
Design Document

for

TRACKit

Version 1.0

Prepared by

Group 5

Group Name: Chill guys

Aditya Gautam	220064	agautam22@iitk.ac.in
Aayush Singh	220024	aayushs22@iitk.ac.in
Dhruv Rai	220365	dhruvrai22@iitk.ac.in
Ved Prakash Vishwakarma	221180	vedprakash22@iitk.ac.in
Sharique Ahmad	221002	asharique22@iitk.ac.in
Dhruv Varshney	220366	vdhruv22@iitk.ac.in
Mayur Agrawal	220641	mayurag22@iitk.ac.in
Akash Verma	220097	akashv22@iitk.ac.in
Rahul Ahirwar	220856	rahula22@iitk.ac.in
Abhijeet Agarwal	210025	abhijeeta21@iitk.ac.in
Aryan Bansal	200198	aryanb20@iitk.ac.in

Course: CS253

Mentor TA: Jeswaanth Gogula

Date: 7 February 2025

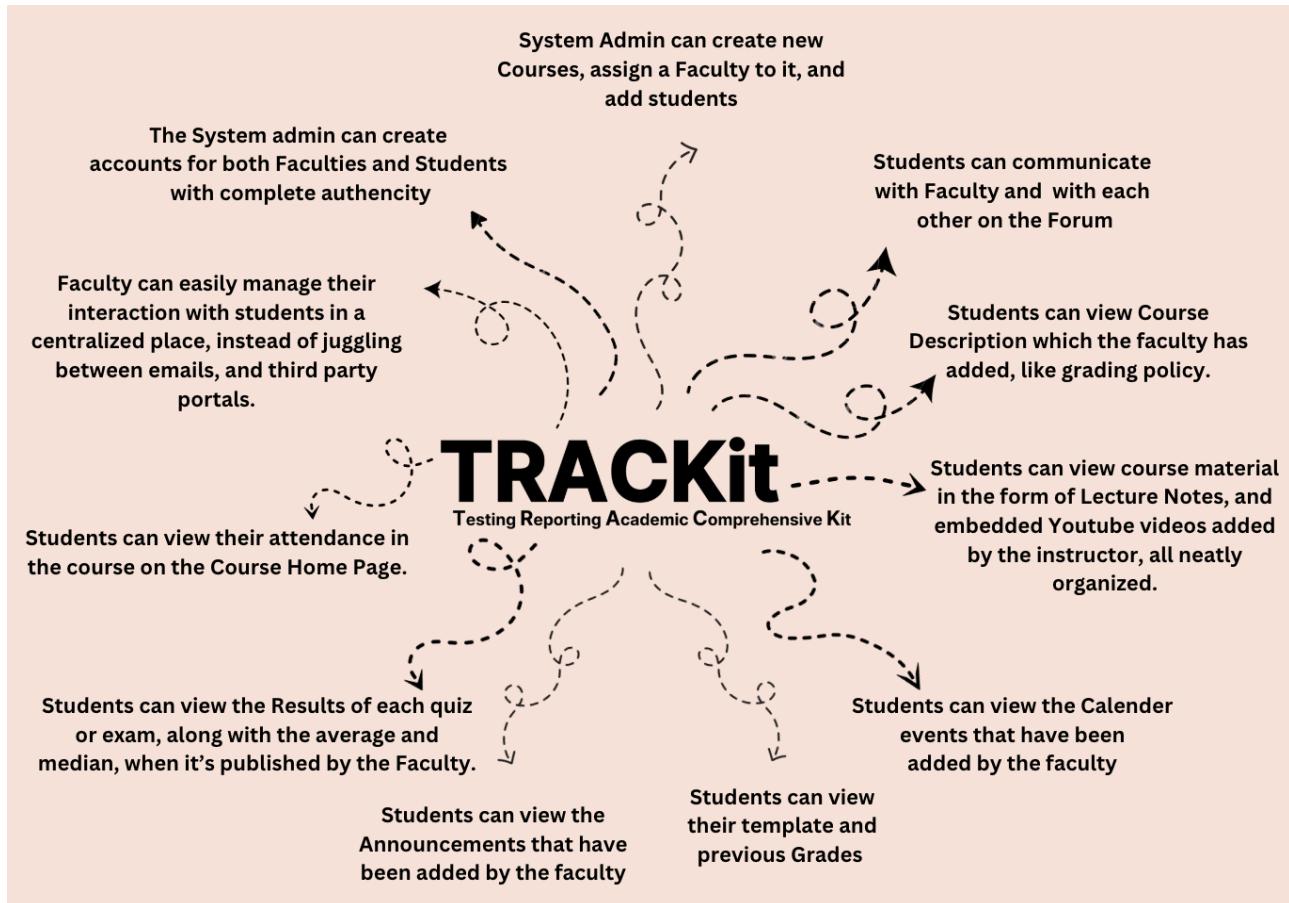
CONTENTS.....	I
REVISIONS.....	II
1 CONTEXT DESIGN.....	1
1.1 CONTEXT MODEL.....	1
1.2 HUMAN INTERFACE DESIGN.....	1
2 ARCHITECTURE DESIGN.....	14
3 OBJECT-ORIENTED DESIGN.....	18
3.1 USE CASE DIAGRAM.....	18
3.2 CLASS DIAGRAM.....	27
3.3 SEQUENCE DIAGRAM	31
3.4 STATE DIAGRAM	35
4 PROJECT PLAN.....	42
APPENDIX A - GROUP LOG.....	43

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
v1.0	All Team Members	Final version of Design Document for TRACKit.	7/2/25

1. Context Design

1.1. Context Model

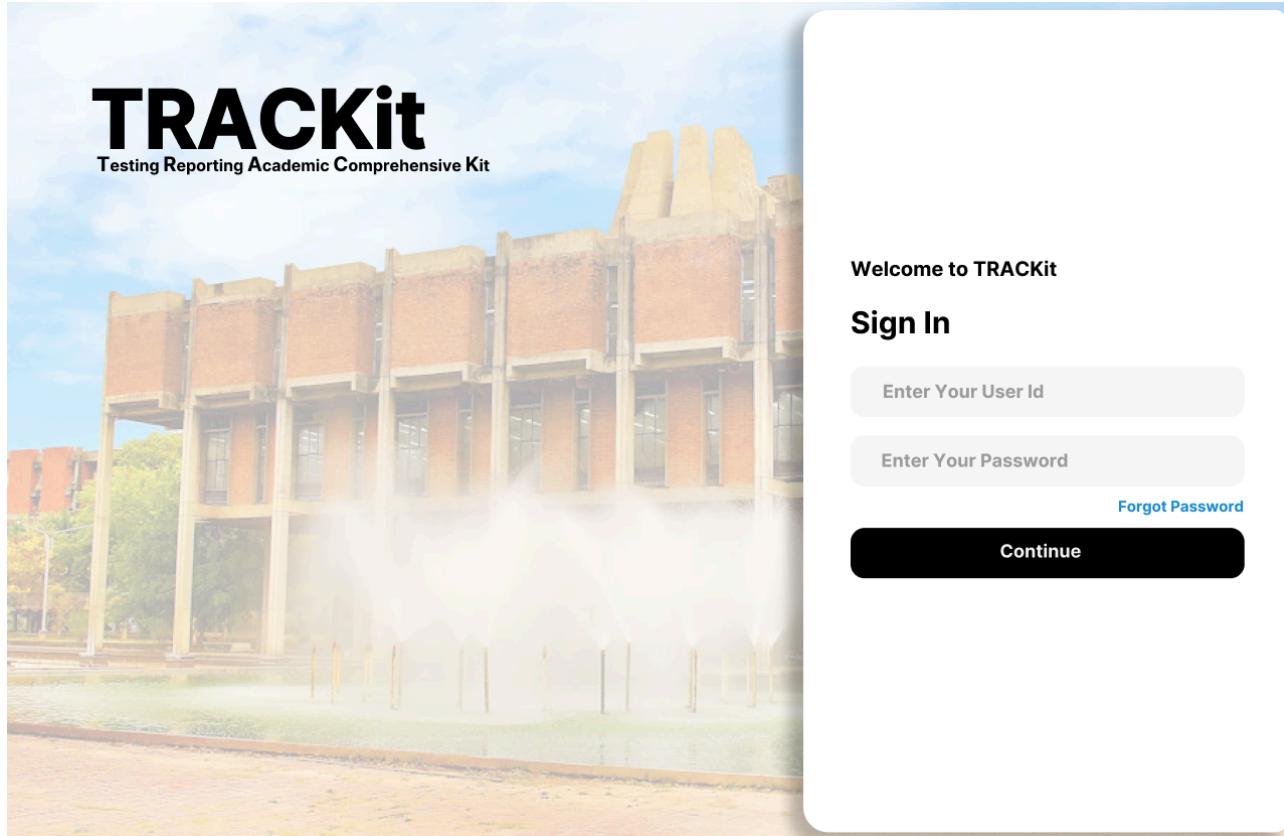


1.2. Human Interface Design

1.2.1 User Interfaces

The Login Page

This is the landing page of our website. This will be the same for all users, who will use their credentials to log in to their accounts. It will have a separate Forgot Password button, which will allow users who have forgotten their Password to regain access to their accounts. An OTP will be created and sent to their emails, and they will be able to use this OTP to set a new password.



The Main Dashboard

The Main Dashboard will be the first screen both students and faculty will see upon signing into TRACKit. The Dashboard for the system admin will be characterised in an upcoming section.

For Students, it will be a place to access the courses which they are doing in the current semester on the My Courses section. The My Courses section will also have a calendar displaying the merged data of events from all courses in which the student is enrolled. They can also access course pages of courses done by them in their previous semesters. There will also be a Performance section in the sidebar which will allow students to view their template and grades in completed courses at a glance. The Profile section on the sidebar would allow for actions like Change Password.

The UI for the Instructors would be exactly similar, except for the fact that they will not see the Performance section on their sidebar. Just like students can view the courses they are enrolled in for the current semester, the Faculty can view the courses which they are teaching in that semester. The calendar which they see will simply be a merged calendar of all the events they have created in the courses they are taking.

The dashboard displays a grid of course cards for EE320, EE370, ESO207, CS253, CS220, and CS771, each with its title, code, and professor. Below the cards is a weekly calendar for August 2024, showing classes from Monday to Saturday. At the bottom are navigation links for different academic years.

The Performance Page

As mentioned above, this page is viewed by clicking the namesake icon on the Main Dashboard's sidebar. This Page is available for students only. Students can see the template of their department as a chart. On this chart information like pending, failed, current & completed courses will also be displayed. The grade sheet of students will also be displayed here.

The performance page features a large chart titled "Template" showing a grid of courses across various subjects and levels. Below the chart is a "Gradesheet" table with columns for S.No., Academic Session, Semester, Course ID, Course Name/Similar Course, Course Type, Grade, and Credits. A note at the bottom of the chart indicates that courses should be compiled as per the template.

S.No.	Academic Session	Semester	Course ID	Course Name/Similar Course	Course Type	Grade	Credits
1	2022-23	1	PHY113	CLASSICAL ELECTRODYNAMICS	IC	A*	11
2	2022-23	1	CHM111	CHEMISTRY LABORATORY	IC	A*	3

The Course Home Page

The Course Home Page is the first page which both students and faculty see after clicking the icon of a course on the Main Dashboard. The Course Home Page contains a small Events list in the top left corner which contains the upcoming events under that course scheduled by the course Instructor. Towards the immediate right of the Events list is the current attendance status box. To the students, it shows their current attendance in the course. To the Instructor, it shows the percentage of students attending lectures, on average.

Right below these are the Course Info sections. These collapsible boxes contain the most important facts and attributes about the course, similar to an FCH. The faculty may choose to add sections for Course Description, Grading Policy, Books and References, and anything else he/she thinks fit. He/She will also have the capability to edit existing sections.

The navigation bar will also get changed compared to the one on the Main Dashboard. Now it will contain six tabs namely Course Home, Lectures, Calendar, Results, Forum. These tabs will show the content of specific course that the instructor/student has currently selected.

Student's Course Home Page

The screenshot displays the 'Course Home' page for a course titled 'CS253'. On the left, a sidebar menu lists 'Course Home' (selected), 'Lectures', 'Announcements', 'Calender', 'Results', and 'Forum'. The main content area features a 'Next Events' section listing three days: 20 JAN, MON (Project Presentation at 2pm); 21 JAN, TUE (Lecture at 10am and Quiz at 5:30pm); and 22 JAN, WED (Lecture at 11am and Doubt Resolution at 2pm). To the right is a 'Your Attendance' box showing '95%' with the message 'You have attended 19/20 classes.' Below these are six collapsible sections: 'Description', 'Syllabus', 'Course Team', 'Class Venue', 'Textbooks and References', and 'Course Grading Policy', each represented by a downward arrow icon.

Instructor's Course Home Page

CS253

Course Home

Next Events

- 20 JAN, MON
 - 2pm Project Presentation
- 21 JAN, TUE
 - 10am Lecture
 - 5:30pm Quiz
- 22 JAN, WED
 - 11am Lecture
 - 2pm Doubt Resolution

Class Attendance

75%

On an average, the class attendance is 75% of the class strength.

Add Section

Description	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Syllabus	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Course Team	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Class Venue	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Textbooks and References	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The Lectures Page

The Lectures Page will be a centralised location for course material. The Faculty will be able to create sections, ideally into weeks, and then further organise the material into subsections. Each subsection will be capable of holding a PDF, text, and an embedded YouTube video.

Student's Lecture Page

CS253

Lectures

Week 1

Software Processes

- Introduction to Software Processes
- Software Process Paradigms
- Agile Processes

Requirement Engineering

- Types of Requirements
- Functional Requirements
- Non-functional Requirements

Week 2

Week 3

Week 4

Faculty's Lecture Page

The screenshot shows the 'Lectures' section of the Faculty's Lecture Page for the course CS253. The sidebar on the left includes links for Course Home, Lectures (which is selected and highlighted in black), Announcements, Calender, Results, and Forum. The main content area is titled 'Lectures' and features a 'Add Section' button. Under 'Week 1', there are two main categories: 'Software Processes' and 'Requirement Engineering'. Each category contains three sub-sections, each with edit, delete, and file icons. The 'Software Processes' category includes 'Introduction to Software Processes', 'Software Process Paradigms', and 'Agile Processes'. The 'Requirement Engineering' category includes 'Types of Requirements', 'Functional Requirements', and 'Non-functional Requirements'. Below 'Week 1' are 'Week 2' and 'Week 3' sections with dropdown arrows.

