Palestine
An-Najah National University
Faculty of Information Technology
Computer Science

دولـــــة فلســـــطين
جامعة النجاح الوطنية
كلية تكنولوجيا المعلومات
قسم علم الحاسوب

MENTORSHIP PROGRAM FOR UNDERGRADUATE IT STUDENTS

Students:

Abdalrhman Marea	**11820796**
Ahmad Saleh	**11819495**
Abdalhakeem Shkal	**11821259**

Supervisor:

Suhad M. Daraghmeh

An-Najah National University
2022

A c k n o w l e d g e m e n t

I would like to thank Ms. Suhad M. Daraghmeh and efforts that they provided for our team. I would like to thank her not only for her guidance and patience, but also for her continuous suggestions and feedbacks to this work.

Abstract

Nowadays, the demands on recommendation system are getting increase quickly, like recommendation for movies, books, and restaurants. In our project the recommendation will be for future job of university student. The aim of our project is to get prediction for the field of the student by match it with forms that the student will fill it. This approach aims to reduce the amount of data that will enter the fuzzy logic algorithm to produce a prediction the efficiency ratio of student In four criteria (Front End , Back End , Full Stack , Quality assurance)

CHAPTER 1

INTRODUCTION

1.1 Introduction

Recommendation it is got more needed which it depends on the application and what user need, like amazon for ecommerce recommendation or net flex for movie recommendation. Our project it is a Mentorship Program for undergraduate IT students that recommend a career or programing field for the student depends on some properties and information that we get from quizzes that the student should answer.

The main reason for creating the system is for helping mentor from companies to find the top student with well knowledge in some criteria like Front End for example and that help companies to save time and effort by avoiding classification of big amount of student. also help student to know what they good at and help them to decide which career they should keep up with in their future life.

After recommend a career or filed or job to the student and show the best student to the mentor they can send an request asking fer mentorship and they can accept so the mentor can mentoring him by giving him a task or he can refuse it to choose a another mentor.

Our system is a web application which is provide several services that offer for companies mentors accounts to view the student list and send mentoring request to the student to create a mentorship relation which allow to the mentor to training his student by using task system and it offers for the applicant or student an account to get into the system which offer a predicted career post for applicant resume.

1.2Project Scope

Mentoring System Is a website that help the companies to find a high-quality student by using algorithm that classified the student and do recommendation for the mentors who’s looking for student and training them through their university life so they can have a full qualified fresh graduated student and also help the student to know what field they good at and the field they don’t.

1.3Problem Specification

With increasing of university student who study programming majors (Cs , Cis , Nis , and Ce) we notice the problem of student who don’t know anything about work environment and companies so the company need to train them and that’s need a lot of time and effort and we try to solve this problem by developing our system.

1.4 Goals and Objectives

Our goals with this project to develop a system with friendly environment for both students and mentors so they can have a flexible mentorship relation to produce a student who ready to enter the Labor market directly.

CHAPTER TWO

METHODOLOGY

2.1 Introduction

While we were working on our project, we try to find the best methodology or algorithm to relay on to find the most accurate result by searching about Evaluation Criteria that we can use as standard to create a dataset we can depend on it to make a prediction but we can't find one, so we decide to use fuzzy logic to make the prediction for our project.

2.2 Fuzzy logic

Fuzzy logic-based recommendation systems are famous and a lot of published works adopt it in the design and implementation of their recommendation systems. Authors of the work presented in propose a personalized recommendation system driven by fuzzy logic technique for products that target to recommending optimal products to prospective buyers, promoting the rate at which customers visit online stores and eventually increase sales for online businesses.

Depend on Fuzzy logic we build our system using some techniques that can be divide in steps:

A-Evaluation Criteria:

To calculate the percentage for prediction we put a three criteria that all student have in common to be fair and these criteria is :

1- University subject (optional subject):

   Which is the subject or courses that the student can select it and its not necessary to take one of them and it varies from one specialty (major) to another.

2- Programming courses: دورات البرمجة

   It's the courses that you can take outside the university and we select the courses from the certificated place.

3- Programming language and skills:
   The language that popular and apart of (Full Stack, Back End, Front End, Quality Assurance)

B- Add Rating:

After collecting all criteria that we want to use in our formula or prediction we want to give it a rating to using it to produce a result as percentage.

These rating present who much the course or language is related to the field like Front End.

To give a very accurate rating as possible we use three ways or method:

1- Asking expert like university student who graduated and working in these field and university doctor.

2- Analyzing the description of university subject.

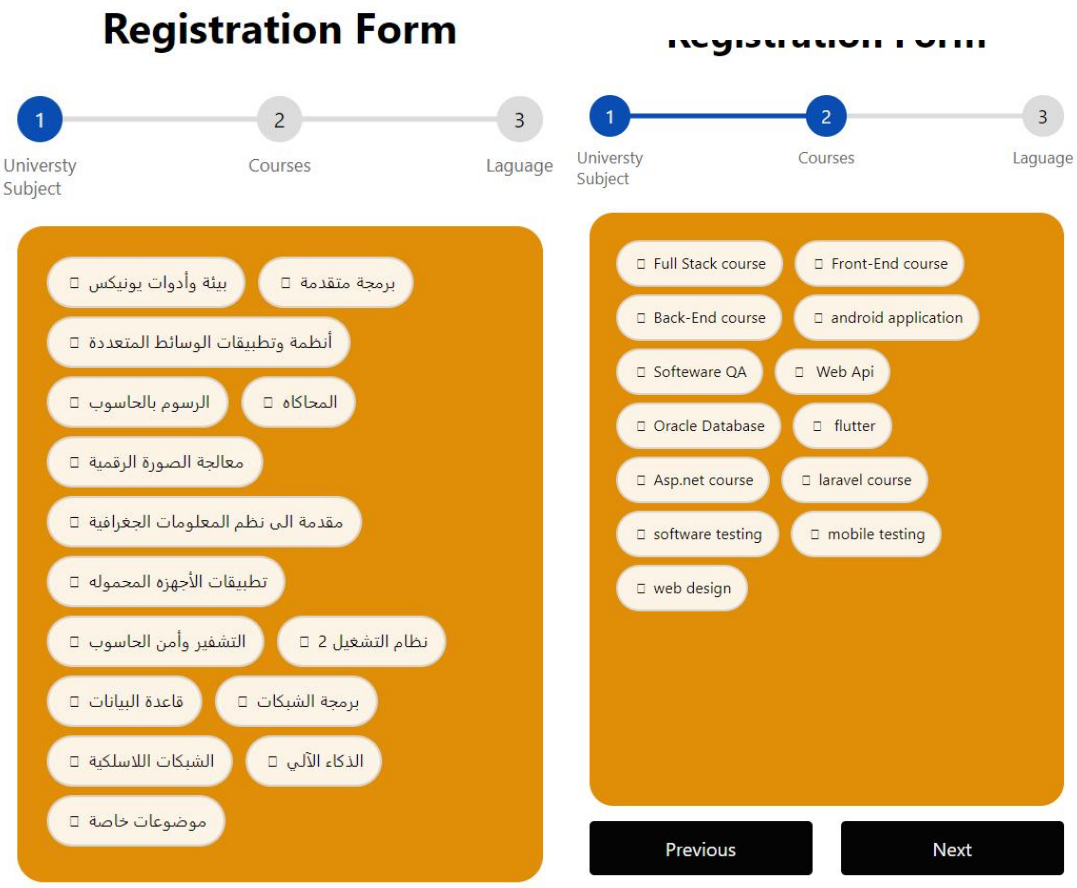
3- Searching in internet looking for previous data in similar project.

C- Evaluation

After student sign up he will fill form of three part

- University subject
- Programing courses
- Programing language and skills

And this who its look like



When the student press submit this form will show as a check box that will set a binary result to the algorithm (if the checkbox is checked it will send 1 else 0)

The value that come from the form will be part of equation or formula:

For all forms:

$$\text{FEP} = \sum_{i=0}^{n} \frac{b_{if} \times r_{if}}{100}$$

$$\text{BEP} = \sum_{i=0}^{n} \frac{b_{ib} \times r_{ib}}{100}$$

$$\text{FSP} = \sum_{i=0}^{n} \frac{b_{if} \times r_{if}}{100}$$

$$\text{QAP} = \sum_{i=0}^{n} \frac{b_{iq} \times r_{iq}}{100}$$

Where :

- FEP is Front End percentage
- BEP is Back End percentage
- FSP is Full Stack percentage
- QAP is Quality assurance percentage
- Bi is the binary value from the form
- Ri is the rate that we give

And for the final result (percentage)

Front end final percentage $=(FEP_{(subject)} + FEP_{(course)} + FEP_{(language)})/3$

Back end final percentage $=(BEP_{(subject)} + BEP_{(course)} + BEP_{(language)})/3$

Full Stack final percentage $=(FSP_{(subject)} + FSP_{(course)} + FSP_{(language)})/3$

Quality assurance final percentage $=(QAP_{(subject)} + QAP_{(course)} + QAP_{(language)})/3$

So, with this equation each student has a 4 percentage that stored in his table in database.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 functional requirement :

A-Mentor:

1- can send a request to student
2- can cancel a request
3- view a table of student sorted depend on their quality
4- add a task
5- delete a task
6- edit a task
7- grade a task for student
8- view student grade percentage for total task

B- Student:

1- can accept a request from mentor

2- can refuse a request from a mentor

3- can rate a mentor

4- upload a solution to mentor task

5- can fill the form

3.2 Non-functional requirement :

- **The system shall ensure that all data is stored securely and that unauthorized access is prevented form borrowing, rating, commenting, or doing anything.**
- **The system will have different types of users and each one has access constraint.**
- **The system shall accommodate high number users without any fault.**

Environment:

Spring framework:

Spring is an open-source application framework that provides infrastructure support for developing Java applications

spring has many traits:

1- Spring makes programming Java quicker, easier, and safer for everybody. Spring's focus on speed, simplicity, and productivity has made it the world's most popular Java framework.

