Design Document

for

SmarTutor

Version 0.1

Prepared by

Group #: 16 **Group Name: Software Avengers**

Sonu Kumar	**211052**	**sonuk21@iitk.ac.in**
Ch Hemanth Kumar	**210277**	**chandaka21@iitk.ac.in**
Sarthak Paswan	**220976**	**sarthakp22@iitk.ac.in**
Yash gothwal	**211189**	**yashg21@iitk.ac.in**
Kantule Ritesh Ramdas	**210488**	**kantulerr21@iitk.ac.in**
Saurav Kumar	**210950**	**sauravk21@iitk.ac.in**
Surendra kumar ahirwar	**211083**	**surendrak21@iitk.ac.in**
Rishit Bhutra	**210857**	**rishitb21@iitk.ac.in**
Krishna Chandu	**220832**	**pkrishna22@iitk.ac.in**
Sonu Kumar	**211052**	**sonuk21@iitk.ac.in**

Course: **CS253**

Mentor TA: ***Sumit Chaduhry***

Date: **9/02/24**

Revisions

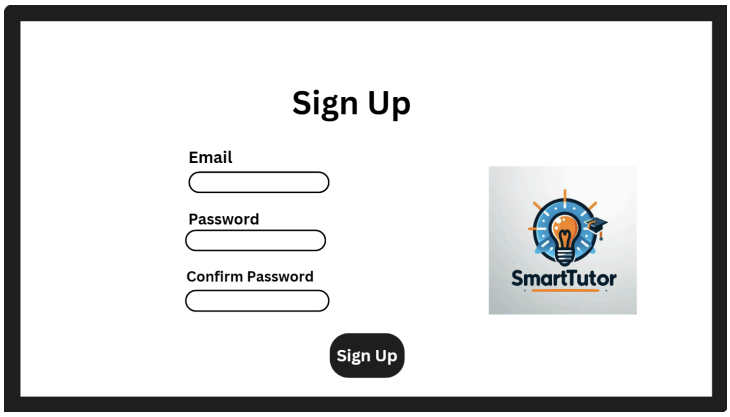
Version	Primary Author(s)	Description of Version	Date Completed
0.1	Sonu Kumar Ch Hemanth Kumar Sarthak Paswan Yash gothwal Kantule Ritesh Ramdas Saurav Kumar Surendra kumar ahirwar Krishna Chandu Rishit Bhutra	First draft	9/02/23

1 Context Design

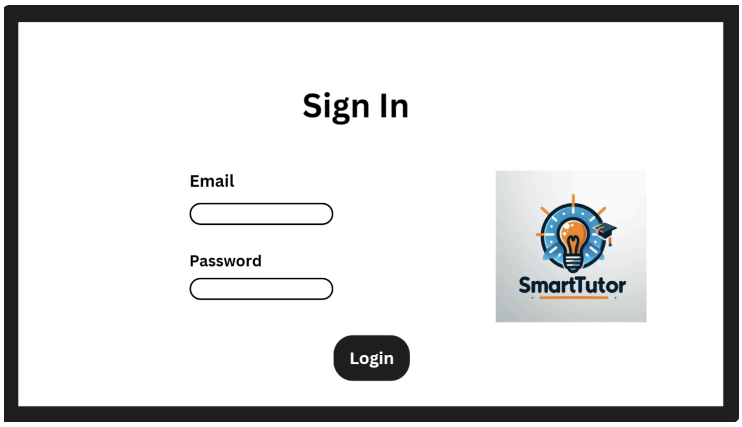
1.1 Context Model

1.2 Human Interface Design

Sign Up Page: The sign-up window will ask for the IITK email and password of the user. The user can sign up after confirming the password.

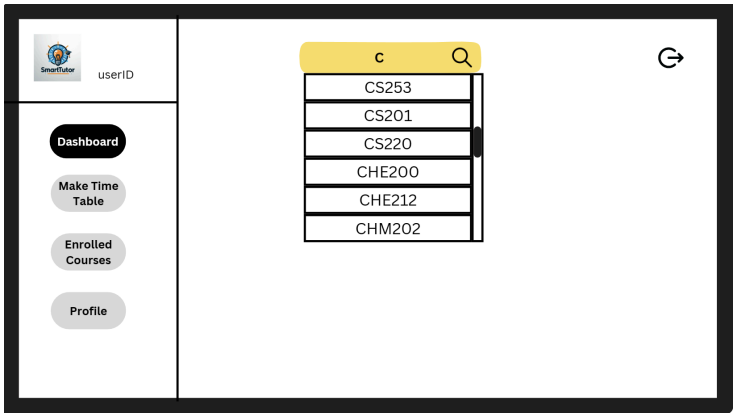


Sign In Page: The sign-in window will ask for the IITK email and password of the user. The user can enter their email and password in the appropriate text fields and click on the Login button to sign in to their account.

