέρει (restore) την βάση του σε μια προηγούμενη κατάσταση πατώντας "Επιλογή αρχείου", διαλέγοντας το αρχείο backup που επιθυμεί και κάνοντας κλικ στο "Restore". Μπορεί να επιστρέψει στην αρχική σελίδα με το "Back".

Backup/Restore your database

Back

Backup database

Restore database

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο.

Figure 3.12: Backup/Restore

### 3.2.6 Schools

Ο admin μεταβαίνοντας στο <http://127.0.0.1:5000/schools> με κλικ στο "Schools" της αρχικής βλέπει τα στοιχεία των σχολείων που είναι εγγεγραμμένα στο σύστημα myLibs. Στο πάνω μέρος της σελίδας μπορεί να αναζητήσει κάποιο σχολείο

με βάση το όνομα. Κάνοντας κλικ στο κουμπί "Edit School", που βρίσκεται κάτω από το πλαίσιο του σχολείου στο οποίο αναφέρεται, οδηγείται στην σελίδα <http://127.0.0.1:5000/school/number> όπου number το αναγνωριστικό του σχολείου (δεν αφορά τον χρήστη της εφαρμογής).

### Browse Schools

Back

#### Filters

School Name

Apply filters

Name	Model Lyceum Anavryta
Address	Kifisias Av. 184
City	Athens
Phone	2108012944
Email	mail@lyk-peir-anavr.att.sch.gr
PrincipalFirstName	Theodoros
PrincipalLastName	Otzakoglou
ManagerNames	Jordon Boswood

Edit School

Figure 3.13: School Profiles

Εκεί μπορεί να επεξεργαστεί τα διάφορα στοιχεία του. Επιστρέφει πίσω κάνοντας κλικ στο "back" στο κάτω μέρος της σελίδας.

### Edit School

Name:

Model Lyceum Anavryta

Address:

Kifisias Av. 184

City:

Athens

Phone:

2108012944

Email:

mail@lyk-peir-anavr.att.sc

Principal First Name:

Theodoros

Principal Last Name:

Otzakoglou

Submit changes

Back

Figure 3.14: School Profile

### 3.2.7 Manager Profiles

Ο admin μεταβαίνοντας στο <http://127.0.0.1:5000/profiles> με κλικ στο "Profiles" της αρχικής βλέπει τα προφίλ διαχειριστών (manager) σχολικών βιβλιοθηκών του συστήματος myLibs. Στο πάνω μέρος της σελίδας μπορεί να αναζητήσει προφίλ με βάση το ονοματεπώνυμο και το username. Κάνοντας κλικ στο κουμπί "Edit Profile" που βρίσκεται κάτω από το πλαίσιο-προφίλ στο οποίο αναφέρεται οδηγείται στην σελίδα <http://127.0.0.1:5000/profileof/username> όπου username το username του χρήστη-προφίλ.

### Browse Profiles

Back

#### Filters

First Name

Last Name

Username

Apply filters

FirstName	Karen
LastName	Stanley
username	k.stanley
Active	YES
UserType	MANAGER

Edit Profile

Figure 3.15: Manager Profiles

Εκεί μπορεί να ενεργοποιήσει/απενεργοποιήσει τον λογαριασμό χρήστη. Επιστρέφει πίσω κάνοντας κλικ στο "back" στο κάτω μέρος της σελίδας.

### Edit profile

Username:

k.stanley

Email:

k.stanley@anatolia.edu.gr

First Name:

Karen

Last Name:

Stanley

Date Of Birth:

15/07/1980

Active:

Yes

Member type:

MANAGER

School:

Anatolia College

Submit changes

Back

Figure 3.16: Manager Profile

### 3.3 Manager

Ο manager κάνοντας login οδηγείται στην αρχική σελίδα <http://127.0.0.1:5000/index>.

Όλες οι διαδικασίες δανεισμού και κρατήσεων γίνονται στο "Books" έπειτα από επιλογή βιβλίου και κλικ του αντίστοιχου κουμπιού "See Details" όπως εξηγείται παρακάτω.

Μπορεί να κάνει logout πατώντας στο "logout".