2- spring has a special feature which is dependency injection

3- Spring Security: is an open source security framework. It supports authentication and authorization

Languages used:

1- **java**
2- **JavaScript**

3- **JPA Injection**

Others:

1. **Html**
2. **CSs**
3. **bootstrap 4**
4. **jQuery**
5. **Git**

Applications used:

1- **WampServer**
2- **IntelliJ IDEA**

Chapter 4: system overview

Section 1: Sign Up page

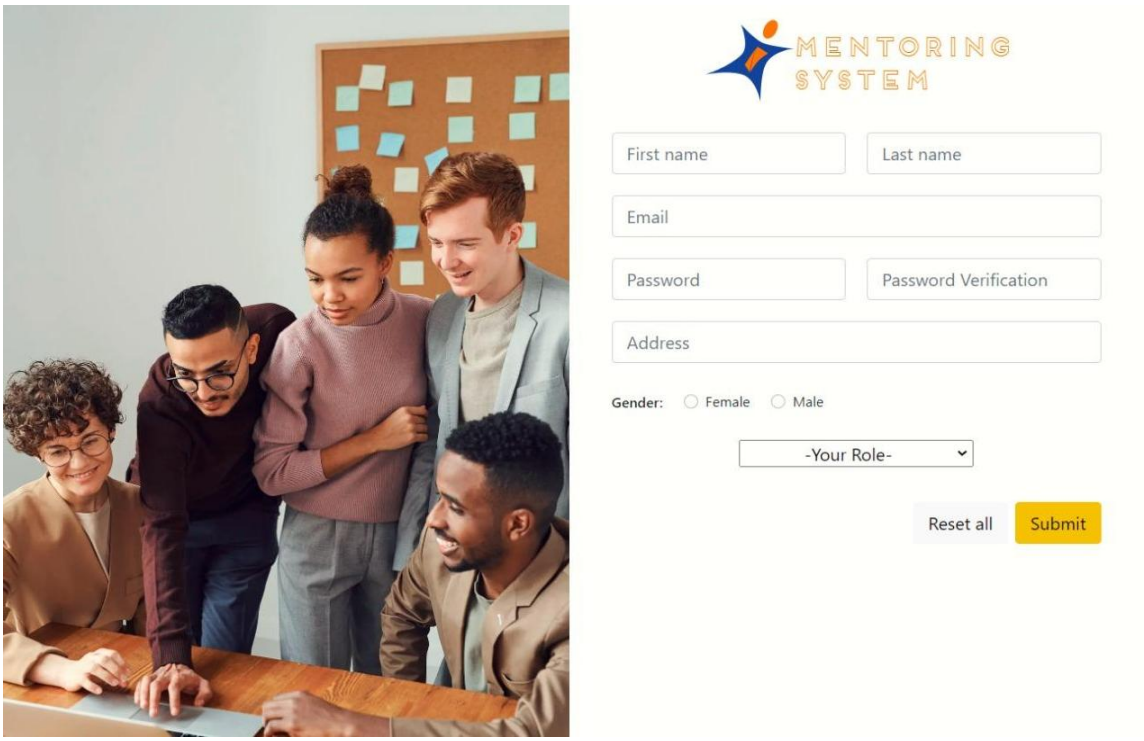


Figure 1:Sign Up page

Where the user will enter his personal information needed to run the application in the way that is designed.

The required information:

1. First and last name.
2. Email.
3. Password.
4. Their address
5. Gender.
6. Their roles ether full-stack, frontend, and backend.

The system can make difference between students and mentors from The Email domain, it can verify if the user has a domain for a company or organization.

Section 2: login page

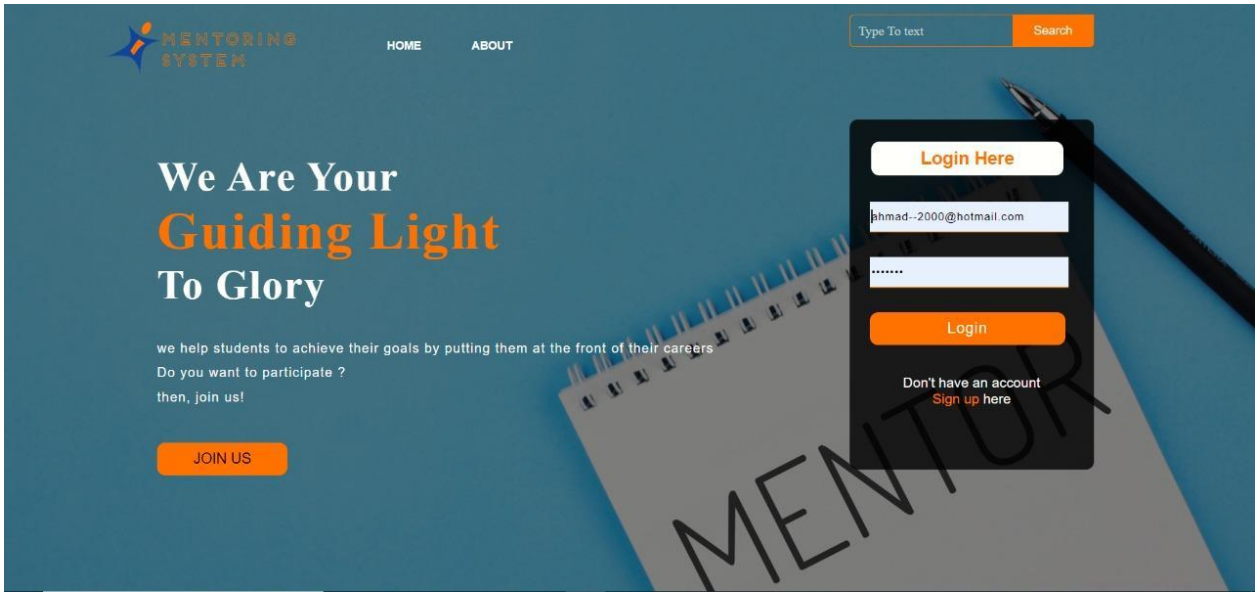


Figure 2: logIn page

Where the user will enter his personal information if they already have sign-up in the way that is designed.

The required information:

1. The email
2. Password

It consists of two text fields, where the user can enter his email and password.

Section 3-A: home page

After the mentor's login, he is greeted with a welcoming page where he can either go to his Account, My Network, Request, or tasks, through the header, if you scroll down you will see the mentor control there they can mentor an available new student, received requests from the students you send application and stop mentoring student.