The Calendar Page

Each faculty member and student will have access to a Calendar page dedicated to every course they are teaching or enrolled in. To access the Calendar, users can navigate to the Main Dashboard, select their desired course, and then click on the Calendar option available in the sidebar.

The Calendar provides a comprehensive overview of the scheduled activities for the course, including lectures, labs, tutorials, quizzes, discussion hours, deadlines for assignments, submissions, and other activities.

While students can view scheduled events and stay updated on course timelines, instructors have additional administrative control over the Calendar. They will be able to add, edit or delete events. Changes made by the instructor get instantly reflected on the calendar, ensuring real-time updates. Additionally, instructors can choose to generate an announcement to notify students about the changes.

To enhance usability, the Calendar offers multiple viewing modes, users can switch between different time ranges, including daily, weekly, monthly, custom range view.

Students Calendar

CS253

- [Course Home](#)
- [Lectures](#)
- [Announcements](#)
- Calender**
- [Results](#)
- [Forum](#)

CALENDAR

today **January 2025** < year month week list >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
W1 29	30	31	1	2	3	4
W2 5	6	7	8	9	10	11
W3 12	13	14	15	16	17	18
2:01a Quiz 1		10a Discussion 1			3:01a Quiz 2	
W4 19	20	21	22	23	24	25
3:01a Quiz 2		10a Discussion 2			11:55 Deadline - Project : Requirement Docu	
W5 26	27	28	29	30	31	1
		10a Discussion 3				
W6 2	3	4	5	6	7	8
		10a Discussion 4			11:59 Deadline - Project : Design Document S	

Instructor's Calendar

CS253

- [Course Home](#)
- [Lectures](#)
- [Announcements](#)
- Calender**
- [Results](#)
- [Forum](#)

CALENDAR

ADD EVENT EDIT EVENT DELETE EVENT < year month week list >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
W1 29	30	31	1	2	3	4
W2 5	6	7	8	9	10	11
W3 12	13	14	15	16	17	18
2:01a Quiz 1		10a Discussion 1			3:01a Quiz 2	
W4 19	20	21	22	23	24	25
3:01a Quiz 2		10a Discussion 2			11:55 Deadline - Project : Requirement Docu	
W5 26	27	28	29	30	31	1
		10a Discussion 3				
W6 2	3	4	5	6	7	8
		10a Discussion 4			11:59 Deadline - Project : Design Document S	

The Announcement Page

Instructors can use this page to post important announcements related to their courses. They can also edit or delete existing announcements. All students enrolled in said course will have access to the announcements made by the instructor. Additionally, the page will display the count of unseen announcements, to each student.

Student's Announcement Page

Instructor's Announcement Page

The Forum Page

The Forum Page will be accessible to all the users enrolled in a particular course. Users can use this page to post their doubts and any user can reply to existing posts. Instructors will have special access to edit or delete any existing post in order to moderate the discussion.

Student's Forum

CS253

FORUM **Add Post**

Use of Laplace Equation

ayush22

In question 5 of assignment 5, can we use the Laplace equation instead of using the boundary value approach. I am having some doubts regarding this, if anybody has solved then please reply. I have put the equation in contention for all to see.

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = \frac{S_y}{K D} \frac{\partial h}{\partial t}$$

Replies

aguptam22

The equation can't be used as it does not satisfy the initial conditions of the problem.
Therefore, we must use the Laplace Equation.
Hope you understood the answer!

Can we use calculator in mid-sem exams?

aguptam22

Replies

aguptam22 (Instructor)

Yes, you may bring a calculator with you.
But the design of questions will be such that you wouldn't need a calculator to solve any question.

Instructor's Forum

CS253

FORUM **Add Post**

Use of Laplace Equation

ayush22

In question 5 of assignment 5, can we use the Laplace equation instead of using the boundary value approach. I am having some doubts regarding this, if anybody has solved then please reply. I have put the equation in contention for all to see.

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = \frac{S_y}{K D} \frac{\partial h}{\partial t}$$

Replies

aguptam22

The equation can't be used as it does not satisfy the initial conditions of the problem.
Therefore, we must use the Laplace Equation.
Hope you understood the answer!

Can we use calculator in mid-sem exams?

aguptam22

Replies

aguptam22 (Instructor)

Yes, you may bring a calculator with you.
But the design of questions will be such that you wouldn't need a calculator to solve any question.

The Results Page

The Result Page will provide tailored functionality for both students and instructors:

- For Students: The Result Page will allow students to view their performance in a specific course. It will display their scores, total marks, average marks of all students, standard deviation, and median for each exam, quiz, and project in the course. Additionally, the page will show the weightage of each component and the corresponding weighted marks, offering a detailed breakdown of their performance.
- For Instructors: The Result Page will enable instructors to analyze the performance of each student in the course. It will display the marks scored by individual students for quizzes, assignments, projects, and other components. Furthermore, the page will include a histogram representing the distribution of overall marks, assisting instructors in grading and evaluating students at the end of the course.

Student's Results Page

CS253

- [Course Home](#)
- [Lectures](#)
- [Announcements](#)
- [Calender](#)
- [**Results**](#)
- [Forum](#)

RESULTS

	Total Weightage	Obtained
Endsem	15	12
Midsem	15	13
Assignment	15	10
Weekly Quiz	10	10
Project	45	44

Weekly Quiz	Your Marks	Total Marks	Average	Stand. Dev.	Median
Assignment	9	10	7.5	2.5	6
Project	10	10	10	10	10
Mid-sem	10	10	10	10	10
End-sem	10	10	10	10	10
Total	10	10	10	10	10

Instructor's Results Page

RESULTS

Student	Roll No.	Marks
Aditya Gautam	220064	5
Dhruv Varshney	220366	10
Ved Prakash Vishwakarma	221180	15

The System Admin's Dashboard

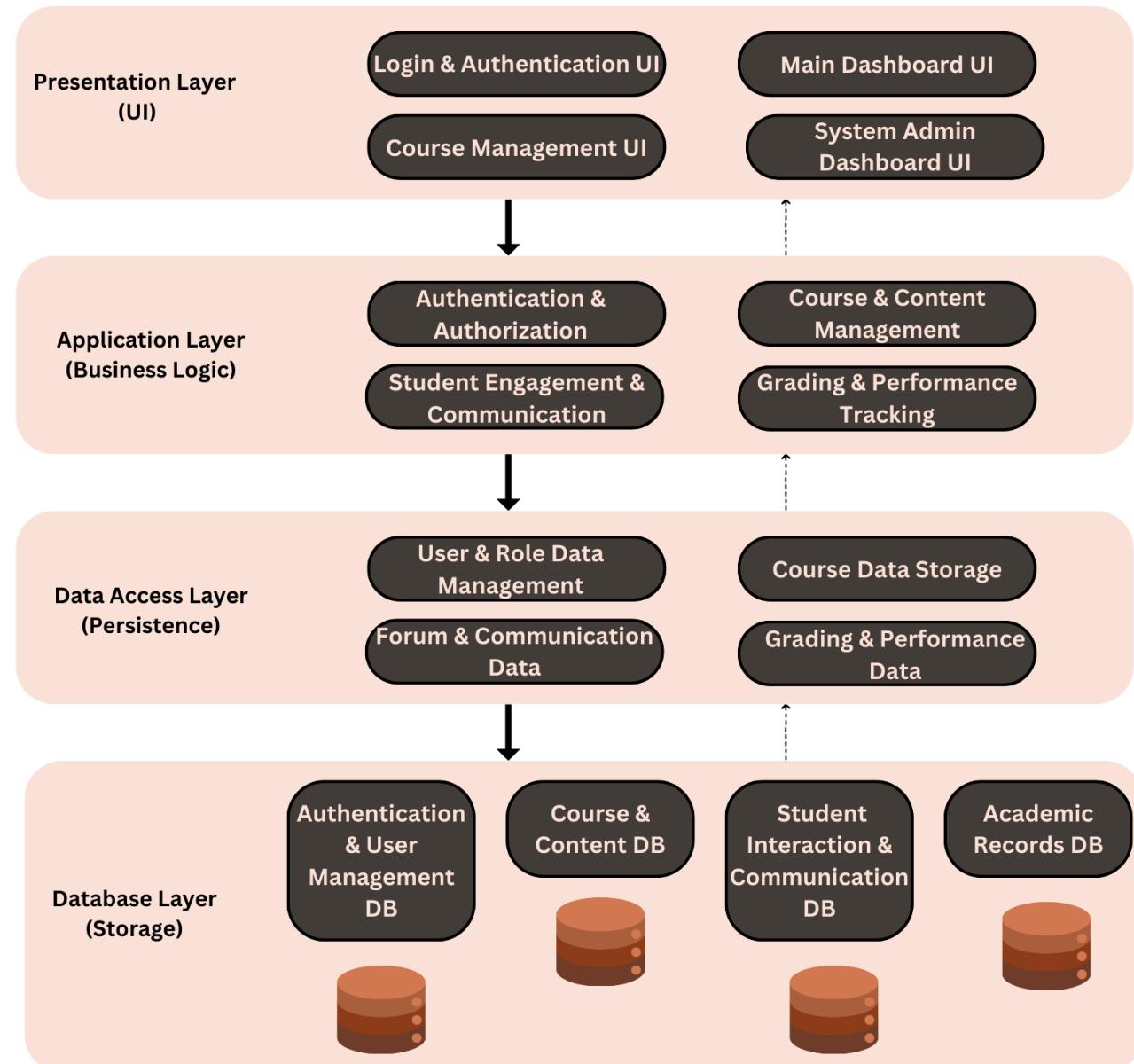
The System Admin will be a unique user. This Dashboard will allow the system administrator to create new courses, faculty, and user accounts, as well as manage them effectively.

TRACKit Welcome System Admin: IIT Kanpur

Add Student	Add Faculty	Create Course
Manage Courses	Manage Users	Contact Developers

2. Architecture Design

The architecture of our proposed system is designed to ensure modularity, scalability, and maintainability. The system follows a **multi-layered architecture**, separating concerns into the Presentation layer, Application layer, Data access layer and Database layer for better organization and performance.



Layer	Modules	Responsibilities
Presentation Layer (UI)	<ul style="list-style-type: none"> • Login & Authentication UI • Main Dashboard UI • Course Management UI • System Admin Dashboard UI 	<ul style="list-style-type: none"> • Renders user interface for Students, Faculty, and Admin. • Collects user input (login, course actions, announcements, etc.). • Displays processed data (grades, attendance, discussion posts). • Manages navigation between different features (courses, forums, admin pages).
Application Layer (Business Logic)	<ul style="list-style-type: none"> • Authentication & Role Management • Course & Content Management • Student Engagement & Communication • Grading & Performance 	<ul style="list-style-type: none"> • Implements core business rules (how courses are created, how grading is performed, etc.). • Enforces role-based access and permissions (Student, Faculty, Admin). • Manages workflows (creating events, announcements, moderating forums etc.). • Handles attendance calculation, marks calculation, performance tracking logic, etc.
Data Access Layer (Persistence)	<ul style="list-style-type: none"> • User & Role Data Management • Course Data Storage • Forum & Communication Data • Grading & Performance Data 	<ul style="list-style-type: none"> • Serves as an abstraction between the business logic and the database. • Handles secure CRUD operations for users, roles, courses, forum posts, etc. • Manages queries, transactions, and error handling. • Shields the Application Layer from knowing the physical data schema or DB details, making it easier to update or switch databases.

Database Layer (Storage)	<ul style="list-style-type: none"> • Authentication & User Management DB • Course & Content DB • Student Interaction & Communication DB • Academic Records DB 	<ul style="list-style-type: none"> • Represents the actual physical storage of data. • Persists all system information, including user credentials, lectures, forum content, grades, etc. • Ensures data security, integrity, and scalability (e.g., backups, replication, sharding).
-----------------------------	---	--

Addressing Non-Functional Requirements (NFRs)

The key NFRs included in our SRS document and the reasons for why layered architecture will abide by these requirements are:

1. Concurrency (handle 4000+ users)

- Since different parts of the system are separated into layers, each layer can scale independently. We can handle more users by adding more servers for the Presentation Layer and using caching to reduce database load.
- The Business Layer can be optimized to handle multiple requests at the same time without slowing down.

2. Resource Efficiency

- We can add more copies of the database (replication) or multiple app servers to balance the load.
- Using caching in the Persistence Layer reduces the number of direct requests to the database, making the system faster and reducing resource usage.

3. Security

- The Presentation Layer never directly accesses the Database; all calls are funneled through the Business logic and Persistence layers.
- All security checks (like verifying user roles and permissions) happen in the Business Layer, ensuring that only authorized users can perform certain actions.
- Sensitive data (passwords, tokens etc.) is handled at the Persistence boundary, and the Database remains secured behind a firewall.

4. Maintainability

- Since each part of the system is separated into layers, we can make changes to one part without affecting the others.
- For example, if we change how course data is stored (e.g., switching databases), it only affects the Persistence Layer, and the UI and Business Layer remain the same.

5. Availability & Reliability

- If any layer fails, layered architecture designs typically allow graceful error handling by fallback strategies, database replication etc. making our system more reliable.

- The architecture allows us to distribute risk across layers that prevents a single point of failure, making the system more robust.

6. Portability

- Since we use modern frameworks like React (for UI) and Node.js/Spring Boot (for Business Logic), the system can run on different platforms without major changes.
- Changes in the database or storage system only affect the bottom layer, making migrations easier.

Although other architectures can be used, we came up with the following arguments as to why they are not better fit for TRACKit.

- **Microservices Architecture:**

Microservices allow independent deployment of small services, which is great for big, complex systems. But for a small-to-medium system like TRACKit, microservices add too much complexity (e.g., managing multiple services, handling network delays). Since we have a single development team, maintaining many independent services would be too much effort compared to a simpler approach.

- **Model-View-Controller (MVC) Architecture:**

MVC is great for organizing UI-based applications, especially when the focus is on separating data (Model), logic (Controller), and UI (View). However, TRACKit is more than just a UI-based app—it involves business logic, security, and multiple system interactions, which MVC doesn't manage well. Also, the layered architecture allows us to improve each layer independently which is a bit difficult in MVC architecture.

