Software Requirements and Design Document: **Rate My Course**

Group 102: Bowen Cheng, Phaneendra Amruthur Ravi, Venkata Sai Srikar Jilla, Yifan Feng, Yifan Wang, Wenjing Xu

1. Software Requirements:

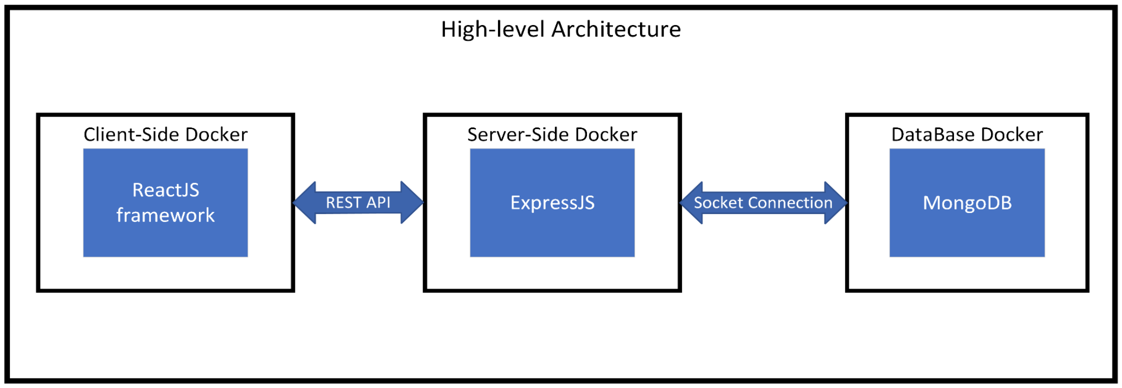
- Development Requirements:
    - NodeJS: for Complying JavaScript outside browser
    - ExpressJS: JavaScript Server
    - ReactJS: Client-side Javascripting
    - MonoDB: NoSQL DB for saving data
    - Docker: containerization of each microservice

- Tool Requirements:
    - IDE: for Rapid development
    - PostMan: for Rapid API development and testing
    - GitHub: Version control
    - Vite: Bundling UI
    - MailRocketAPI: To send emails for verification purpose

2. System Architecture

2.1. Architectural Design

As shown in this diagram, we use the ReactJS framework for the front-end and the ExpressJS framework for the back-end. The front-end communicates with the back-end through REST API calls, which allows for a clear separation between Front-End and Back-End. The back-end is connected to a MongoDB database to fetch and manage data. Both the front-end, back-end, and database are deployed inside Docker containers, ensuring cross-platform compatibility and deployment.



2.2. Component Design

2.2.1. Client-side Docker

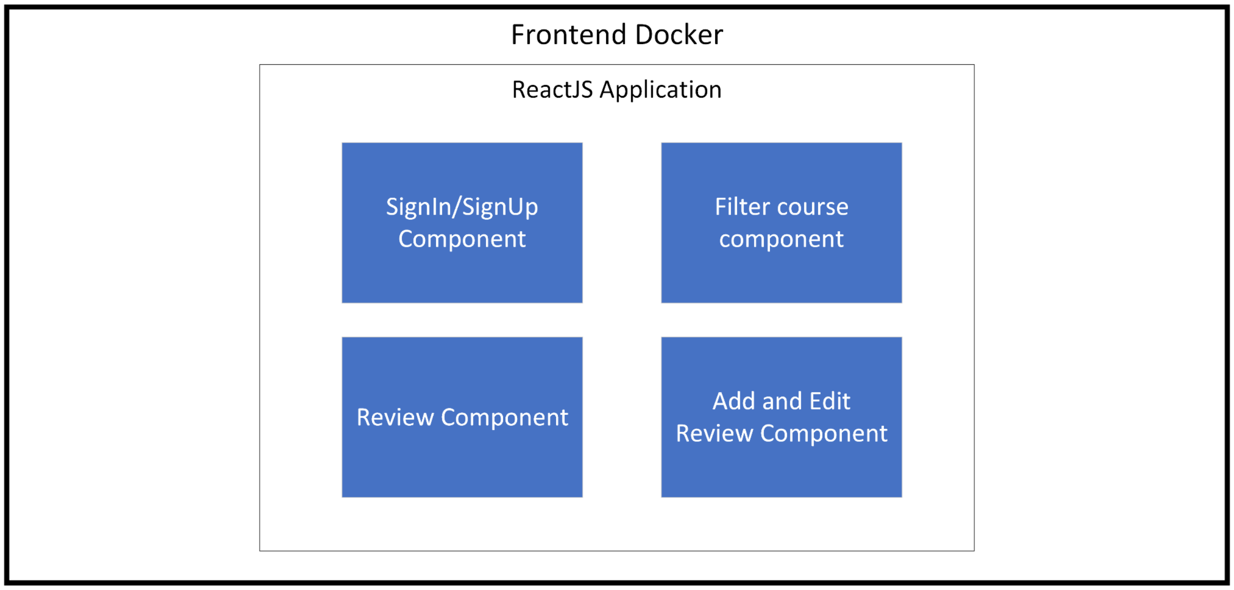
The front-end mainly consists of four components.

1. The first one is the SignIn/SignUp component which students can use their campus email to register or login to the system.

