ARPO

Design Document

Version 1.1

Prepared by

Group #: 14   **Group Name: Anonymous**

Shubhan R	200971	shubhan.ravi@gmail.com
Akansh Agrawal	180050	akanshcs2020@gmail.com
Akshan Agrawal	180061	akshancs2020@gmail.com
Lakshay Rastogi	180378	rastogilakshay31@gmail.com
Atharv Tyagi	170174	atharvlalittyagi@gmail.com
Jahnavi Kairamkonda	200482	jahnavi26k@gmail.com
Pranav Singh	190622	pranavsingh560@gmail.com
Manoj Kumar	180409	manojkumar.nvsd@gmail.com
Keshav Kumar	180353	keshavkr54321@gmail.com
Piyush Senwar	180512	psenwar@gmail.com

Course: **CS253**
Mentor TA: **Shatroopa Saxena**
Date: **28 April 2022**

CONTENTS

Revisions

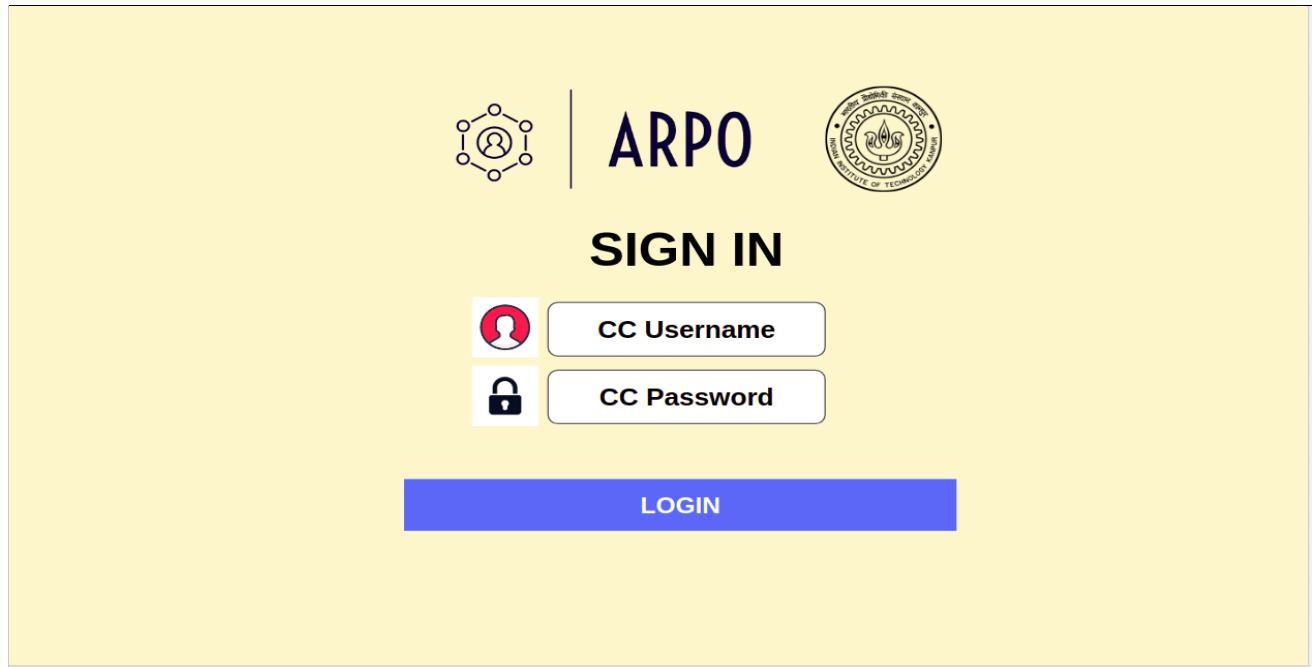| Version	Primary Author(s)	Description of Version	Date Completed
1.0	Akansh Agrawal Jahnavi Kairamkonda Shubhan R Akshan Agrawal Pranav Singh Manoj Kumar Atharv Tyagi Keshav Kumar Piyush Senwar Lakshay Rastogi	First Version (Version 1.0): First Design Document/Draft for ARPO	15/02/2022
1.1	Akansh Agrawal Jahnavi Kairamkonda Shubhan R Akshan Agrawal Pranav Singh Manoj Kumar Atharv Tyagi Keshav Kumar Piyush Senwar Lakshay Rastogi	Updated content as per Version 1.1 of SRS and added log to the front page	27/04/2022

1 Context Design

1.1 Context Model

The context diagram as shown in the figure explains what are the basic entities (systems) which our system connects and is connected to. The query status and notifications are sent from the system to various other systems like admins, technical staff, students, instructors, etc. Students send requests for query details, update status and the responses and view profile to the system. Instructors, Tutors/TAs and Admins send information regarding resolving queries and status updation to the system for the queries requests received from the system. Even Instructors, Tutors and TAs can send requests for posting query & announcements to the system. The admins too can send requests for posting queries regarding technical issues and create general announcements to the system. Technical staff sends updating information and information related to resolving technical issues to the system.
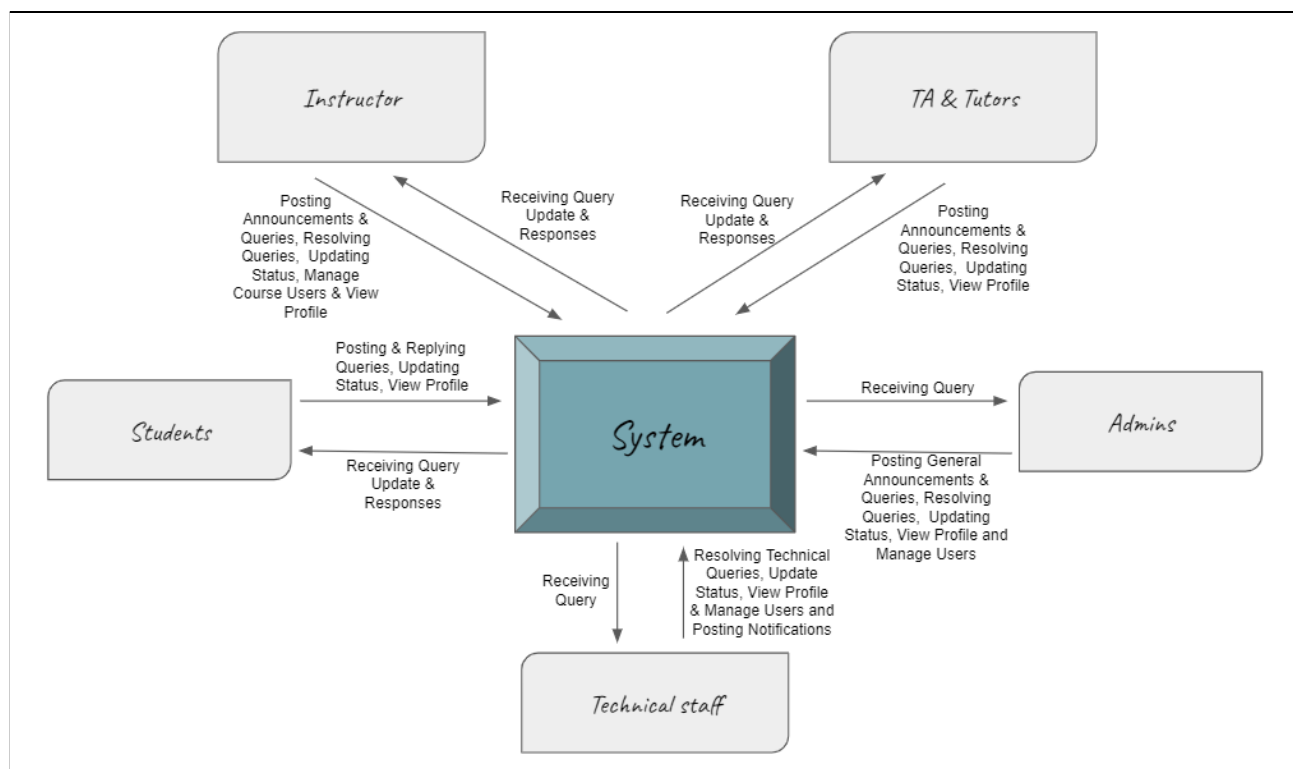
1.2 Human Interface Design

