

Εργασία μαθήματος
«ΔΙΑΔΙΚΤΥΑΚΑ ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ»

Τεύχος σχεδιασμού της Εφαρμογής

Ομάδα **12**

Μακρής Στράτος 21736

Μπούσουλα Ιωάννα 21748

Ξένου Άννα 21751

Θέμα 2: Αιτήσεις φοιτητών σε μεταπτυχιακό

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	Error! Bookmark not defined.
Περιγραφή Συστήματος	3
Παραδοχές	4
Class Diagram	5
Use Case Diagram	6
Sequence Diagrams	7
Σχήμα με την συνολική αρχιτεκτονική	8
Component Διάγραμμα	10
Εγχειρίδιο χρήσης	11

Περιγραφή Συστήματος

Πρόκειται για ένα σύστημα αιτήσεων για μεταπτυχιακό. Οι εμπλεκόμενοι είναι:

- Ο φοιτητής
- Η γραμματεία
- Ο καθηγητής

Όλοι οι ρόλοι κάνουν login στο σύστημα με username και password (θα διαχωρίζονται στην βάση με το role_id) και μετά ο καθένας θα μπορεί να κάνει συγκεκριμένα πράγματα. Ο φοιτητής θα μπορεί να κάνει μόνο αίτηση για μεταπτυχιακό καθώς και να δει αν έγινε δεκτός ή όχι.

Η γραμματεία θα μπορεί να δει όλες τις αιτήσεις και αν πληρούν τα κριτήρια (δηλαδή αν έχει η κάθε αίτηση τα τρία δικαιολογητικά που χρειάζονται, προπτυχιακό πρόγραμμα σπουδών, βαθμός πτυχίου και 2 καθηγητές που έδωσαν συστατικές επιστολές στον φοιτητή) θα τις επιβεβαιώνει ή θα τις απορρίπτει.

Ο καθηγητής θα βλέπει μόνο τις επιβεβαιωμένες αιτήσεις και σύμφωνα με τον βαθμό πτυχίου θα τις βάζει σε σειρά κατάταξης.

Εκτός των προαναφερόμενων χρηστών υπάρχει και ένας διαχειριστής του συστήματος, ο οποίος δεν είναι κανένας από τους παραπάνω, και ο ρόλος του είναι να βάζει χρήστες στην βάση δεδομένων.

Επίσης ο administrator διαχειρίζεται έναν κατάλογο που διατηρεί πληροφορίες για τους ρόλους που ορίζονται. Σε αυτή εκτός από χρήστες/ρόλους, θα διατηρείται και κατάλογος όλων των υποστηριζόμενων υπηρεσιών (πχ αίτηση μεταπτυχιακού, επιβεβαίωση αίτησης κλπ), καθώς και των δικαιωμάτων εκτέλεσης που έχουν οι χρήστες για κάθε υπηρεσία ανάλογα με το ρόλο τους. Αυτή η υπηρεσία καταλόγου χρηστών είναι ανεξάρτητη και θα πρέπει να υποστηρίζει:

α) καταχώρηση/διαγραφή ρόλων

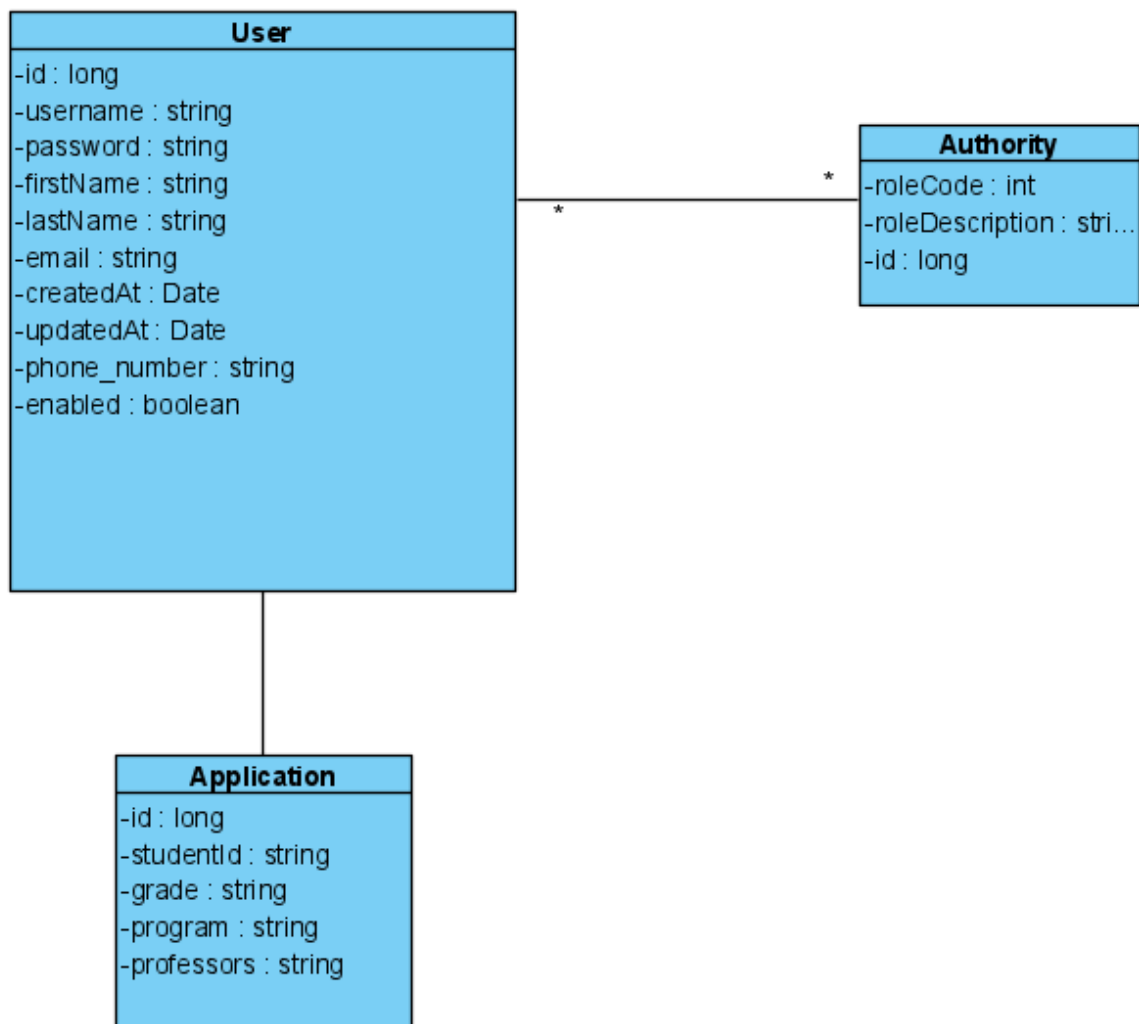
β) καταχώρηση/διαγραφή/ενημέρωση στοιχείων χρηστών και

γ) καταχώρηση/διαγραφή υποστηριζόμενων υπηρεσιών και ενημέρωση των δικαιωμάτων εκτέλεσης που έχουν οι ρόλοι σε αυτές.