2. The second one is Filter Course Component which allows students to select various prerequisites for filtering courses. For example, students can choose to display only courses worth two credits, only courses in the computer science major, or only graduate courses. They can even combine these filters to achieve advanced search and filtering functionality,

3. The third one is Review Component which can be used to view reviews for each course, which accurately reflect students' evaluations of the class.

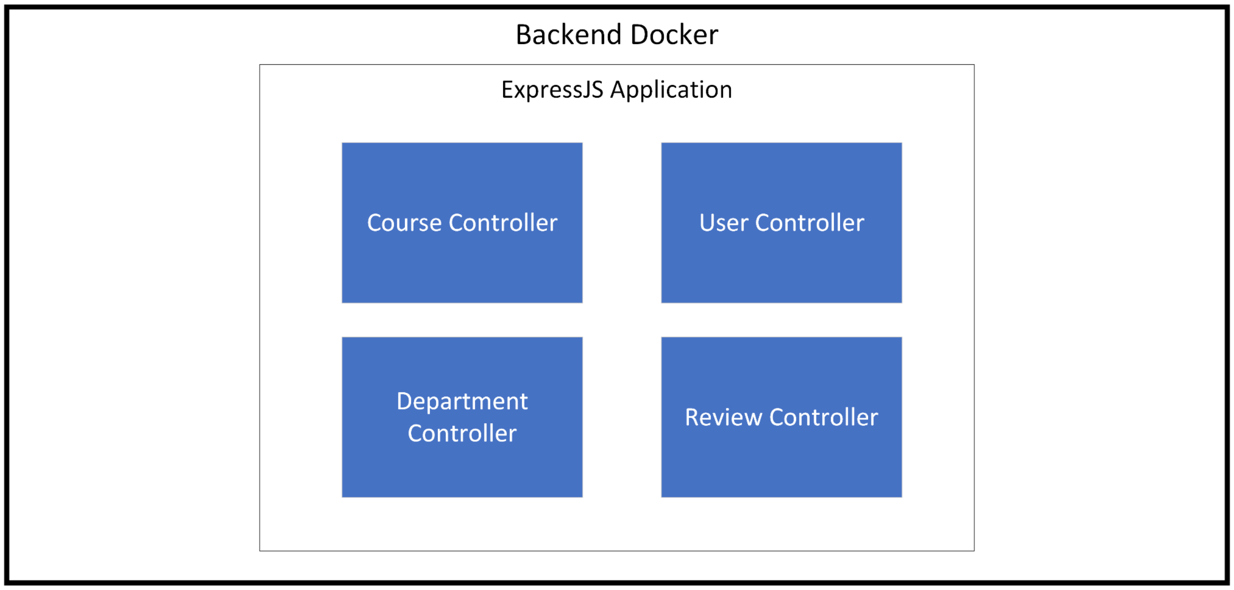
4. The fourth one is Add and Edit Review Component which allows students to edit and submit their course evaluations.



2.2.2. Server-side Docker

In the back-end, we have adopted the MVC (Model-View-Controller) pattern, which consists of three components: Model, View, and Controller.
We have four Controllers, which are Course Controller,User Controller,Department Controller and Review Controller.

1. Course Controller:Responsible for adding and deleting course operations.
2. User Controller:Responsible for user login, registration, and course enrollment operations.
3. Department Controller:Responsible for managing courses across different majors.
4. Review Controller:Responsible for adding and editing course review operations.



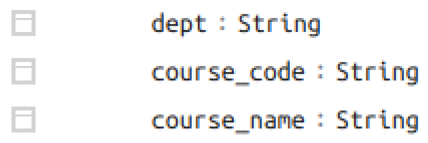
2.2.3. Database Design

We are using MongoDB, a NoSQL DB for our software, because our data needs no structure and we can incorporate rapid prototyping as we can add new fields and not worry about structure.

Users Collection Schema

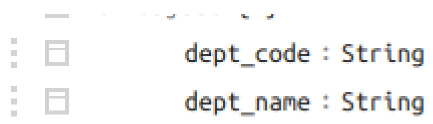
- ☐ `email : String`
- ☐ `date_created : DateObject`
- ☐ `hash_password : String`

Course Collection Schema



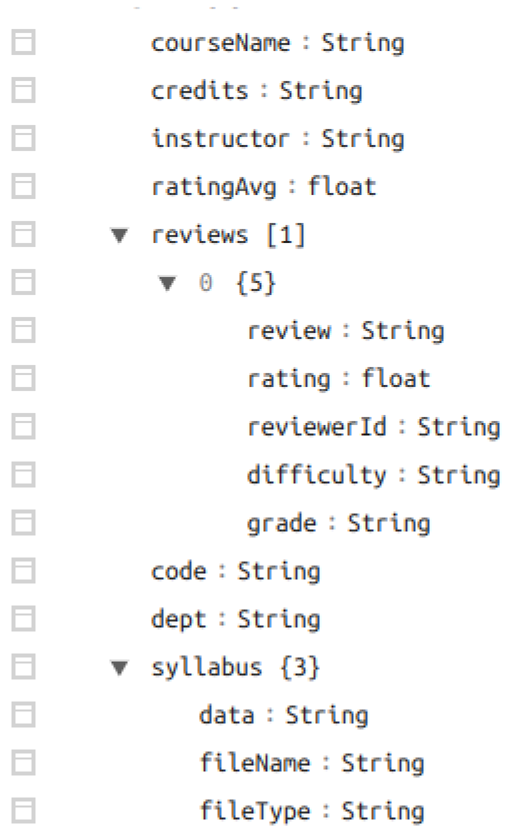
```
dept : String
course_code : String
course_name : String
```