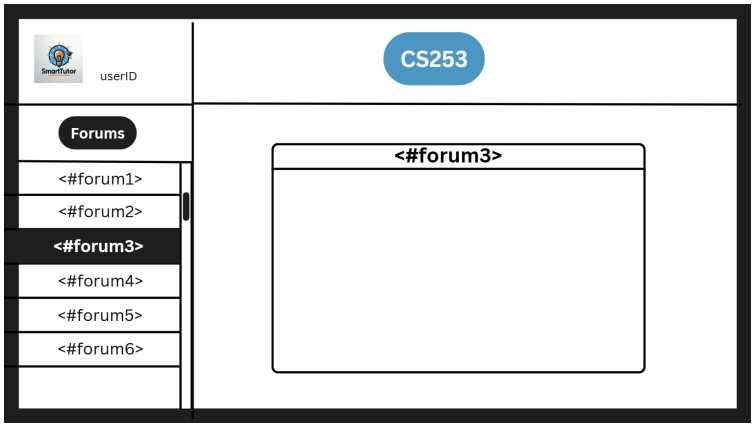


Student's Interface: It should display the portfolio, enrolled courses, time table, dashboard.

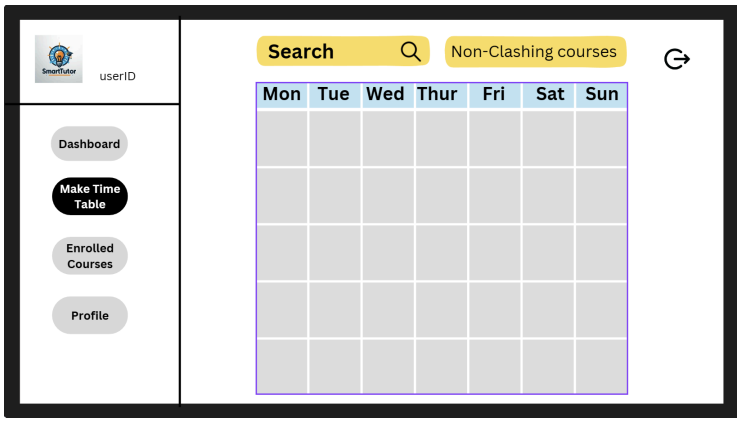
Dashboard: It should display a search bar with all the listed published courses.



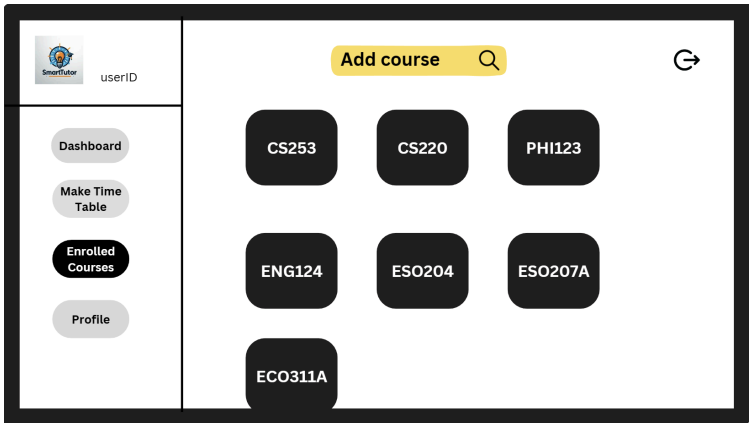
Course Forum Page: The page will contain all the forums asked by the students of a particular course. There will be a general discussion area where the queries are asked and the mentors can answer accordingly.



