*Μάνος Πλατάκης 4730 Στέλιος Σφακιανάκης 4513*

Αναφορά Προτζεκτ ΗΥ-359

1. Οι κλάσεις που χρησιμοποιήσαμε ουσιαστικά ήταν οι ήδη υπάρχουσες (User, Student, Librarian, Borrowing, Book, BookInLibrary, JSONConverter, Review) και όλα τα databasetables σε κάποια από τα οποία κάναμε κάποιες αλλαγές/προσθήκες. Πιο συγκεκριμένα, στις EditStudentsTable, EditLibrarianTable, EditBooksTable, EditBooksInLibrary προσθέσαμε κάποιες συναρτήσεις όμοιες με τις ήδη υπάρχουσες που ουσιαστικά το μόνο που διαφέρει είναι οι παράμετροι σε μερικές περιπτώσεις η το τι επιστρέφουν (Σε μερικά προσθέσαμε και συναρτήσεις που επιστρέφουν όλους τους μαθητές/βιβλιοθηκάριους κλπ). Επίσης, προσθέσαμε και μια συνάρτηση delete στα ΕditStudentsTable και EditLibrarianTable, έτσι ώστε να διαγράφει ο admin τους users που επιθυμεί.
2. \* Χρησιμοποιήσαμε πολλά servlets μιας και δε κάναμε καθόλου χρήση του REST.

Πιο συγκεκριμένα, όσον αφορά το login, έχουμε 3 servlets: 1. Login (για τους students, κρατάει session) 2. LoginLib (για τους librarians, κρατάει session) 3. LoginAdmin (για τους admins, δε κρατάει session για λόγους ασφάλειας).

Για το register έχουμε 1 servlet όπως ακριβώς στην 3η Σ.Α.

Για την εμφάνιση όλων των βιβλίων έχουμε 2 servlets (Guest, SeeBooks (για επισκέπτες και μαθητές αντίστοιχα)) που παίρνουν όλα τα βιβλία από το database και τα εμφανίζουν το ένα κάτω από το άλλο, δημιουργώντας ταυτόχρονα ένα navigation bar έτσι ώστε να μπορούμε να δούμε και κάθε είδος ξεχωριστά (θα το εξηγήσω παρακάτω). Επιπρόσθετα, στους μαθητές έχουμε και 2 επιπλέον κουμπιά που δείχνουν σε ποιες βιβλιοθήκες υπάρχει διαθεσιμότητα του βιβλίου που διάλεξαν και το άλλο είναι για παραπάνω πληροφορίες σχετικά με το βιβλίο (ShowAvailability, ShowMore). Με το ShowAvailability, εμφανίζονται οι βιβλιοθήκες που έχουν διαθέσιμο το βιβλίο και δίπλα από την κάθε μια ένα κουμπί Borrow το οποίο όταν πατηθεί δημιουργεί νέο δανεισμό και το attribute status του βιβλίου σε requested.

Όσον αφορά τα βιβλία έχουμε δύο servlets για κάθε είδος βιβλίου(για τον επισκέπτη και τον μαθητή αντίστοιχα) (NovelG, NovelS, RomanceG, RomanceS κλπ), 10 στο σύνολο. Το κάθε ένα απ’ αυτά ξαναφτιάχνει το navigation bar και εμφανίζει όλα τα βιβλία του εκάστοτε είδους με τη μόνη διαφορά ότι αυτά που αφορούν τους μαθητές τοποθετούν και τα κουμπιά που αναφέραμε παραπάνω.

Έχουμε άλλα 2 servlets (GetInfo, GetInfoLib) για να εμφανίζονται οι πληροφορίες του εκάστοτε user (Student, Librarian αντίστοιχα). Πέρα από τις πληροφορίες εμφανίζεται και ένα κουμπί edit που κάνει enable 2 άλλα servlets (UpdateS, UpdateL) για την επεξεργασία στοιχείων του εκάστοτε χρήστη (Εμφάνιση φόρμας στην οποία όποια στοιχεία συμπληρωθούν θα γίνουν update ενώ όποια μείνουν κενά θα μείνουν ίδια).