Department Collection Schema



```
dept_code : String
dept_name : String
```

Review Collection Schema



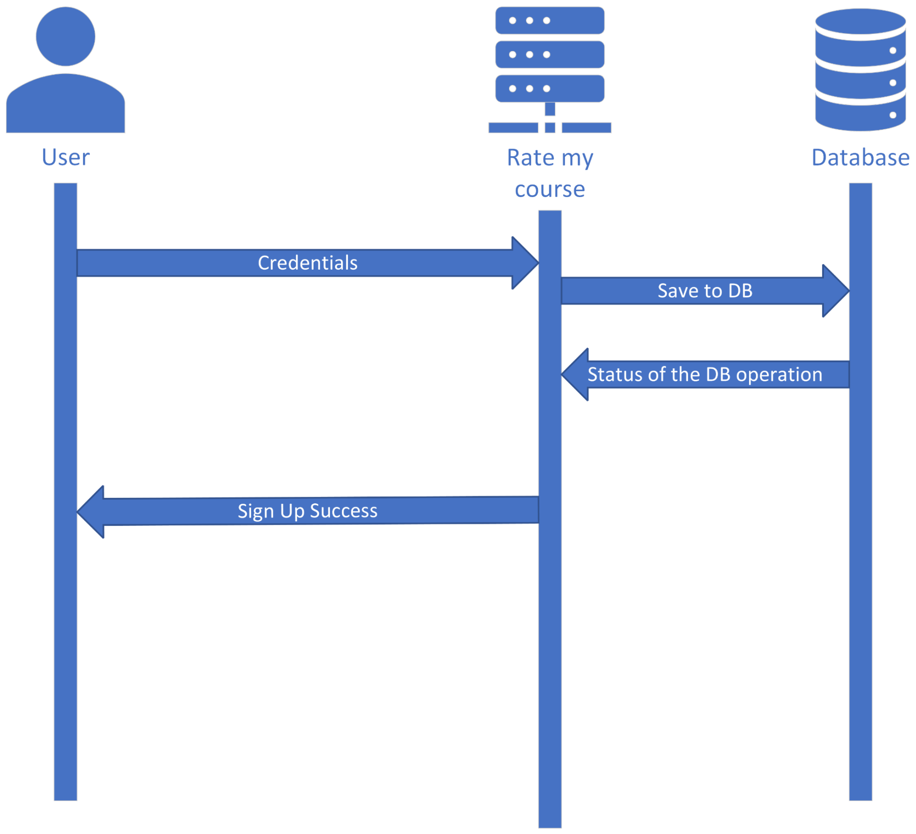
```
courseName : String
credits : String
instructor : String
ratingAvg : float
▼ reviews [1]
    ▼ 0 {5}
        review : String
        rating : float
        reviewerId : String
        difficulty : String
        grade : String
code : String
dept : String
▼ syllabus {3}
    data : String
    fileName : String
    fileType : String
```