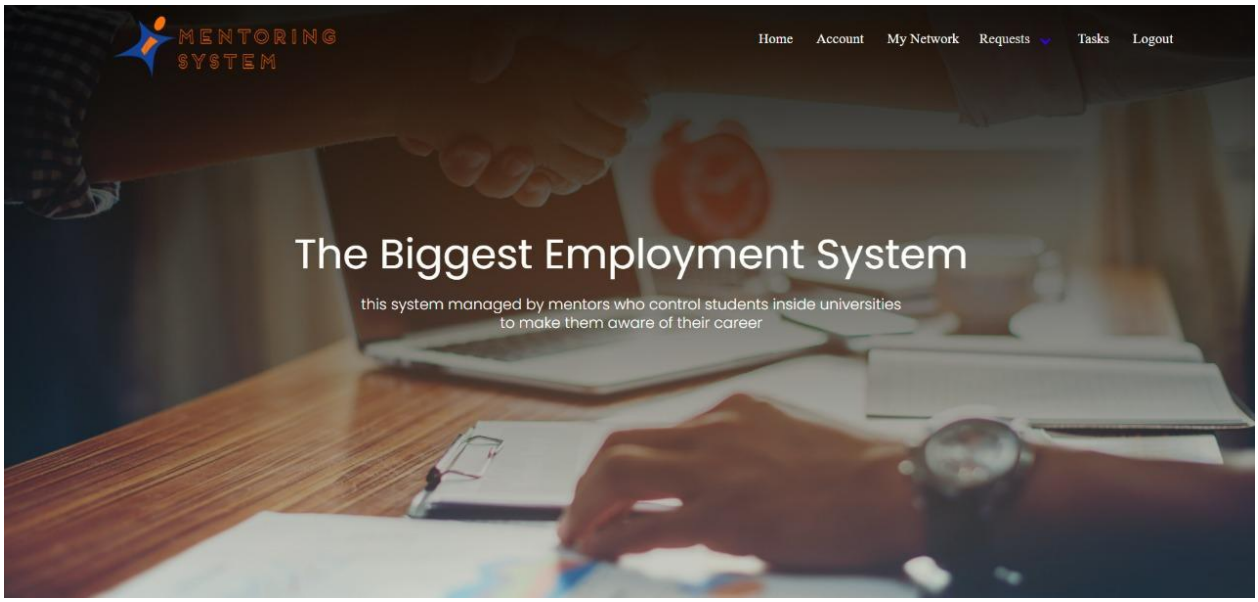


Figure 3: homepage

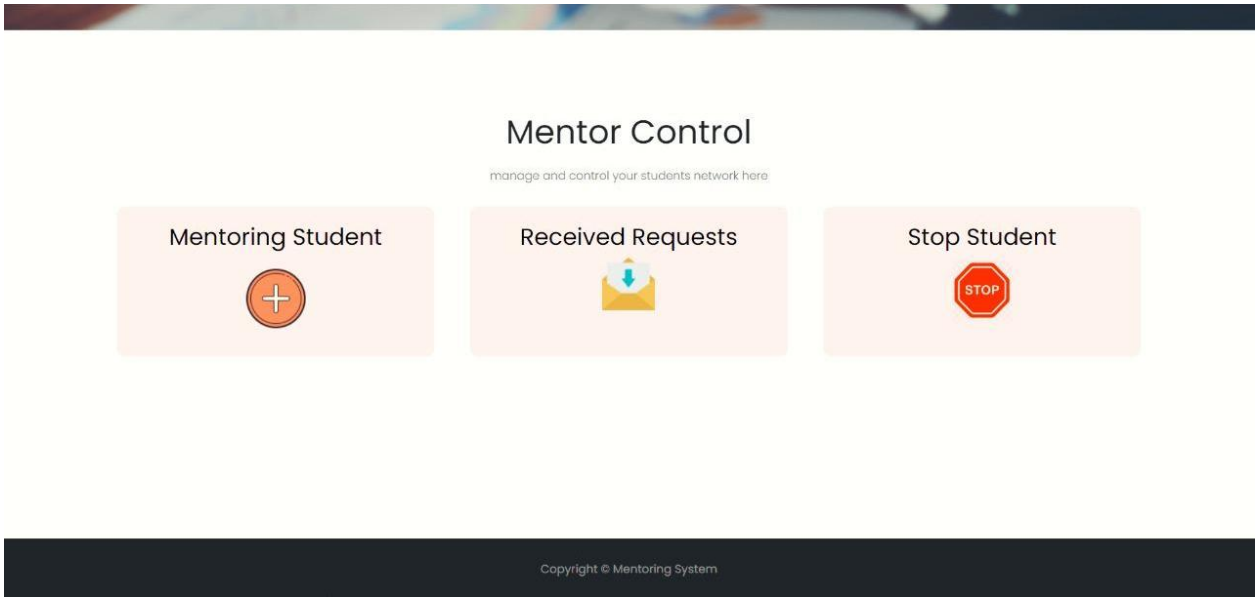


Figure 4: Mentor Control

Section 3-B: Mentor Network view

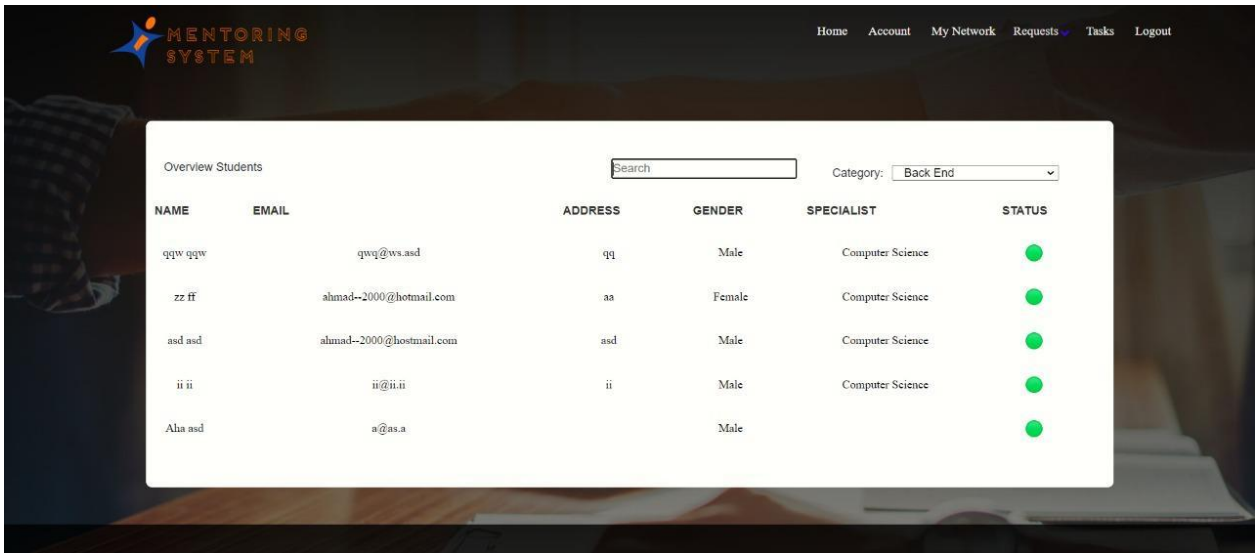


Figure 5: network view

When the mentor wants to search through students, first he chooses what category the students want to specialize in, which are backend, frontend, full-stack, and quality assurance.

Then the table will show him who is in that category, and it will show them in descending order, depending on the value that the fuzzy algorithm calculated.

The info of the student:

1. Name.
2. Email
3. Address
4. Gender
5. Specialist
6. Statues

Section 3-C: Request Network view

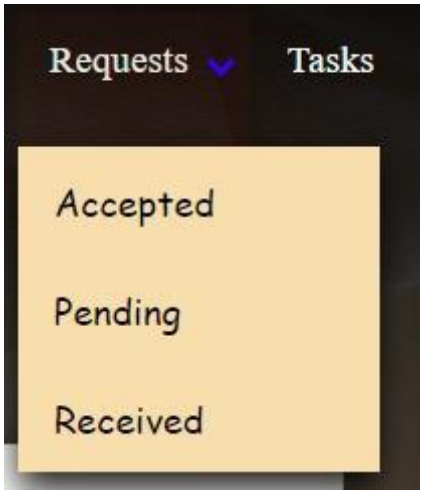


Figure 6: Request view

When the mentor wants to check Request he can lock through Accepted, Pending, or Received.

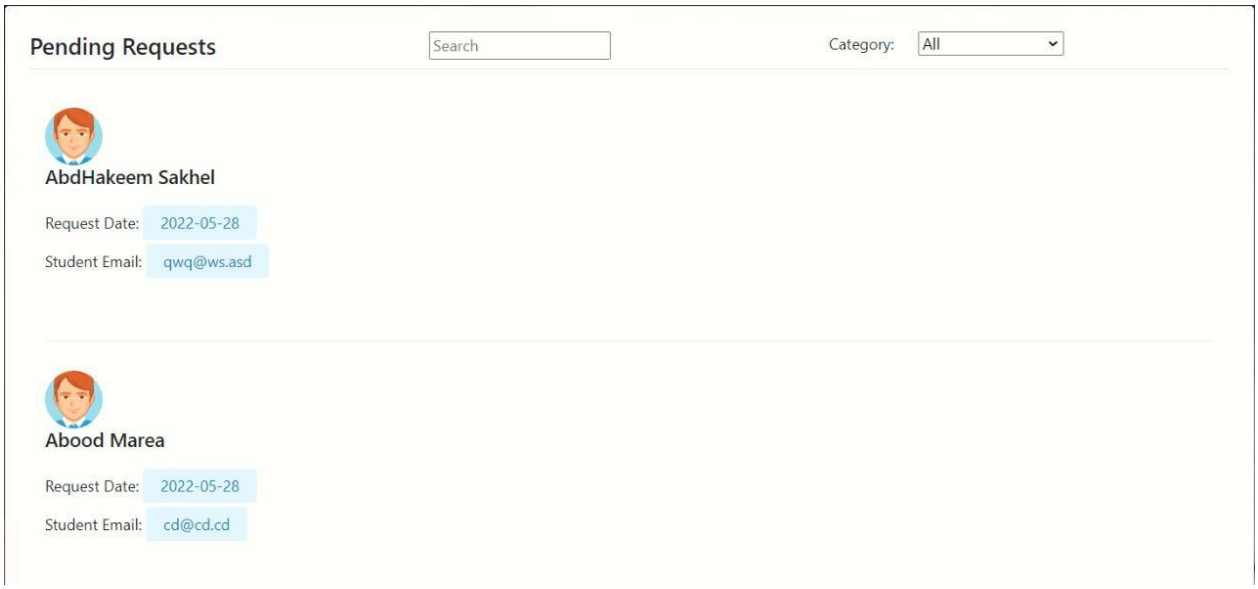


Figure 7: Pending request

First pending Requests are the request of the mentor that has been sent to the student and after the student accepts the request, they will be added to the accepted request.

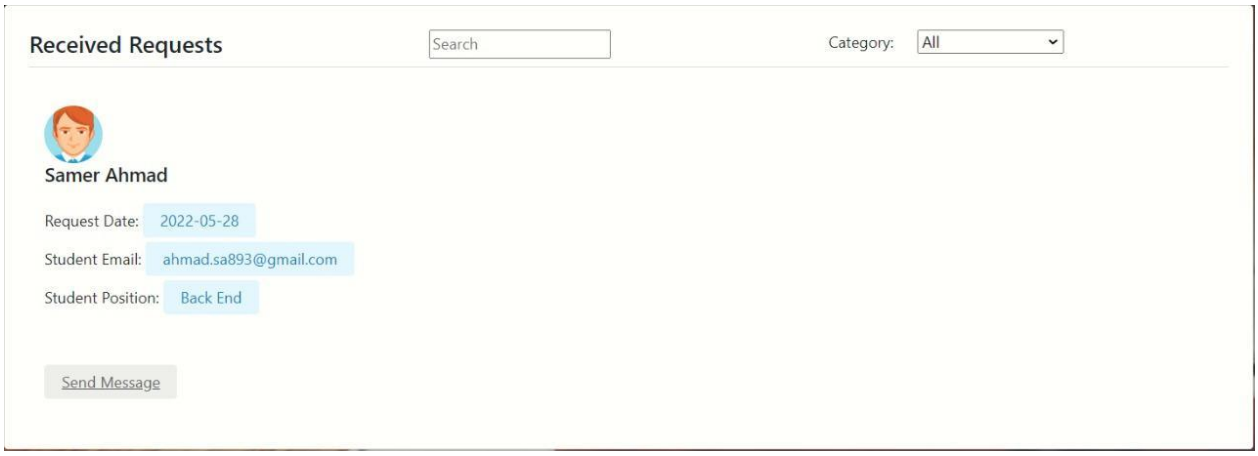


Figure 8: Received request

Secondly Received Requests: requests that have been sent by students to this mentor, and if the mentor accepted them, the student immediately become under his supervision, it will not be under the pending request.

