



Εργασία μαθήματος
«ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ»

ΖΑΛΑΧΩΡΗΣ ΧΡΗΣΤΟΣ-ΚΥΠΡΙΑΝΟΣ, it21922

ΑΠΟΣΤΟΛΙΔΗΣ ΜΥΡΙΑΝΘΗΣ, it219119

ΠΑΠΑΜΙΧΑΗΛ ΧΑΡΗΣ, it219122

Ημερομηνία



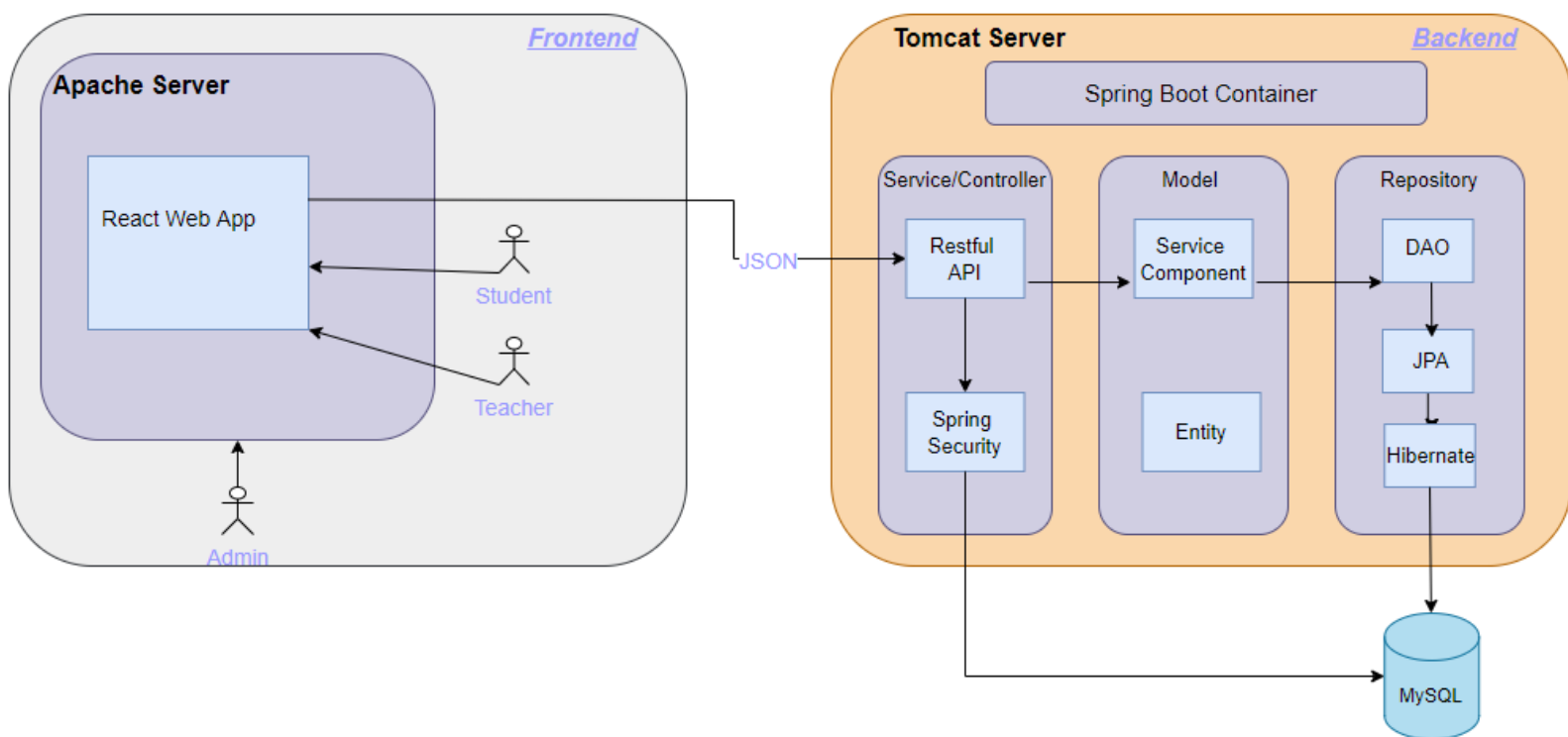
Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	2
Backend	3
System Architecture Diagram.....	3
Use Case Diagram	4
Use Case Description Table	5
Class Diagram.....	7
Sequence Diagram.....	8
Component Diagram	9
ER Diagram.....	10
Παραδοχές.....	11
Εγχειρίδιο χρήσης.....	12
Frontend	13
Τεύχος σχεδιασμού	14