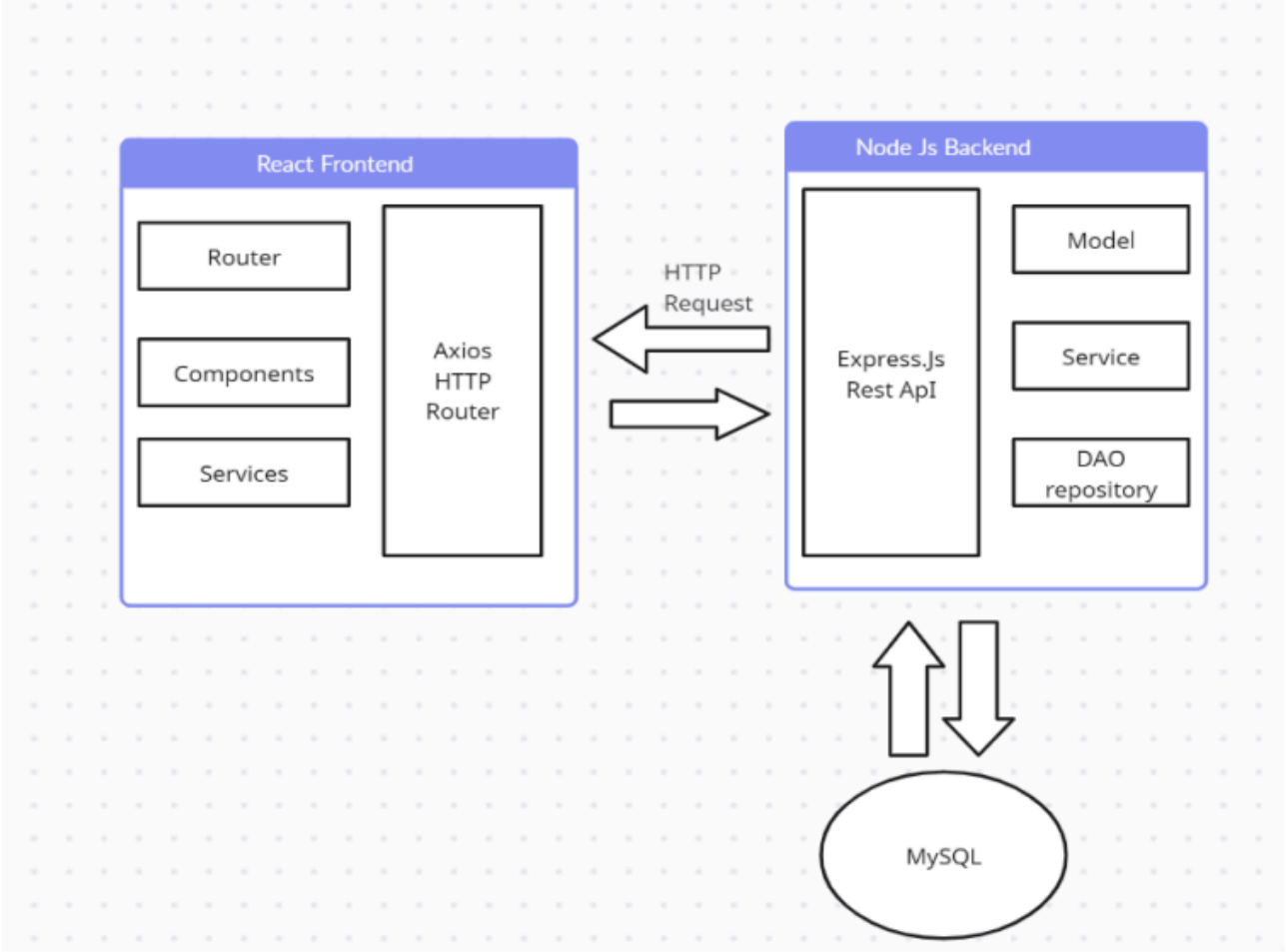
Make Time Table: It should display the time table and provide option to add/ remove a course from timetable.



Enrolled Courses: It will show all the courses the student is enrolled in during the sem



2 Architecture Design



Why are we using this: We are using this model since we would be interacting with the data in multiple ways, like publishing and adding/editing courses, maintaining and updating discussion threads according to requests, and storing student-course data. Also, in the near future, one might think of adding some functionalities to it, which makes the Model-View-Controller model the best fit.

Advantages: One great advantage of this model is that the data can be represented in various manners. This will help us in maintaining discussion threads, segregating courses according to published/ unpublished, student-courses enrolled/ unenrolled etc..

Disadvantages: One disadvantage is that we need to write some additional code for full functionality, even if we know that our implementation is simple in the starting stages