Figure 9: Accepted request

Second Accepted requests: pending requests that have been accepted by students themselves, or received requests from the student accepted by the mentor will be added here.

Section 3-D: Task system for the mentor

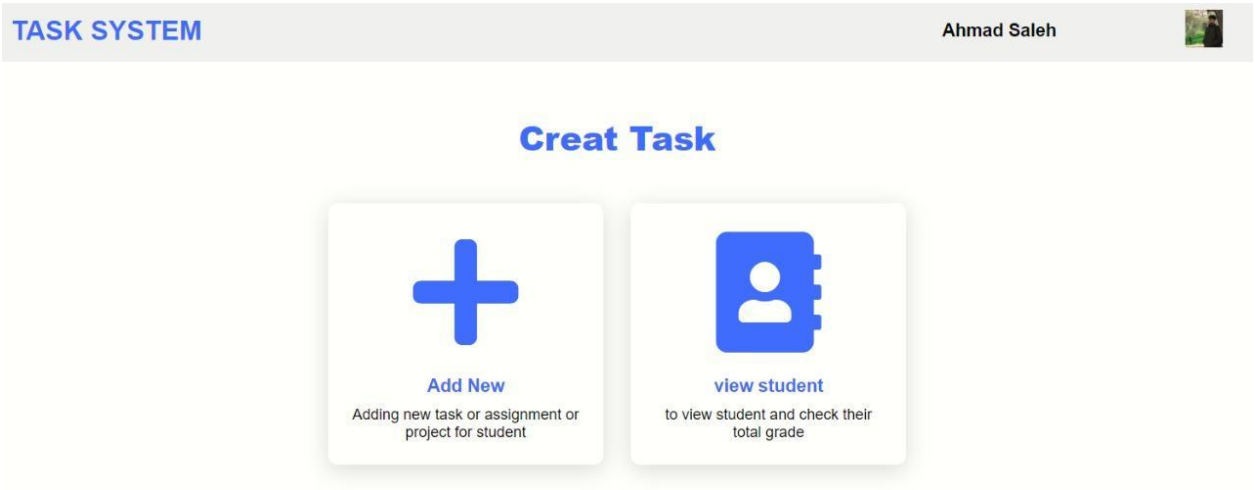


Figure 10: Task System

This is the task system for the mentors, here they can control tasks by adding, editing, or removing them, they can grade submitted tasks from students, and they can check the total student grades.

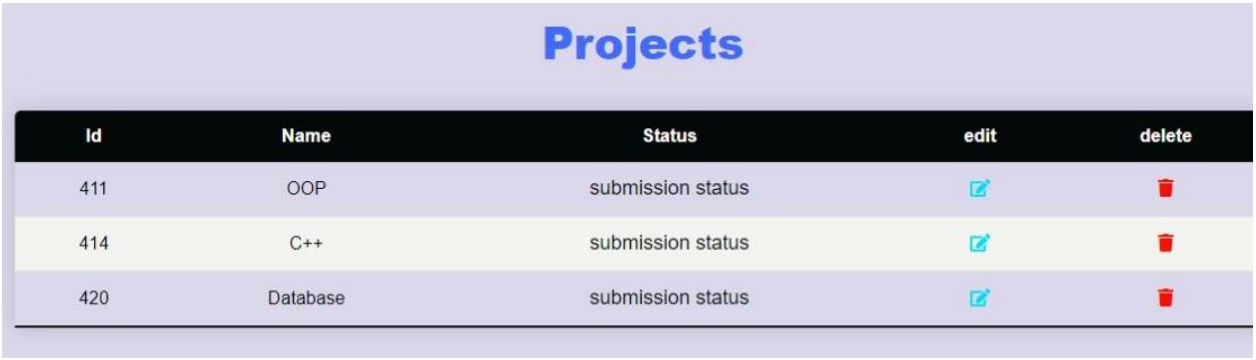


Figure 11: added project by mentors

These projects or tasks that have been added by mentors to students, you can delete one of them or edit some info, The info of the task are:

1. Id.
2. Name of the task.
3. The statues, ether they are submitted or not.

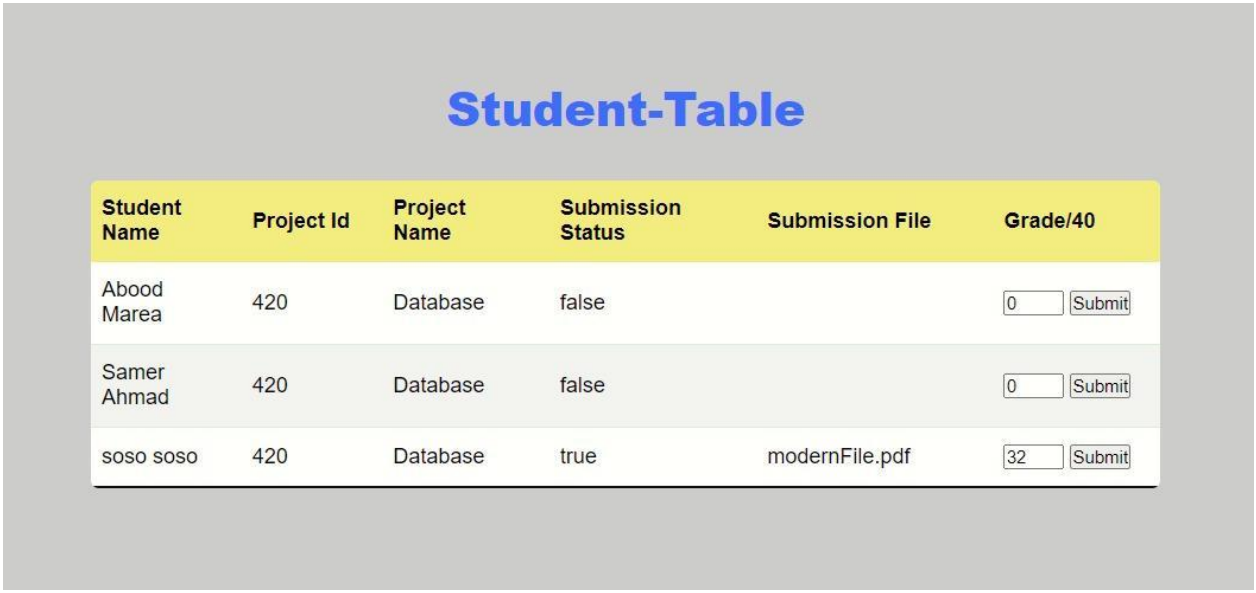


Figure 12: submission table from student

Here the mentor can see the student table for tasks, here the mentor can check his student task, to see who has submitted, and submit the grade of their solution for the tasks.

The info of the table:

1. Student name.
2. Project Id.
3. Project name.
4. Submission status.
5. Submission File.
6. Grade.

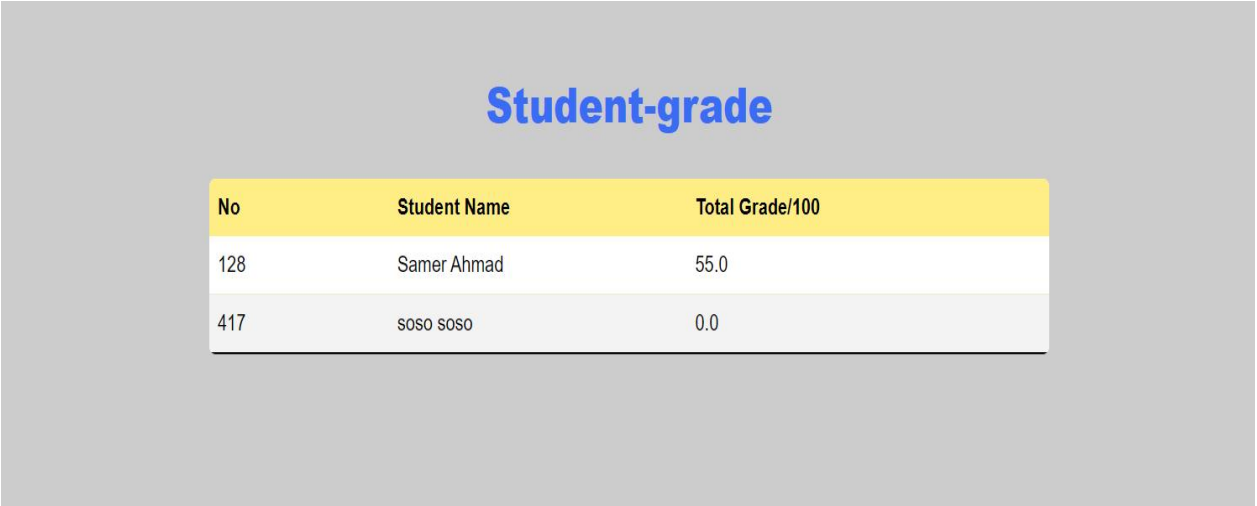


Figure 13: student-grade table from

Here the mentor can see the student table for total grade percentage, where the system calculate the percentage for all task for the student and display it In a table

The info of the table is:

1-The id of the student

2-Name of the student

3-The percentage of total grade for student

Section 4-A: home page

After the student login, he is greeted with a welcoming page where he can either go to his Account, My Network, Request, or tasks, through the header, if you scroll down you will see the mentor control there they can mentor an available

new student, received requests from the students you send application and stop mentoring student.