Backend

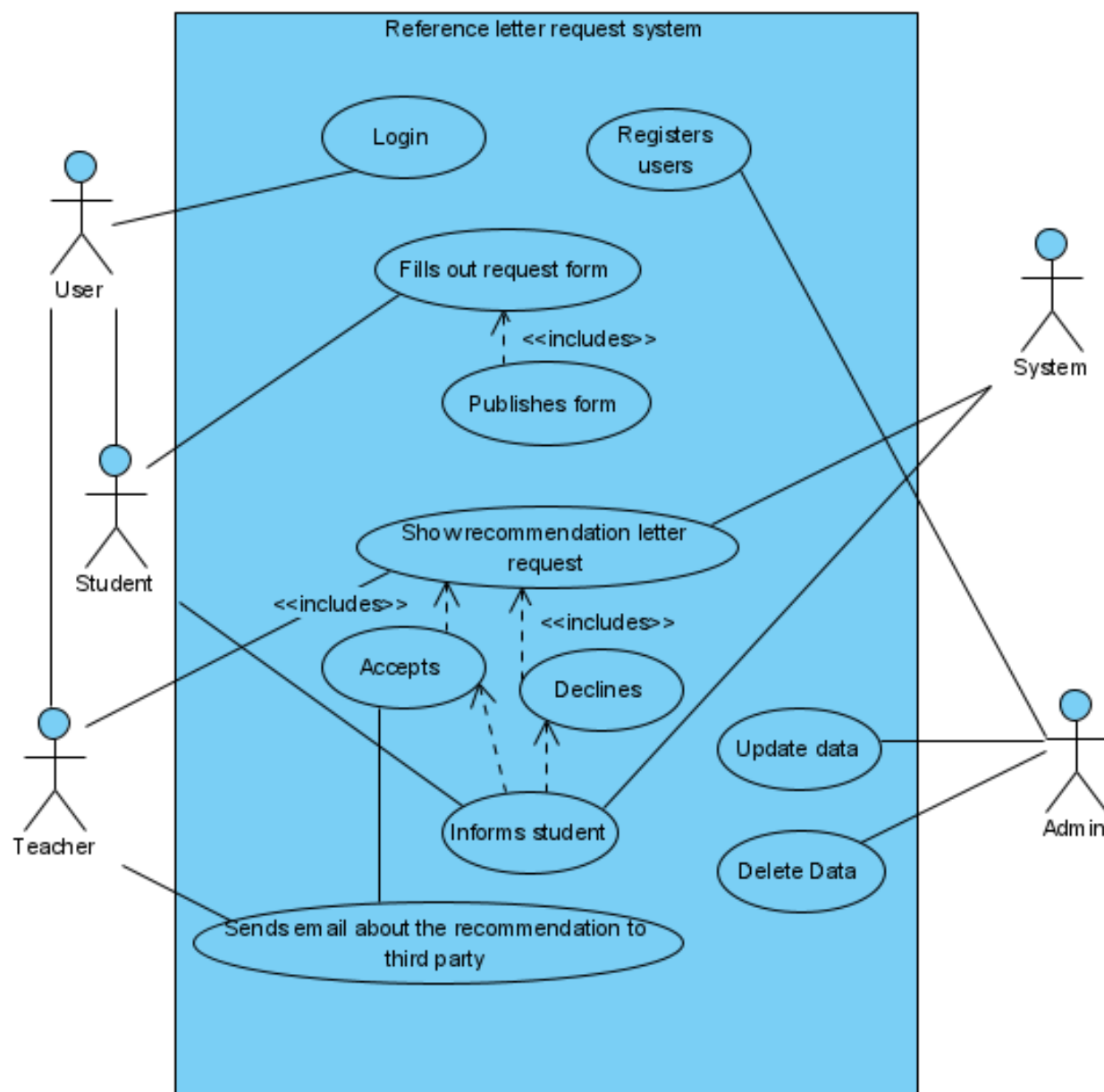
System Architecture Diagram



1-Μεσω του σχήματος συνολικής αρχιτεκτονικής του συστήματος αναπαριστούμε περιληπτικά την μορφή του backend και του frontend.



Use Case Diagram



2-To Use Case διάγραμμα ουσιαστικά παρουσιάζει και περιγράφει την δυναμική συμπεριφορά του συστήματος και χρησιμοποιείται για να καθοδηγήσει την ανάπτυξη του συστήματος σε όλες τις φάσεις του.