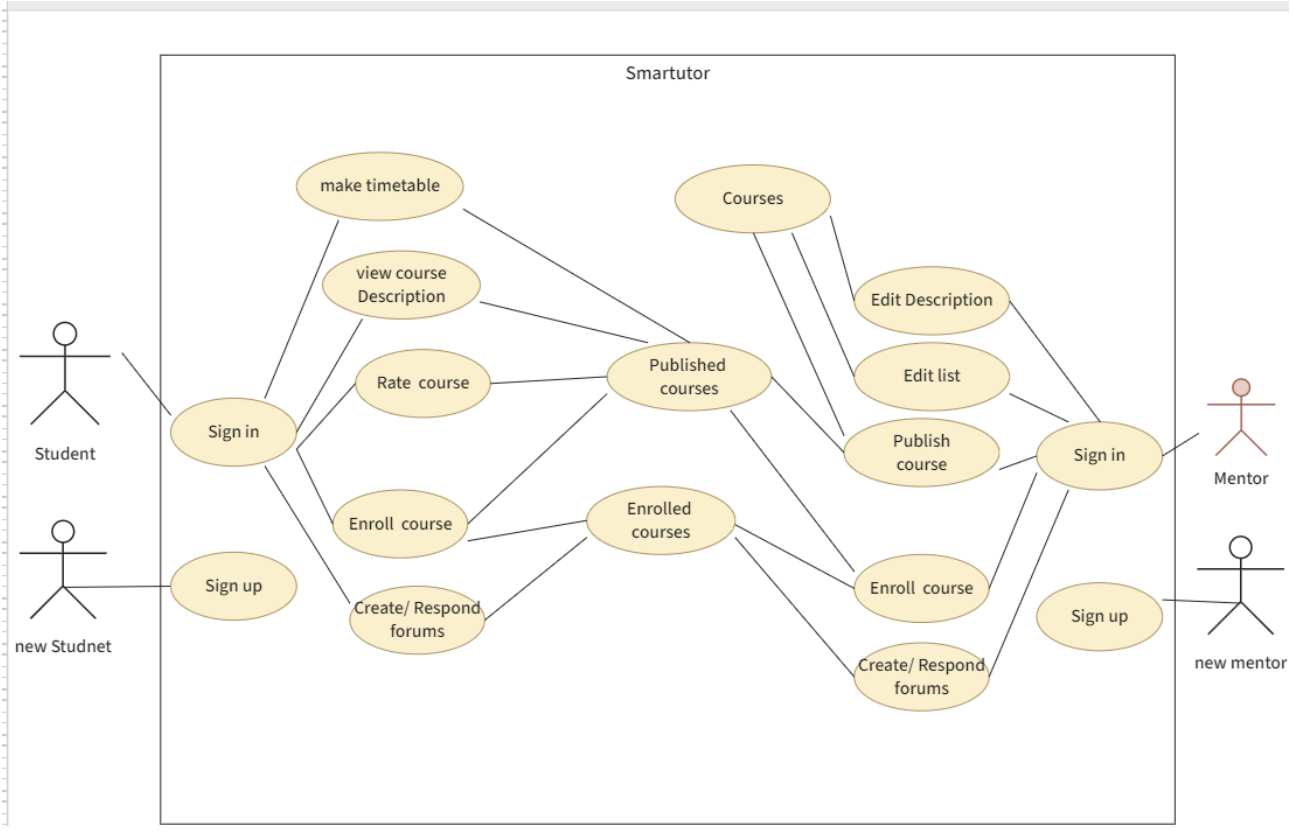
3 Object Oriented Design

3.1 Use Case Diagrams

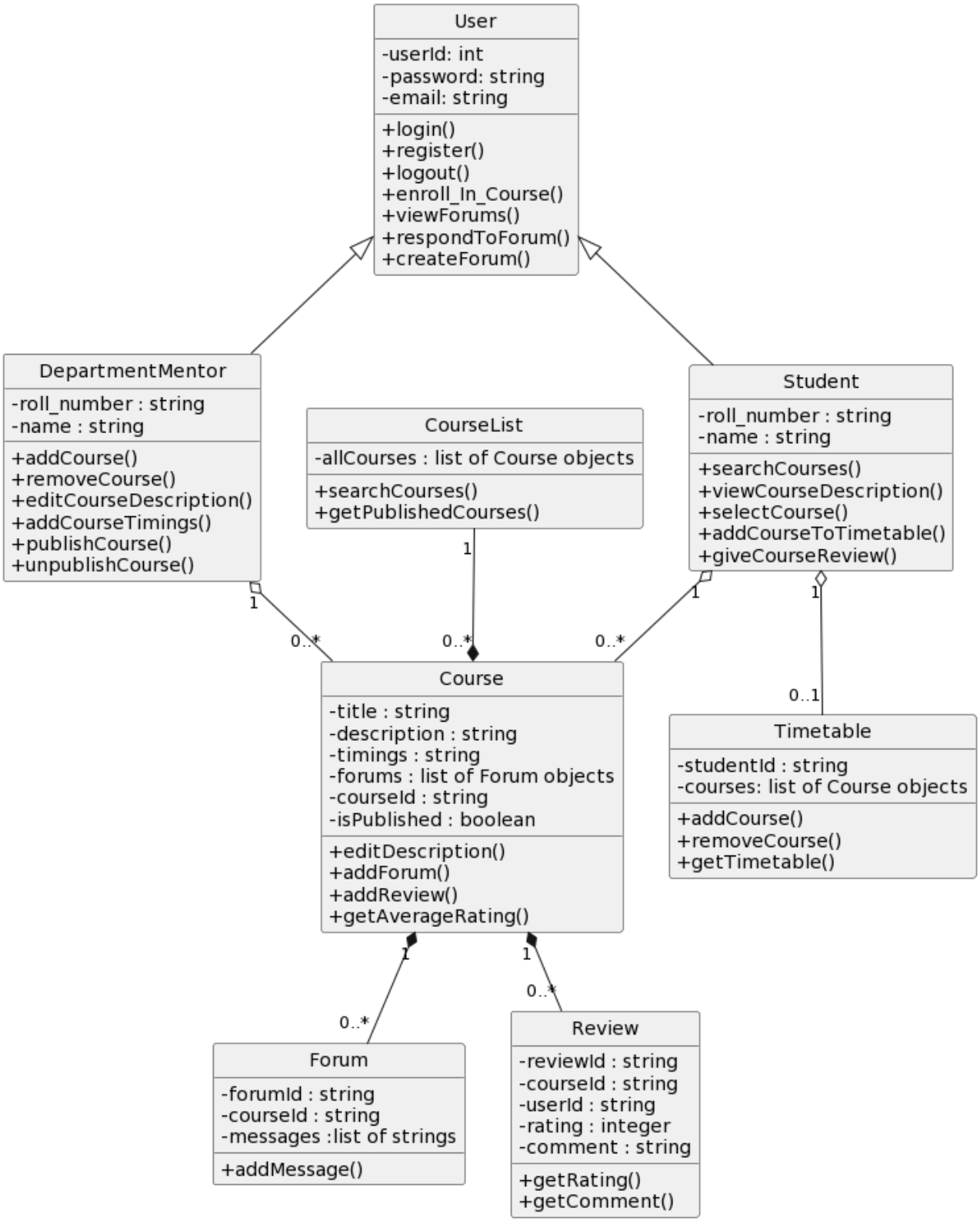
1. Our intended users are of IITK students, who will be using it. Initially, they will have to first make their account through the 'Sign up' option by using their IITK email-id, and after successful authentication they will be able to sign in.