LOGIN PAGE:
The first page of our platform is a login interface from where all the actors of the platform can login to their respective platform dashboards. This particular page may look something like as shown below.

Here, participants or actors can login using their CC username and password. Major actors here are Students, Mentors, Academic Admins and Technical Support Staff.

STUDENTS' INTERFACE:

After a student logs into the platform, she enters his "Home Page" or "Dashboard" as shown below.

The student profile home page displays some general information like Notifications, Important Links and News from the Institute. These are not course specific, rather more institute level relevant information.

From options given in top right, the student can access
- The courses he/she is enrolled in.
- The personal student profile.
- The Academic Admins list and send queries to them.
- The technical support staff.

Courses Tab:
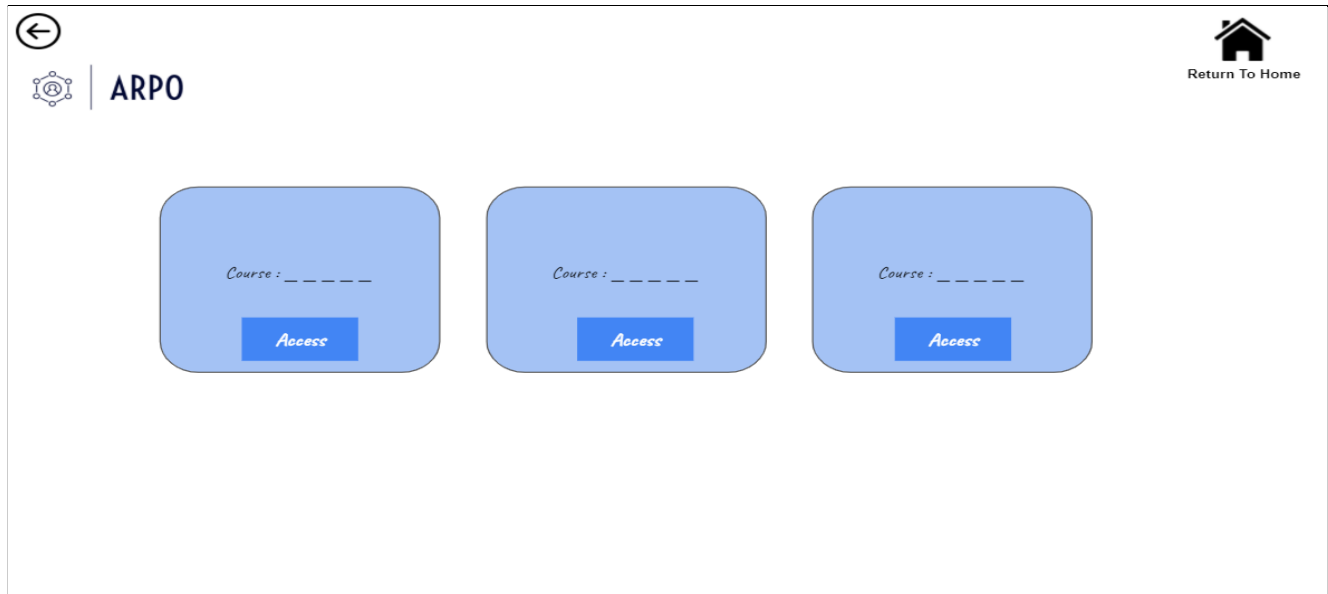
This is how the courses section in the student dashboard looks like. She can access any of the courses she's enrolled in from here and it displays the following view.

Course Home Page:

After entering in any of the courses, the student can access the course notifications, General Forums, Private Queries and her Course Profile. The following pictures show views of each of these interfaces.
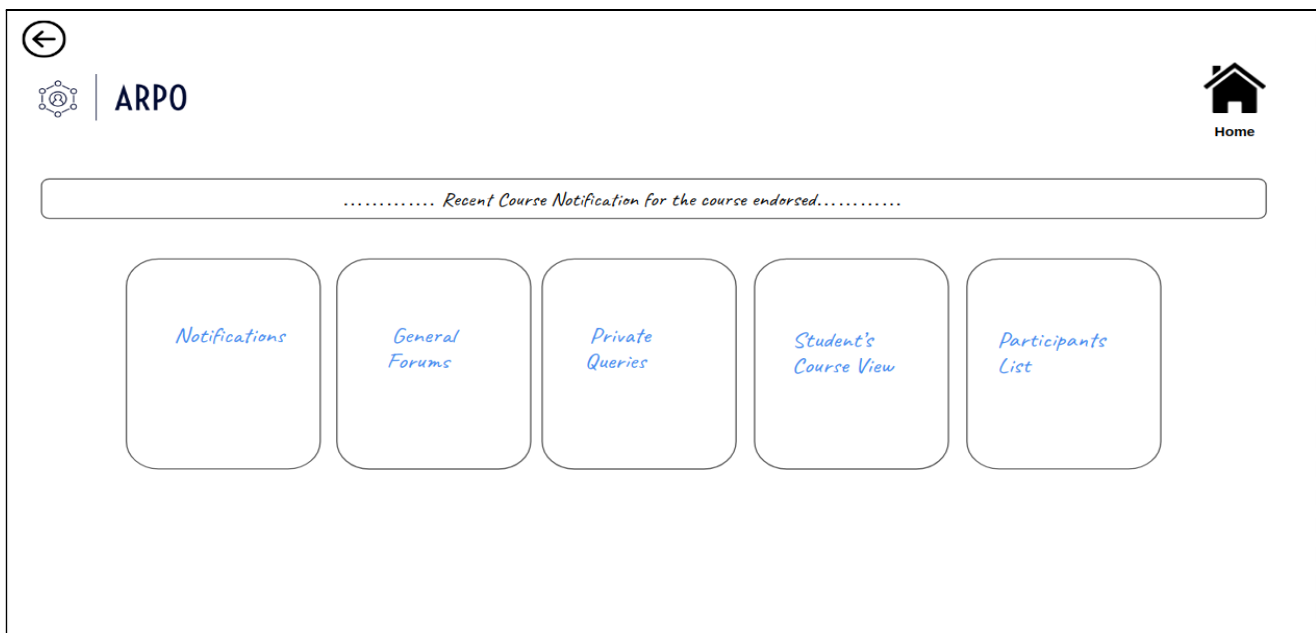
Notifications Tab:

The overall course announcements and query notifications have been separated as shown.

General Forum/ Private Query Tab:
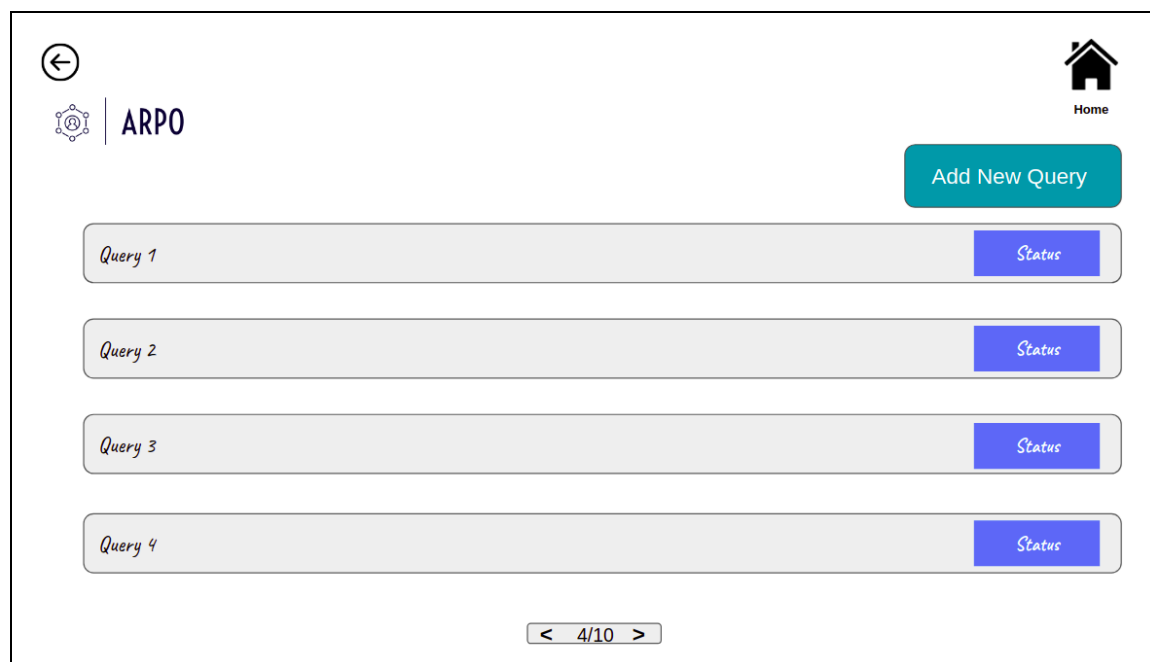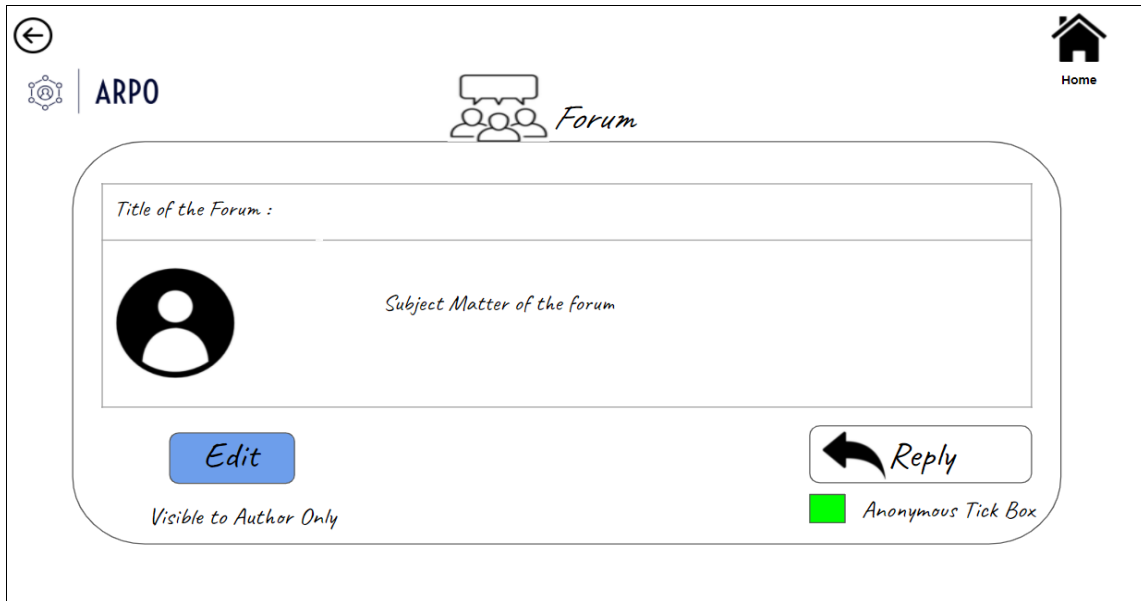
Add Query tab for General and Private Queries:

There is an anonymous tick box also to hide identity.

View of Received/Posted Query Forum (General/Private):

Only the author of the post can edit the forum posted. The viewers can reply to it anonymously or with their original identity as they wish for general forum.

General:

Private :

Students Course Profile:

It displays statistics of queries like number of queries asked etc. and marks of assessments.

Technical Support and Admin Queries:

Instructors can also raise queries/concerns to the technical staff team and admins in a manner very similar to students as shown below.