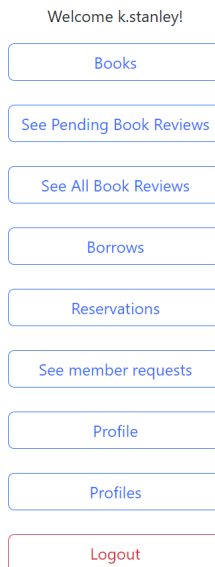


Figure 3.17: Manager Start Page

### 3.3.1 Book Browser

Ο manager μεταβαίνοντας στο <http://127.0.0.1:5000/books> με κλικ στο "Books" της αρχικής βλέπει στο μεσαίο τμήμα (2 στην φωτογραφία) μια λίστα όλων των καταχωρημένων βιβλίων. Στο αριστερό τμήμα (1 στην φωτογραφία) φίλτρα κατηγορίας, συγγραφέα, τίτλου, αριθμού αντιτύπων (2 πεδία ένα για ελάχιστο και μέγιστο) και διαθεσιμότητας στο σχολείο. Η επιβολή του γίνεται με το "apply filters".

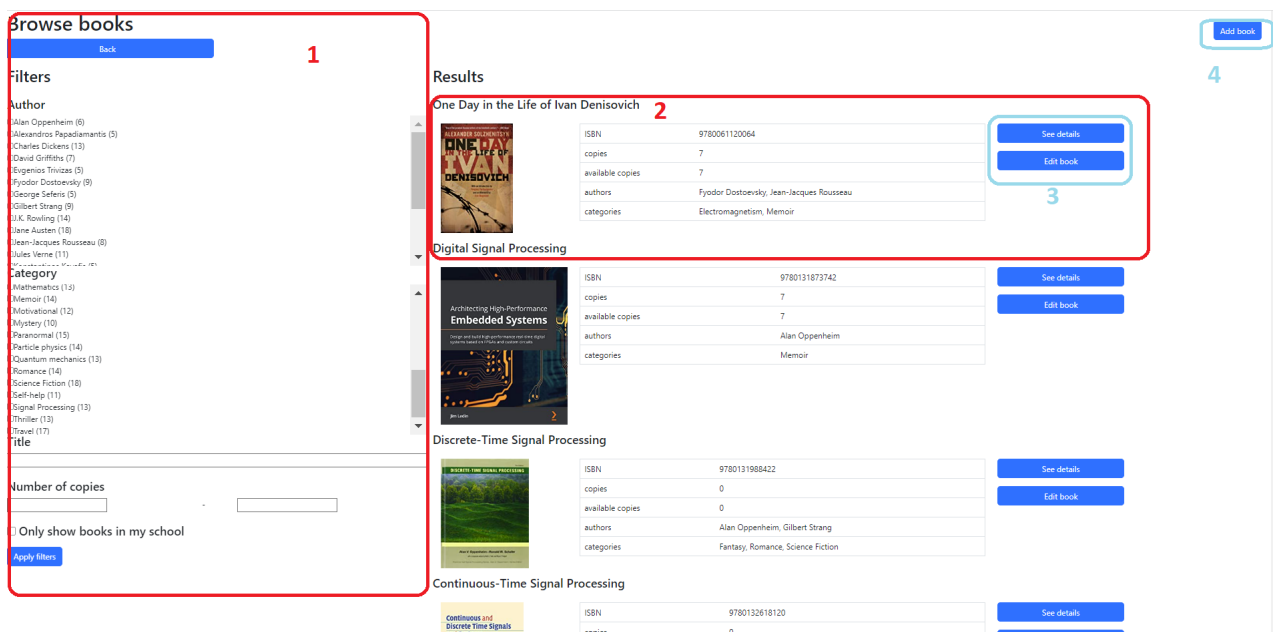


Figure 3.18: Book Browser

Στο πλαίσιο του κάθε βιβλίου υπάρχουν τα κουμπιά "See Details" και "Edit Book" (3 στην φωτογραφία). Συγχρόνως, με το κουμπί "Add Book" στο πάνω δεξιά μέρος της σελίδας (4 στην φωτογραφία) δίνεται η επιλογή προσθήκης βιβλίου. Οι τρεις αυτές λειτουργίες περιγράφονται παρακάτω.