Figure 14: student home page

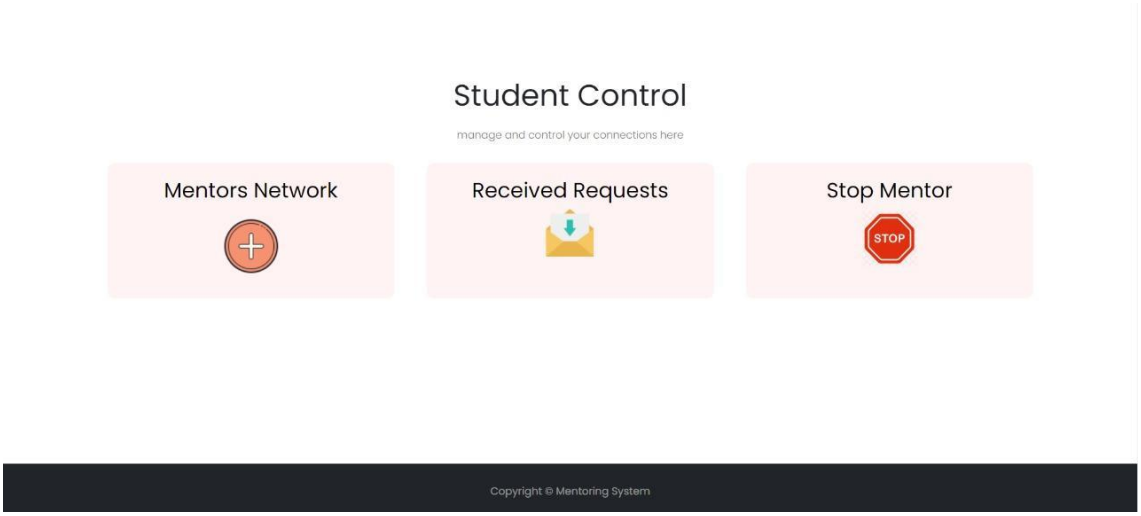


Figure 15: student Control

Section 4-B: Registration form

Figure 16: Registration form

This form is filled out by students after login to examine which position most likely to be in. subjects are different from specialist to a specialist but the others are the same.

The first form of content is the university subject for example advanced programming, android developing image proccing, or AI, etc...

The second form asks about courses you take such as full-stack courses, software testing, Laravel courses, web design, etc....

The final form is the programing language you have a good understanding for instance C#, CSS, Html, JavaScript, etc....

After the student fills these forms the algorithm calculates the four majors, which are backend, frontend, full-stack, and

quality assurance, numbers between 0 and 1, with the highest one set as their major.

Section 4-C: Request Network view

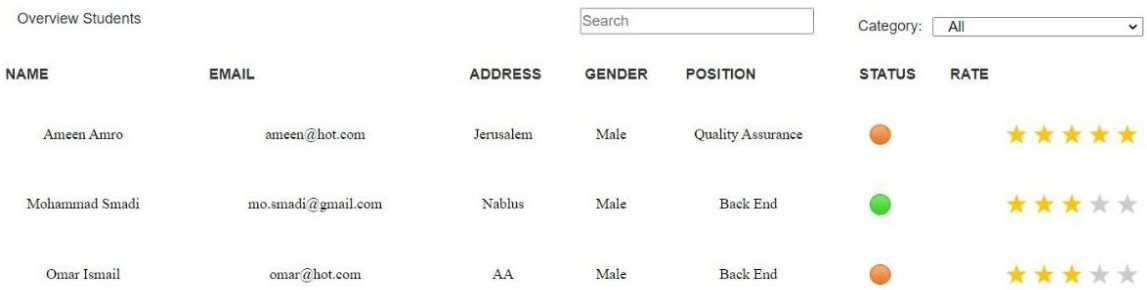


Figure 17: student Network view

This is the network used by students to review the mentors available in the network sorted depending on their rates, the rate is submitted by students who were under their supervision, and it displays the average rate of the mentor, on this view the mentor info that is displayed is:

1. Name.
2. Email.
3. Address.
4. Gender.
5. Position.
6. Statues.
7. Rate.

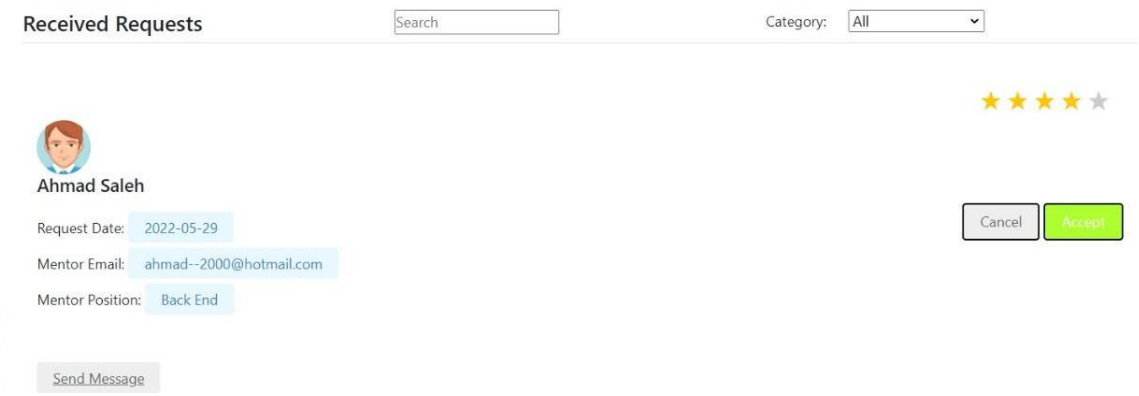


Figure 18: received request

These are the received request, that was sent by the mentor to the student, a student can receive many requests, not only that he can accept or deny the request, although he can't accept more than one mentor after he chooses, the student is supervised under the mentor he chooses, and the rest of the request from other mentors are removed.

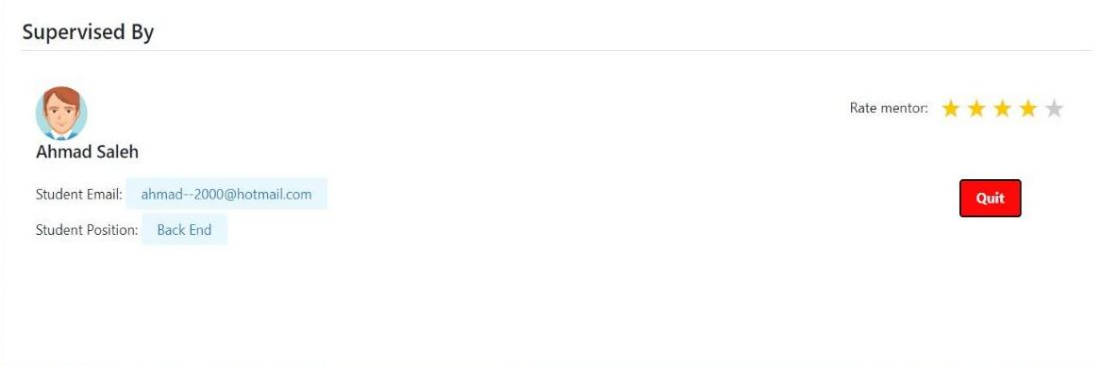


Figure 19: student supervisor

Here the student can rate his mentor or quit being under their supervision, and show the mentor they are under.
The students also can send a request to the mentors, through pending request tabs, in this situation, the mentor can accept or deny the request for supervision, and the student can cancel the request or resend it.

Section 4-D: Task system for the student

When the student opens the tasks he's welcomed with the student task, displayed on the screen with the id and the name of the project, that meteor, when you click the tasks you go to its submission

status.