INSTRUCTORS' INTERFACE:

The instructor's Dashboard looks very similar to the students' dashboard as shown below.
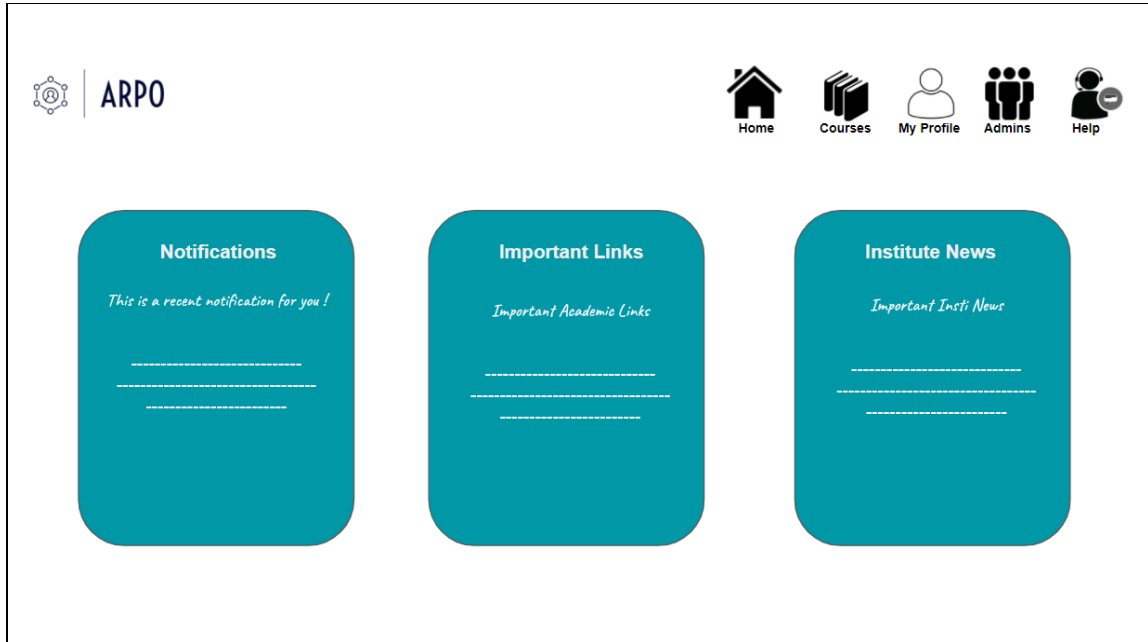

The courses section in the top right corner in the instructor's profile offers more authorities which are not provided to the students. To see this, look at the following images which shows the all courses page and course home page for an instructor.

Courses Page:
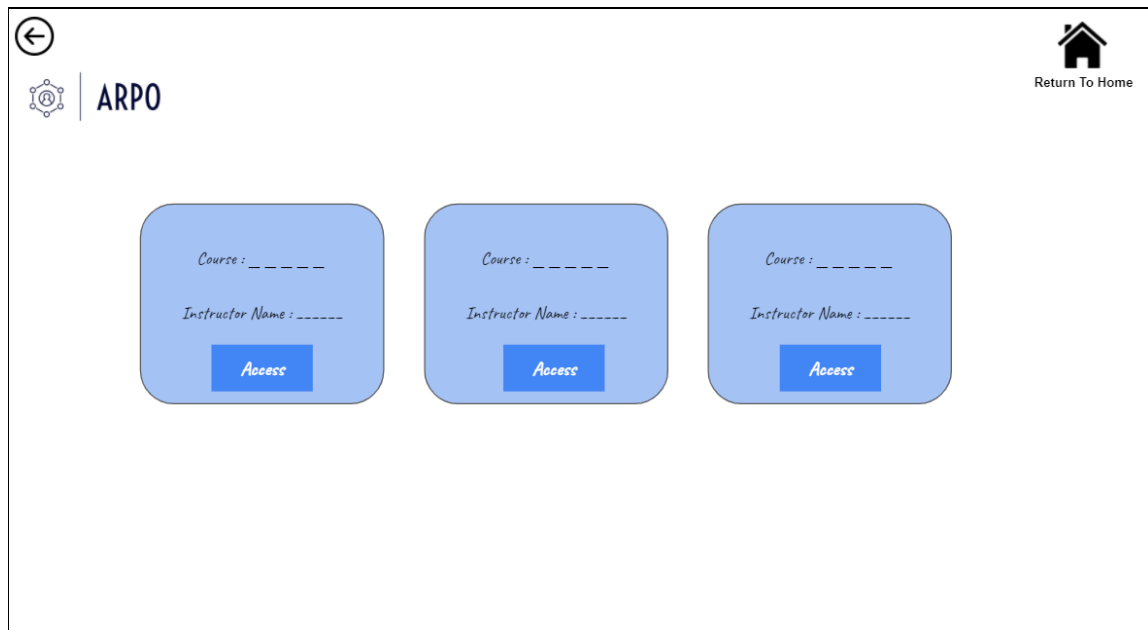
Course Home Page:

Manage :