#### Κρατήσεις Δανεισμοί/See Details

Όταν ο manager κάνει κλικ στο "See Details" οδηγείται στο <http://127.0.0.1:5000/book/ISBN> όπου ISBN είναι το ISBN του βιβλίου που επέλεξε. Στο τμήμα 1 της σελίδας παρατίθενται τα στοιχεία και το εξώφυλλο του βιβλίου, καθώς και κουμπί προς το "Edit Book" για επεξεργασία τους. Με κλικ στο "Back to books" πηγαίνει πίσω στον book browser.

Στο τμήμα 2 στα αριστερά προβάλλονται μια μια οι κρατήσεις για το εν λόγω βιβλίο, ενώ στα δεξιά μπορεί ο manager να

δημιουργήσει κράτηση γράφοντας το username του μέλους της βιβλιοθήκης και πατώντας το "Grant Reserve" μετέπειτα. Η κράτηση εκτελείται εφόσον πληρούνται οι περιορισμοί του συστήματος myLibs.

Στο τμήμα 3 για κάθε item (αντίτυπο) του βιβλίου εμφανίζονται περιληπτικά τα στοιχεία δανεισμού αν αυτό είναι δανεισμένο. Αν δεν είναι δανεισμένο, υπάρχει δυνατότητα δανεισμού γράφοντας το username του μέλους στο πεδίο δεξιά και πατώντας "Grant borrow" στο πλαίσιο του item. Ο δανεισμός εκτελείται εφόσον πληρούνται οι περιορισμοί του συστήματος myLibs. Συγχρόνως, υπάρχει δυνατότητα διαγραφής αντιτύπου (item) με κλικ στο αντίστοιχο "Remove" και προσθήκης αντιτύπου με κλικ στο "add item"

Στο τμήμα 4 εμφανίζονται τα reviews για το συγκεκριμένο βιβλίο και ο μέσος όρος αξιολογήσεων τους.

The screenshot shows a web interface for a library system. It is divided into four main sections, each highlighted with a red box and a red number:

- 1. Book Details:** Displays the book cover for 'emma' by Jane Austen. It includes metadata: ISBN: 9780140434261, Publisher: Penguin Classics, Pages: 512, Summary: A novel about Emma Woodhouse and her misguided attempts at matchmaking, Language: English, Keywords: romance, Regency era, Authors: Jane Austen, and Categories: Memoir, Thriller, History, Humor, Childrens, Quantum mechanics. There are 'Edit book' and 'Back to books' buttons.
- 2. Reservations:** A table with columns 'username', 'Date reserved', and 'Reservation expires'. It includes a 'Grant reserve' button and a form to 'Grant reservation to member with username:'.
- 3. Items:** A table with columns 'Item id', 'Borrowed by', 'To be returned by', 'Grant borrow', and 'Remove item'. It lists five items (1022-1026) and includes 'Grant borrow' and 'Remove' buttons for each. To the right is an 'Add item' button and a form to 'Grant borrow to member with username:'.
- 4. Reviews:** Shows an 'Average: 1.0000' and a table with columns 'username' and 'rating'. It lists one review by 'apapadopoulos' with a rating of '1'.

Figure 3.19: See Details/Borrows/Reservations

## Edit Book

Όταν ο manager κάνει κλικ στο "Edit Book" οδηγείται στο [http://127.0.0.1:5000/edit\\_book/ISBN](http://127.0.0.1:5000/edit_book/ISBN) όπου ISBN είναι το ISBN του βιβλίου που επέλεξε. Στην σελίδα αυτή μπορεί να επεξεργαστεί τα στοιχεία του βιβλίου και να αλλάξει φωτογραφία εξωφύλλου. Έπειτα κάνοντας κλικ στο "Submit Changes" αποθηκεύονται οι αλλαγές. Με το "Back" πηγαίνει πίσω στον book browser.