2. Once Signed in, they will be enrolled in any of the Published courses. In that registered course they can see the timetable of published courses, course content and can also rate it.

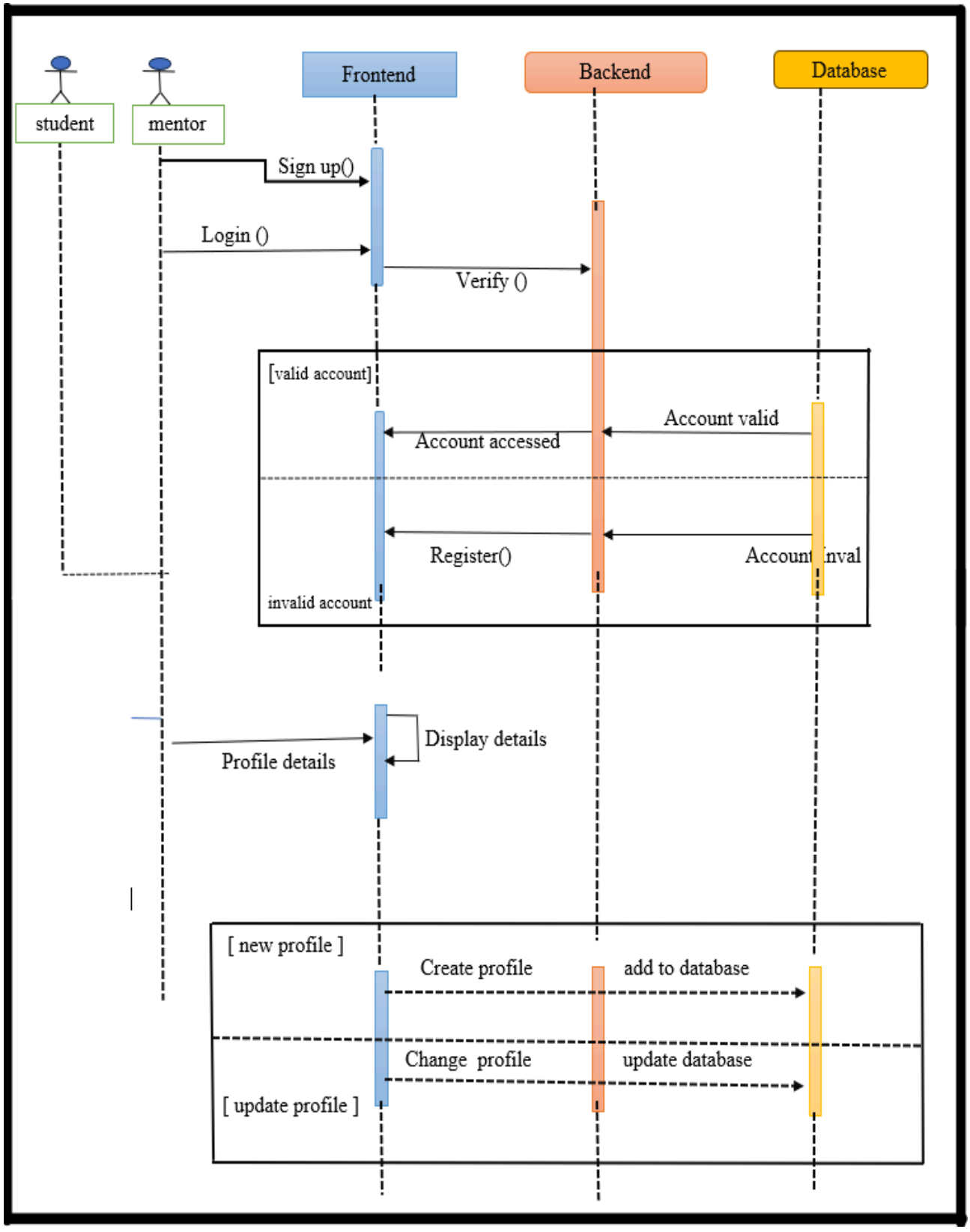
3. As a department mentor, they can add a particular course of their department, update the course list of their dept., and can also change the course Description.