- **Pipe-and-Filter Architecture:**

This architecture works well when data flows through multiple steps in a sequence, like data processing pipelines or compilers. But TRACKit is not just about sequential data processing—it needs structured data storage, business logic, and user interactions. Using Pipe-and-Filter here would make things too rigid and not flexible enough to handle different types of user actions so we will not use it for implementation.

Hence, we choose Layered Architecture for TRACKit, as it is well-suited to meet all the required non-functional requirements and allow us to scale or modify each layer independently without impacting the entire system which provides us enough flexibility.

3. Object Oriented Design

3.1. Use Case Diagrams

3.1.1 Use Case #1 (Secure Account Creation and Login Password Management)

Author: Dhruv Rai and Rahul Ahirwar

Purpose: To authenticate and create accounts for valid users via system admin, and to illustrate options for users to change their password.

Requirements Traceability: Authenticated creation of user accounts on the software. Secure ways for users to change their password.

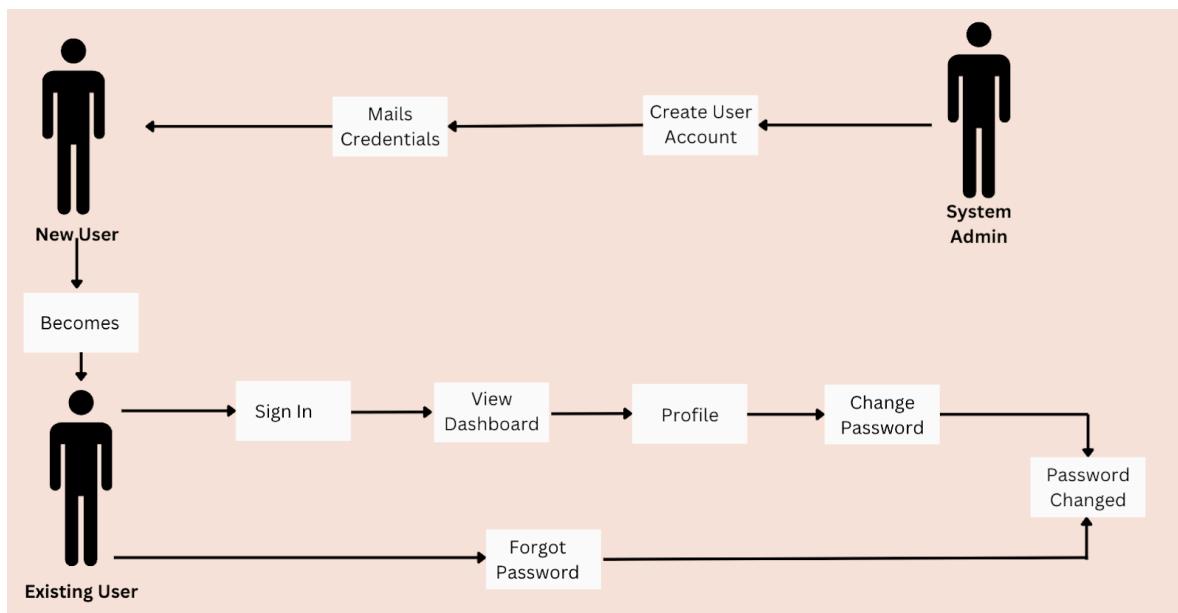
Priority: Very High

Preconditions: The user is a real student/ faculty, and has a valid Email ID.

Postconditions: The user is logged in and is able to view the dashboard and use the software.

Actors: System Admin, Students, Faculty.

Exceptions: The entered User ID should be valid, and be recalled by User.



3.1.2 Use Case #2 (Forum)

Author: Aditya Gautam and Dhruv Varshney

Purpose: To enable users to ask questions and comment on various topics covered in the class.

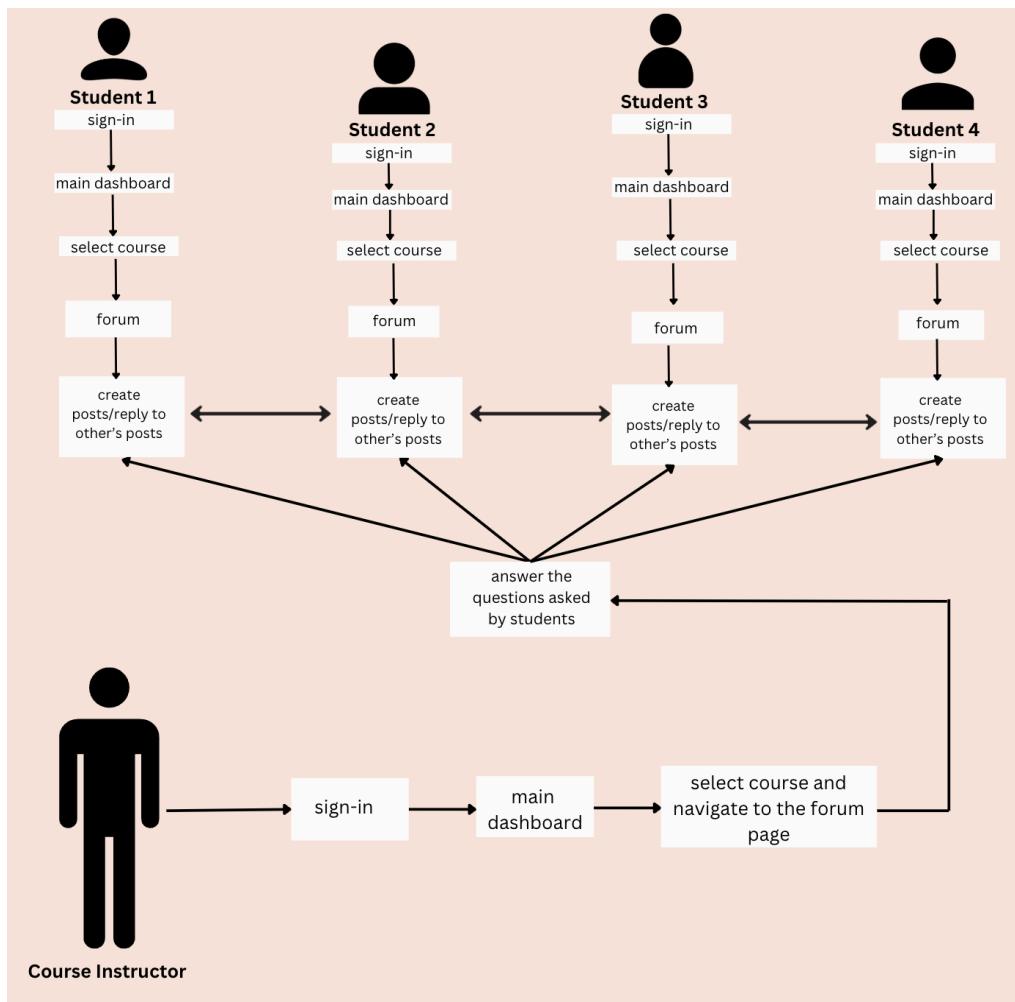
Requirements Traceability: This feature provides a collaborative platform for students to ask questions and discuss class topics, fostering interactive learning, peer engagement, and clarity. Faculty and registered students can create and comment on forum posts.

Priority: Medium

Pre-conditions: The user must have access to the specific course, either as an instructor or a student enrolled in the course.

Post-conditions: After posting a new question, follow up or answer the user and all the other stakeholders will be able to see it. The number of unseen posts also increases for all the other users.

Actors: The primary actors in this use case are students and course instructors



3.1.3 Use Case #3 (Calendar and Event List)

Author: Dhruv Rai and Sharique Ahmad

Purpose: To allow users to manage their day more

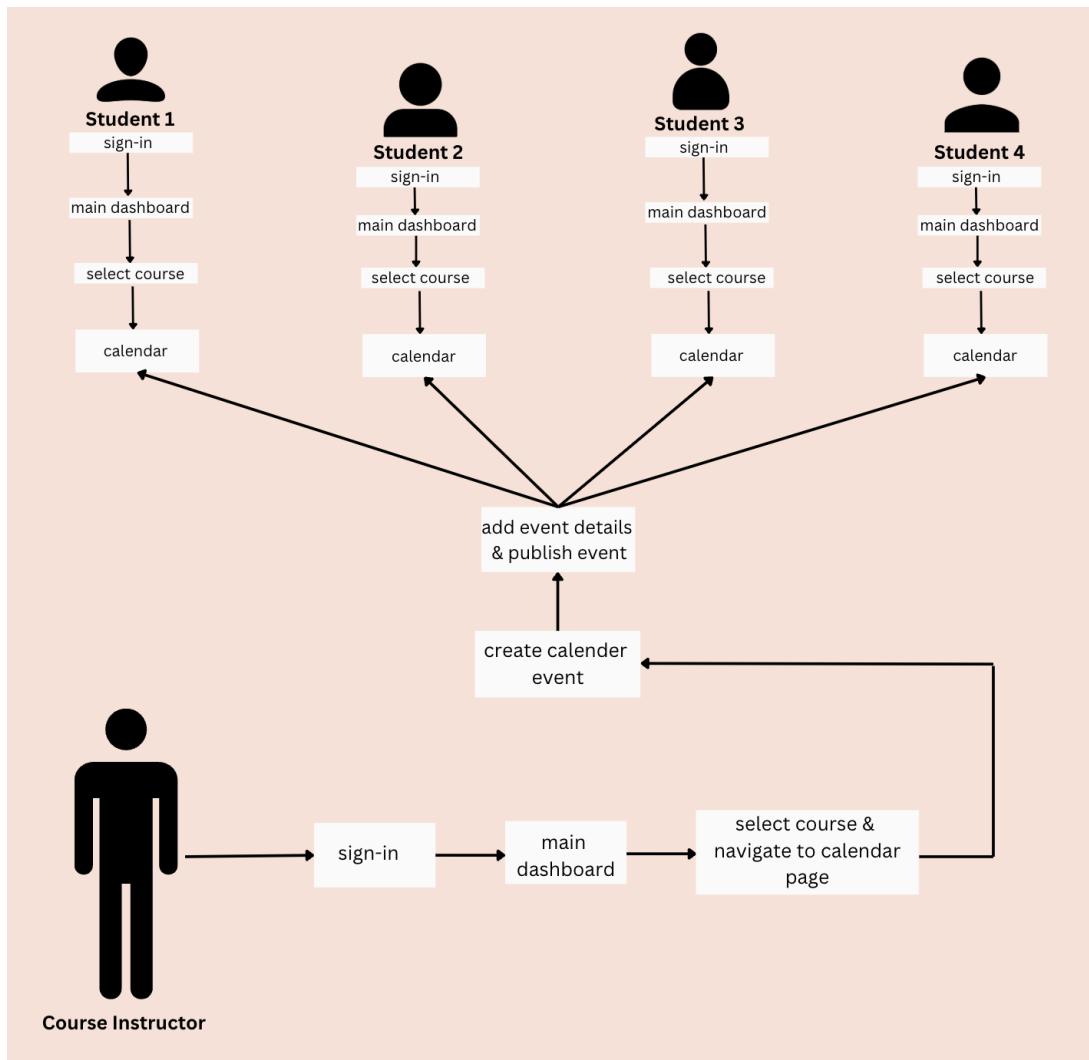
Requirements Traceability: This feature addresses the problem of poor management of daily tasks.

Priority: High

Pre-conditions: The user must have access to the system and be logged in.

Post-conditions: The user will be able to manage their daily tasks successfully.

Actors: The primary actors in this use case are students, who act as the users of the system. The events are created by the faculty.



3.1.4 Use Case #4 (Course Home)

Author: Rahul Ahirwar and Aayush Singh

Purpose: To enable instructors to upload, update, and manage course-related information, such as grading schemes, syllabus, and other critical components, ensuring effective communication and accessibility for students.

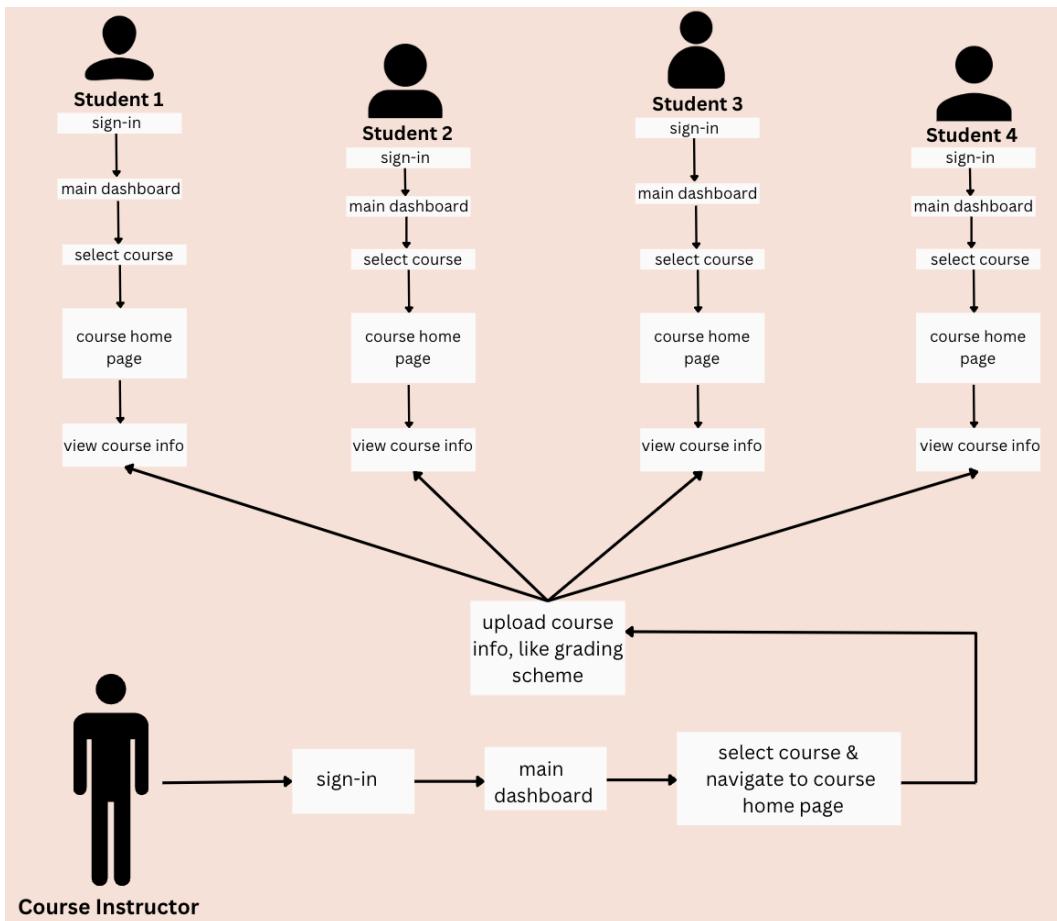
Requirements Traceability: This use case ensures instructors can share essential course details with students in a structured and accessible format, reducing manual effort and ensuring clarity for students regarding their coursework and performance metrics.