The screenshot shows the 'Edit book' page with a 'Submit changes' button at the top right and a 'Back' button. The page is divided into two columns of form fields:

- Left Column:** Fields for ISBN (9780061120064), Title (One Day in the Life of Ivar), Publisher (Harper Perennial Modern), Pages (182), Language (Russian), Keywords (Classic Fiction, Historical F), and Summary (One Day in the Life of Ivan Denisovich brilliantly portrays a single day, any day, in the life of a single Russian soldier who was captured by the Germans in 1945 and who managed to escape...).
- Right Column:** Fields for Cover (with a file upload button and text 'Επιλογή αρχείου | Δεν επιλέχθηκε κανένα αρχείο.'), Copies in your school (7), Authors (Jean-Jacques Rousseau and Fyodor Dostoevsky, with 'Remove author' buttons and an 'Add author' button), and Categories (Memoir and Electromagnetism, with 'Remove category' buttons and an 'Add Category' button).

Figure 3.20: Edit Book

## Add Book



Όταν ο manager κάνει κλικ στο "Add Book" οδηγείται στο [http://127.0.0.1:5000/add\\_book](http://127.0.0.1:5000/add_book). Εκεί καλείται να προσθέσει τα στοιχεία και την φωτογραφία του νέου βιβλίου που θέλει να εισάγει στην βιβλιοθήκη και να πατήσει "Add Book". Μπορεί να επιστρέψει στην αρχική σελίδα με το "Back to index"

Figure 3.21: Add Book

### 3.3.2 Review Acceptance

Ο manager μεταβαίνοντας στο <http://127.0.0.1:5000/reviews> με κλικ στο "See Pending Book Reviews" της αρχικής βλέπει τα διάφορα reviews προς έγκριση από μαθητές. Για να εγκρίνει ή απορρίψει κάποιο από αυτά πατάει το αντίστοιχο κουμπί Accept/Reject. Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.

See pending member book reviews						
ISBN	Username	Name	Member type	Rating	Review Body	Accept
9782070443089	jjones	John Jones	STUDENT	2	Not my cup of tea	Accept Reject
9782253098574	swilson	Sophie Wilson	STUDENT	3	Was fine	Accept Reject
9780521865716	tjackson	Thomas Jackson	STUDENT	4	Was great	Accept Reject
9781472950249	anguyen	Alex Nguyen	STUDENT	5	Loved it	Accept Reject
9780141439662	cmiller	Catherine Miller	STUDENT	1	Hated it	Accept Reject
9789601632617	rharris	Ryan Harris	STUDENT	2	Not my cup of tea	Accept Reject
9780138147570	ktaylor	Katie Taylor	STUDENT	3	Was fine	Accept Reject
9780856463183	brodriguez	Ben Rodriguez	STUDENT	4	Was great	Accept Reject
9780571329076	nmartinez	Natalie Martinez	STUDENT	5	Loved it	Accept Reject
9780571299461	mparker	Matthew Parker	STUDENT	1	Hated it	Accept Reject

Figure 3.22: Pending Reviews

### 3.3.3 Book Reviews

Ο manager μεταβαίνοντας στο [http://127.0.0.1:5000/reviews\\_all](http://127.0.0.1:5000/reviews_all) με κλικ στο "See All Book Reviews" της αρχικής βλέπει στο κάτω μέρος της σελίδας μέσω όρο αξιολογήσεων και τις αντίστοιχες αξιολογήσεις από μέλη του σχολείου του που τον παρήγαγαν. Μπορεί να επιβάλει φίλτρα στο πάνω μέρος της σελίδας ως προς το username του χρήστη που έγραψε τα reviews, τις κατηγορίες και το αν τα reviews που εμφανίζονται και λαμβάνονται υπόψη είναι από χρήστες που έχουν δανειστεί τα βιβλία για τα οποία γράφουν αξιολόγηση.