Use Case Description Table

Use Case: Writes and publishes request form/ Waits for response		
Brief description	Writes all the information in order to make a request about the recommendation letter	
Actors	Student	
Normal Flows	Tasks	Data
	After the student logs in the system he creates a referral letter, writing personal info, detailed grades, the teacher that w either accept or decline the letter and th place that is addressed to.	personalData[] transcript teacher addressedTo
	Clicks a button to make the request	
	Waits some days to see the status of the request	

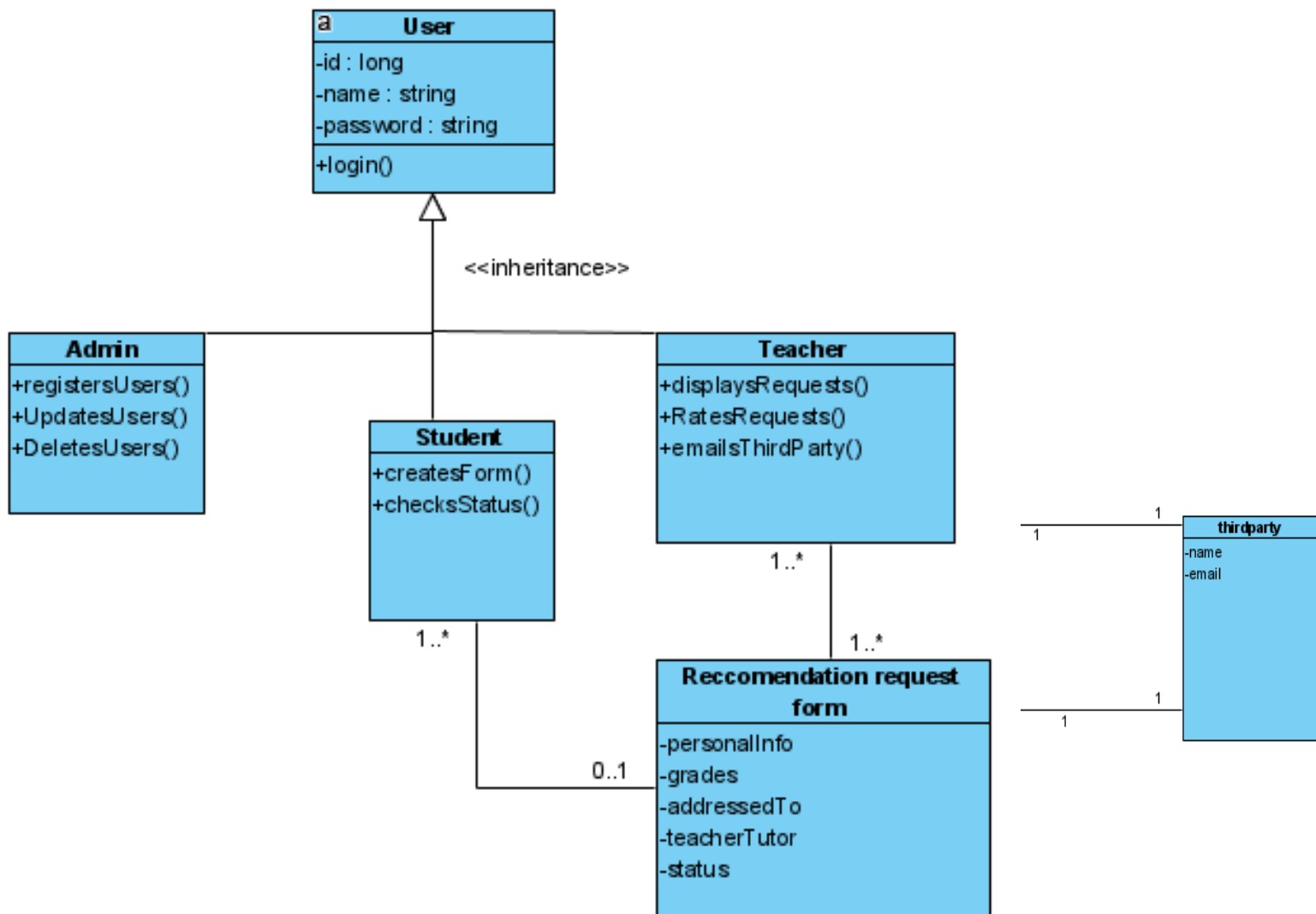


Use Case: Teacher checks requests, rates them and takes action		
Brief description	Checks the request and declines or accepts	
Actors	Teacher	
Normal Flows	Tasks	Data
	Teacher logs in the system and checks the recommendation letters that are targeted to him. He then decides if he will accept or decline the request	id password
Alternative flows	If he accepts then status is set and he proceeds to email the third party that is attached and addressed to the letter	status email
	If he declines then status is set to declined	status

Use Case: Creates, updates and deletes users		
Brief description		
Actors	admin	
Normal Flows	Tasks	Data
	After admin is logged in the system a window will pop up with some actions he can choose.	id password
	Create/update/delete a new user with n. password and role	name, id, password role