3. Detailed Design

3.1. Sequence Diagrams

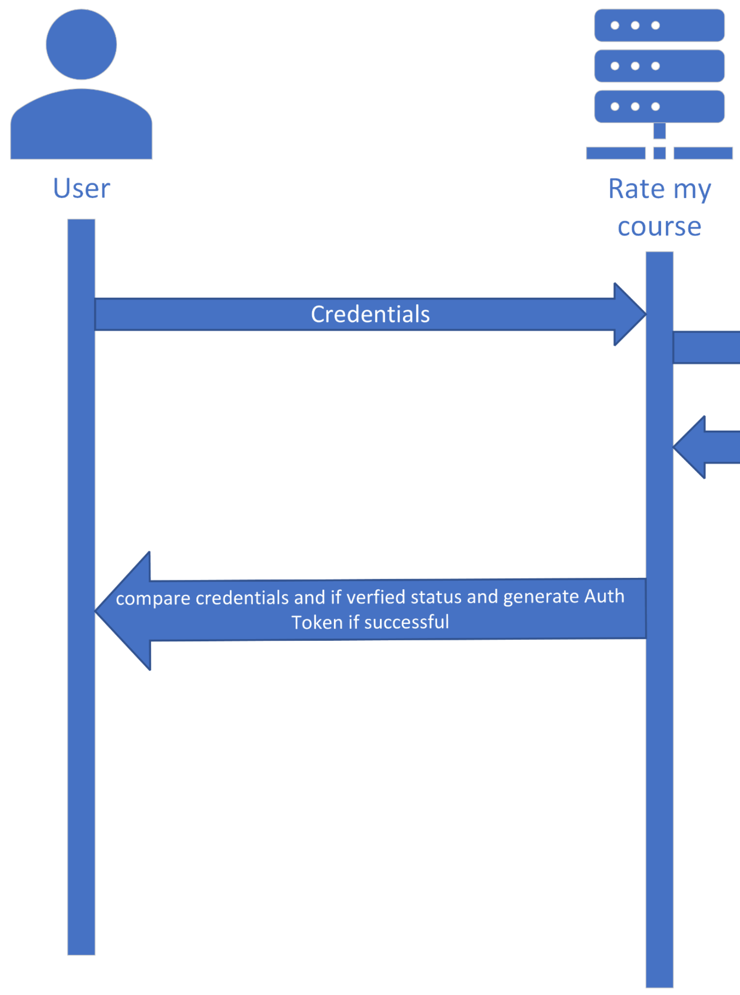
3.1.1. SignIn/SignUp

Sign Up



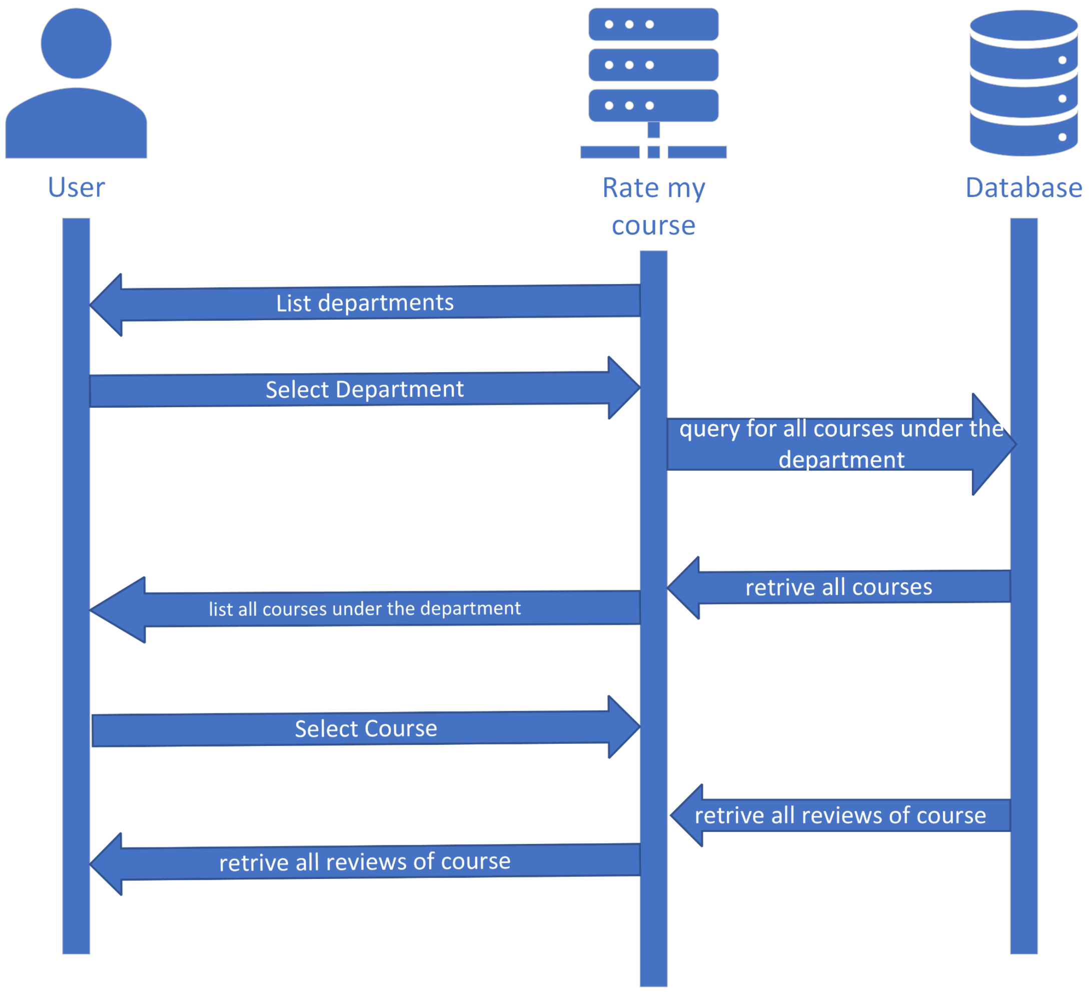
As shown in the diagram, users interact with the front-end interface to call the back-end API, sending their credentials (usually email and password) to the back-end. The back-end then stores these credentials into the database. Afterward, the back-end returns a boolean value to the front-end, indicating whether the registration was successful or not.