In addition to having access to the course view from students' side, instructors are also allowed to manage people in the course including both TAs and Students. Instructors can also register a new person in the course, either as a student , instructor or a TA/Tutor.
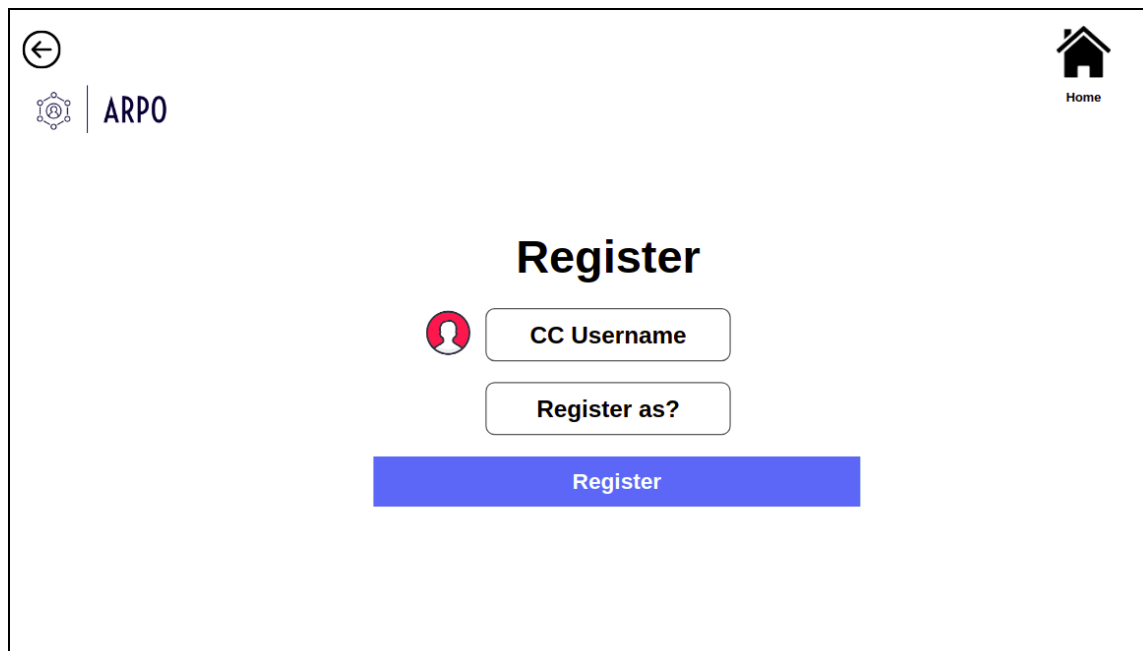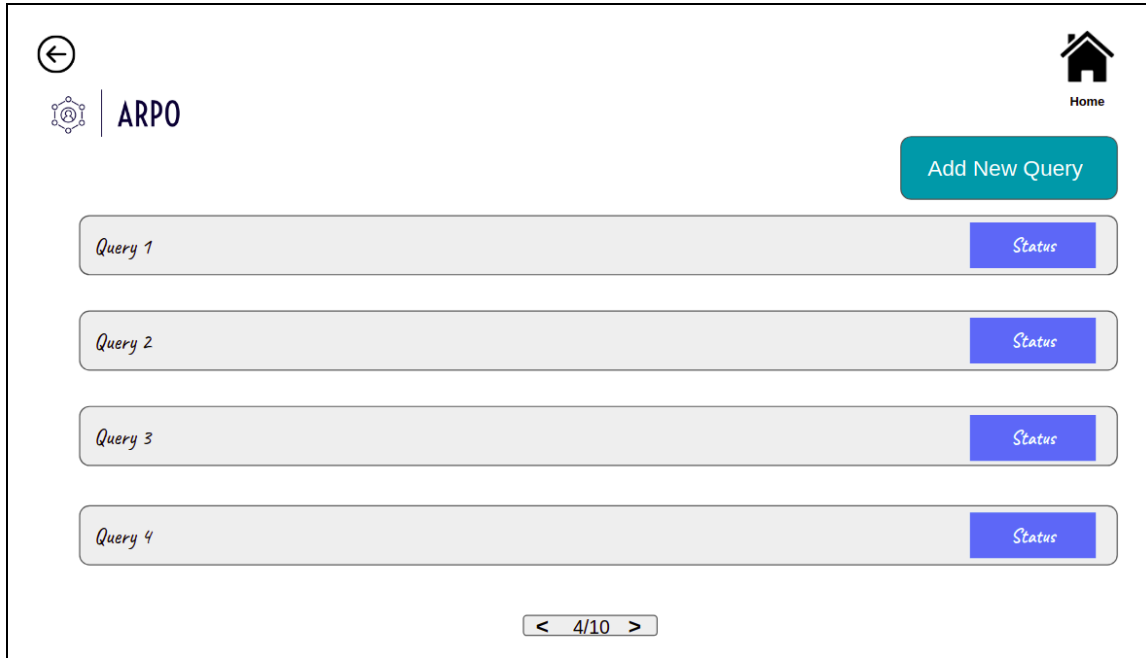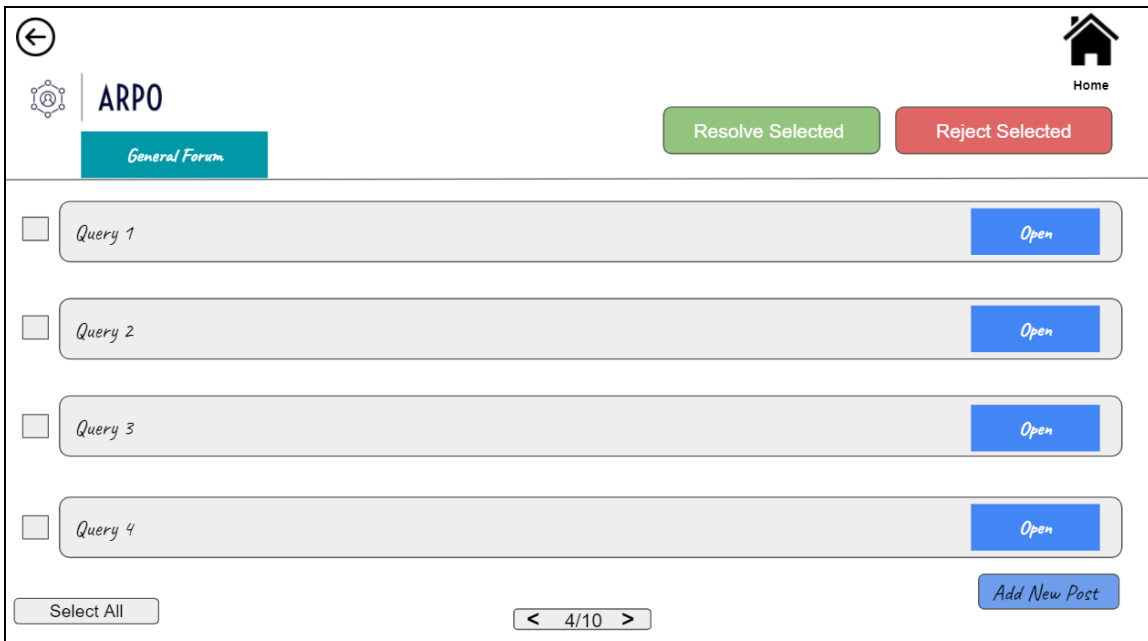
Notifications Tab:

The notifications section offers **additional authority** to mentors for adding any new course announcements.

For general forums as well as private queries, the interface is mostly the same where instructors can answer the query. Additionally, in private query they can change the status of query too. The interface for this is shown in the following images.

General query tab:

Private Query tab:

View of Received/Posted Query Forum:

Only the author of the post can edit the forum posted. The viewers can reply to it anonymously or with their original identity as they wish. The feature of anonymity won't be available in private forums.

General:

Private :

Participants List:

Instructors can also access the participants list, which consists of all the members related to the course including students as well as TAs.

Technical Support and Admin Queries:
Instructors can also raise queries/concerns to the technical staff team and admins in a manner very similar to students as shown below.

TUTORS'/TAs' INTERFACE:

The initial dashboard for TAs is similar to students as well.

Most of the functionalities here are similar to an instructor except the fact that TA/Tutors are not allowed to manage other TA/Tutors in the course. The courses section in the TA's profile looks like the following.