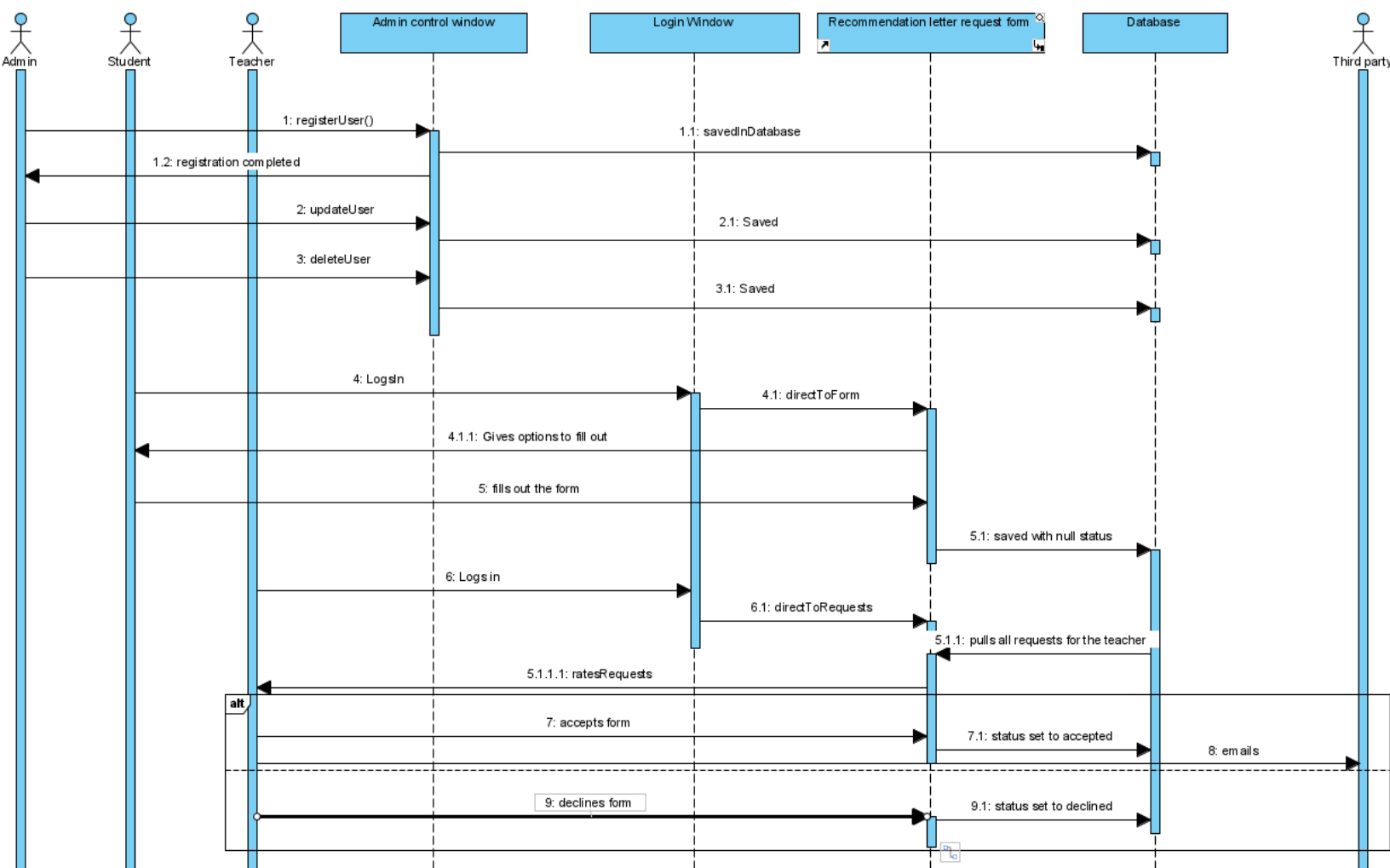
Class Diagram



3-Ουσιαστικά μέσω του διαγράμματος κλάσης παρουσιάζεται η στατική δομή των αντικειμένων του συστήματος και των σχέσεων μεταξύ τους.