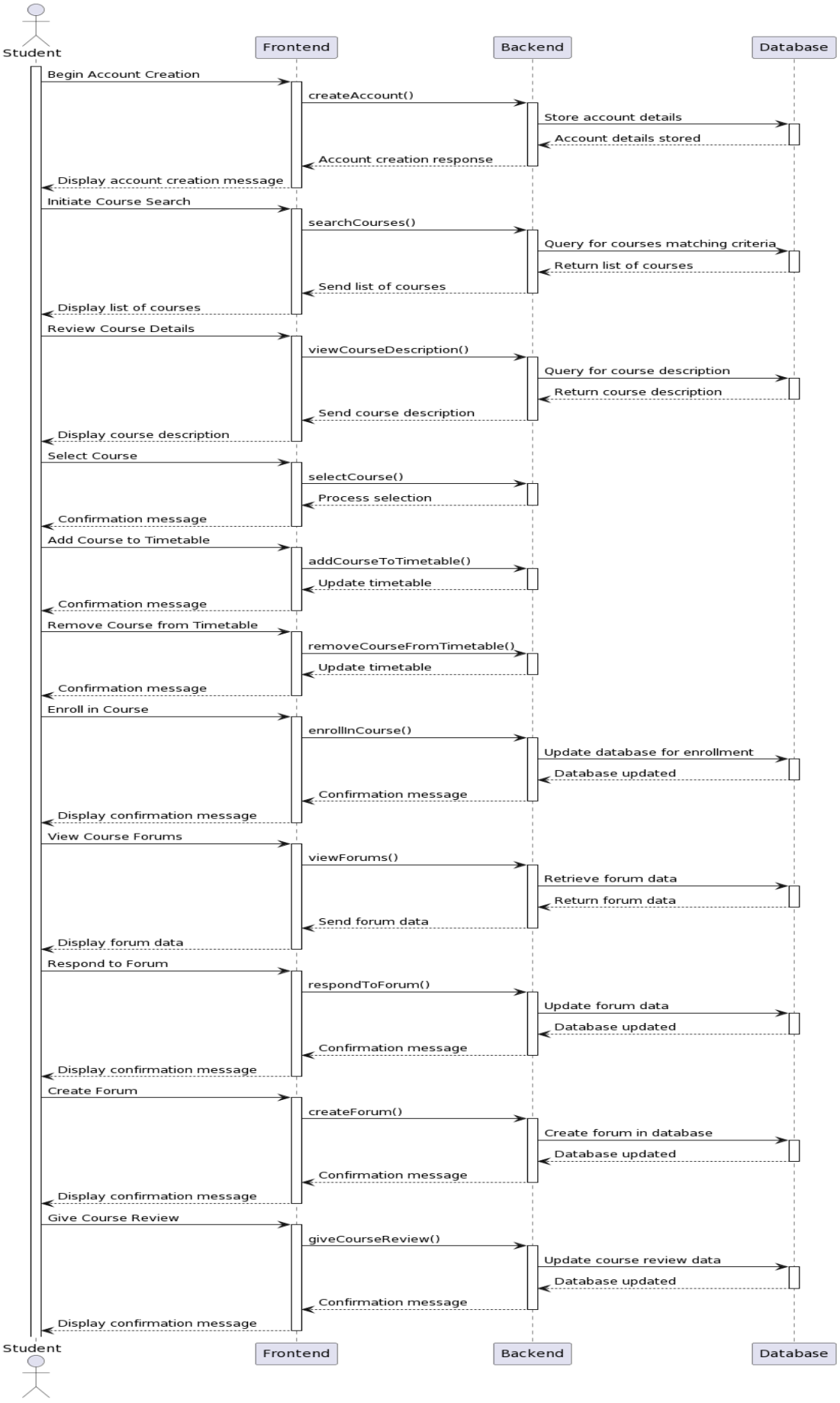


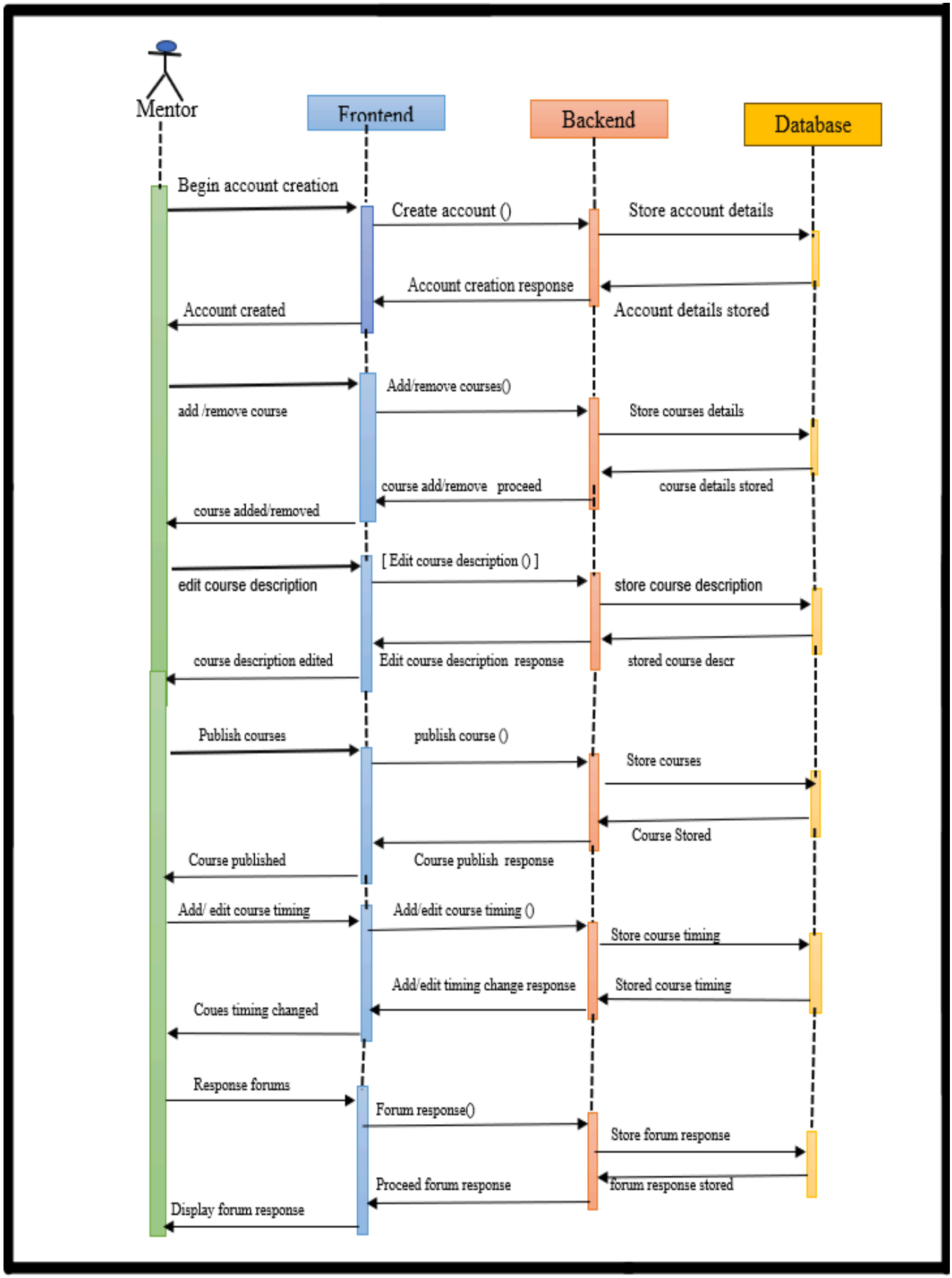
3.2 Class Diagrams



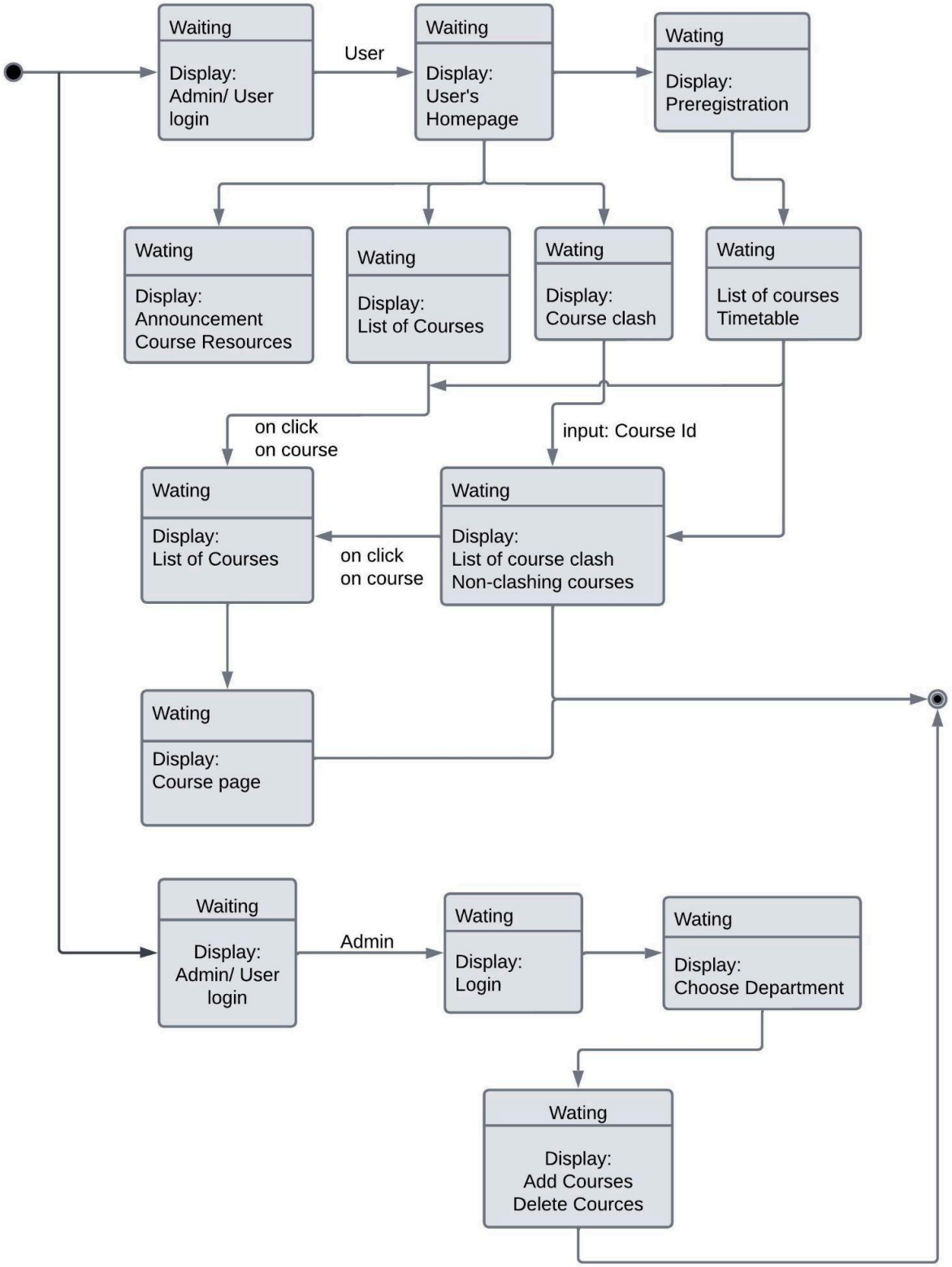
3.3 Sequence Diagrams




Student
Frontend
Backend
Database
Begin Account Creation
createAccount()
Store account details
Account details stored
Account creation response
Display account creation message
Initiate Course Search
searchCourses()
Query for courses matching criteria
Return list of courses
Send list of courses
Display list of courses
Review Course Details
viewCourseDescription()
Query for course description
Return course description
Send course description
Display course description
Select Course
selectCourse()
Process selection
Confirmation message
Add Course to Timetable
addCourseToTimetable()
Update timetable
Confirmation message
Remove Course from Timetable
removeCourseFromTimetable()
Update timetable
Confirmation message
Enroll in Course
enrollInCourse()
Update database for enrollment
Database updated
Confirmation message
Display confirmation message
View Course Forums
viewForums()
Retrieve forum data
Return forum data
Send forum data
Display forum data
Respond to Forum
respondToForum()
Update forum data
Database updated
Confirmation message
Display confirmation message
Create Forum
createForum()
Create forum in database
Database updated
Confirmation message
Display confirmation message
Give Course Review
giveCourseReview()
Update course review data
Database updated
Confirmation message
Display confirmation message
Student
Frontend
Backend
Database


Mentor
Frontend
Backend
Database
Begin account creation
Create account ()
Store account details
Account creation response
Account details stored
Account created
Add/remove courses()
add /remove course
Store courses details
course details stored
course add/remove proceed
course added/removed
[Edit course description ()]
edit course description
store course description
stored course descr
Edit course description response