Priority: High

Pre-conditions: The instructor must have administrative access to the specific course in the system. The course must be created and active on the platform. The user must be enrolled in that course.

Post-conditions: Course information, including grading schemes, is successfully uploaded and visible to enrolled students.

Actors: Instructor and Students



3.1.5 Use Case #5 (Attendance)

Author: Akash Verma and Dhruv Rai

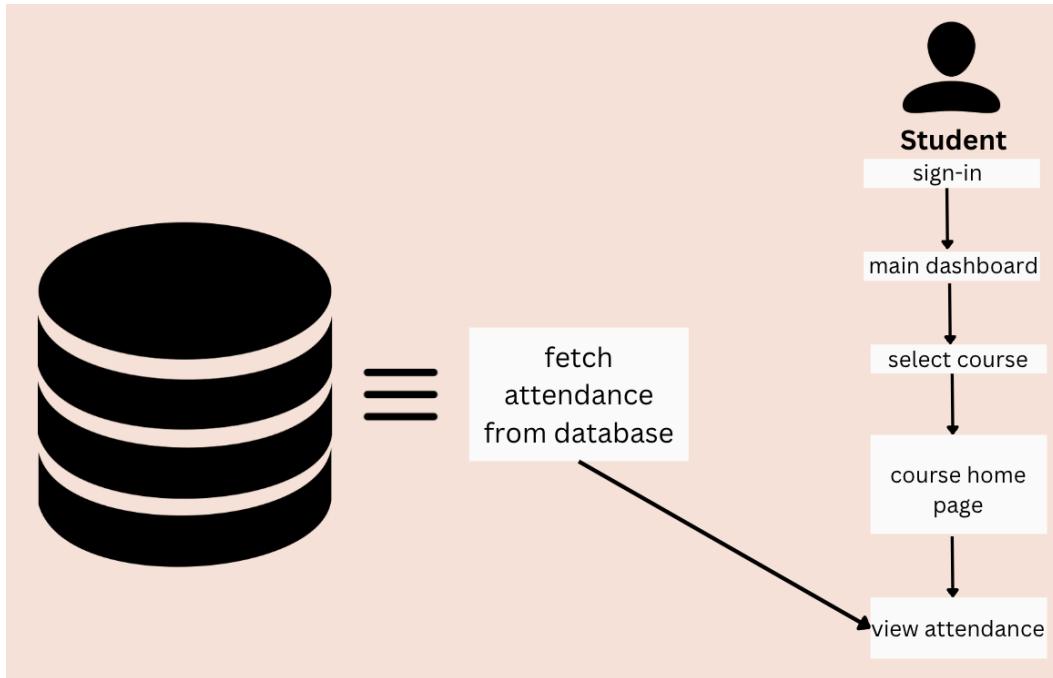
Purpose: Enable users to constantly check and monitor their attendance in different courses they are enrolled in.

Requirements Traceability: This use case is required so that students can check their attendance without visiting any office. It also saves time for biometric staff and at the same time, instructors can keep track of student's attendance in their courses.

Priority: Medium

Pre-conditions: The user should be enrolled in that particular coursework. The biometric office must diligently maintain the attendance database.

Post-conditions: Users can view their current attendance in every course seamlessly. Actors: Student and Instructor



3.1.6 Use Case #6 (Announcements)

Author: Abhijeet Agarwal and Mayur Agrawal

Purpose: To allow users to access announcements for a course.

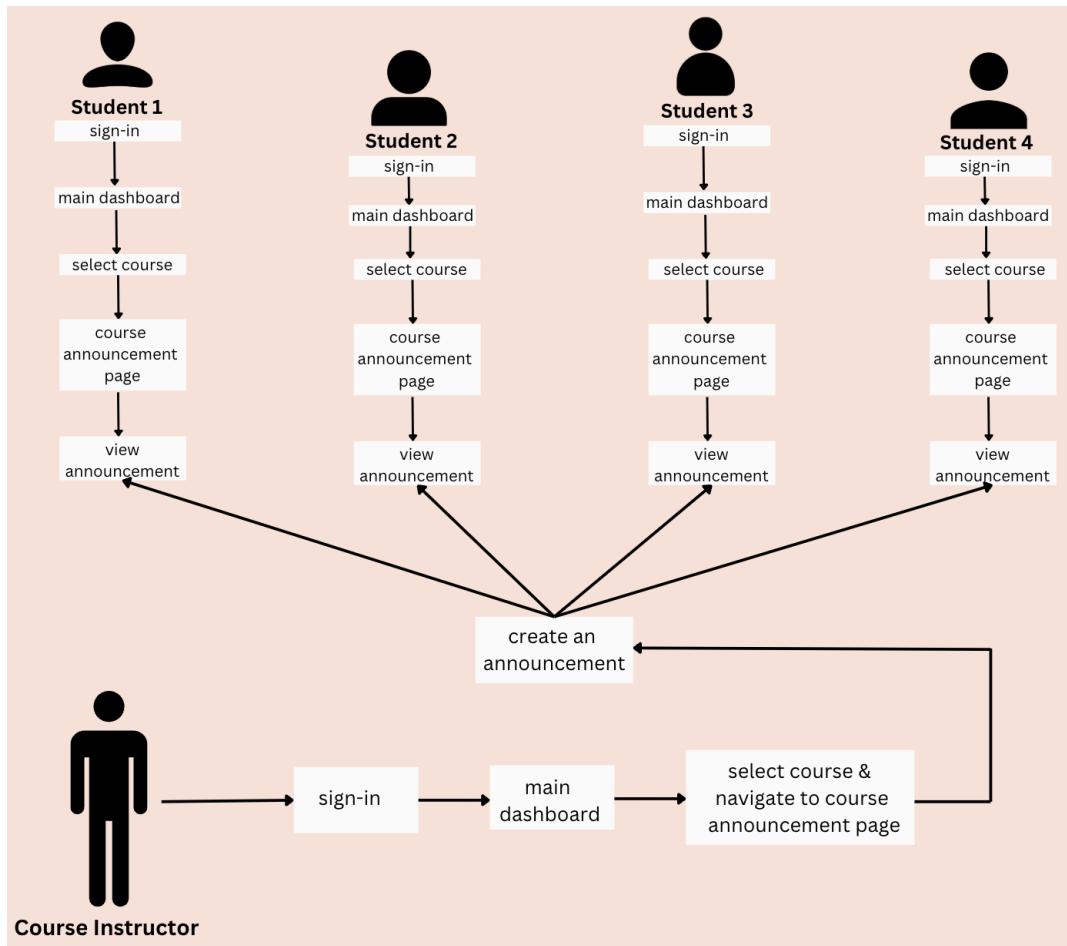
Requirements Traceability: This feature addresses the problem of missing major announcements for a course and helps users manage all important notifications easily.

Priority: High

Pre-conditions: Users are currently enrolled in the course and interested in knowing any general announcement or update about a particular course.

Post-conditions: Users will be informed of any new announcements.

Actors: The primary actors in this use case are the students acting as users, and the faculty publishing an announcement.



3.1.7 Use Case #7 (Results)

Author: Dhruv Varshney and Ved Prakash Vishwakarma

Purpose: To enable instructors to upload and manage student results and view overall course statistics, such as averages and performance trends.

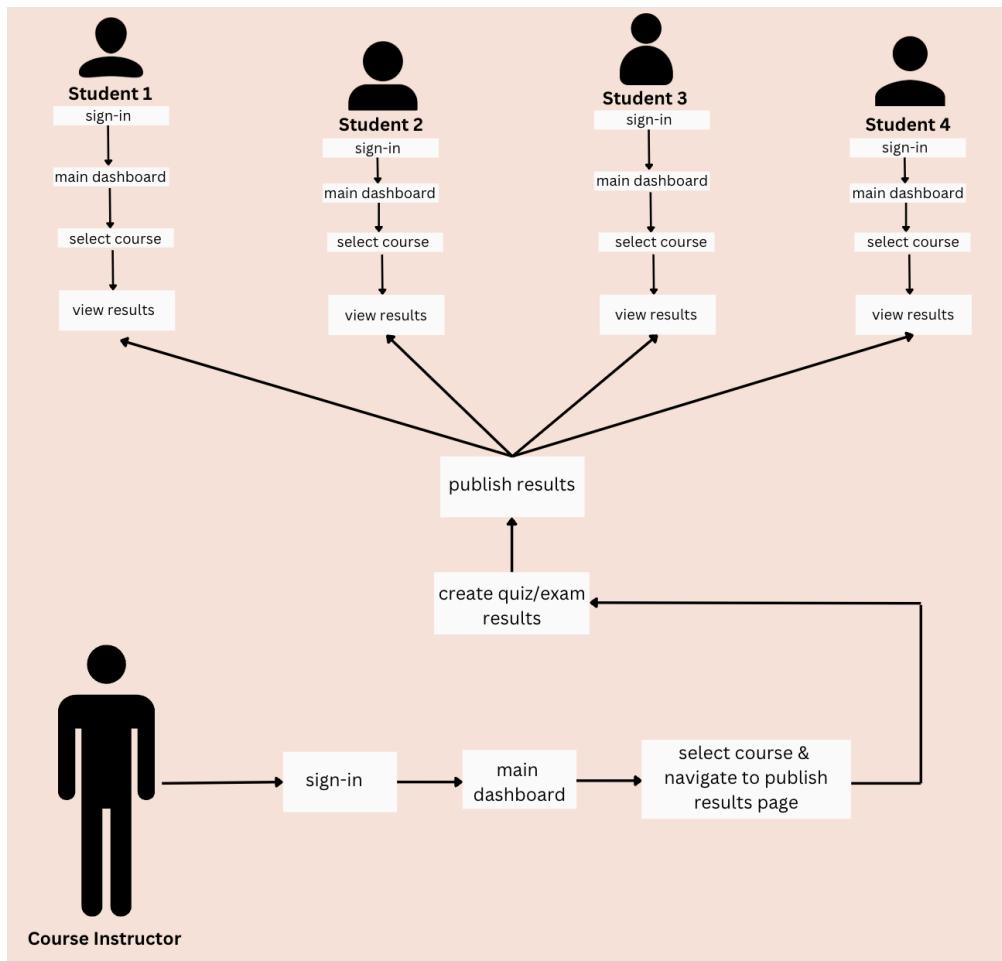
Requirements Traceability: This feature ensures instructors can analyze students' results and course performance to enhance assessments and delivery strategies.

Priority: High

Pre-conditions: The course must be active with enrolled students, grading criteria predefined, and the instructor authorized to manage results.

Post-conditions: Results are uploaded and stored, overall course statistics are displayed, and insights into student performance are provided for analysis. The students can also view graphs and other visual analyses on the basis of their results data.

Actors: The primary actors in this use case are the instructors acting as users.



3.1.8 Use Case #8 (Lectures)

Author: Akash Verma and Aditya Gautam

Purpose: To allow students to access lectures and resources of the course at a single place, at the same time allowing easy and structured distribution of course material for course

instructors. Instructors and students often encounter significant challenges in managing course materials in a structured and organized manner, which our product will solve.

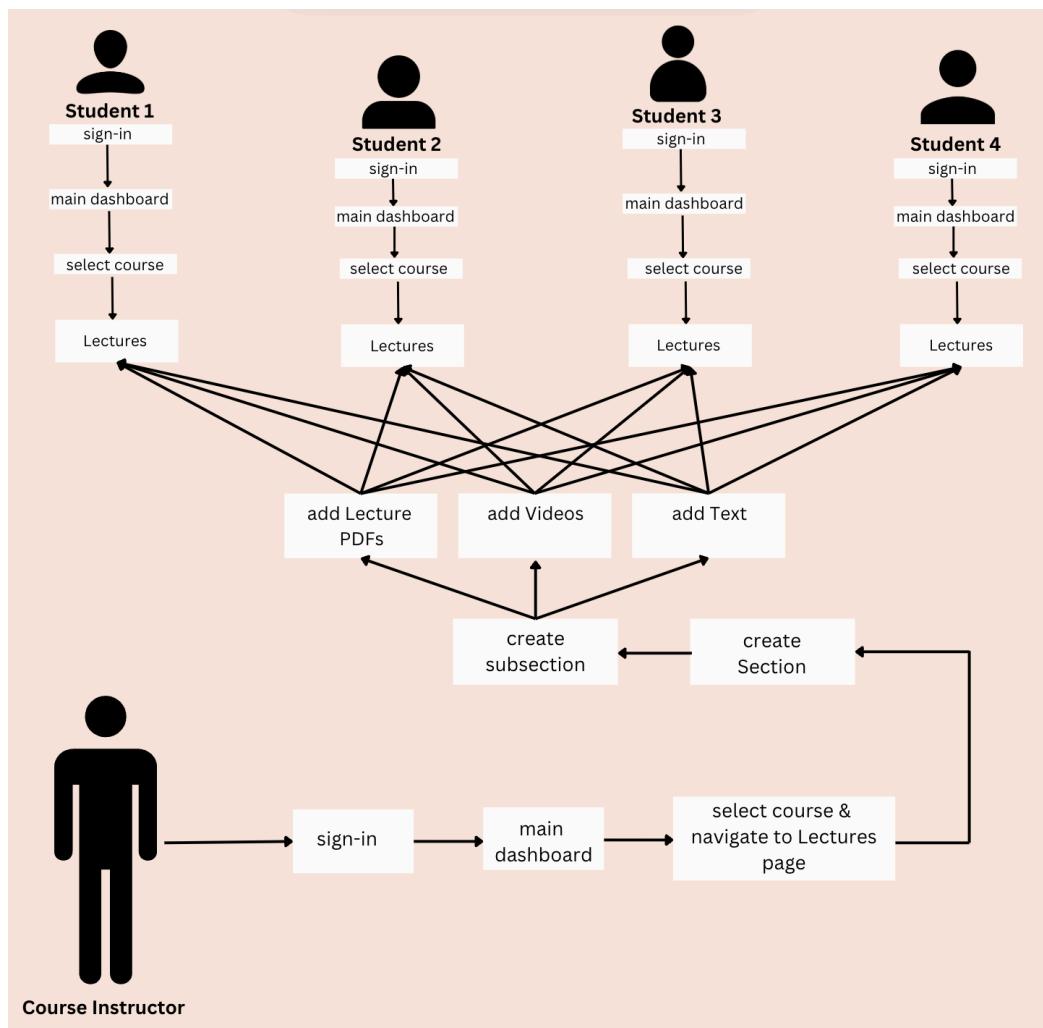
Requirements Traceability: Students should be able to view PDFs, text, and videos(tentatively embedded YouTube videos) uploaded by the Faculty, all of which should be neatly organized in sections and subsections.