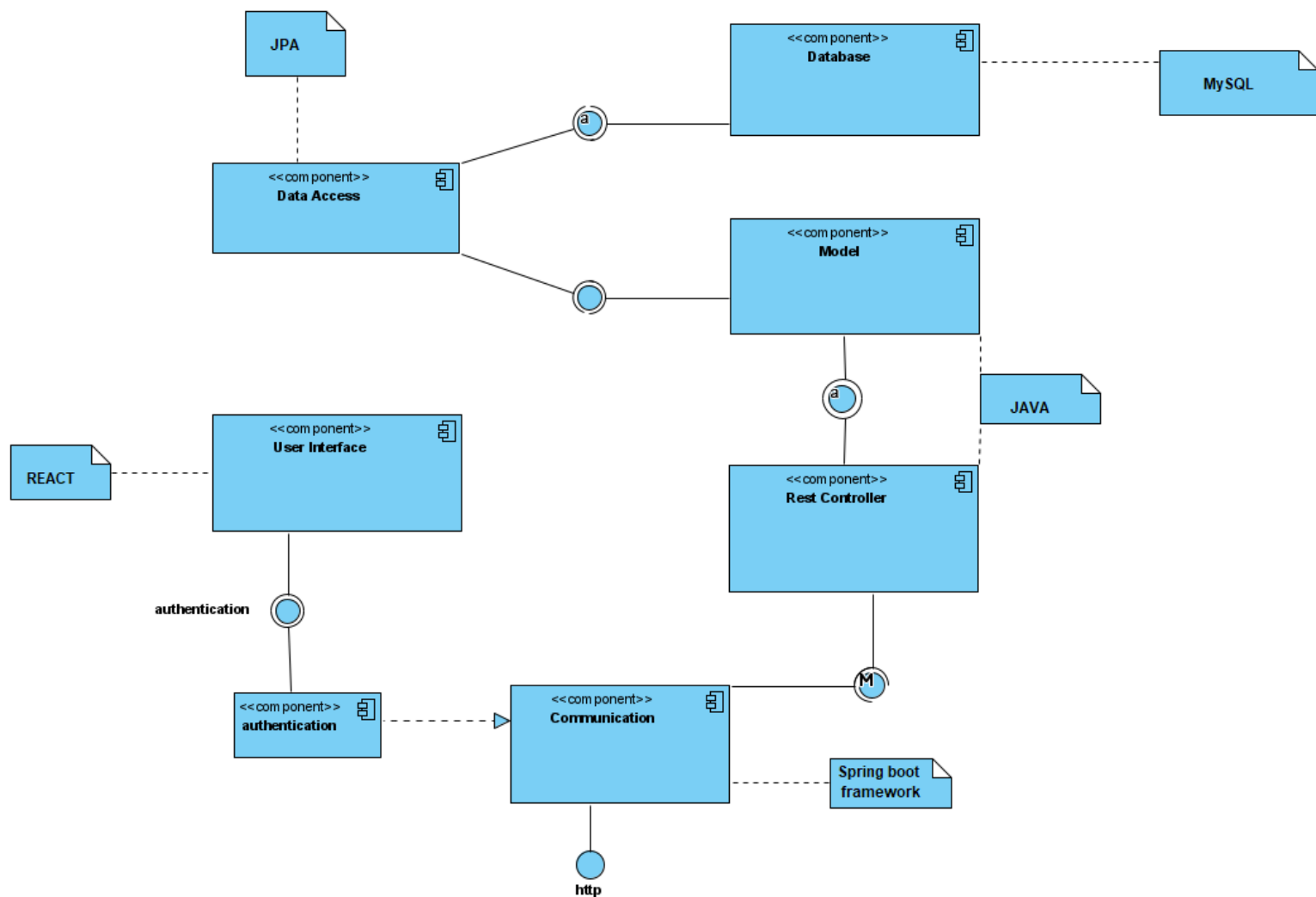
Sequence Diagram



4-Τα διαγράμματα ακολουθίας παρουσιάζουν την αλληλεπίδραση των αντικειμένων μέσω της ανταλλαγής μηνυμάτων και χρησιμοποιούνται για να την περιγραφή του κύκλου ζωής των αντικειμένων. Στην συγκεκριμένη περίπτωση έχουμε προσθέσει όλες τις διαδικασίες σε μικρά κομμάτια στο ίδιο διάγραμμα.