course description edited
Publish courses
publish course ()
Store courses
Course Stored
Course publish response
Course published
Add/ edit course timing
Add/edit course timing ()
Store course timing
Stored course timing
Add/edit timing change response
Coues timing changed
Response forums
Forum response()
Store forum response
forum response stored
Proceed forum response
Display forum response

3.4 State Diagrams



4 Project Plan

We have set up a thorough JIRA board as part of our project management approach to make project planning and the creation of a Gantt chart easier.

Project Organization:

Tales and Epics:
Our project is organized into Epics, or high-level functionalities, which are further divided into more detailed Stories.

JIRA Board: The JIRA board is our main project management tool. This platform offers a centralized hub for collaboration and progress monitoring, facilitating the effective tracking and management of Epics, Stories, and tasks.

Planning for Sprints

Weekly sprints are the approach we've chosen to maintain an agile development process. I, along with every other team member, have been given well-defined duties for the sprint, which guarantees targeted and doable workloads.
Task Distribution:

During sprint planning sessions, my teammates and I work together on task assignments. Task distribution is made sure to be based on individual capabilities and expertise thanks to this cooperative effort.

Top Techniques:

Detailed Task Breakdown and User Stories:
I place a lot of emphasis on writing user stories that are precisely stated and have accepted standards. Furthermore, segmenting stories into tasks offers a detailed perspective that helps with precise effort estimation and tracking of progress.

Reviews of Code and CI/CD:

Crucial components of my job include putting in place Continuous Integration/Continuous Deployment (CI/CD) pipelines and a strong code review procedure. These procedures guarantee both a smooth deployment process and high-quality code.