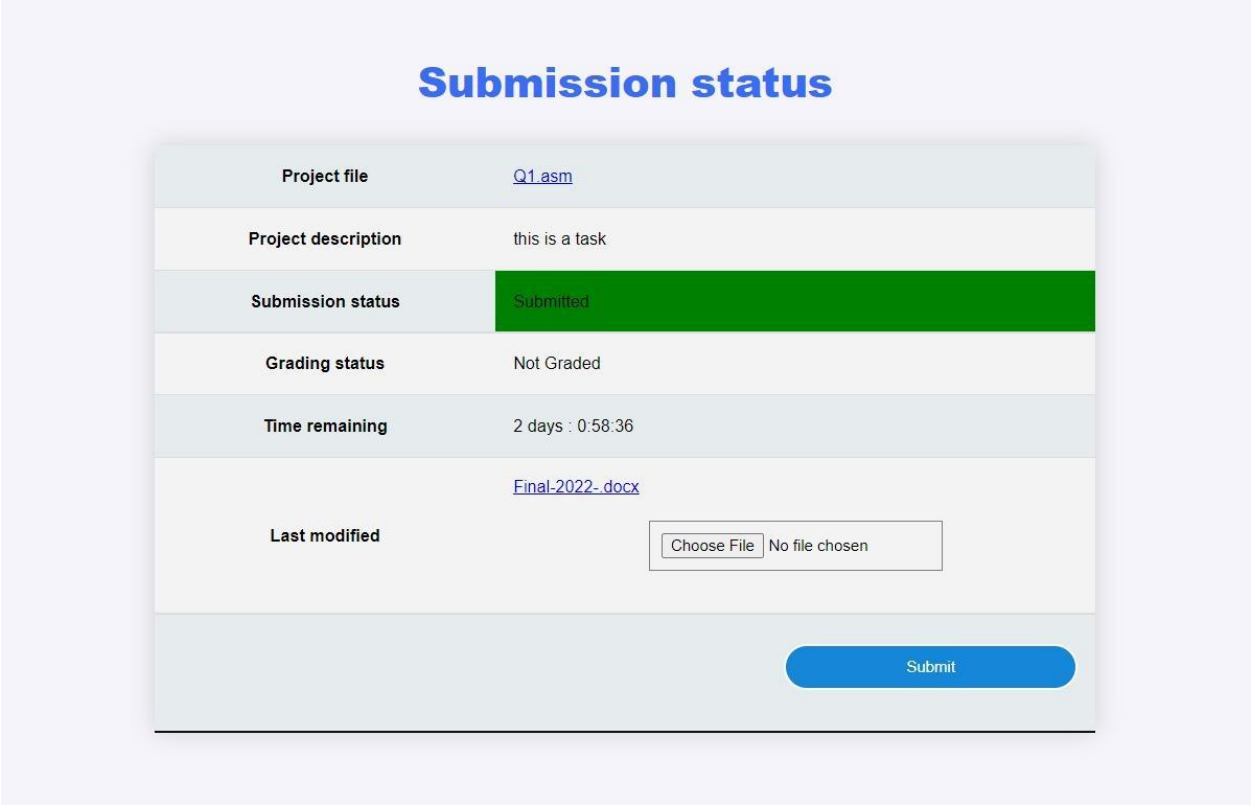


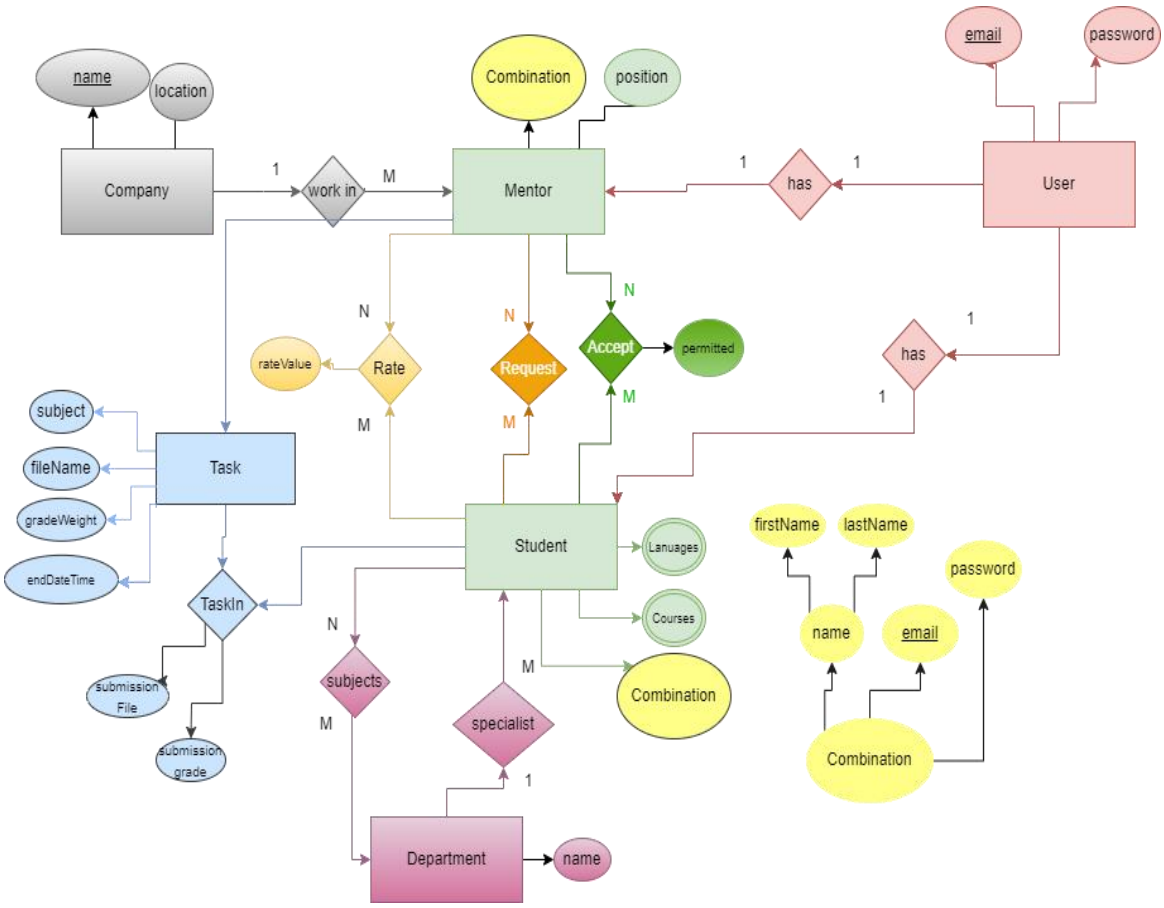
Figure 19: submission status

Here the student can review the task and submit their file solution, when the student doesn't submit the file before the deadline, the submit button will not work.