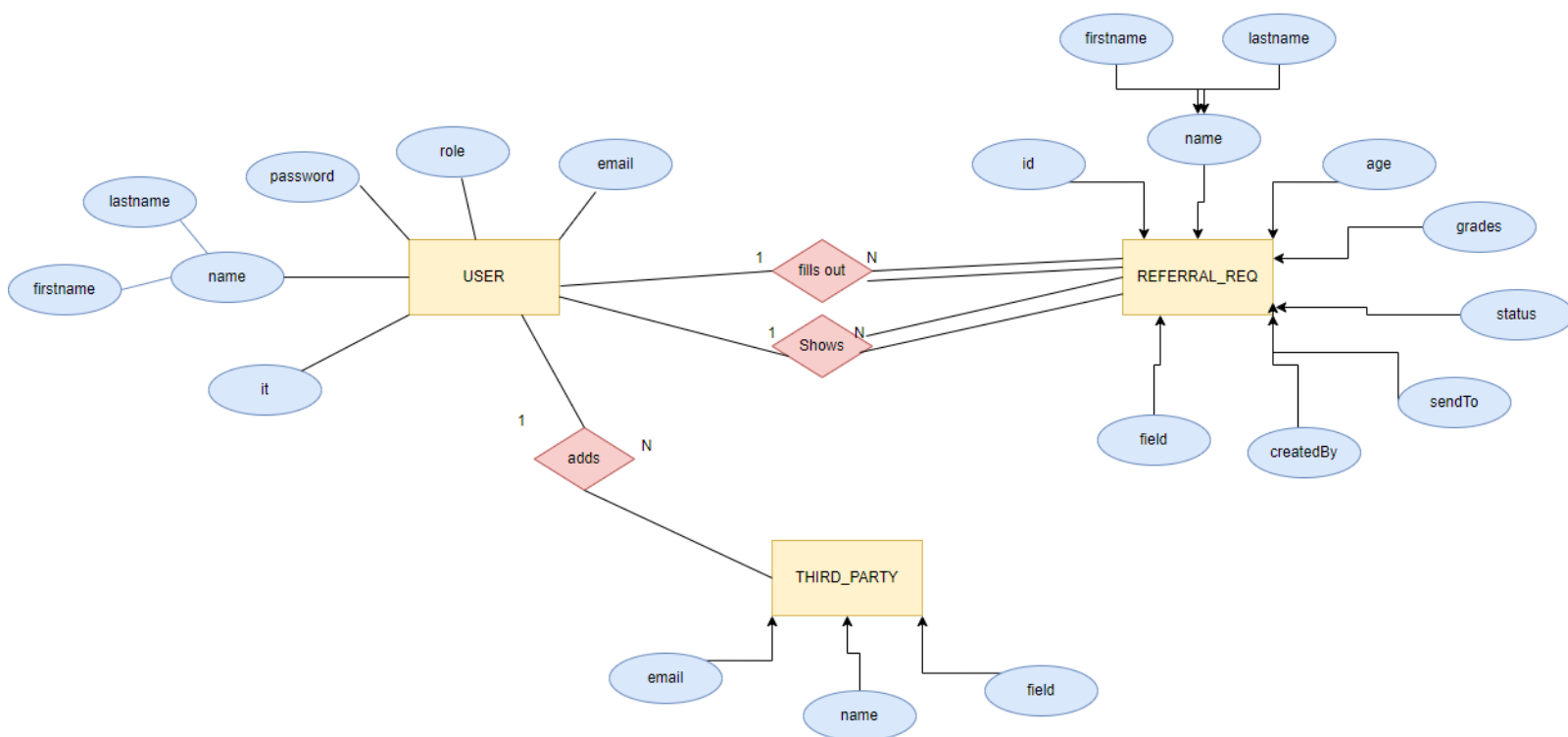
Component Diagram



5- Τα component διαγράμματα χρησιμοποιούνται για να περιγράψουν την φυσική διάσταση του συστήματος με όρους εκτελέσιμων αρχείων, βιβλιοθηκών κώδικα, σελίδων html, εγγράφων, πίνακες βάσεων δεδομένων κ.λπ. Στο συγκεκριμένο απεικονίζουμε το σύστημα μας με τους όρους και τα στοιχεία που χρησιμοποιεί.