[Back](#)

## Review Average

### Filters

Review Author (username):

Category

- ☐Adventure
- ☐Art
- ☐Childrens
- ☐Computer science
- ☐Contemporary
- ☐Cookbook
- ☐Development
- ☐Romance
- ☐Science Fiction
- ☐Self-help
- ☐Signal Processing
- ☐Thriller
- ☐Travel

Only show borrowed ☐

[Apply filters](#)

Average rating: 2.9565

username	johndoe
title	Introduction to Applied Mathematics
rating	2

Figure 3.23: Reviews

### 3.3.4 Borrows

Ο manager μεταβαίνοντας στο <http://127.0.0.1:5000/borrows> με κλικ στο "Borrows" της αρχικής βλέπει τους διάφορους δανεισμούς από μέλη της βιβλιοθήκης (μαθητές/εκπαιδευτικοί). Στους μη ολοκληρωμένους δανεισμούς (το μέλος έχει στην κατοχή του το βιβλίο) εμφανίζεται ένα κουμπί "return" δεξιά της εν λόγω εγγραφής. Αυτό το πατάει ο manager μόλις γίνει η επιστροφή του βιβλίου. Συγχρόνως με χρήση των φίλτρων στο πάνω μέρος της σελίδας μπορεί να αναζητήσει δανεισμούς με βάση το ονοματεπώνυμο και τις ημέρες καθυστέρησης. Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.

## Borrows in your school

### Filters

First name:

Last name:

☐Only show overdue

Days overdue:

 - 

[Apply filters](#)

ername	First name	Last name	Item id	Book title	ISBN	Date borrowed	To be returned by	Returned ?	Overdue ?
indoe	John	Doe	43	Digital Signal Processing	9780131873742	2023-01-01 00:00:00	2023-01-08 00:00:00	2023-01-05 00:00:00	No
indoe	John	Doe	43	Digital Signal Processing	9780131873742	2022-07-01 00:00:00	2022-07-08 00:00:00	2022-09-05 00:00:00	No
esmith	Jane	Smith	91	Signals, Systems, and Inference	9780262033848	2022-01-01 00:00:00	2022-01-08 00:00:00	2022-01-05 00:00:00	No

Figure 3.24: Borrows

### 3.3.5 Reservations

Ο manager μεταβαίνοντας στο <http://127.0.0.1:5000/reservations> με κλικ στο "Reservations" της αρχικής βλέπει τις διάφορες κρατήσεις από μέλη της βιβλιοθήκης (μαθητές/εκπαιδευτικοί). Συγχρόνως με χρήση των φίλτρων στο πάνω μέρος της σελίδας μπορεί να αναζητήσει κρατήσεις με βάση το ονοματεπώνυμο. Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.

Reservations in your school

Filters

First name:

Last name:

Apply filters

Username	First name	Last name	Book title	ISBN	Begin Date	End Date
johndoe	John	Doe	L'Â®le mystÃ©rieuse	9782253098604	2023-05-29 00:00:00	2023-06-06 00:00:00
janesmith	Jane	Smith	Signals, Systems, and Inference	9780262033848	2023-05-28 00:00:00	2023-06-05 00:00:00
robertjohnson	Robert	Johnson	Digital Signal Processing	9780131873742	2023-05-27 00:00:00	2023-06-04 00:00:00
karenjackson	Karen	Jackson	Michel Strogoff	9782253098611	2023-05-30 00:00:00	2023-06-07 00:00:00
peterwang	Peter	Wang	De la Terre Ã la Lune	9782253098581	2023-05-29 00:00:00	2023-06-06 00:00:00
emilynguyen	Emily	Nguyen	Between the Acts	9780156949606	2023-05-28 00:00:00	2023-06-05 00:00:00
jamesmith	James	Smith	Digital Signal Processing	9780131873742	2023-05-27 00:00:00	2023-06-04 00:00:00
susanbrown	Susan	Brown	Michel Strogoff	9782253098611	2023-05-30 00:00:00	2023-06-07 00:00:00

Figure 3.25: Reservations

### 3.3.6 Member Requests

O manager μεταβαίνοντας στο <http://127.0.0.1:5000/requests> με κλικ στο "See member requests" της αρχικής βλέπει τις διάφορες αιτήσεις εγγραφής από μαθητές ή εκπαιδευτικούς του σχολείου του. Για να εγκρίνει ή απορρίψει κάποια από αυτές πατάει το αντίστοιχο κουμπί Accept/Reject. Έπειτα από αποδοχή εμφανίζεται προς εκτύπωση η κάρτα μέλους και αποστέλλεται αυτόματα email ενεργοποίησης στον αιτούντα. Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.