Επίσης έχουμε ένα κουμπί See Borrowings που κάνει enable 2 servlets (SeeBorrowings, SeeBorrowingsLib για μαθητές και βιβλιοθηκάριους αντίστοιχα), τα οποία εμφανίζουν active και ended borrowings του μαθητή και όλα τα borrowings της βιβλιοθήκης. (δε δουλεύουν αλλά ο κώδικας είναι γραμμένος, είπαμε τον λόγο στην προφορική εξέταση).

Ένα τελευταίο κουμπί στην αρχική του Librarian κάνει enable το servlet AddBook το οποίο εμφανίζει μια φόρμα που πρέπει να συμπληρώσει ο Librarian για να προσθέσει ένα βιβλίο στο database.

Τέλος όσον αφορά στον admin, με το που γίνεται το login εμφανίζονται κάποια κουμπιά. Πιο συγκεκριμένα, το Show Libraries και Show Students, τα οποία κάνουν enable 2 servlets (SeeLibs, SeeStudents) που εμφανίζουν όλους του εκάστοτε είδος χρήστη με ένα κουμπί Delete δίπλα στον κάθε ένα. Το κουμπί αυτό κάνει enable το τελευταίο servlet (DelUser) το οποίο διαγράφει τον χρήστη.

\*JavaScript libraries και jsp δε χρησιμοποιήσαμε αλλά είχαμε 3 js files , ένα για να ελέγχει το registration(code.js), και άλλα 2 για τα ajax requests (ajax\_1.js(περισσότερο για τον guest), ajax.js).

Επίσης, έχω 10 html files (κάποια buttons, registration form, login forms for every type of user, edit data forms for every type of user, add book form και το βασικό html file index.html).

\* Έχουμε 1 CSS file (css1) το οποίο φτιάχνει ουσιαστικά το κύριο κομμάτι της σελίδας μας (τα 2 δίπλα το ένα στο άλλο πλαίσια: Το αριστερό για τα κουμπιά και τα choices και το δεξί για τη συμπλήρωση φορμών και για την εμφάνιση των βασικών και όχι μόνο λειτουργειών του προτζεκτ μας.) και φτιάχνει τα κουμπιά το navigation bar, τις εικόνες κ.α.

\*Ουσιαστικά, από την πλευρά του client, με το ajax (JavaScript) αλλάζουμε το html και τον βοηθάμε να επικοινωνήσει επιτυχώς με τον server και μέσω αυτού με τη βάση δεδομένων.

\*Τα ajax τμήματα της εφαρμογής μας συνδέονται με όλα τα buttons για να αλλάζουν το html file (δηλαδή το front-end) και επικοινωνούν με τον server (δηλαδή το back-end) από τον οποίο παίρνουν δεδομένα που επίσης θα αλλάξουν το html file. Άρα τα ajax βοηθάνε τόσο στο front-end όσο και στο back-end κομμάτι της ιστοσελίδας μας.

\*Δεν χρησιμοποιήσαμε πουθενά REST requests/APIs.

\*Όσον αφορά αλλαγές στη βάση δεδομένων, το μόνο που πήγαμε να αλλάξουμε είναι να προσθέσουμε ένα attribute στο table borrowings (isbn) έτσι ώστε να μπορέσουμε από την General Queries να πάρουμε ένα arraylist τύπου borrowing με όλους τους δανεισμούς. Δυστυχώς, όμως, το δοκιμάσαμε, κάναμε τις απαραίτητες αλλαγές στις συναρτήσεις που έπρεπε και πάλι δε δούλευε, δε περνούσε στο database. Γι’ αυτό το αλλάξαμε ξανά ως ήταν και απλά δε μας δουλεύει το servlet SeeBorrowings (παρόλο που ο κώδικας έχει λογική και συνοχή αν δούλευε το παραπάνω.  
Παρόλα αυτά δεν κάναμε κάποια αλλαγή στη βάση δεδομένων.