Παραδοχές

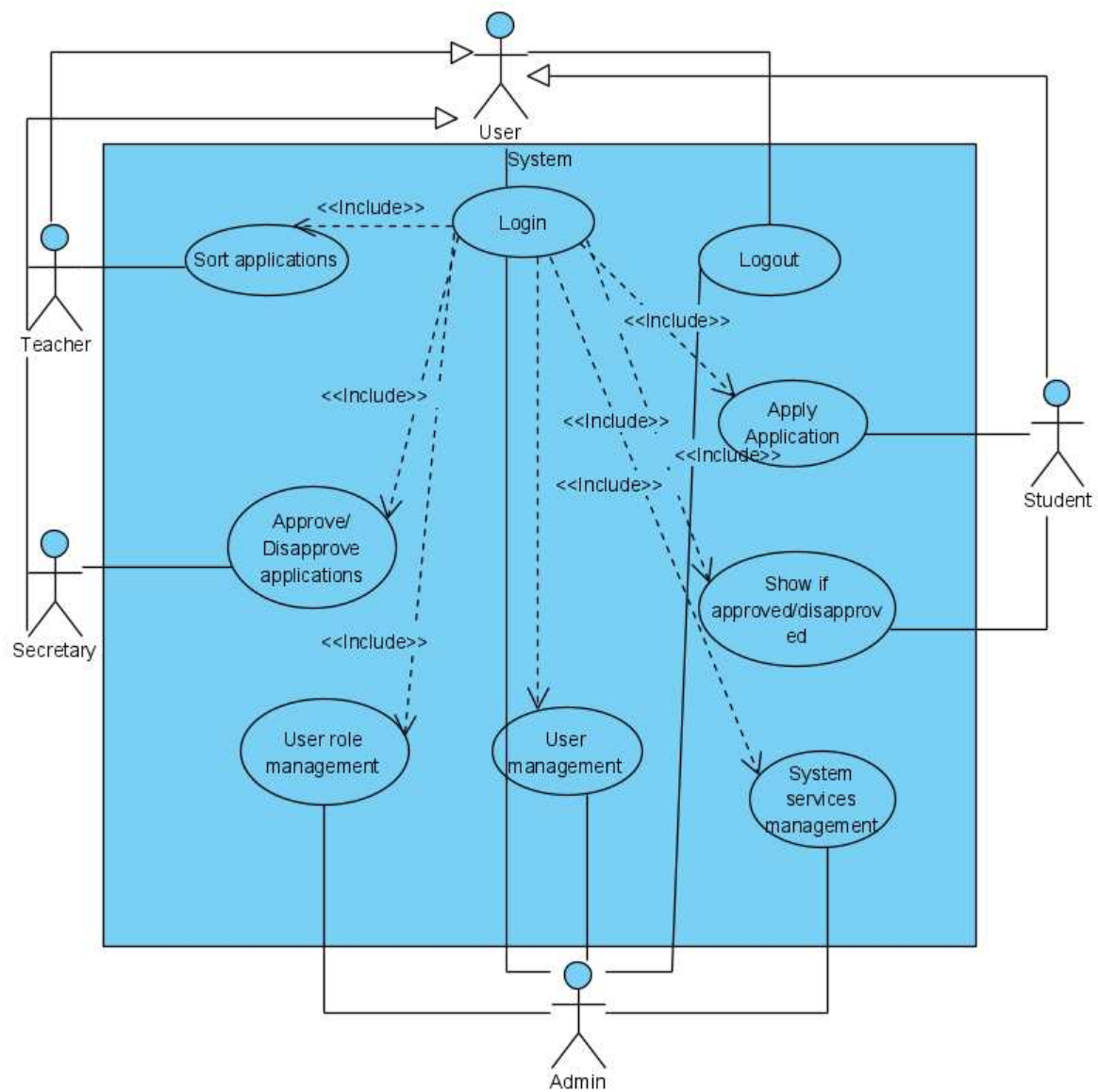
1. Μόνο μία αίτηση για μεταπτυχιακό μπορεί να κάνει ο φοιτητής κάθε φορά. Δηλαδή αν δεν γίνει αποδεκτός μόνο τότε μπορεί να ξανακάνει αίτηση για μεταπτυχιακό, όχι δύο αιτήσεις ταυτόχρονα.
2. Ο διαχειριστής δημιουργεί τους χρήστες στην βάση. Επίσης αυτός ανανεώνει τα στοιχεία τους, τους διαγράφει κλπ.
3. Ο φοιτητής ειδοποιείται ακόμα και αν δεν έγινε δεκτός.
4. Όλοι οι εμπλεκόμενοι χρήστες ακόμα και ο διαχειριστής πρέπει να κάνουν login στο σύστημα με το username και το password τους και μπορούν να κάνουν logout όποτε θελήσουν.
5. Η γραμματεία είτε επικυρώνει μια αίτηση βιογραφικού είτε την απορρίπτει.
6. Η γραμματεία δίπλα από κάθε αίτηση, στην βάση δεδομένων απλά θα βάζει σε μία στήλη αν είναι η αίτηση επιβεβαιωμένη (yes) αν όχι (no) σύμφωνα με το αν πληροί τα κριτήρια δηλαδή αν υπάρχουν τα τρία έγγραφα που χρειάζονται.
7. Ο κωδικός χρήστη θα έχει μια συγκεκριμένη μορφή ανάλογα τον ρόλο του χρήστη καθώς και το ID τους.