Sign In


User
Rate my course
Database
Credentials
ask for user object
retrive user object
compare credentials and if verfied status and generate Auth Token if successful

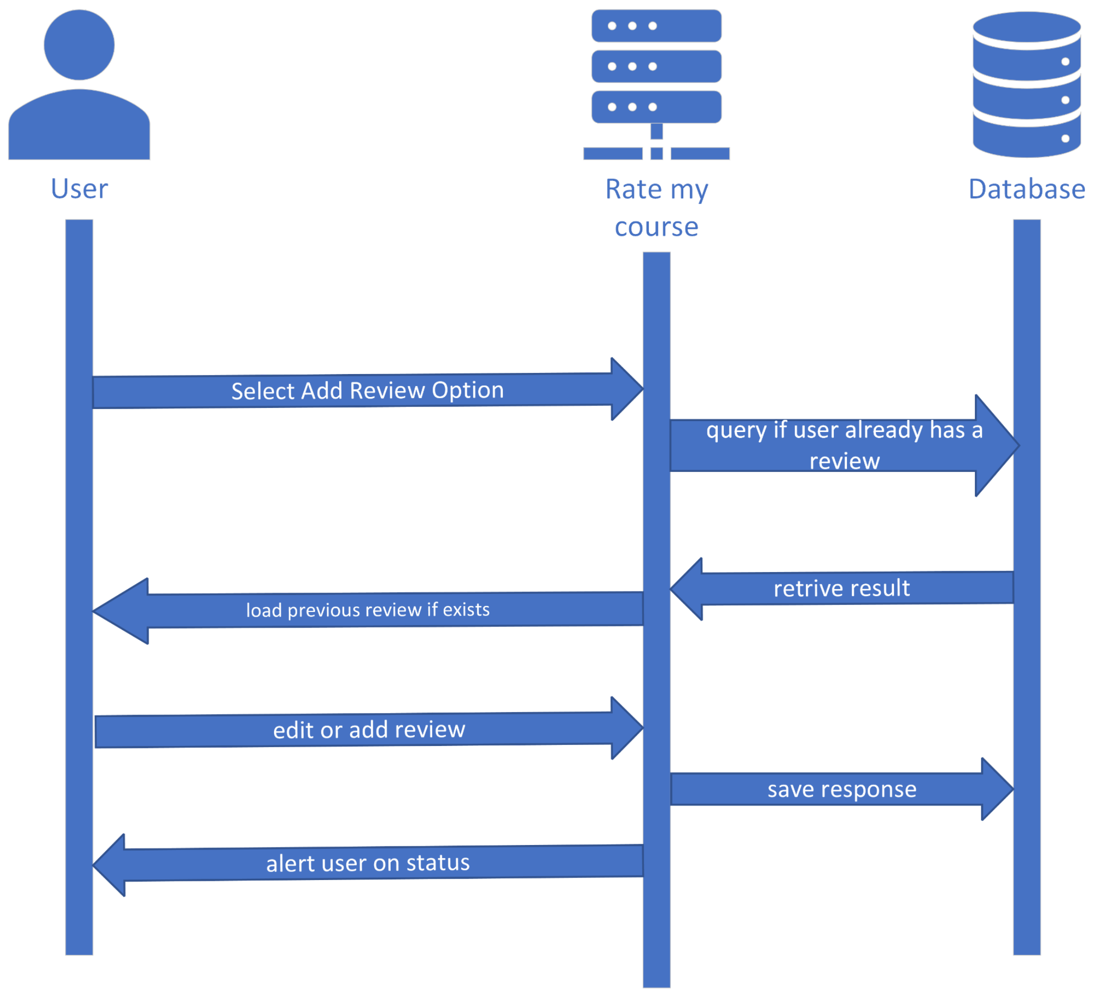
Users input their credentials (email and password) on the front-end login interface, which are then sent to the back-end. The back-end searches the database using the email to find the corresponding user object. The user object is retrieved from the database, and the password is compared to the stored one. If the passwords match, the login is successful; otherwise, the login fails.

3.1.2. Filter


User
Rate my course
Database
List departments
Select Department
query for all courses under the department
retrive all courses
list all courses under the department
Select Course
retrive all reviews of course
retrive all reviews of course

After users log into the system, a list of departments is displayed for them to choose from. Once a department is selected, the system presents all the courses offered by that department. When users choose a specific course, the system displays all the reviews related to that course.