Courses Page:
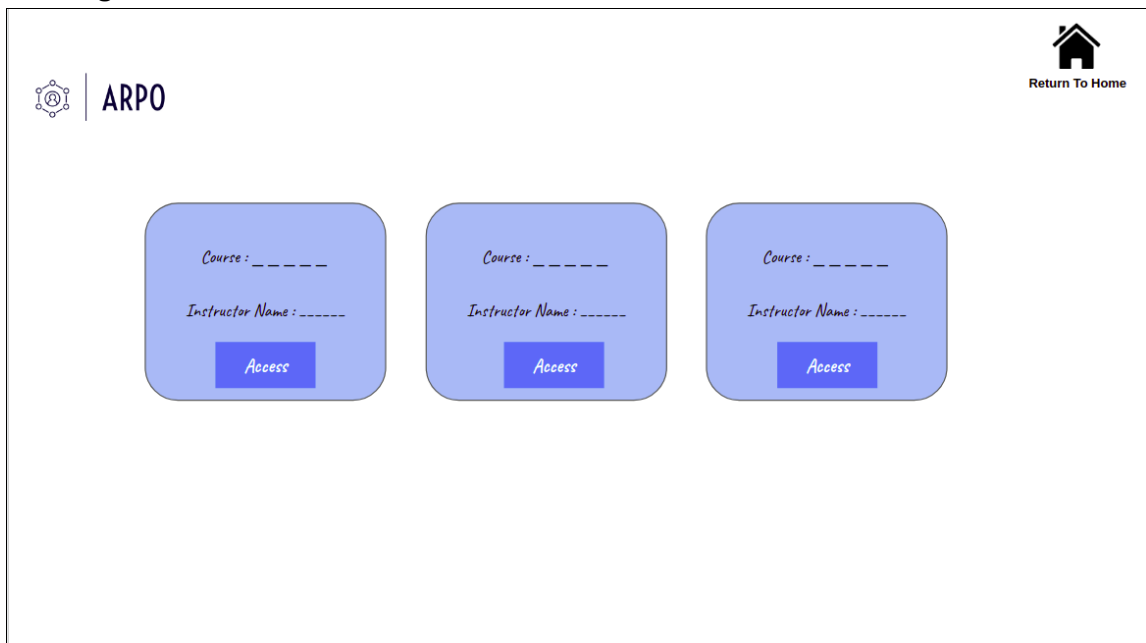
Course Home Page:

On the inside of a particular course module, we see the following view(similar to what an instructor sees, but with certain functionalities like manage users removed)
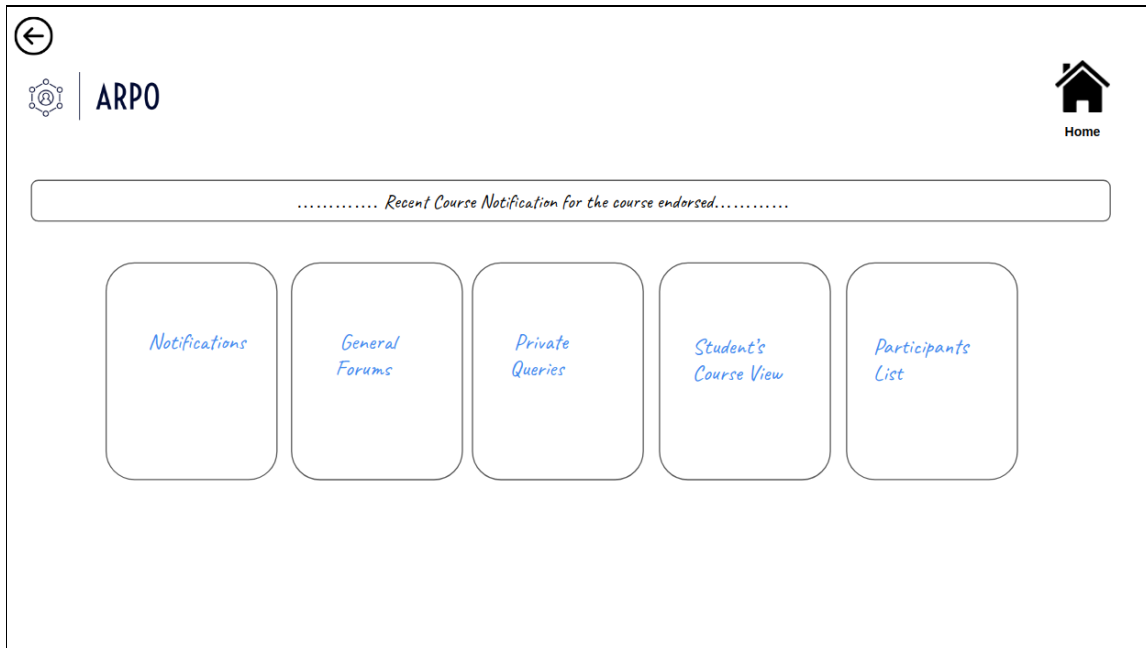
Notifications Tab:
They can view and add new notifications regarding the course.
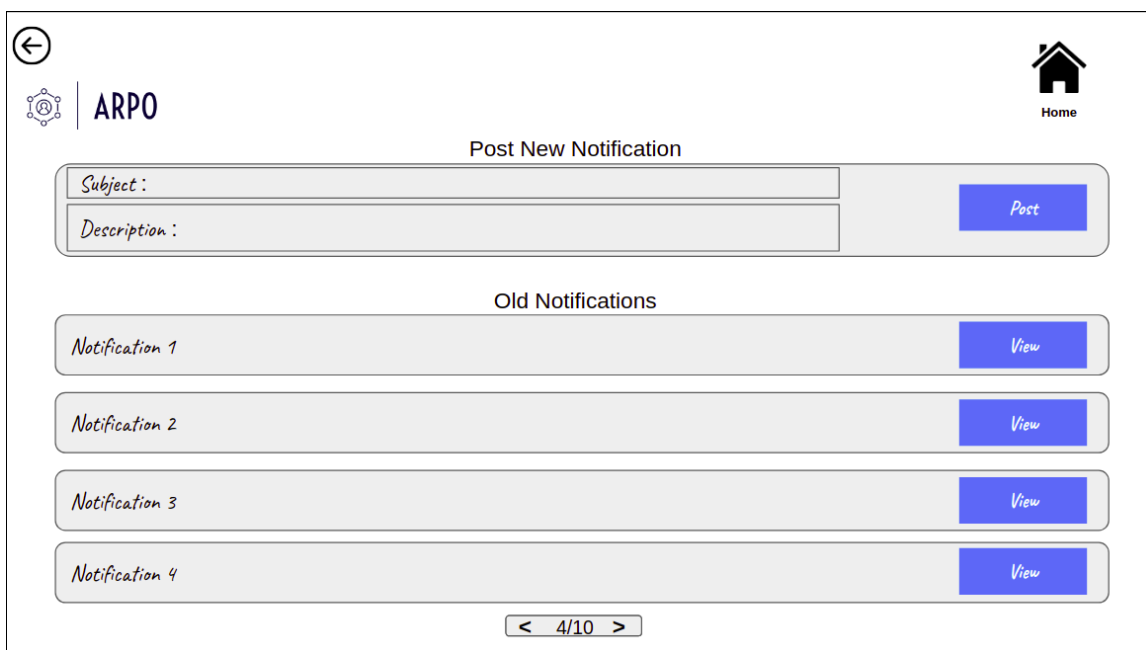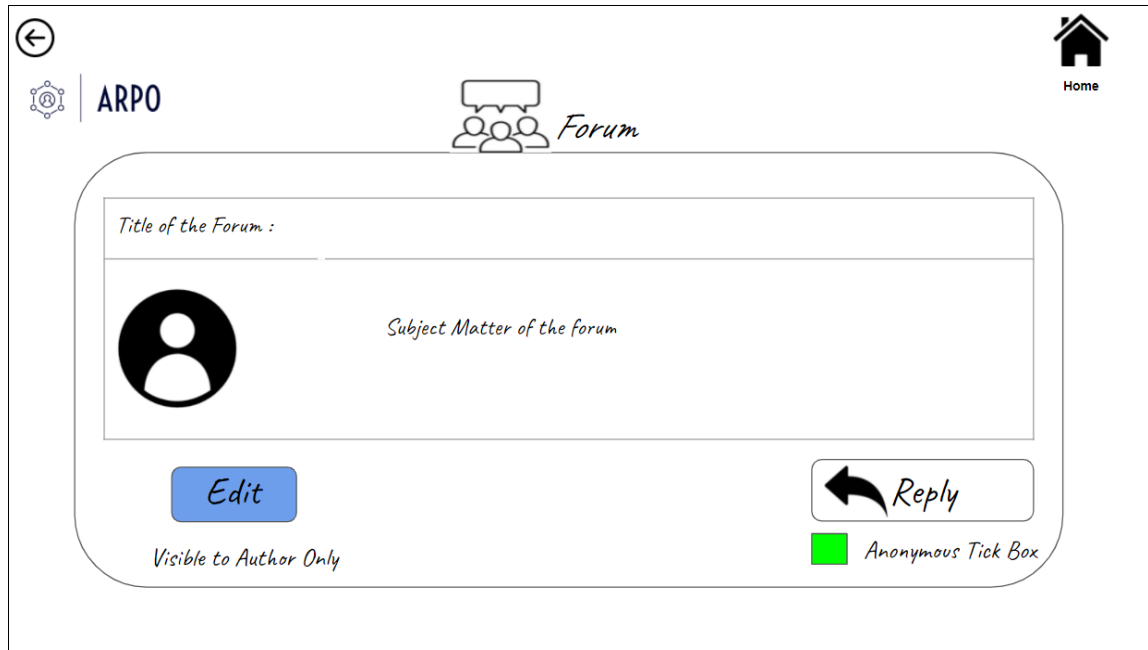
Similar to an instructor, they can also respond to general and private queries and resolve them, the status of which will be reflected on the student's profile.
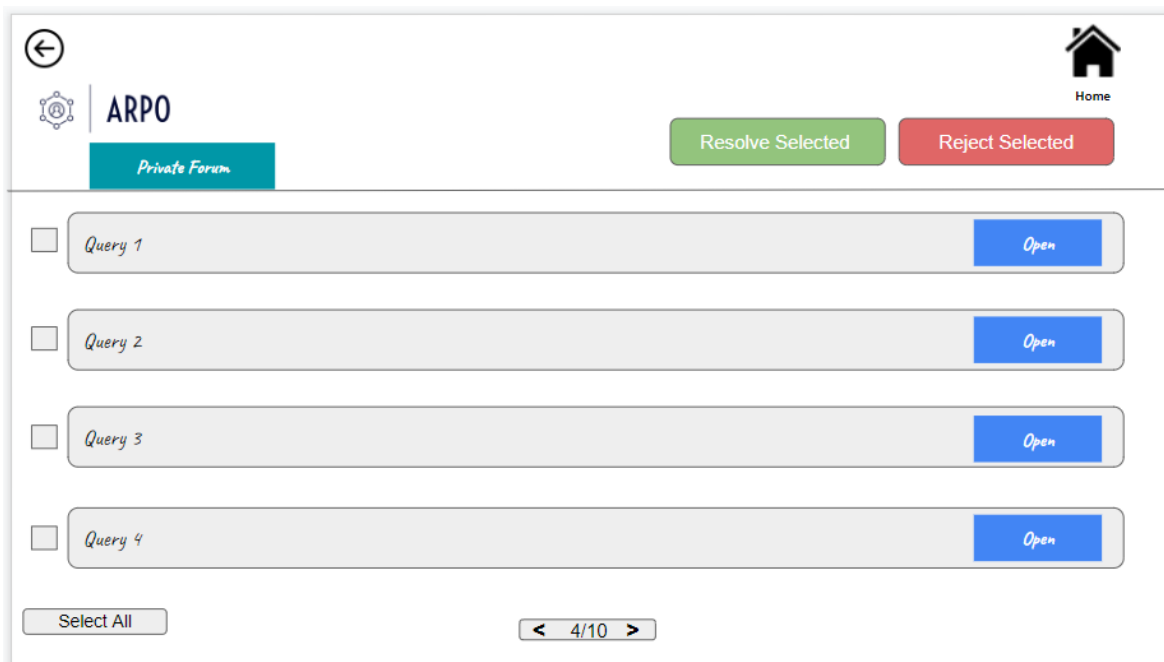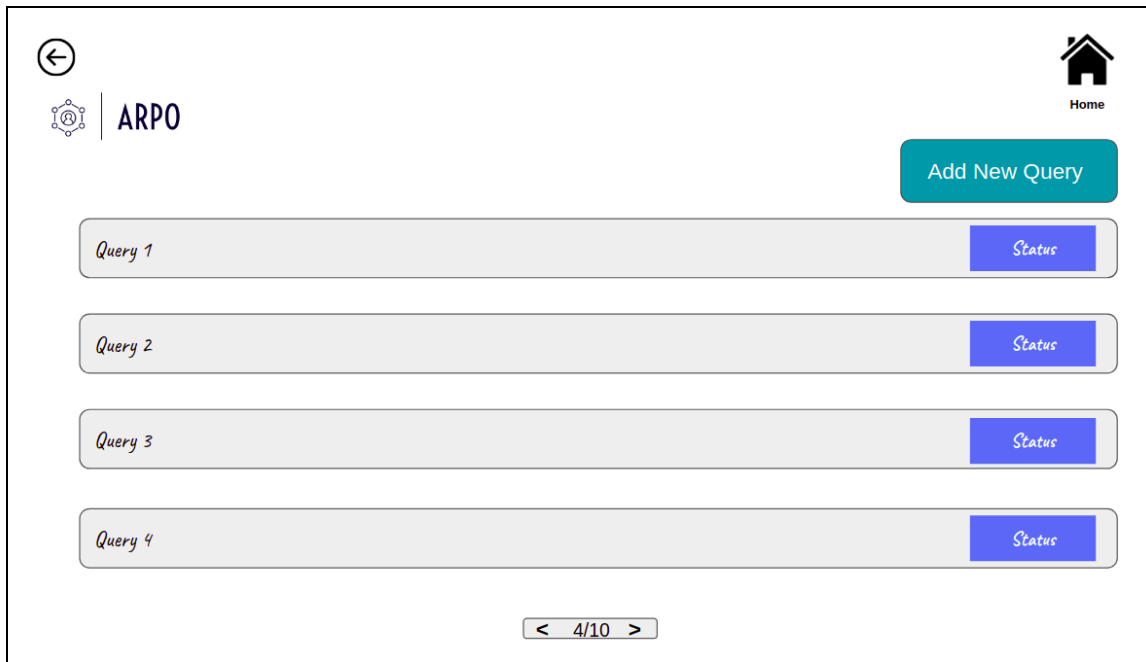
View of Received/Posted Query Forum :