Class Diagram



Στο παραπάνω Class Diagram φαίνονται οι κλάσεις που θα υπάρξουν στο σύστημα, καθώς και οι συσχετίσεις μεταξύ τους.

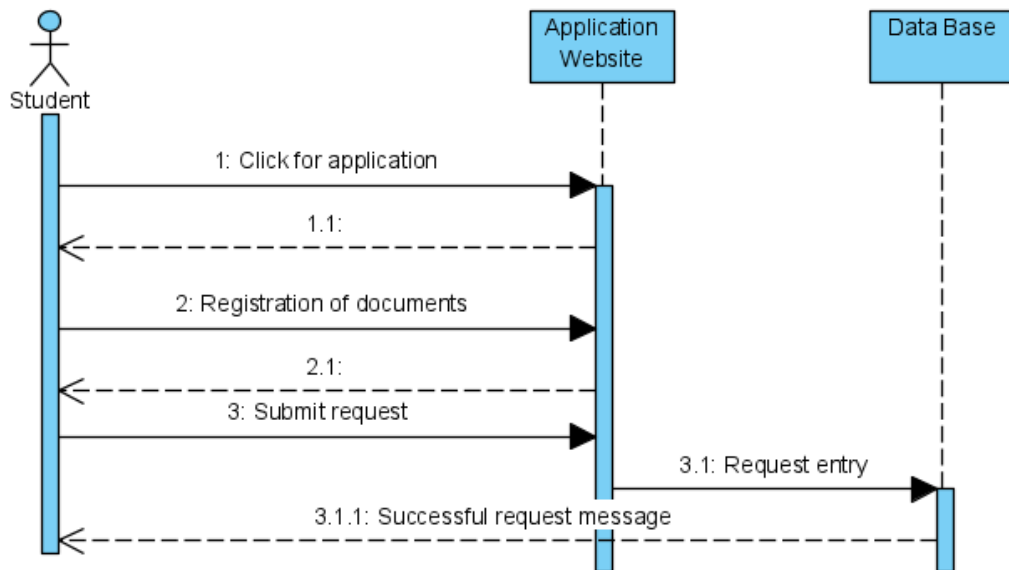
Use Case Diagram



Στο παραπάνω use case φαίνεται πως έχει γίνει generalization του User σε Teacher, Secretary και Student.