3.1.3. Add Review

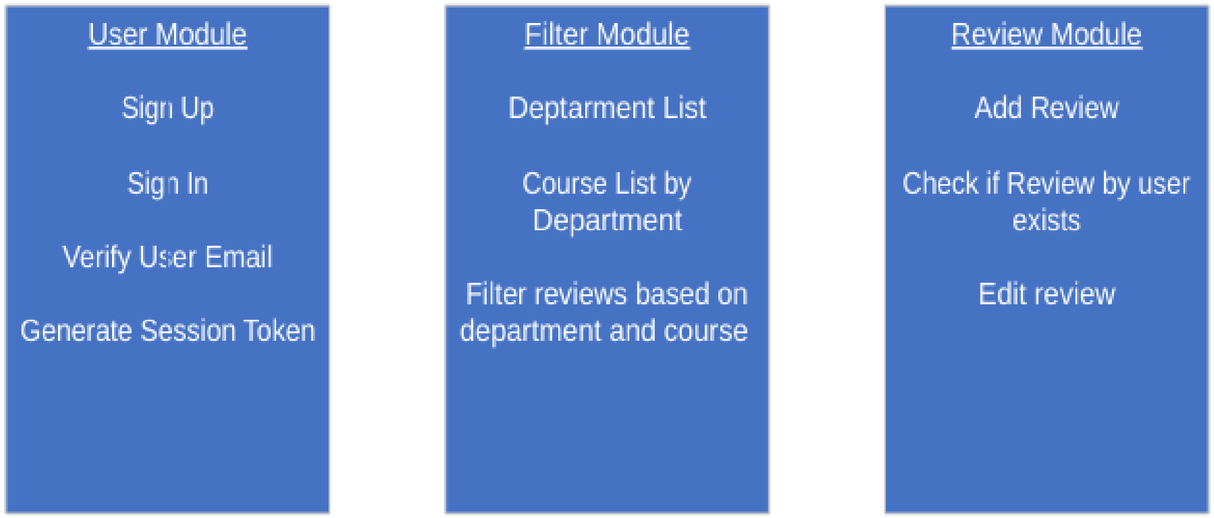


When users click the 'Add Review' button, the back-end sends a request to the database to check if the user has previously reviewed the course. If a review exists, the database returns the previous review, allowing the user to edit it based on the existing content. Once the user finishes editing, they can save the changes. The updated review is then stored in the database, and the system returns the operation result.

3.2. Modules Design

The functionalities of the application have been modularized into 3 modules.

- User Module
- Filter Module
- Review Module



3.2.1. User Module

The module is responsible for the functionalities of Sign Up, Sign In, Verify user's SCU email, and generate session token. This includes the UI section which is the Sign In/Up page, and the corresponding APIs on the backend which handle mongodb transactions.

3.2.2. Filter Module

The module is responsible for the functionalities of the Department List population, get course list by department, and filter reviews based on the department and course selection. This includes the UI section which is the Filter page, and the corresponding APIs on the backend which handle mongodb transactions.

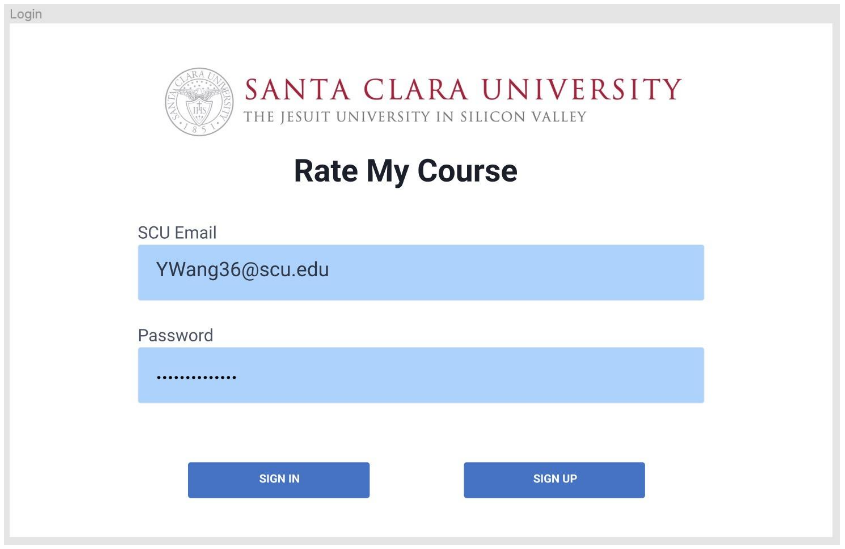
3.2.3. Review Module

The module is responsible for the functionalities of the Add Review, Check if review by a user exists and Edit review. This includes the UI section which is the Reviews Page, and the corresponding APIs on the backend which handle mongodb transactions.

3.3. User Interface Design

Here is an elaboration of the user interface designs that we have created for Rate My Courses, utilizing the powerful design tool Figma to meticulously craft each element for an optimal user experience.

- Easy to use and navigate User Interface for signing in and registration for the user to use. Show as Fig 1.

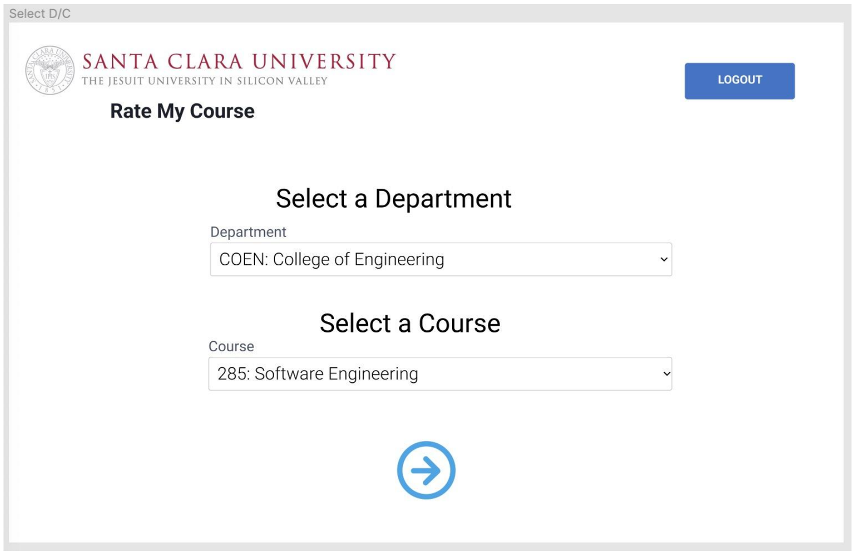


SCU Email: YWang36@scu.edu

Password: ....

Fig 1: Sign In/Up

- The User Interface for selecting department and course for filtering the courses. Show as Fig 2.