ER Diagram



6- Μέσω του διαγράμματος παρουσιάζουμε την δομή της βάσης μας



Παραδοχές

- Σκεφτήκαμε πως θα ήταν καλύτερο το σύστημα να λειτουργεί όπως το eclass καθώς θεωρήσαμε ότι θα είναι κάτι που χρησιμοποιούν φοιτητές και καθηγητές. Οπότε υποθέσαμε ότι ο admin θα ελέγχει εξ ολοκλήρου το register page όπου θα μπορεί ουσιαστικά να δημιουργεί νέους χρήστες, είτε φοιτητές(students) είτε καθηγητές(teachers). Αντίστοιχα οι χρήστες αυτοί θα μπορούν να κάνουν login με τα στοιχεία που τους έχουν δοθεί από τον admin.
- Μιλώντας για χρήστες και ρόλους αποφασίσαμε πως οι εξωτερικοί φορείς δεν θα αποτελούν ρολό του συστήματος, δηλαδή δεν θα έχουν κάποιο λογαριασμό και κάποια λειτουργία στο σύστημα. Βασική τους ύπαρξη θα είναι στην βάση δεδομένων(ο admin θα έχει κάποια λειτουργία που θα τους προσθέτει) όπου θα υπάρχει κάποιο table με τα στοιχεία τους(email , name , and fieldofwork) ώστε ο μαθητής να μπορεί να στοχεύσει κάποιο πεδίο που τον ενδιαφέρει όπως μεταπτυχιακό ή επαγγελματικούς σκοπούς και για την συστατική του και αντίστοιχα ο καθηγητής να μπορεί να δει το πεδίο αυτό και να διασταυρώσει σύμφωνα με τα emails τους σε ποιον τελικά να στείλει την επιστολή.
- Επίσης προσθέσαμε τους καθηγητές ως στοιχείο του recommendation request form table ώστε να γνωρίζει ο κάθε καθηγητής πια αίτηση απευθύνεται σε αυτόν.
- Αντίστοιχα στο table με την συστατική οι βαθμοί θα αποτελούν ένα string από τους μέσους όρους κάθε εξαμήνου του φοιτητή χωρισμένοι με κόμμα.
- Τέλος προσθέσαμε μια λειτουργία η οποία ουσιαστικά θα παρουσιάζει το status της κάθε αίτησης ώστε ο μαθητής να μπορεί να βλέπει αν έχει γίνει δεκτή ή όχι.



Εγχειρίδιο χρήσης

Link του Project: <https://github.com/chriszal/referenceLetterApp>

- First Path can be accessed by **everyone** in order to login to the system

<https://localhost:8080/login>

- ✓ After login, depending on the logged in user the system will redirect to the according dashboard.
- ✓ Every User Role has its own Dashboard.

- Endpoints and Paths with **admin** role authorization:

<https://localhost:8080/admin/users> (GET)

- ✓ Returns all the users in the system
- ✓ Used to access the users when admin needs to update one or delete them.

<https://localhost:8080/admin/users/{id}> (GET)

- ✓ Returns exact user with id==it
- ✓ Used to select an exact user to modify or delete

<https://localhost:8080/admin/users/register> (POST)

- ✓ Post new user in the system
- ✓ Used to register a new user in the system

<https://localhost:8080/admin/users/modify/{id}> (PUT)

- ✓ Update a user in the system

<https://localhost:8080/admin/users/modify/{id}> (DELETE)

- ✓ Delete a user in the system

<https://localhost:8080/admin/thirdparty/add> (POST)

- ✓ Add a new third party in the system



➤ Endpoints and Paths with **student** role authorization:

<https://localhost:8080/request/form> (POST)

- ✓ Posts the request letter data

<https://localhost:8080/request/owned> (GET)

- ✓ Returns the request of the logged in student in order to check the status

➤ Endpoints and Paths with **teacher** role authorization:

<https://localhost:8080/requests/sendTo> (GET)

- ✓ Returns all requests with null status that are directed to the logged in teacher
- ✓ Used by teacher to check all his requests before approving or declining them

<https://localhost:8080/requests/sendTo/{id}> (GET)

- ✓ Returns request that is directed to the logged in teacher and has null status
- ✓ Used to pick an exact request

<https://localhost:8080/requests/sendTo/{id}> (PUT)

- ✓ Updates status of request

<https://localhost:8080/requests/approved> (GET)

- ✓ Gets all requests that are approved and belong to the logged in teacher

<https://localhost:8080/requests/approved/{id}> (GET)

<https://localhost:8080/requests/approved> (POST)

- ✓ Gets approved request and email Data about the field of the request as well as the email of the logged in teacher
- ✓ Posts Email with sendGrid API

Frontend



Τεύχος σχεδιασμού

The screenshots bellow are design concepts and will change in the future

Login Page:

The image shows a login page design concept. It features a dark blue gradient background. In the center is a white rounded rectangle containing the title "REFERRAL" in bold black letters. Below the title are two input fields: "Username" with a user icon and "Password" with a lock icon. Each field has a placeholder text "Type your username" and "Type your password" respectively. At the bottom of the white box is a blue rounded button with the text "LOGIN" in white.

- Every user has access to login page at the moment he enters the website

Admin Dashboard:



























- The admin dashboard has its own navigation bar with the actions of an admin. In our case an admin can register a new user, modify an existing one and add a new party to the system that is directly connected with the field that a user will choose to write his referral letter.

The image shows a 'Create New User' form with a dark background. The form contains the following fields and elements:

- Title:** IT
- First Name:** Type your first name
- Last Name:** Type your last name
- Password:** Type your password
- Email:** Type your email
- Role:** A dropdown menu with 'Admin' selected and a downward arrow.
- REGISTER Button:** A blue button with a gradient and rounded corners.