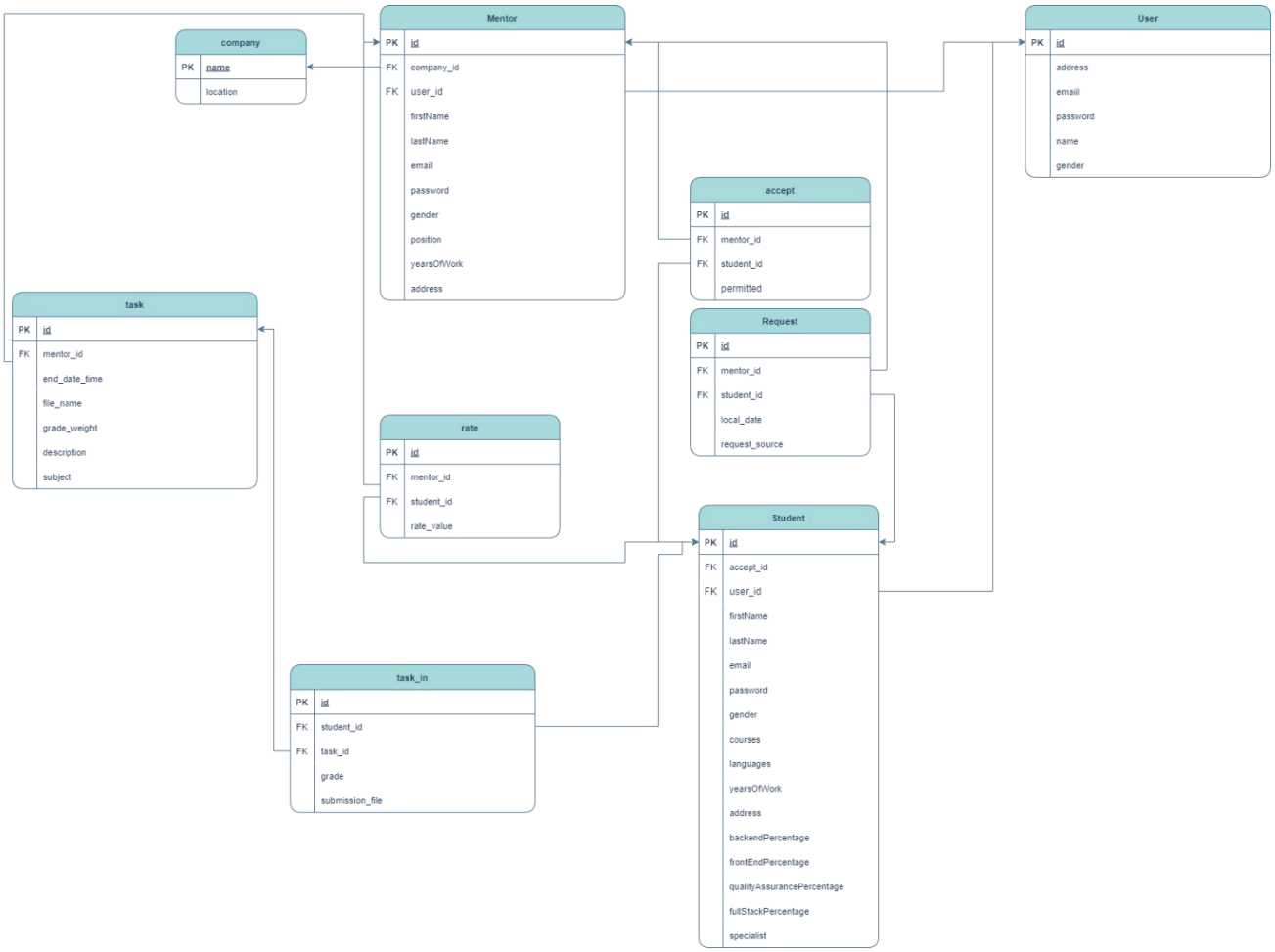
CHAPTER FIVE

DIAGRAMS

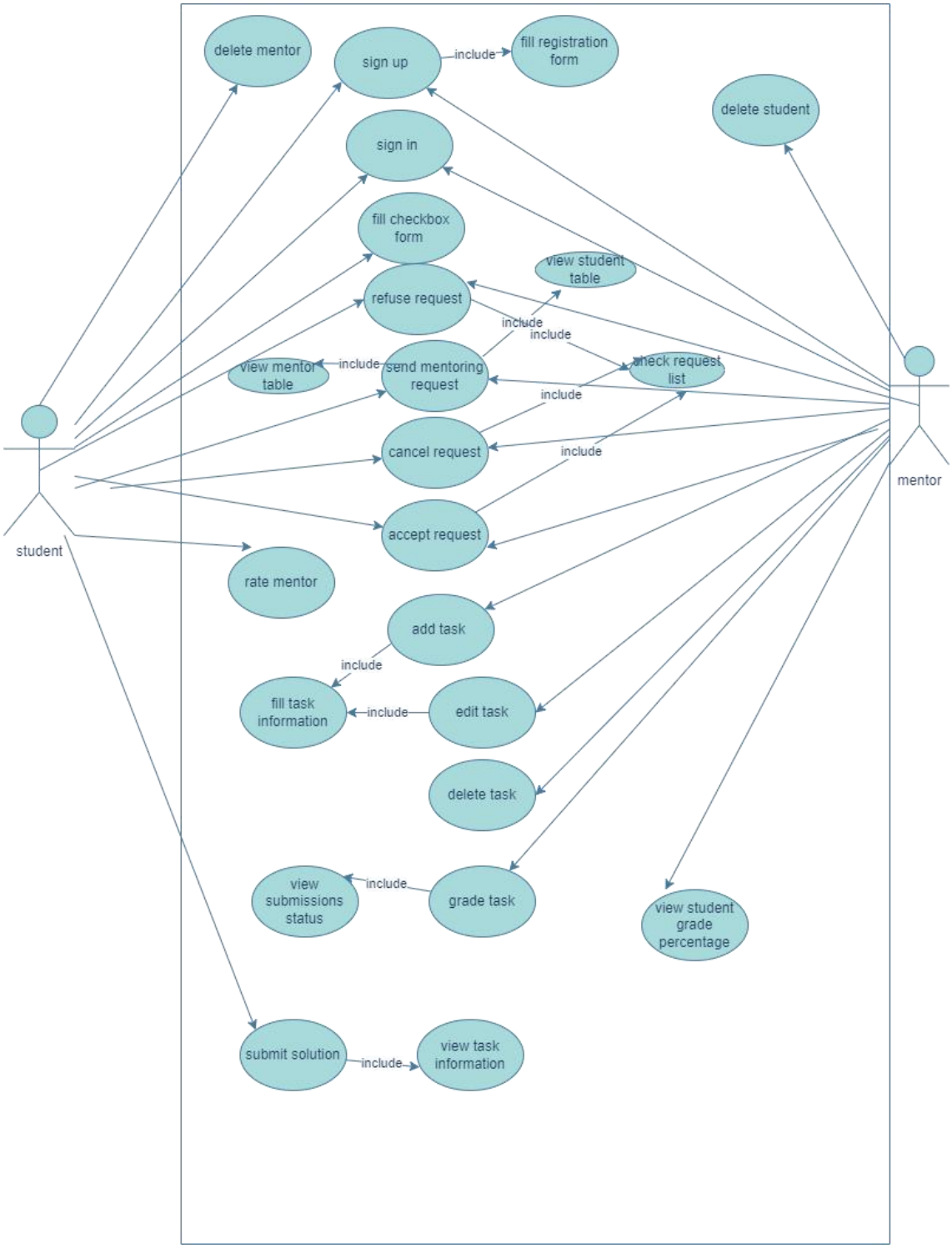
A-EER- Diagram



B-Schema

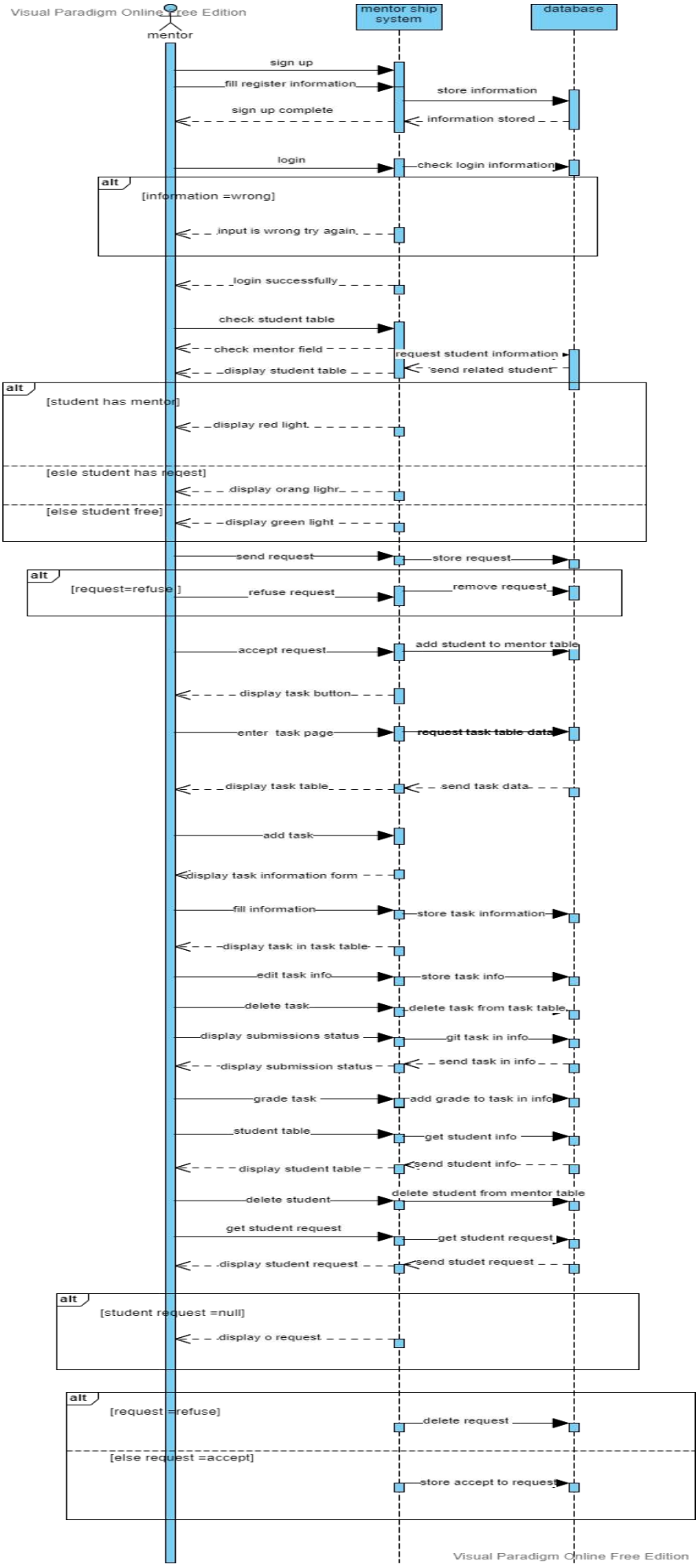

company
PK name
location
Mentor
PK id
FK company_id
FK user_id
firstName
lastName
email
password
gender
position
yearsOfWork
address
User
PK id
address
emaill
password
name
gender
accept
PK id
FK mentor_id
FK student_id
permitted
task
PK id
FK mentor_id
end_date_time
file_name
grade_weight
description
subject
Request
PK id
FK mentor_id
FK student_id
local_date
request_source
rate
PK id
FK mentor_id
FK student_id
rate_value
Student
PK id
FK accept_id
FK user_id
firstName
lastName
email
password
gender
courses
languages
yearsOfWork
address
backendPercentage
frontEndPercentage
qualityAssurancePercentage
fullStackPercentage
specialist
task_in
PK id
FK student_id
FK task_id
grade
submission_file