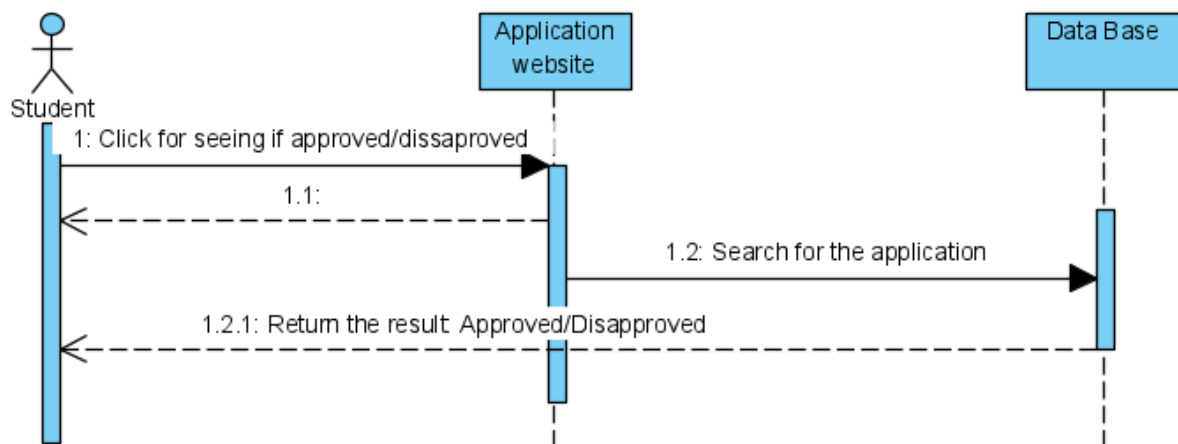
Sequence Diagrams

Apply Application



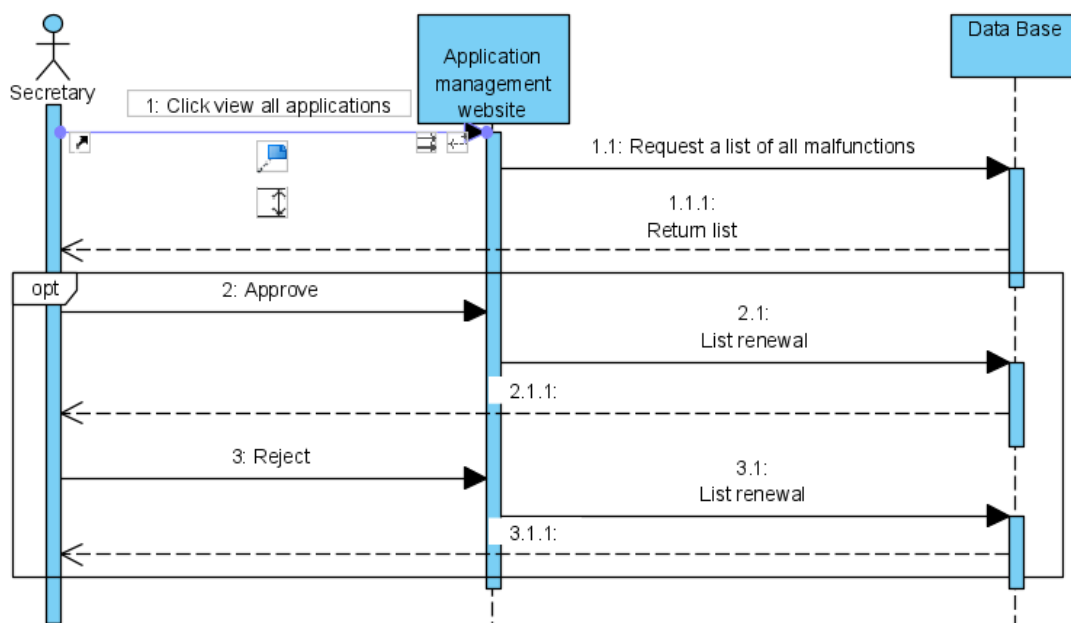
Στο παραπάνω sequence φαίνεται η διαδικασία που πραγματοποιείται κατά την αίτηση για μεταπτυχιακό από τον φοιτητή. Η αίτηση καταχωρείται στο τέλος της διαδικασίας στην βάση δεδομένων του συστήματος.

Show if approved/disapproved



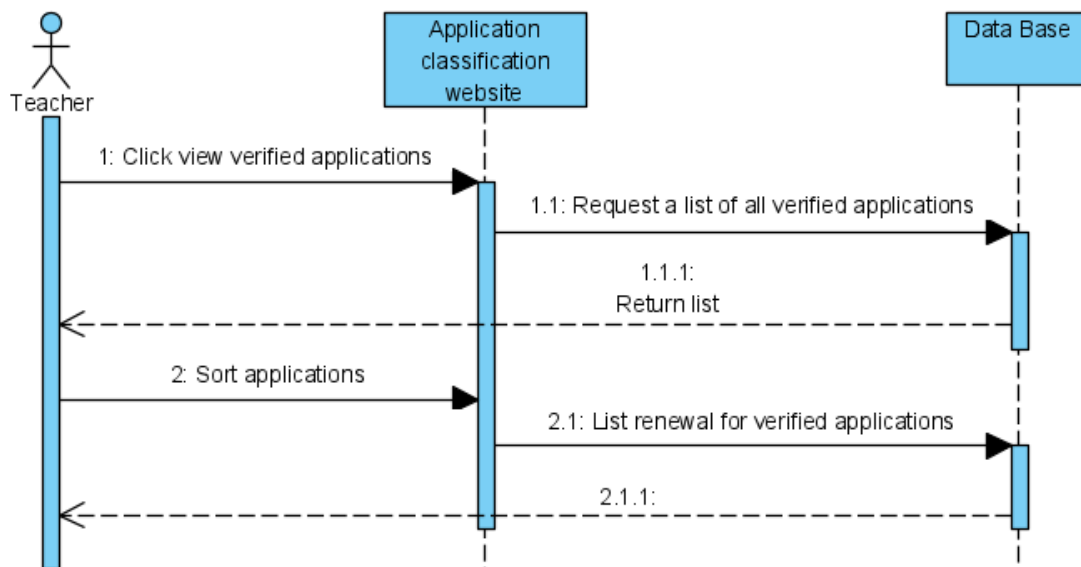
Στο παραπάνω sequence φαίνεται η διαδικασία κατά την οποία ο φοιτητής βλέπει αν έγινε δεκτός ή όχι.