- Admin gets redirected to his dashboard after the log in ,where he firstly sees the register tab.
- Admin can register a new user with all the above attributes and the system will check if the user already exists.



Users List						
it	First Name	Last Name	Password	Email	Role	Actions
admin	admin	admin	\$2a\$12\$K/sVWqkI9HCodU8W5t35sujQuieTapOT1GtAaDm6phMgMwX8WP5ea	admin@gmail.com	ADMIN	 
it1	chris	zalachoris	\$2a\$12\$QhowltiSHmITHRJmHe0r4OtS78acaxPB9yf9zplBoO4HKTMfotVga	christoszal@gmail.com	STUDENT	 
it2	Peter	Jones	\$2a\$12\$QhowltiSHmITHRJmHe0r4OtS78acaxPB9yf9zplBoO4HKTMfotVga	peterjon@gmail.com	STUDENT	 
it3	James	Casper	\$2a\$12\$QhowltiSHmITHRJmHe0r4OtS78acaxPB9yf9zplBoO4HKTMfotVga	jamescasper@gmail.com	STUDENT	 
it4	Bob	Morley	\$2a\$12\$QhowltiSHmITHRJmHe0r4OtS78acaxPB9yf9zplBoO4HKTMfotVga	bobmor@gmail.com	STUDENT	 
it5	Meg	Lawrence	\$2a\$12\$QhowltiSHmITHRJmHe0r4OtS78acaxPB9yf9zplBoO4HKTMfotVga	meglaw@gmail.com	STUDENT	 
itM1	Nick	Koutas	\$2a\$12\$MeyFMSrYO5gnqALy2YfHFeKTX.44ExtX40UoSzwF0YYKN0s.7mG	Nickkout@gmail.com	TEACHER	 
itM2	Steven	Water	\$2a\$12\$MeyFMSrYO5gnqALy2YfHFeKTX.44ExtX40UoSzwF0YYKN0s.7mG	StivWater@gmail.com	TEACHER	 
itM3	Bill	Cosby	\$2a\$12\$MeyFMSrYO5gnqALy2YfHFeKTX.44ExtX40UoSzwF0YYKN0s.7mG	bilco@gmail.com	TEACHER	 
itM4	Shane	White	\$2a\$12\$MeyFMSrYO5gnqALy2YfHFeKTX.44ExtX40UoSzwF0YYKN0s.7mG	ShaneWhite@gmail.com	TEACHER	 
itM5	Barack	Obama	\$2a\$12\$MeyFMSrYO5gnqALy2YfHFeKTX.44ExtX40UoSzwF0YYKN0s.7mG	barack@gmail.com	TEACHER	 
itM6	Dora	Black	\$2a\$12\$MeyFMSrYO5gnqALy2YfHFeKTX.44ExtX40UoSzwF0YYKN0s.7mG	doar@gmail.com	TEACHER	 

- Admin can see a list of all the users in the system when he clicks on the modify tab , where he can either delete or modify a user
- Admin can modify a user by changing all the fields or he can delete the user

Add Third Party

Email
Type the organization's email

Name
Type the organization's name

Field
Job

ADD

- Admin can add a third party



Student Dashboard:

REFERRAL Form Status

Create a referral

First Name
Type your first name

Last Name
Type your last name

Age
Type your age

Average grade in each year

1st	2nd	3rd	4th
-----	-----	-----	-----

Receiver
Type your receiver's it

Field
Job

SUBMIT

- Student gets redirected to his dashboard after login where he can create a new recommendationreferral letter with the attributes shown above

Status Form			
id	Receiver	Field	Status
1	itM1	Job	APPROVED
2	itM3	Postgraduate	Pending

- Student can see all his requests and the status(approved, pending, declined)



Teacher Dashboard:

REFERRAL Requests itM1 Logout

Requests									
it	First Name	Last Name	Age	1st Year Grade	2nd Year Grade	3rd Year Grade	4th Year Grade	Field	Actions
it1	chris	zalachoris	20	6.5	4.9	8	9	Job	APPROVE DECLINE
it3	James	Casper	18	5	4	7	6	Postgraduate	EMAIL
it5	Meg	Lawrence	19	6	4	7	9	Job	DECLINED

- Teacher gets redirected to his dashboard where he can view all the requests that are sent to him
- He can also send email on another tab that is being implemented
- He can decide after reviewing the letter if he wants to approve or decline
- If he approves an email button will redirect him in an email page
- If he declines the status going to change to "declined"

Create Referral Email

User Info
id: 4
first name: James
last name: Casper
age: 18
grades: 5,4,7,6
field: Postgraduate

Receiver Email
Select an email

Subject
Type a subject

Message

SEND