Priority: High

Pre-conditions: The student must be enrolled on the course. The Lectures and course works of all courses must be recorded and updated in the system database.

Post-conditions: Users are able to access various Lecture materials.

Actors: The primary actors in this use case are the students and course instructors



3.1.9 Use Case #9 (Overall Performance History)

Author: Aryan Bansal and Ved Prakash Vishwakarma

Purpose: To allow students to see their templates and grades.

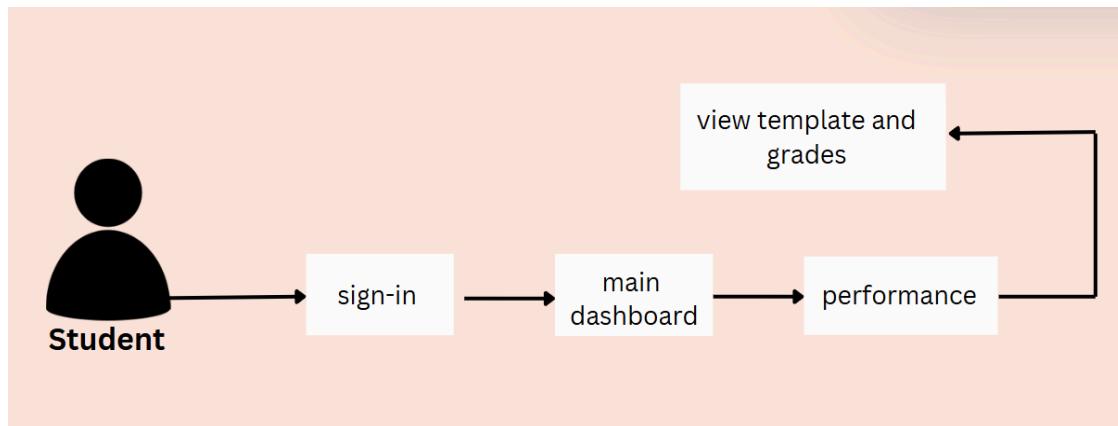
Requirements Traceability: Templates of the department of students are displayed with crucial information like their pending, failed, current, and completed courses, along with grades from completed courses, helping them manage courses and evaluate their performance effectively.

Priority: Medium

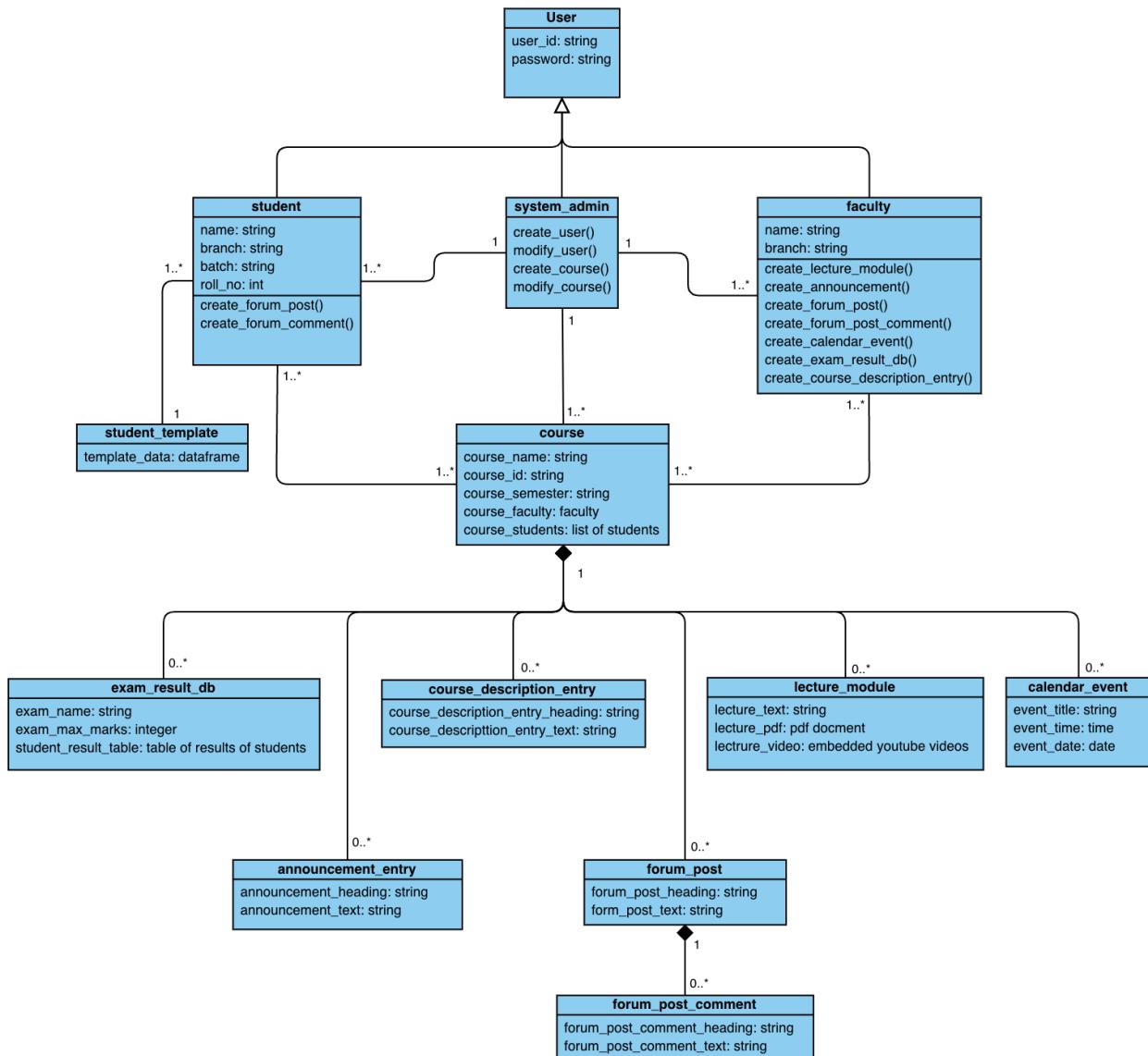
Pre-conditions: The student must be registered.

Post-conditions: The user can see his template and grades.

Actors: The primary actors in this use case are the students acting as users.



3.2. Class Diagram



The classes used in our Product have been described below:

Classes	Description	Methods and Attributes	Datatype
user	The overarching class which defines users of the product. Other user classes inherit its attributes like user_id, and password.	user_id	string
		password	string

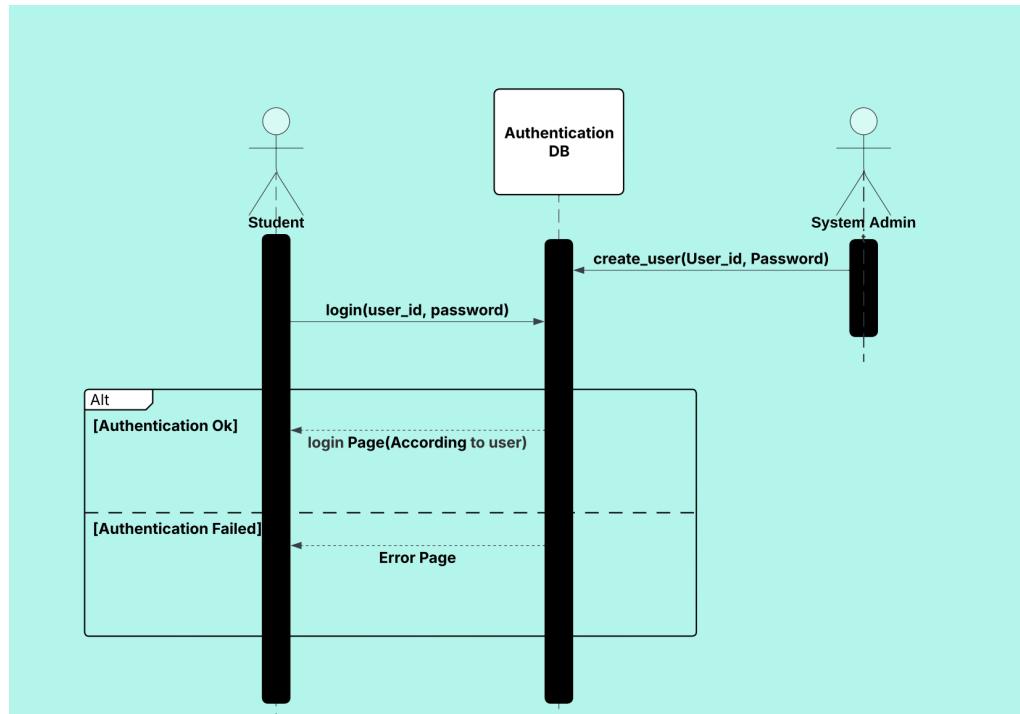
system_admin	The controlling user class capable of creating new users, modifying existing users, creating courses and assigning faculty and students to them.	create_user	function
		modify_user	function
		create_course	function
		modify_course	function
student	The end-user class which consumes other data published by the faculty user class, and can communicate with other students via the forum. The student can exist in the states of being enrolled in particular courses.	name	string
		branch	string
		batch	string
		roll_no	int
		create_forum_post	function
		create_forum_post_comment	function
		enrolled_courses	list of course
faculty	The end-user class which produces other data consumed by the student user class. The faculty can exist in the states of being the instructor of particular courses.	name	string
		branch	string
		create_lecture_module	function
		create_announcement	function
		create_forum_post	function
		create_forum_post_comment	function

		ent	
		create_calender_event	function
		create_exam_result_db	function
		create_course_description_entry	function
course	A paramount Class Variable which contains within itself a multitude of other Class Variables like lecture_module, calender_event etc. It has two primary fields: one is a variable of the Class faculty, while the other is a list of variables of the Class student.	course_name	string
		course_id	string
		course_semester	string
		course_faculty	faculty
		course_students	list of students
lecture_module	A Class Variable created by a faculty which appears in the <i>Lectures</i> section and can contain PDFs, text, and embedded videos.	lecture_text	string
		lecture_pdf	pdf document
		lecture_video	embedde d youtube video
calender_event	A Class Variable created by a faculty which appears in the <i>Calendar</i> section and describes an event's title, time and date.	event_title	string
		event_time	time
		event_date	date

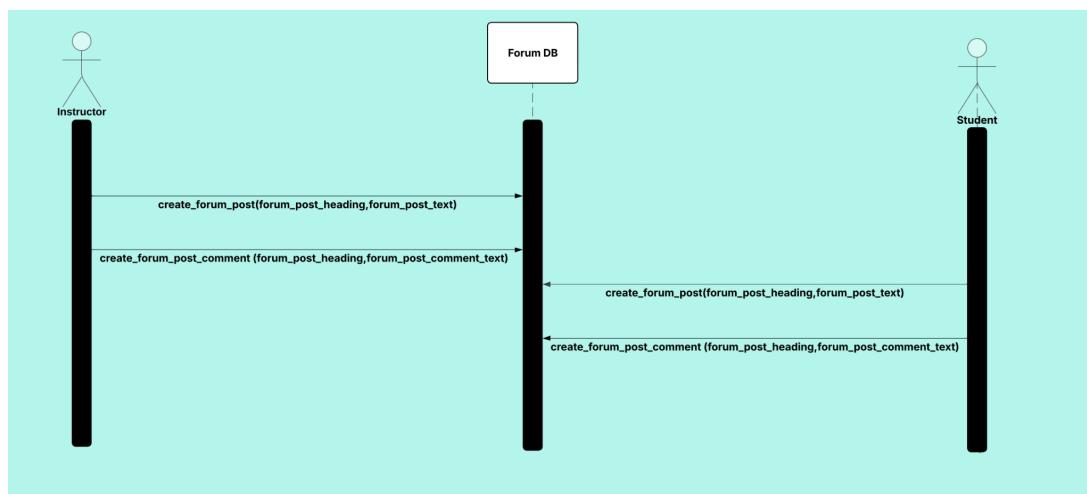
announcement_entry	A Class Variable created by a faculty which appears in the <i>Announcements</i> section and describes an Announcements title and text.	announcement_heading	string
		announcement_text	string
exam_result_db	A Class Variable created by a faculty which defines the results of all students for a particular exam in a particular course.	exam_name	string
		exam_max_marks	integer
		student_result_table	table of results of students
course_description_entry	A Class Variable created by a faculty which appears in the <i>Course Home</i> section, and describes aspects of a course like grading scheme.	course_description_entry_heading	string
		course_description_entry_text	string
forum_post	A Class Variable created by a student or a faculty which defines the title and text of a forum post.	forum_post_heading	string
		forum_post_text	string
forum_post_comment	A Class Variable created by a student or a faculty which defines the title and text of a forum post's comment.	forum_post_comment_heading	string
		forum_post_comment_text	string
student_template	A static predefined table-like Class which details a student's template as decided by his/her branch.	template_data	dataframe