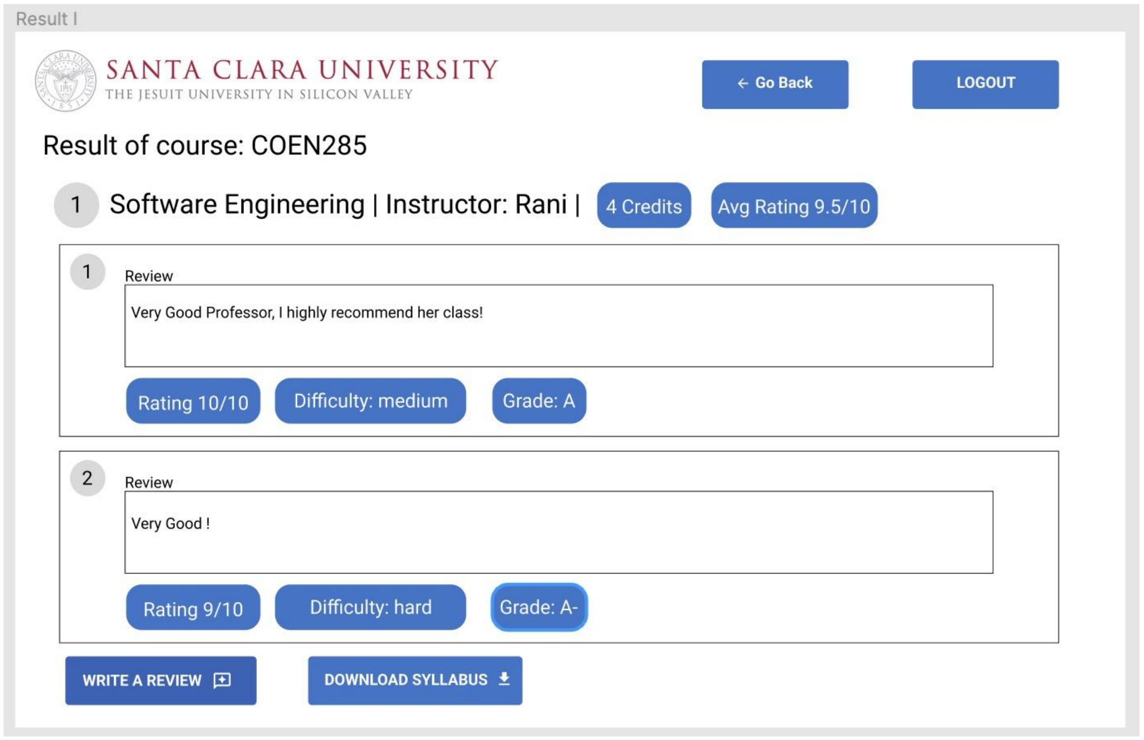


Department: COEN: College of Engineering

Course: 285: Software Engineering

Fig 2: Course filter

- Easy to navigate and view the reviews of the selected course by the user. Show as Fig 3 and fig 4.



Result I

SANTA CLARA UNIVERSITY
THE JESUIT UNIVERSITY IN SILICON VALLEY

Result of course: COEN285

1 Software Engineering | Instructor: Rani | 4 Credits | Avg Rating 9.5/10

1 Review

Very Good Professor, I highly recommend her class!

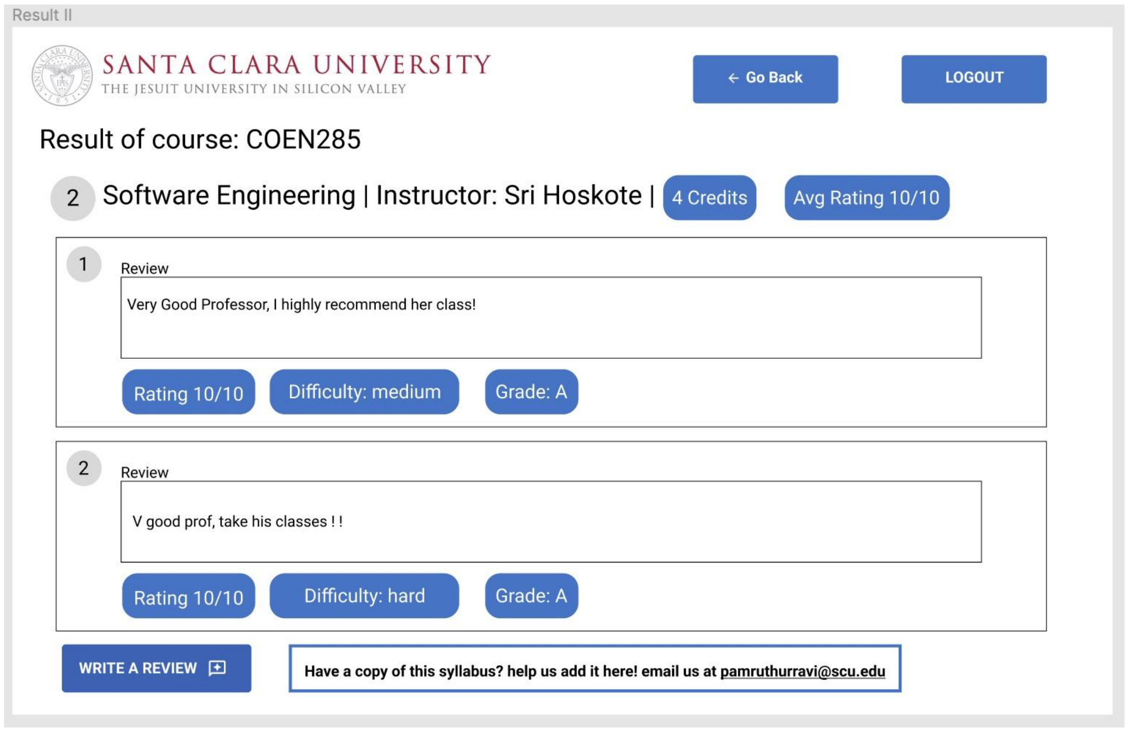
Rating 10/10 | Difficulty: medium | Grade: A