See pending member requests					
Username	Email	Name	Date of Birth	User Type	Manage
JohnK	johnk@gmail.com	Johnathan Kyle	2000-02-10	STUDENT	<input type="button" value="Accept"/> <input type="button" value="Reject"/>

[Back](#)

Figure 3.26: Member Requests

### Member Card

**First name:** Johnathan  
**Last name:** Kyle  
**E-mail:** johnk@gmail.com  
**Date of birth:** 2000-02-10  
**School:** Anatolia College  
**Member type:** STUDENT

Figure 3.27: Member Card

### 3.3.7 Profile

O manager μεταβαίνοντας στο <http://127.0.0.1:5000/profile> με κλικ στο "Profile" της αρχικής βλέπει και επεξεργάζεται τα στοιχεία του στο σύστημα myLibs σύμφωνα με τις προδιαγραφές της βάσης. Επιστροφή στην αρχική γίνεται με το "back"

στο κάτω μέρος της σελίδας.

### Edit Profile

Username:

Email:

First Name:

Last Name:

Date Of Birth:

School:

### Change Password:

Current Password:

New Password:

Confirm Password:

[Back](#)

Figure 3.28: Manager Profile

### 3.3.8 Member Profiles

Ο manager μεταβαίνοντας στο <http://127.0.0.1:5000/profiles> με κλικ στο "Profiles" της αρχικής βλέπει τα προφίλ μελών της βιβλιοθήκης του σχολείου του. Στο πάνω μέρος της σελίδας μπορεί να αναζητήσει προφίλ με βάση το ονοματεπώνυμο και το username. Κάνοντας κλικ στο κουμπί "Edit Profile", που βρίσκεται κάτω από το πλαίσιο-προφίλ στο οποίο αναφέρεται, οδηγείται στην σελίδα <http://127.0.0.1:5000/profileof/username> όπου username το username του χρήστη-προφίλ.

### Browse Profiles

Filters

First Name

Last Name

Username

FirstName	John
LastName	Doe
username	johnndoe
Active	YES
UserType	TEACHER

FirstName	Jane
LastName	Smith
username	janessmith
Active	YES
UserType	TEACHER

Figure 3.29: Member Profiles

Εκεί μπορεί να επεξεργαστεί τα διάφορα στοιχεία του και να απενεργοποιήσει/ενεργοποιήσει τον λογαριασμό χρήστη. Συγχρόνως με το κουμπί "Delete User" του δίνεται η δυνατότητα διαγραφής user εφόσον αυτός δεν έχει κρατήσεις, δανεισμούς ή αξιολογήσεις βιβλίων. Επιστρέφει πίσω κάνοντας κλικ στο "back" στο κάτω μέρος της σελίδας.


### Edit profile

Username:  
johndoe

Email:  
john.doe@anatolia.edu.gr

First Name:  
John

Last Name:  
Doe

Date Of Birth:  
01/01/1980 

Active:  
Yes

Membertype:  
TEACHER

School:  
Anatolia College

[Submit changes](#) [Delete user](#)

[Back](#)

Figure 3.30: Member Profile

## 3.4 Member-Teacher/Student

Το μέλος της βιβλιοθήκης κάνοντας login οδηγείται στην αρχική σελίδα <http://127.0.0.1:5000/index>.

Όλες οι διαδικασίες κρατήσεων γίνονται στο "Books" έπειτα από επιλογή βιβλίου και κλικ του αντίστοιχου κουμπιού "Re-serve" όπως εξηγείται παρακάτω.

Μπορεί να κάνει logout πατώντας στο "logout".

Welcome janesmith!

[Books](#)

[Borrows](#)

[Borrowed books](#)

[Reservations](#)

[Profile](#)

[Logout](#)

Figure 3.31: Member Index

### 3.4.1 Book Browser

Το μέλος μεταβαίνοντας στο <http://127.0.0.1:5000/books> με κλικ στο "Books" της αρχικής βλέπει στο μεσαίο τμήμα (2 στην φωτογραφία) μια λίστα όλων των καταχωρημένων βιβλίων. Στο αριστερό τμήμα (1 στην φωτογραφία) φίλτρα κατηγορίας, συγγραφέα, τίτλου, αριθμού αντιτύπων (2 πεδία ένα για ελάχιστο και μέγιστο) και διαθεσιμότητας στο σχολείο. Η επιβολή τους γίνεται με το "apply filters".