3.3. Sequence Diagrams

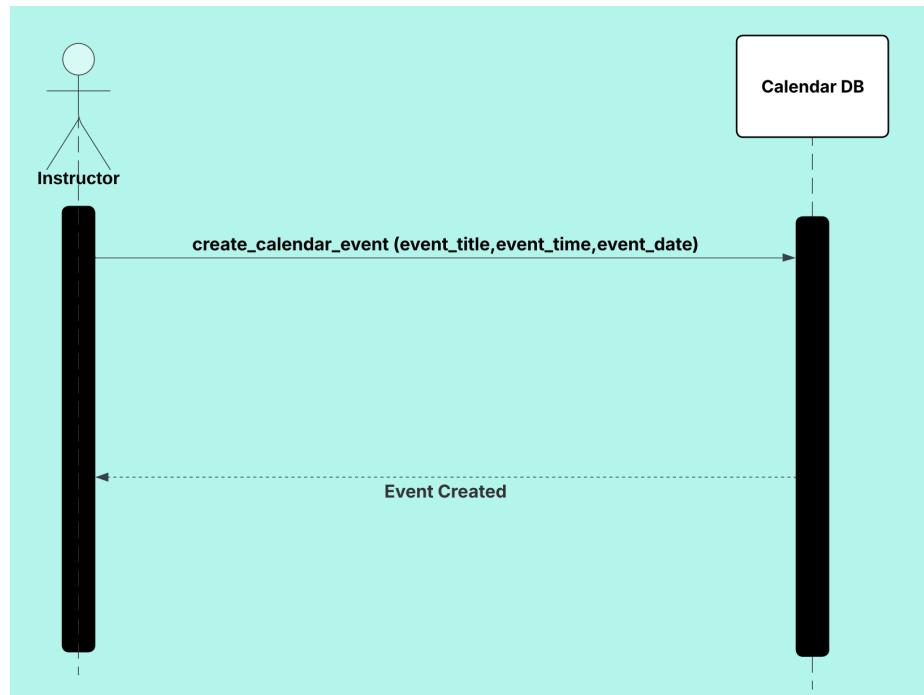
3.3.1 Log-in



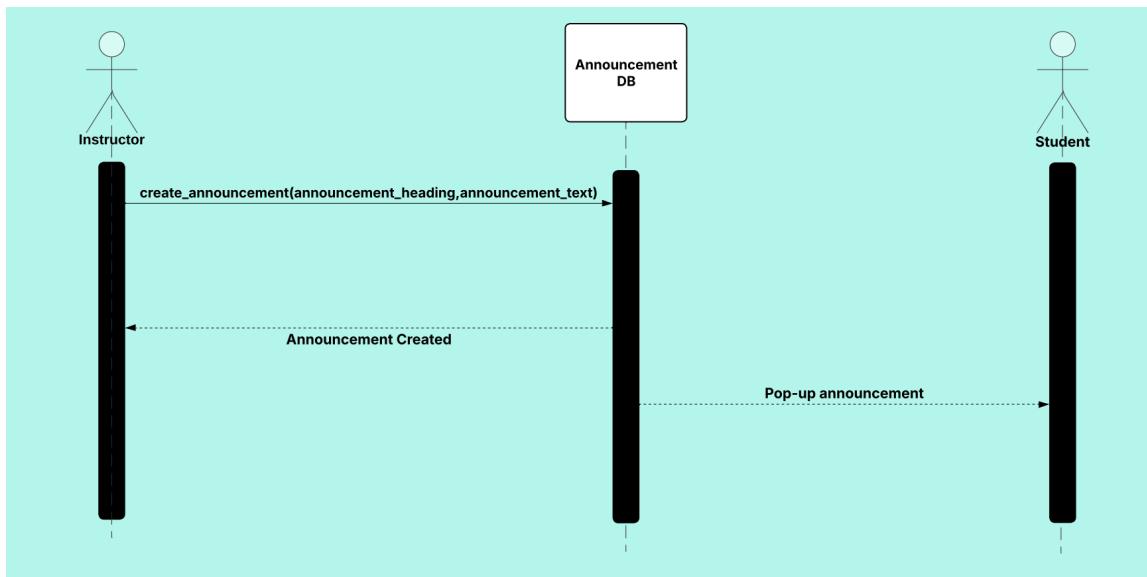
3.3.2 Forum



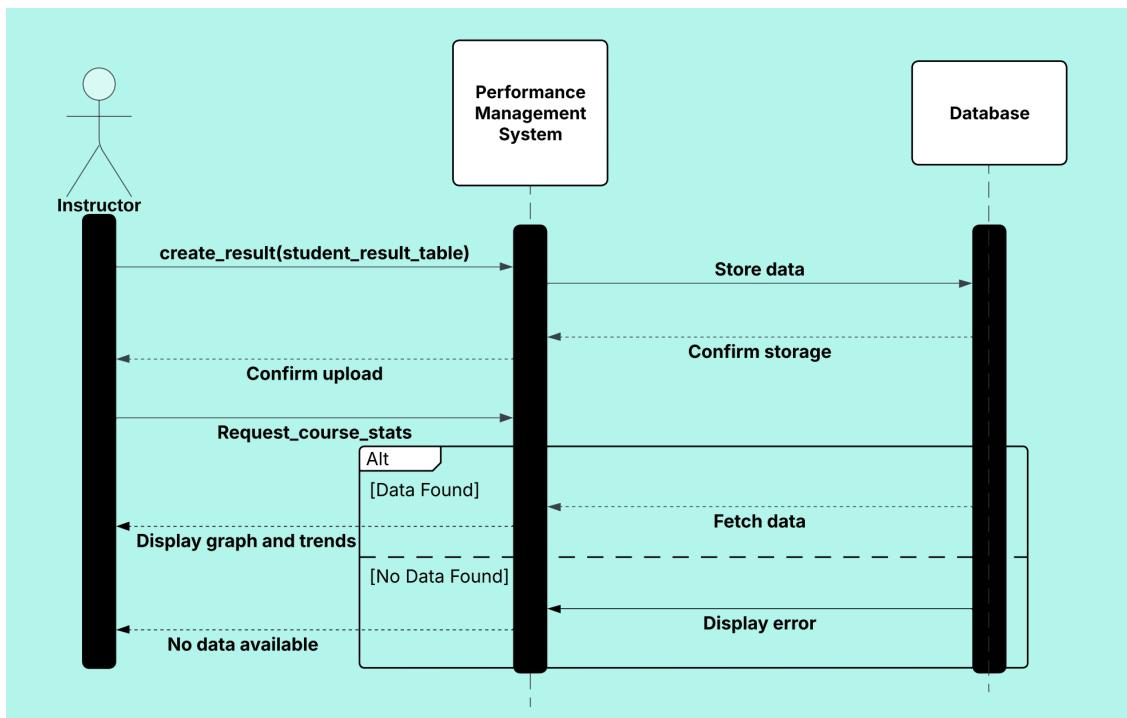
3.3.3 Calendar



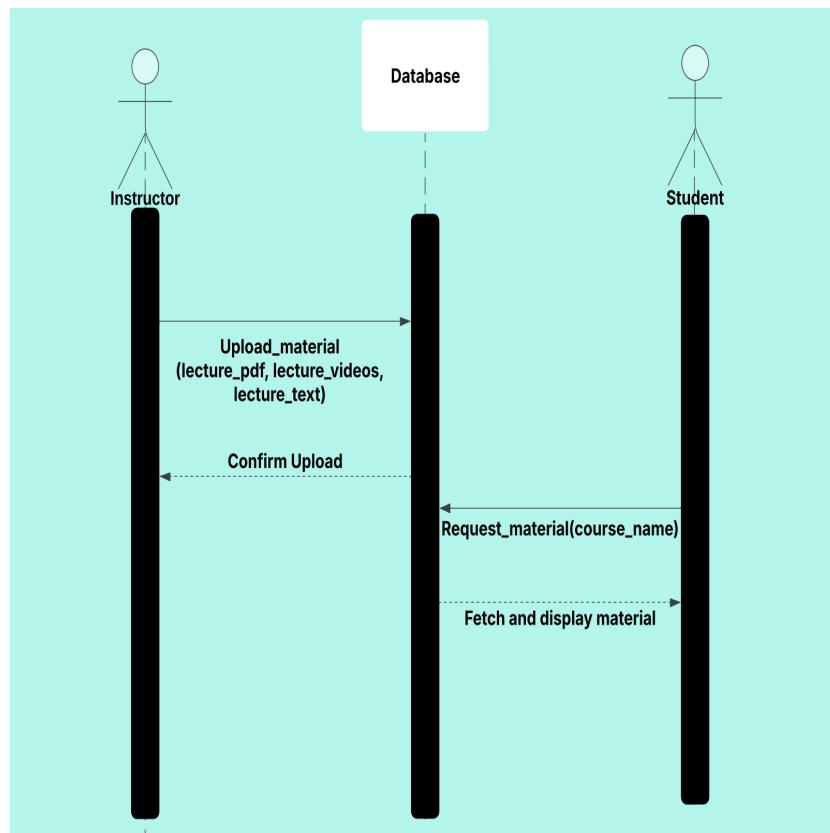
3.3.4 Announcements



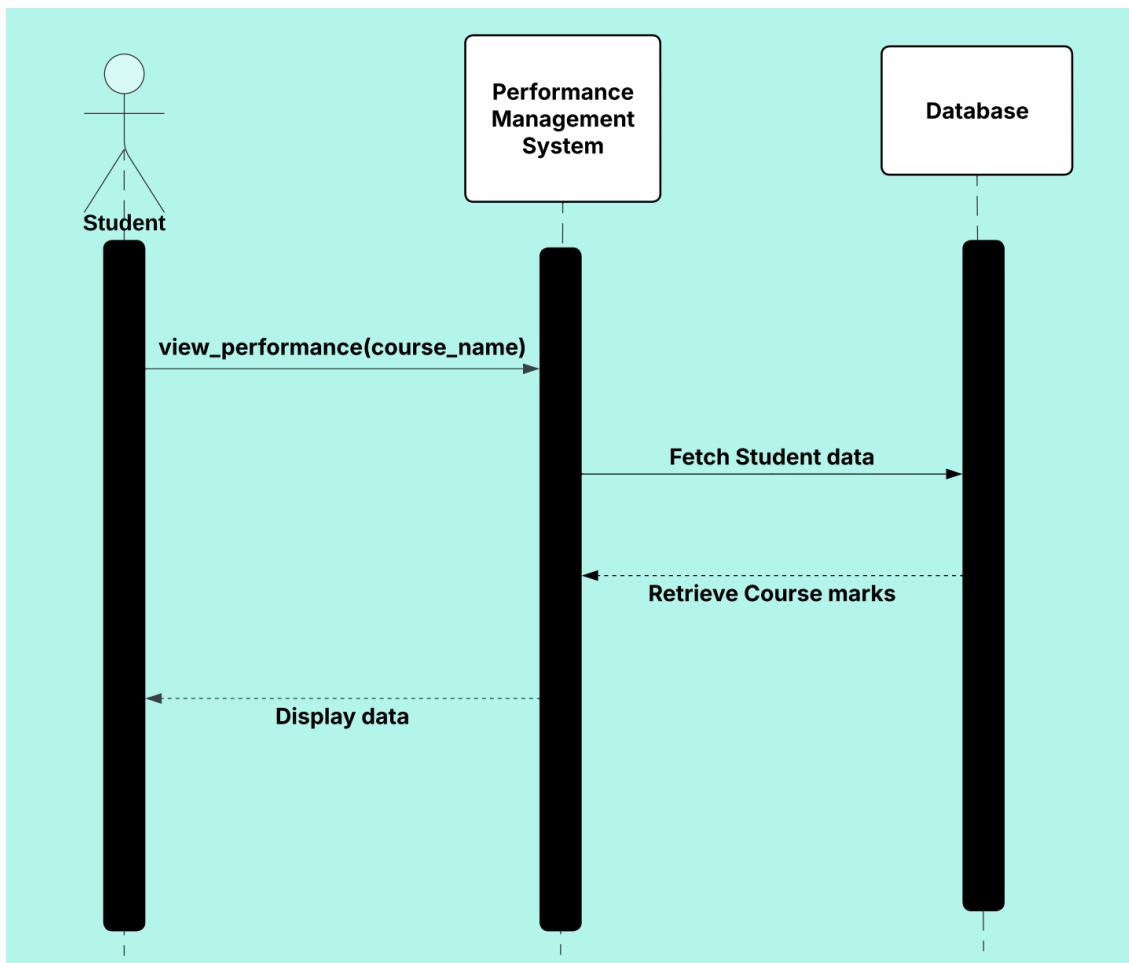
3.3.5 Results



3.3.6 Resources

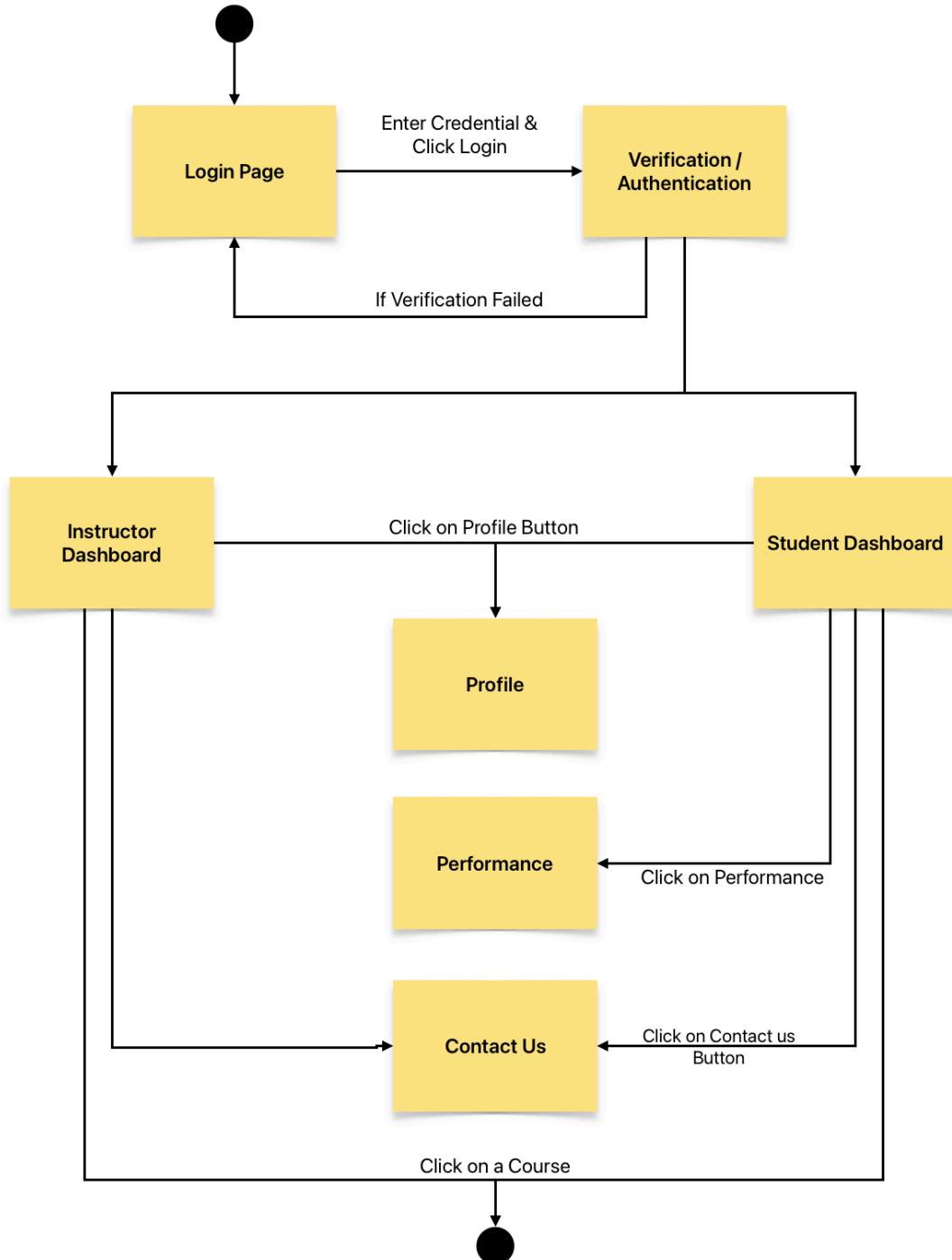


3.3.7 Overall Performance

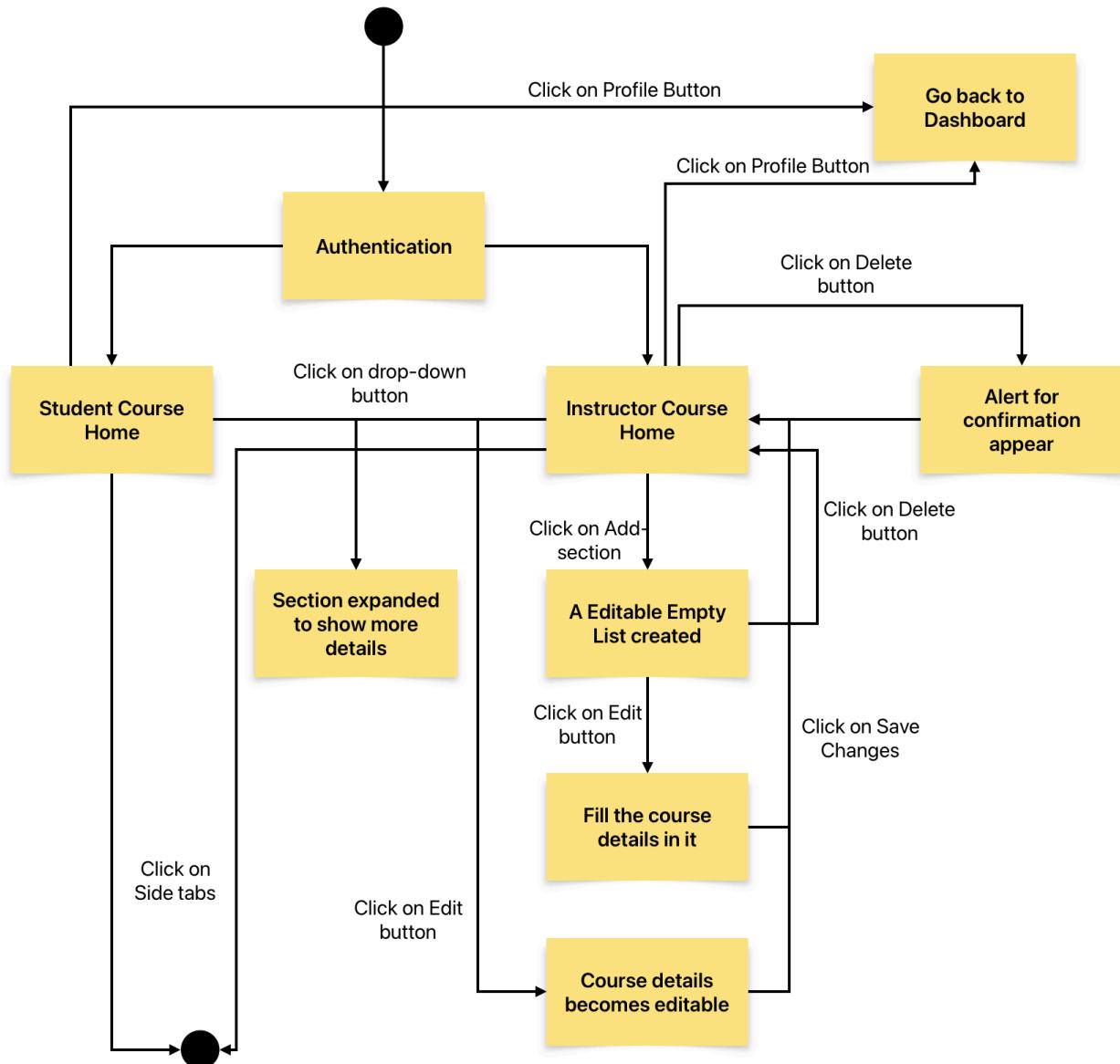


3.4. State Diagrams

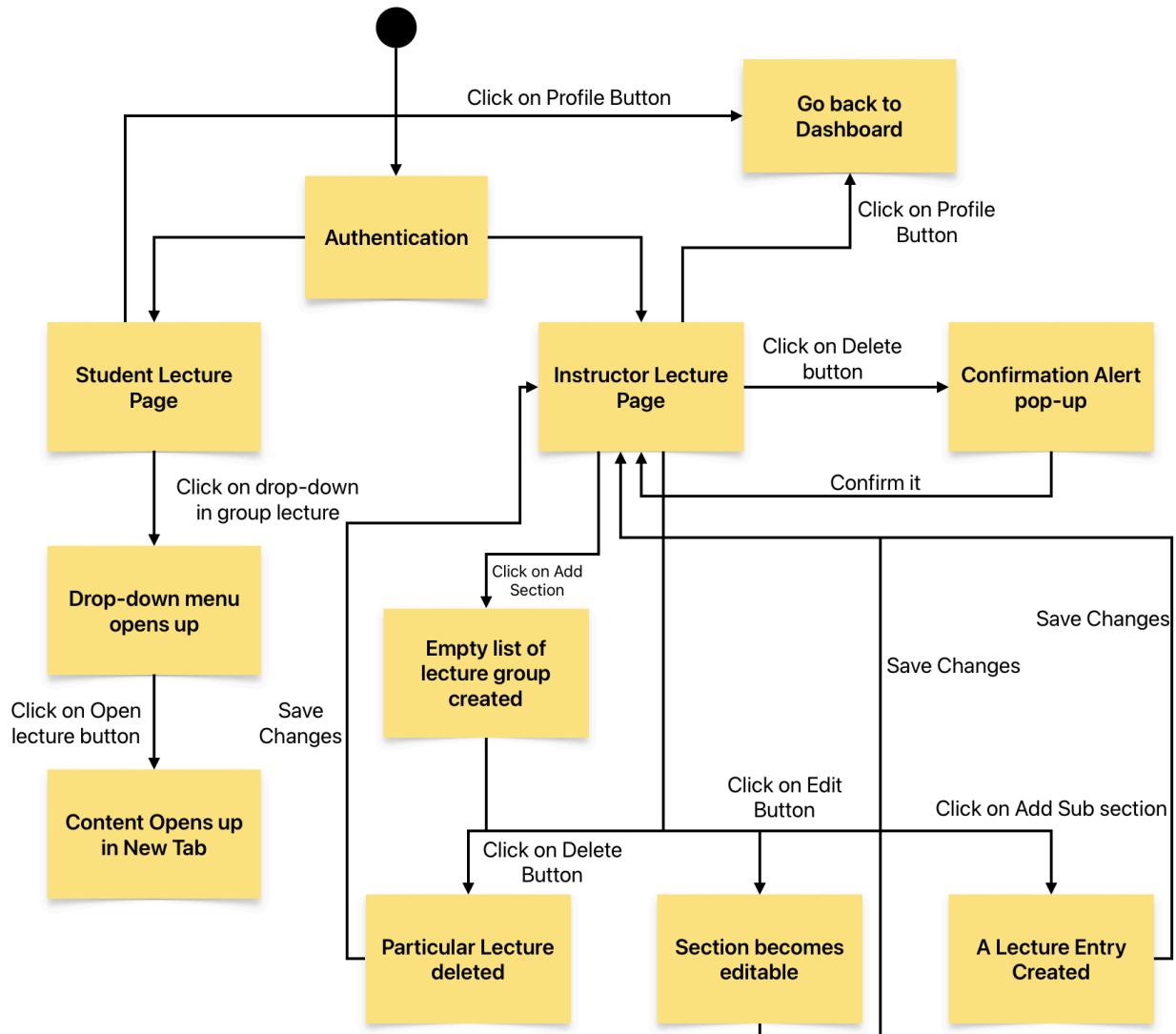
3.4.1 Login Page and Dashboard



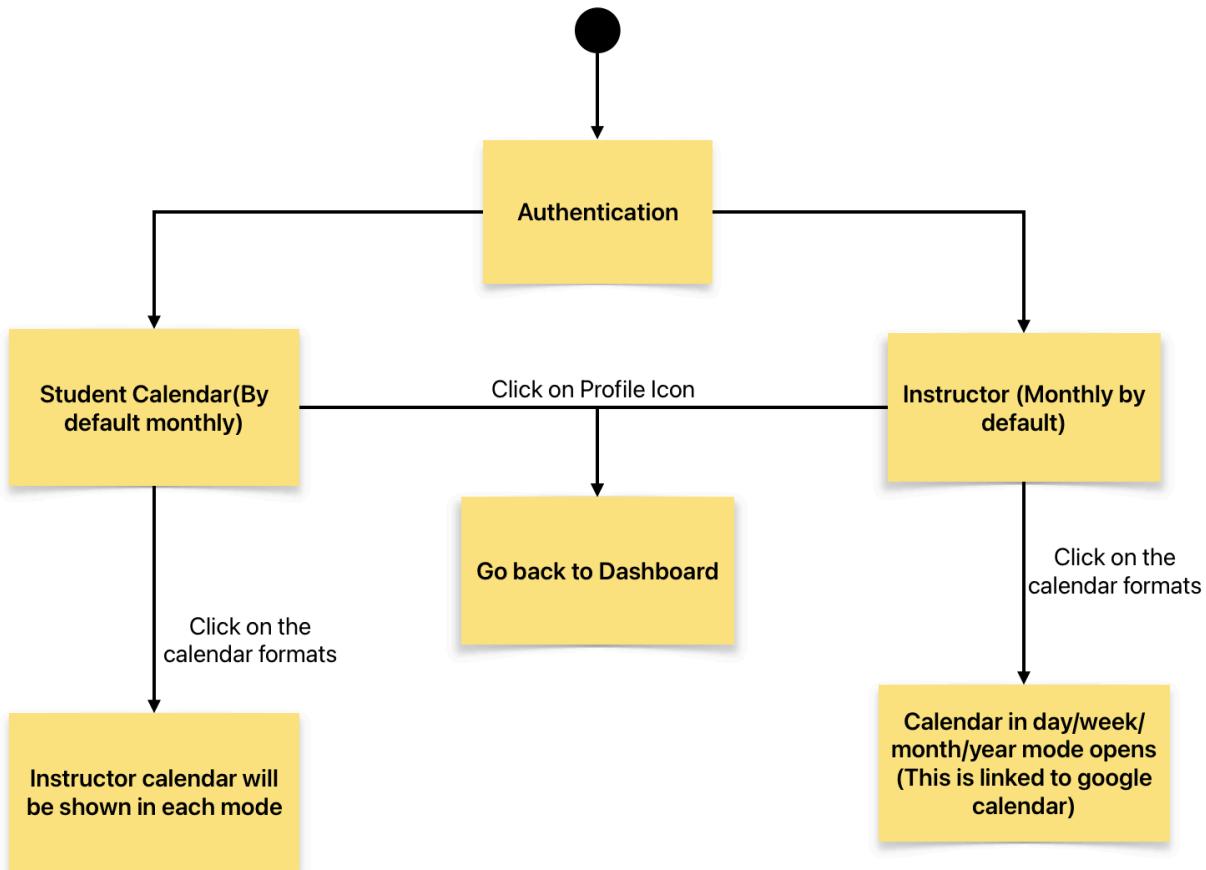
3.4.2 Course Home Page



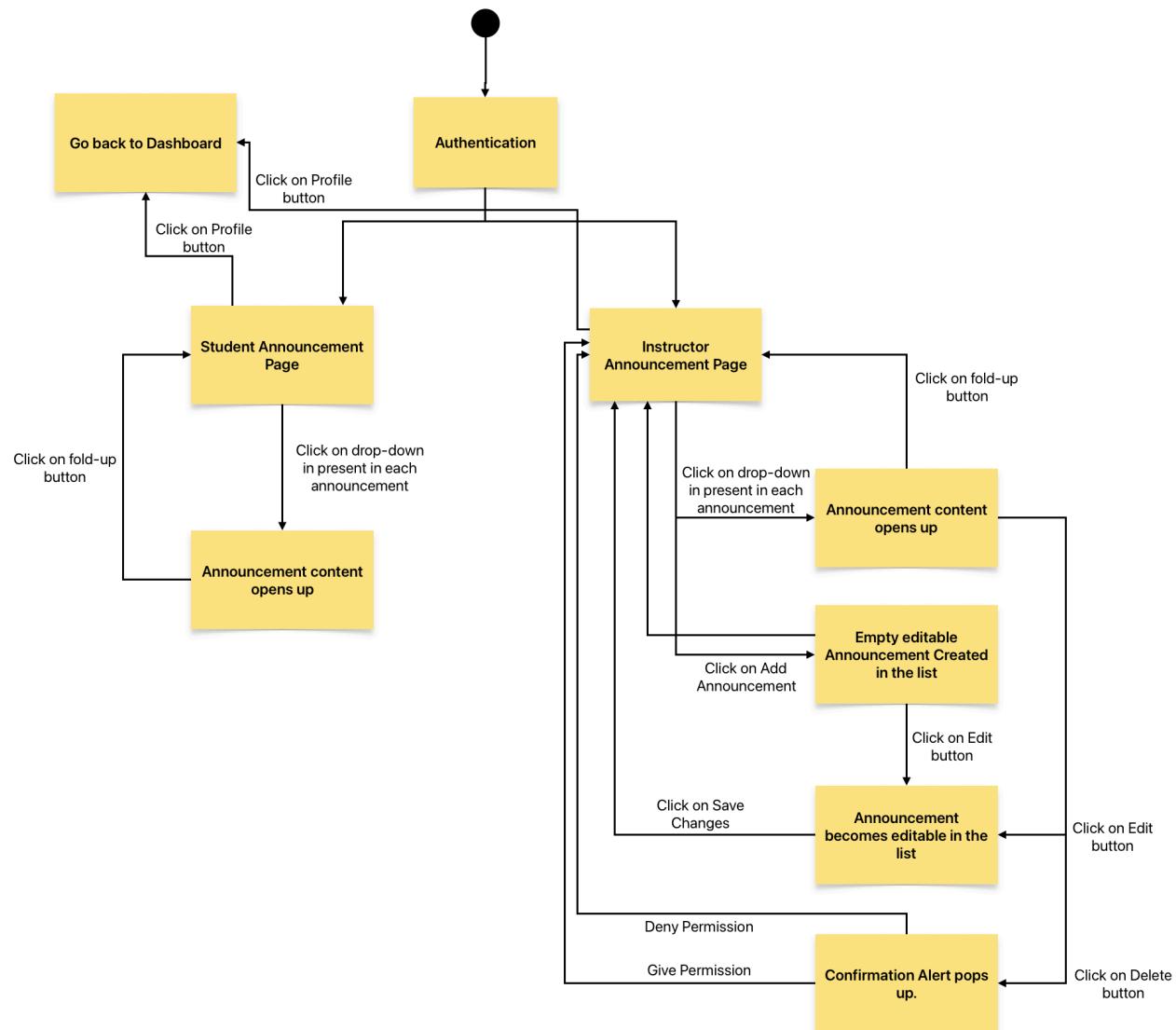
3.4.3 Lectures Page



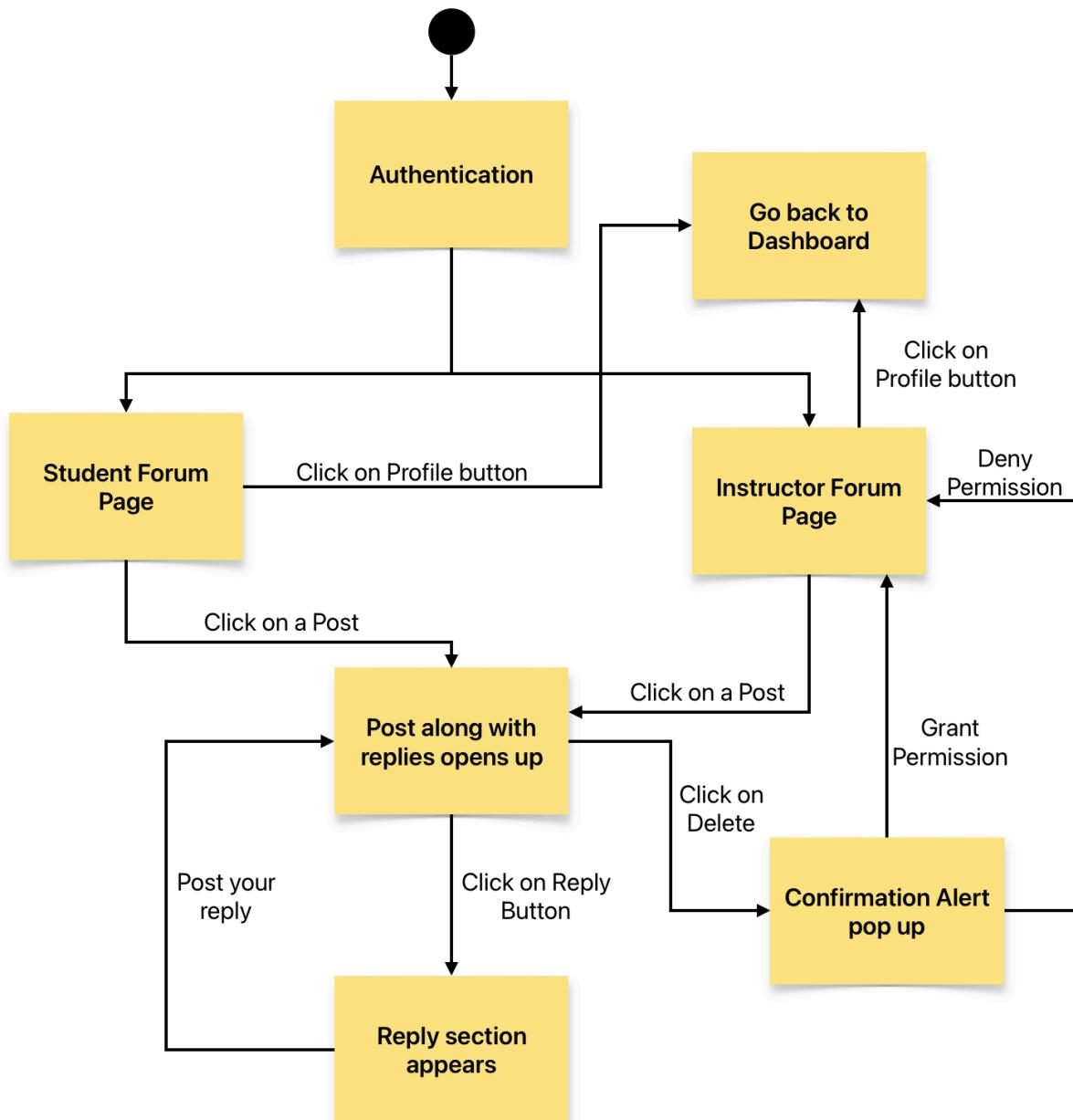
3.4.4 Calendar Page



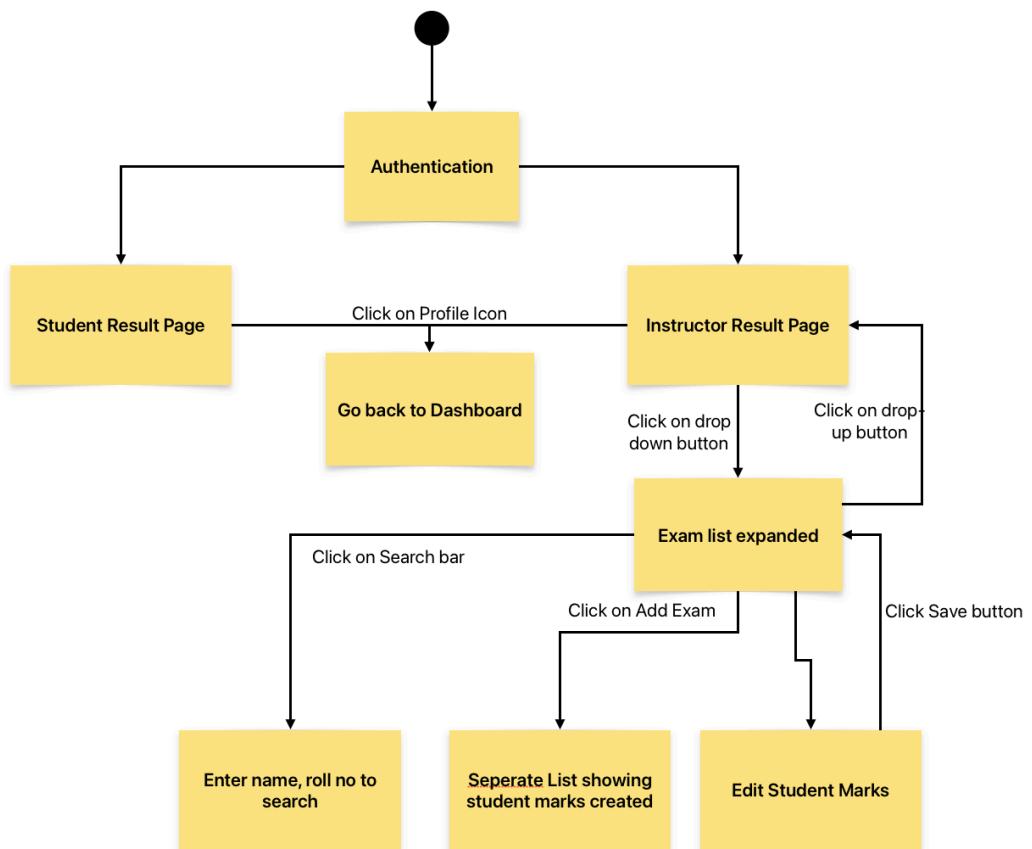
3.4.5 Announcement Page



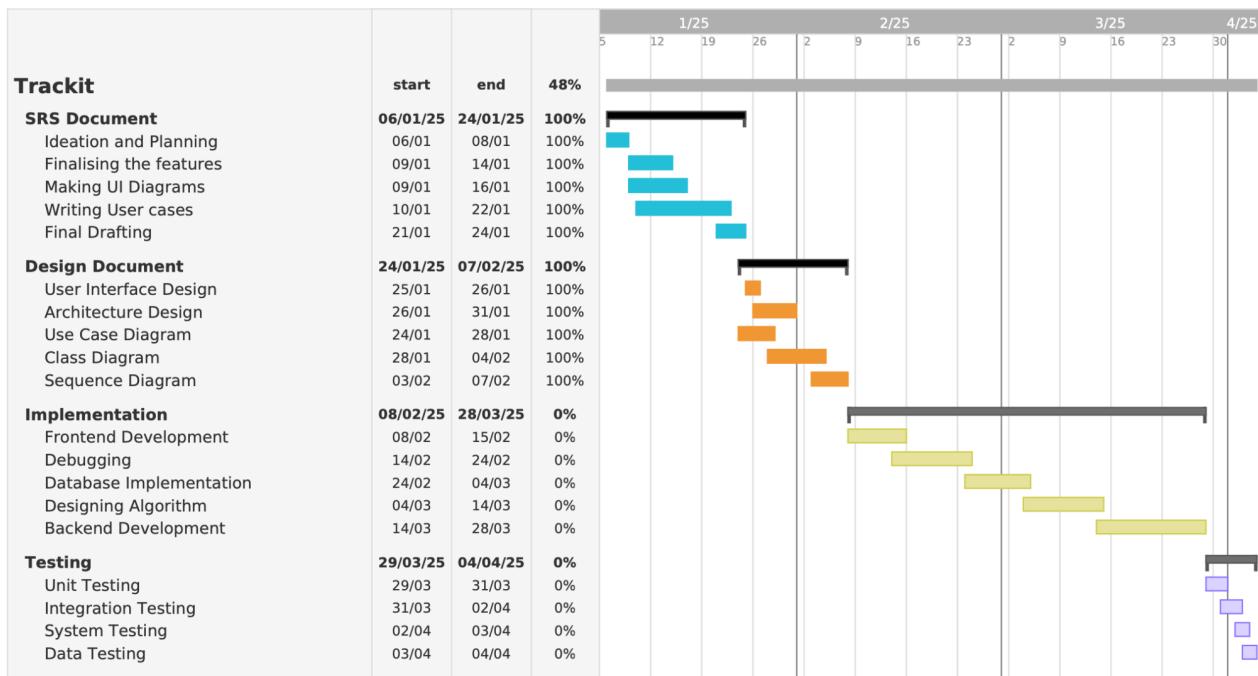
3.4.6 Forum Page



3.4.7 Result Page



4. Project Plan



Team Member	Task
Aditya Gautam	Full Stack development, Algorithm designing, Alpha testing