Approve/Disapprove Applications



Στο παραπάνω sequence φαίνεται η διαδικασία κατά την οποία η γραμματεία επικυρώνει/απορρίπτει τις αιτήσεις. Ανάλογα με την επιλογή της, ενημερώνεται η λίστα.

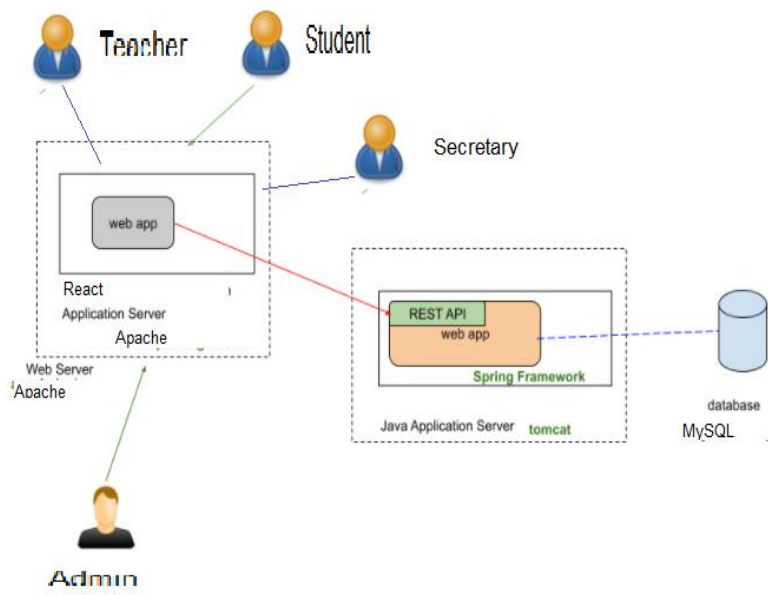
Sort Applications



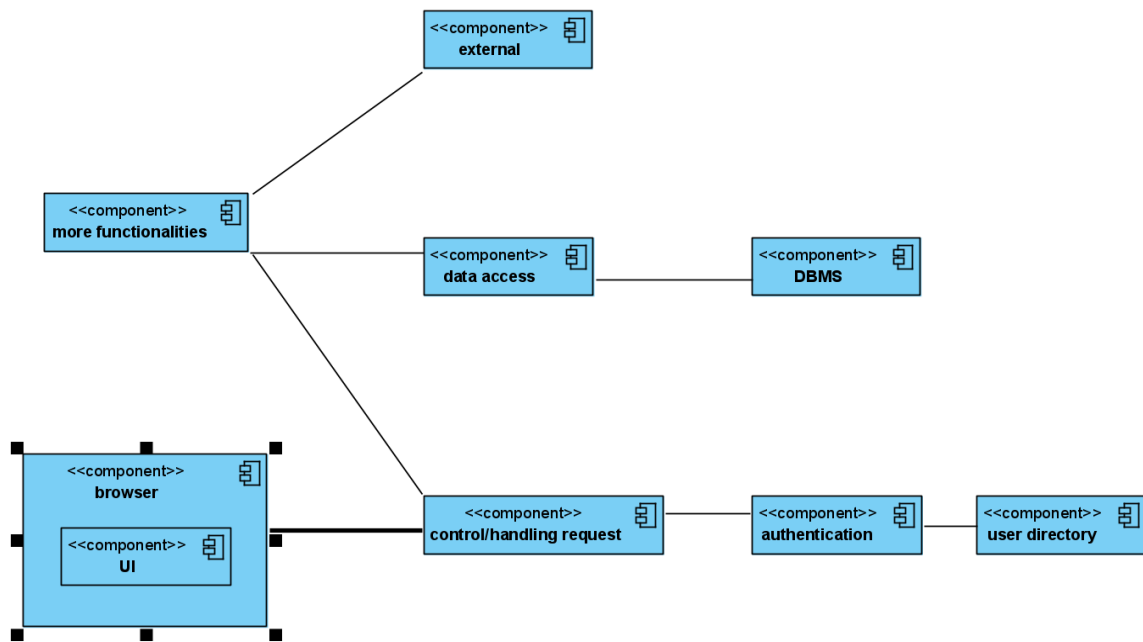
Στο παραπάνω sequence φαίνεται η διαδικασία κατά την οποία ο καθηγητής βάζει σε σειρά κατάταξης τις αιτήσεις.

Σχήμα με την συνολική αρχιτεκτονική

Ενδεικτική



Component Διάγραμμα



Εγχειρίδιο χρήσης

Κατάλογος REST API:

Για application

Base URL: /api/applications

GET:/api/applications/ (Επιστρέφει όλες τις αιτήσεις που έχουμε φτιάξει).

POST:/api/applications/ (Φτιάχνουμε αίτηση στέλνοντας με json τα πεδία για τις αιτήσεις, δηλαδή: studentId, program, grade, professors).

GET: /student/{studentId} (Επιστρέφει την αίτηση σύμφωνα με το studentId του φοιτητή).

GET: {id} (Επιστρέφει την αίτηση σύμφωνα με το id αίτησης που δώσαμε).

PUT: {id} (Κάνουμε update την αίτηση σύμφωνα με το id της).

DELETE: {id} (Σβήνουμε μια αίτηση σύμφωνα με το id της).

Για authentication

Base URL: /api/v1

POST: /auth/login (Κάνουμε login στέλνοντας με json τα πεδία userName και password).

GET: /auth/userinfo (Επιστρέφει τις πληροφορίες του χρήστη που είναι συνδεδεμένος.