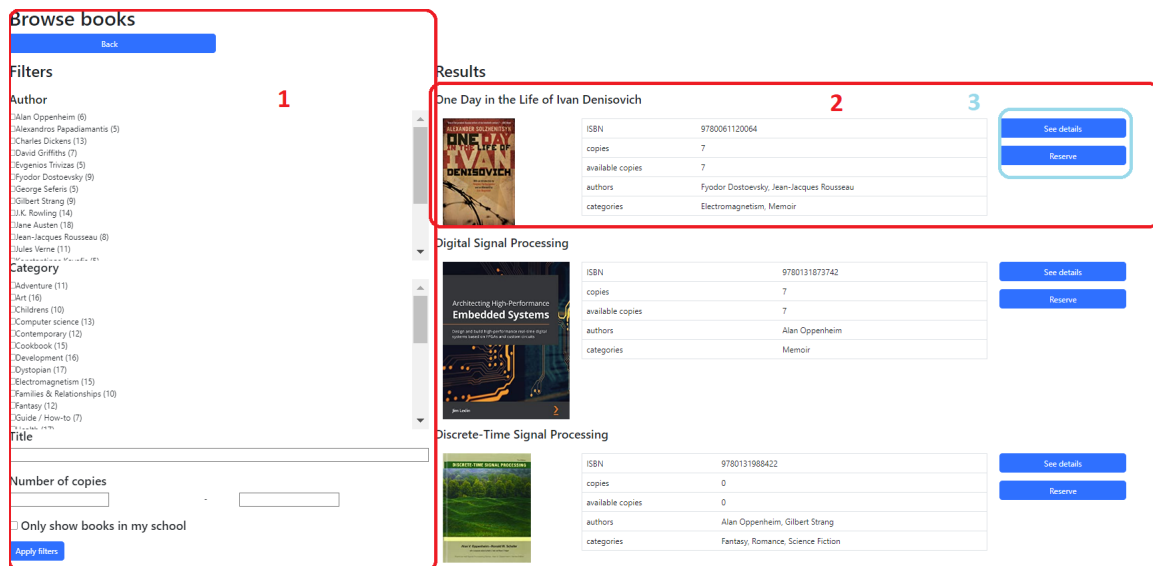
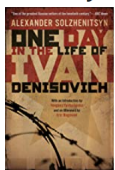


Figure 3.32: Book Browser

Στο πλαίσιο του κάθε βιβλίου υπάρχουν τα κουμπιά "See Details" και "Reserve" (3 στην φωτογραφία). Με κλικ στο "Reserve" εκτελείται η κράτηση του αντίστοιχου βιβλίου από το μέλος, εφόσον πληρούνται οι προϋποθέσεις της βιβλιοθήκης. Με κλικ στο "See Details" το μέλος μεταβαίνει στο <http://127.0.0.1:5000/book/ISBN> όπου ISBN είναι το ISBN του βιβλίου που επιλέχθηκε. Στα αριστερά βλέπει τις αξιολογήσεις για το βιβλίο αυτό. Συγχρόνως μπορεί να συνθέσει δικιά του. Αν είναι μαθητής θα αποσταλεί για έγκριση στον αρμόδιο διαχειριστή, διαφορετικά δημοσιεύεται άμεσα. Δεξιά εμφανίζονται τα στοιχεία του βιβλίου. Μπορεί με κλικ στο "reserve" να εκτελέσει κράτηση. Με κλικ στο "Back to books" επιστρέφει στο books.

### One Day in the Life of Ivan Denisovich



ISBN: 9780061120064

Publisher: Harper Perennial Modern Classics

Pages: 182

**Summary:** One Day in the Life of Ivan Denisovich brilliantly portrays a single day, any day, in the life of a single Russian soldier who was captured by the Germans in 1945 and who managed to escape...