C-Use case

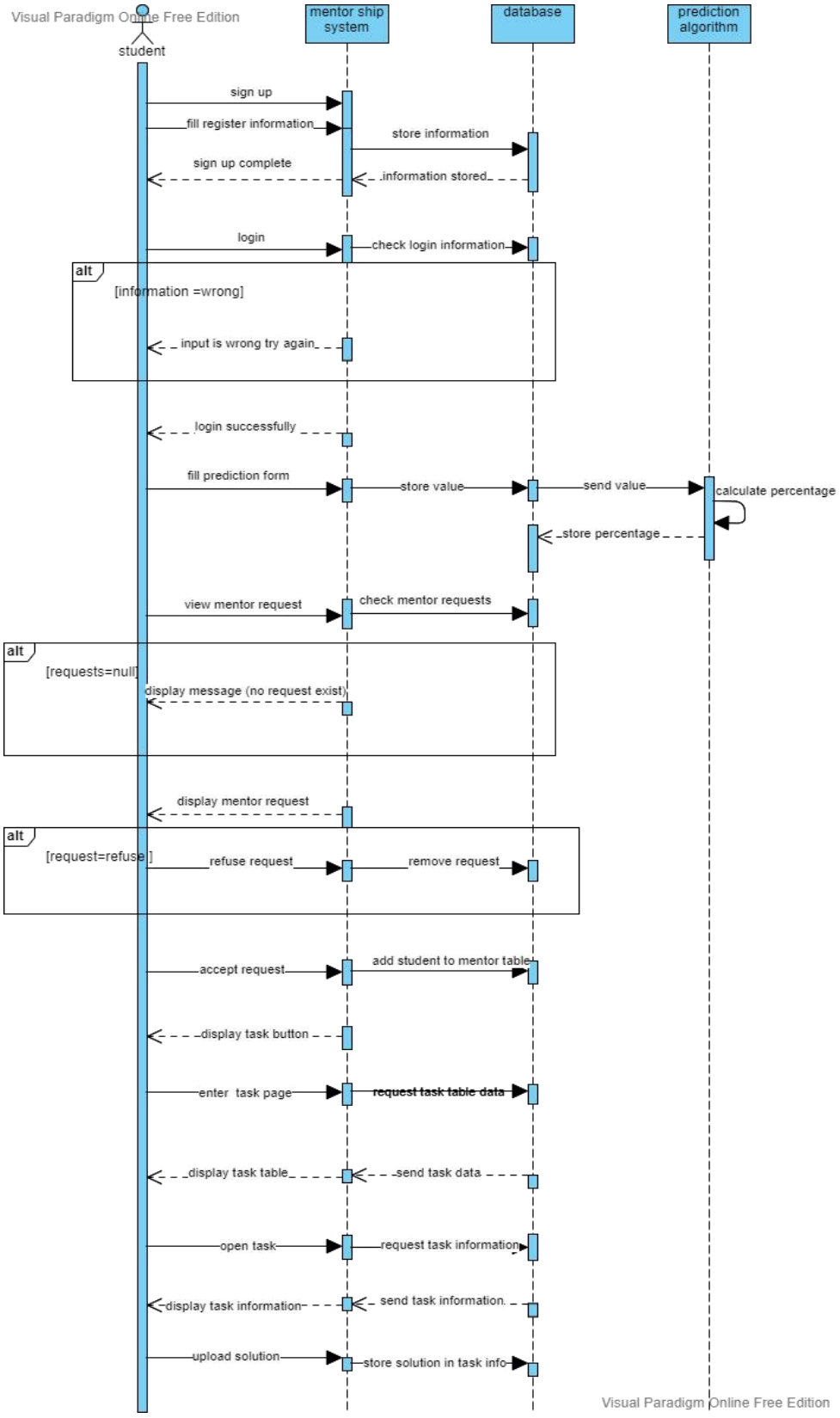


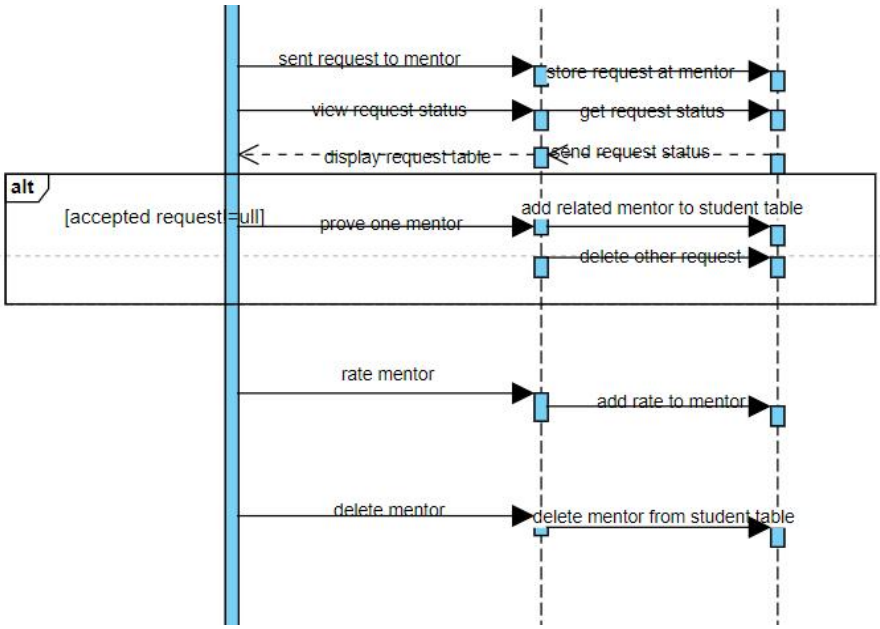
D-Sequence diagram

1- Mentor

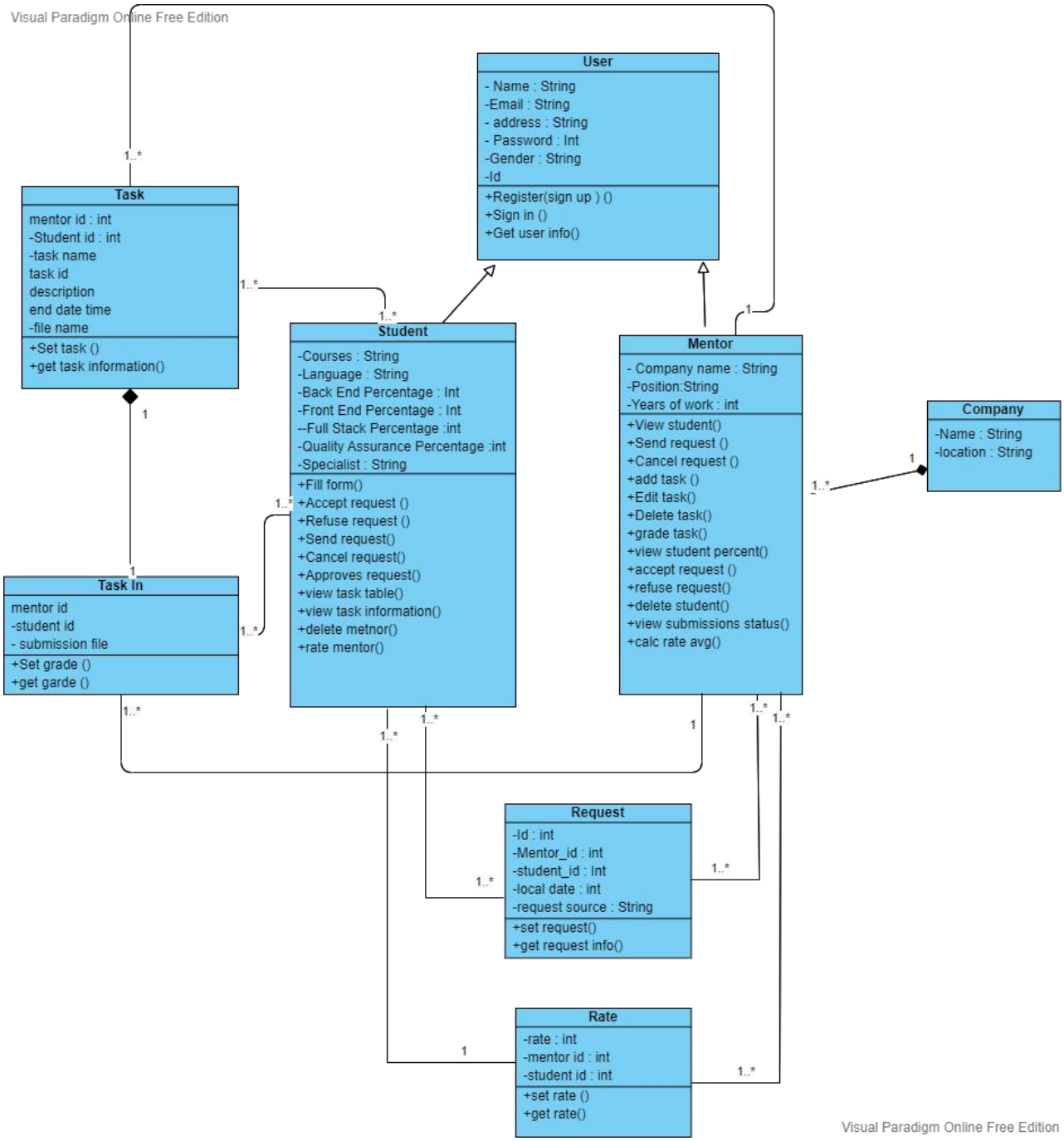

Visual Paradigm Online Free Edition
mentor
mentor ship system
database
sign up
fill register information
store information
sign up complete
information stored
login
check login information
alt
[information =wrong]
input is wrong try again
login successfully
check student table
check mentor field
request student information
display student table
send related student
alt
[student has mentor]
display red light
[esle student has request]
display orang lighr
[else student free]
display green light
send request
store request
alt
[request=refuse]
refuse request
remove request
accept request
add student to mentor table
display task button
enter task page
request task table data
display task table
send task data
add task
display task information form
fill information
store task information
display task in task table
edit task info
store task info
delete task
delete task from task table
display submissions status
git task in info
display submission status
send task in info
grade task
add grade to task in info
student table
get student info
display student table
send student info
delete student
delete student from mentor table
get student request
get student request
display student request
send studet request
alt
[student request =null]
display o request
alt
[request =refuse]
delete request
[else request =accept]
store accept to request
Visual Paradigm Online Free Edition

2- Student


Visual Paradigm Online Free Edition
student
mentor ship system
database
prediction algorithm
sign up
fill register information
store information
sign up complete
information stored
login
check login information
alt
[information =wrong]
input is wrong try again
login successfully
fill prediction form
store value
send value
calculate percentage
store percentage
view mentor request
check mentor requests
alt
[requests=null]
display message (no request exist)
display mentor request
alt
[request=refuse]
refuse request
remove request
accept request
add student to mentor table
display task button
enter task page
request task table data
display task table
send task data
open task
request task information
display task information
send task information
upload solution
store solution in task info
Visual Paradigm Online Free Edition


sent request to mentor
store request at mentor
view request status
get request status
display request table
send request status
alt
[accepted request != null]
prove one mentor
add related mentor to student table
delete other request
rate mentor
add rate to mentor
delete mentor
delete mentor from student table

E-Class Diagram



Chapter 6: Conclusion

Future work :

- Improve the algorithm get more accurate recommendation
- Support reading pdf file to get the grade of student in university course (اجباري التخصص)

  To add it to the algorithm
- Adding the ability to add quiz from mentor to student
- Adding more major to the system not only it major.

Thanks