Aayush Singh	Backend development, Debugging, Unit testing
Dhruv Rai	Full Stack development, Code improvement, Alpha testing
Ved Prakash Vishwakarma	Full Stack Development, Debugging, Debugging
Sharique Ahmad	Frontend development, Code improvement, Unit testing
Dhruv Varshney	Frontend development, Algorithm designing, Debugging
Mayur Agrawal	Frontend development, Debugging, System testing
Akash Verma	Frontend development , Unit testing, Addressing feedback
Rahul Ahirwar	Backend development, Unit testing, System testing
Aryan Bansal	Full Stack development, Code improvement, Alpha testing
Abhijeet Agarwal	Backend development, Debugging, Unit testing

Appendix A - Group Log

Since the beginning of the project, our entire team has been very enthusiastic. Apart from following offline meetings, we have formed a Whatsapp group and Discord server for effective communication. Log Maintained by Akash Verma.

DATE	TIMINGS	DURATION	AGENDA AND OUTCOMES
Jan 27, 2024	14:00 - 17:00	3 hrs.	<ul style="list-style-type: none"> • Distribution of tasks for design document • Divided into 4 teams for working on the design document.
Jan 30, 2024	9:00 - 11:00	2 hrs.	<ul style="list-style-type: none"> • Explained approach of respective work to each other. • Discussed doubts among team members.
Feb 01, 2024	20:00 - 21:00	1 hrs.	<ul style="list-style-type: none"> • Discussed & clarified our doubts with TAs.
Feb 02, 2024	14:00-16:00	2 hrs.	<ul style="list-style-type: none"> • Reviewed the work done by the four teams and suggested improvements. • Clarified any irregularities in different tasks
Feb 04, 2024	15:00 - 16:30	1hr 30 min	<ul style="list-style-type: none"> • Discussed project plan among team members.
Feb 06, 2024	17:00 - 18:45	1hr 45 mins	<ul style="list-style-type: none"> • Proofread the document and correct any irregularities.
Feb 07, 2024	18:00 - 19:30	1hr 30mins	<ul style="list-style-type: none"> • Finalised the Document