Language: Russian

Keywords: Classic Fiction, Historical Fiction

Authors: Fyodor Dostoevsky, Jean-Jacques Rousseau

Categories: Electromagnetism, Memoir

Reserve

Back to books

### Write a review :)

Rating: 1

Review:

Write a review of max 500 chars

Submit Review

### Reviews

Average: None

Figure 3.33: Book Details

## 3.4.2 Borrows

Το μέλος μεταβαίνοντας στο <http://127.0.0.1:5000/borrows> με κλικ στο "Borrows" της αρχικής βλέπει τους δανεισμούς βιβλίων του. Συγχρόνως μπορεί να αναζητήσει δανεισμούς με βάση τις ημέρες καθυστέρησης.

## Your Borrows

### Filters

☐ Only show overdue

Days overdue:

Apply filters

Book title	ISBN	Date borrowed	To be returned by	Returned ?	Overdue ?
One Day in the Life of Ivan Denisovich	9780061120064	2023-06-03 19:19:31	2023-06-10 19:19:31	No	No

[Back](#)

Figure 3.34: Borrows

### 3.4.3 Borrowed Books

Το μέλος μεταβαίνοντας στο [http://127.0.0.1:5000/borrowed\\_books](http://127.0.0.1:5000/borrowed_books) με κλικ στο "Borrowed Books" βλέπει όλα τα βιβλία που έχει δανειστεί μέσω του συστήματος myLibs. Μπορεί να δει λεπτομέρειες για το καθένα από αυτά με κλικ στο αντίστοιχο "See Details". Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.

Books you have borrowed		
Book title	ISBN	See details
9780262033848	Signals, Systems, and Inference	<a href="#">See details</a>
9780387982588	Linear Algebra, Geodesy, and GPS	<a href="#">See details</a>

[Back](#)

Figure 3.35: Borrowed Books

### 3.4.4 Reservations

Το μέλος μεταβαίνοντας στο <http://127.0.0.1:5000/reservations> με κλικ στο "Reservations" της αρχικής βλέπει τις διάφορες ενεργές κρατήσεις βιβλίων του και μπορεί με το "cancel" να ακυρώσει κάποια από αυτές. Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.

### Your reservations

Book title	ISBN	Date reserved	Reservation expires	Available for borrow	
One Day in the Life of Ivan Denisovich	9780061120064	2023-06-03 18:54:08	2023-06-10 18:54:08	Yes	<a href="#">Cancel</a>
Digital Signal Processing	9780131873742	2023-06-03 18:54:43	2023-06-10 18:54:43	Yes	<a href="#">Cancel</a>

[Back](#)

Figure 3.36: Reservations

### 3.4.5 Profile

Το μέλος μεταβαίνοντας στο <http://127.0.0.1:5000/profile> με κλικ στο "Profile" της αρχικής βλέπει και επεξεργάζεται τα στοιχεία του στο σύστημα myLibs σύμφωνα με τις προδιαγραφές της βάσης. Οι μαθητές μπορούν να επεξεργαστούν μόνο τον κωδικό τους. Όλα τα μέλη έχουν πρόσβαση στην κάρτα μέλους πατώντας το "See member card". Επιστροφή στην αρχική γίνεται με το "back" στο κάτω μέρος της σελίδας.

### Edit Profile

Username:  
janesmithh

Email:  
jane.smith@anatolia.edu.gr

First Name:  
Jane

Last Name:  
Smithh

Date Of Birth:  
15/02/1985

Usertype:  
TEACHER

School:  
Anatolia College

### Change Password:

Current Password:

New Password:

Confirm Password:

[See member card](#) [Submit changes](#)

[Back](#)

Figure 3.37: Teacher Profile

### Edit Profile

Username:  
johnny

Email:  
johnny@gmail.com

First Name:  
johnny

Last Name:  
smith

Date Of Birth:  
2005-06-04

Usertype:  
STUDENT

School:  
Anatolia College

### Change Password:

Current Password:

New Password:

Confirm Password:

[See member card](#) [Submit changes](#)

[Back](#)

Figure 3.38: Student Profile