2 Review

Very Good !

Rating 9/10 | Difficulty: hard | Grade: A-

Fig 3: Review List page (for professor Rani)



Result II

SANTA CLARA UNIVERSITY
THE JESUIT UNIVERSITY IN SILICON VALLEY

Result of course: COEN285

2 Software Engineering | Instructor: Sri Hoskote | 4 Credits | Avg Rating 10/10

1 Review

Very Good Professor, I highly recommend her class!

Rating 10/10 | Difficulty: medium | Grade: A

2 Review

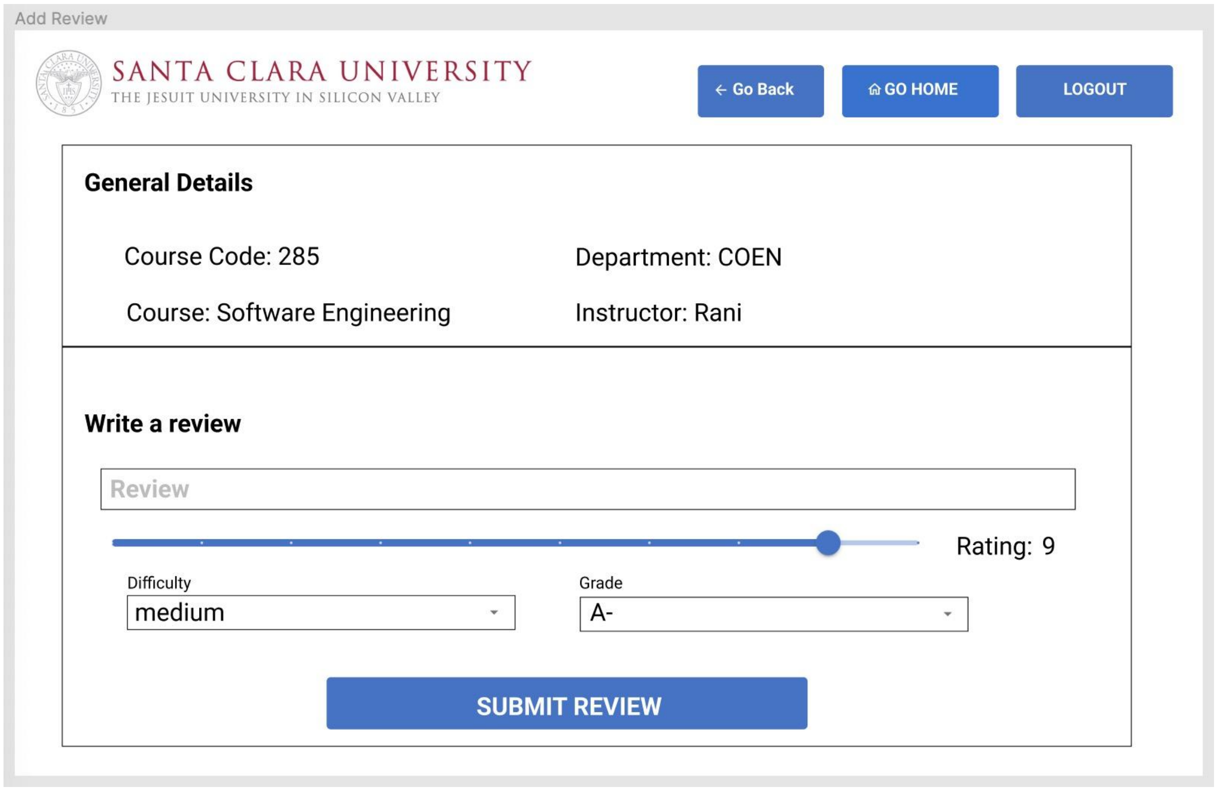
V good prof, take his classes ! !

Rating 10/10 | Difficulty: hard | Grade: A

Have a copy of this syllabus? help us add it here! email us at pamruthurravi@scu.edu

Fig 4: Review List page (for professor Hoskote)

- User Interface to add reviews. Show as fig 5.


Add Review

SANTA CLARA UNIVERSITY
THE JESUIT UNIVERSITY IN SILICON VALLEY

General Details

Course Code: 285

Department: COEN

Course: Software Engineering

Instructor: Rani

Write a review

Review

Rating: 9

Difficulty: medium

Grade: A-

Fig 5: Add Review