Constant Enhancement:

I take an active part in sprint retrospectives, giving feedback on how we performed and offering suggestions for improvement. The team's ongoing progress is aided by this iterative feedback loop.
Group Cooperation:

It is part of my job to promote open communication and a pleasant team environment. In order to create a coherent and productive development environment, I encourage information exchange and teamwork among team members.

Appendix A - Group Log

Why are we using this: We are using this model since we would be interacting with the data in multiple ways, like checking and receiving requests, managing orders,complaints and rebates , and storing transactions. Also, in the near future, one might think of adding some functionalities to it, which makes the Model-View-Controller model the best fit.
Advantages: One great advantage of this model is that the data can be represented in various manners. Which will prove helpful to us while sorting students based on dues, or from the same academic year or in some other fashion. Also, it allows the data to change independently, it would be helpful to add or remove students from the database.
Disadvantages: One disadvantage is that we need to write some additional code for full functionality, even if we know that our implementation is simple in the starting stages

Date	Timings	Minutes
27/ 1/24	8:30pm - 9:30pm	Strategize the future plan of how to progress in its execution
31/ 1/24	5:00pm - 6:30pm	We divided our work among us as of who will be
1/ 2/24	9:00 pm - 11:30 am	We discussed major functional requirements as per the data we collected from students at IITK about the software.
4/ 2/24	7:30pm - 8:00pm	Introductory meeting with TA and work delegation in the team to work on several sections of the SRS.
9/ 2/24	6:30 pm - 7:30pm	Updated TA with our current progress and final discussion on the remaining few sections of SRS.