Αυτά που επιστρέφονται είναι firstName, lastName, userName και τους ρόλους του).

Για User

Base URL: /api/user

POST: /add (Δημιουργεί νέους χρήστες στέλνοντας με json userName, password, auth, firstName, lastName, email, phone)

Τεύχος υλοποίησης εφαρμογής

Github Repository

<https://github.com/AnnaXenou/DSEBackend.git>

README

Κατεβάζουμε το zip αρχείο από το Github και το κάνουμε unzip. Στη συνέχεια ανοίγουμε Eclipse και κάνουμε δεξί κλικ και Import project. Περιμένουμε να κάνει build και στην συνέχεια στην main class κάνουμε δεξί κλικ run as-> Java Application. Η βάση που έχουμε χρησιμοποιήσει είναι MySQL. Ανοίγουμε Postman και στέλνουμε σαν πρώτη εντολή POST <http://localhost:8080/api/v1/auth/login>. Στο body βάζουμε raw -> json ->

```
{  
  "userName": "admin",  
  "password": "admin"  
}
```

και κάνουμε login ως admin.

Παίρνουμε το token που μας δίνεται και αν πχ θέλουμε να φτιάξουμε μια αίτηση στέλνουμε νέο POST <http://localhost:8080/api/applications/>

Βάζουμε στα Headers ένα ακόμα Authorization -> Bearer token given

Και φτιάχνουμε και εδώ body -> raw -> json

```
{  
  "studentId": "21751",  
  "program": "IT",  
  "grade": "7",  
  "professors": "tsadimas, nikolaidi"  
}
```

και έτσι φτιάχνουμε μία αίτηση.