Only the author of the post can edit/ delete the forum posted. The viewers can reply to it anonymously or with their original identity as they wish but this feature is available only for general forum.

General:

Private:

Participants List:

TAs can also see the participants' list of the course to access information about a particular student or a fellow TA.

Technical Support and Admin Queries:
TAs can also contact the technical system staff for any concerns.

TECHNICAL STAFF'S INTERFACE:

The system dashboard for technical staff members (TSM) looks like the following. All the institute level notifications, links and news is also accessed by TSM.
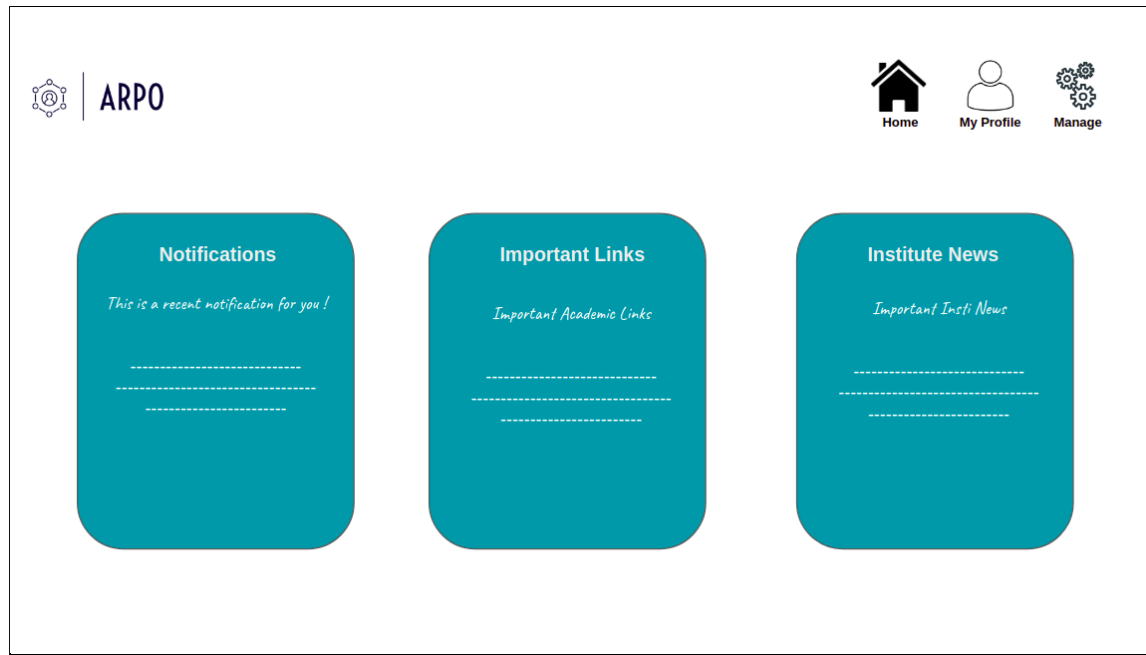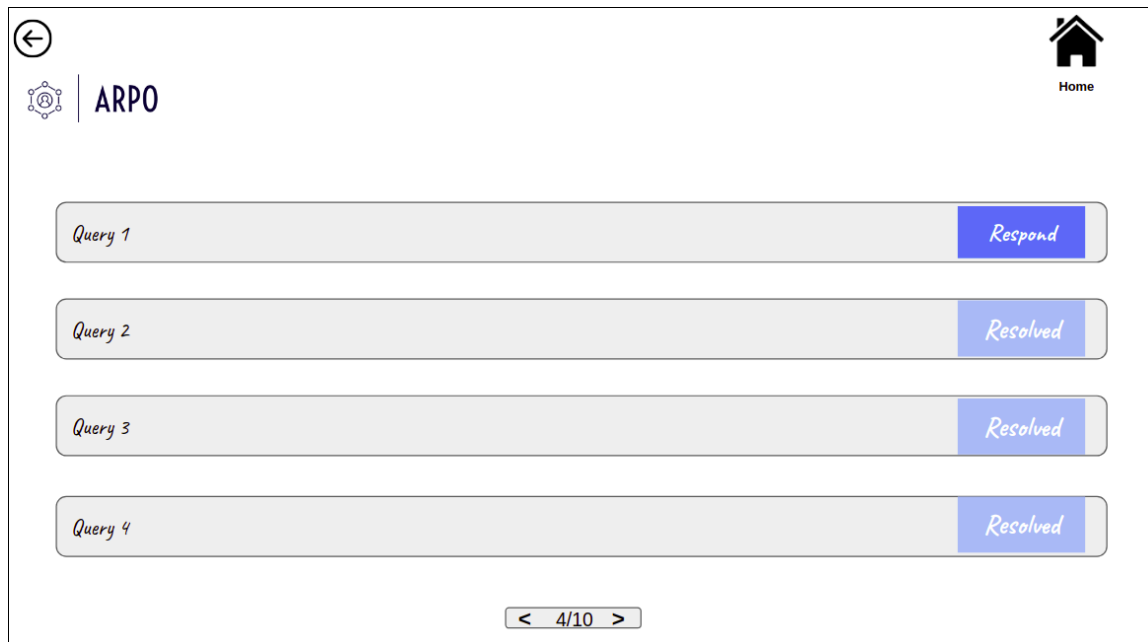
The main section for TSM is the manage section where they can register a new user for the platform, manage received queries, post notifications regarding the platform which will be reflected to every user on their Dashboard and view any user's profile, as shown below.

Manage:
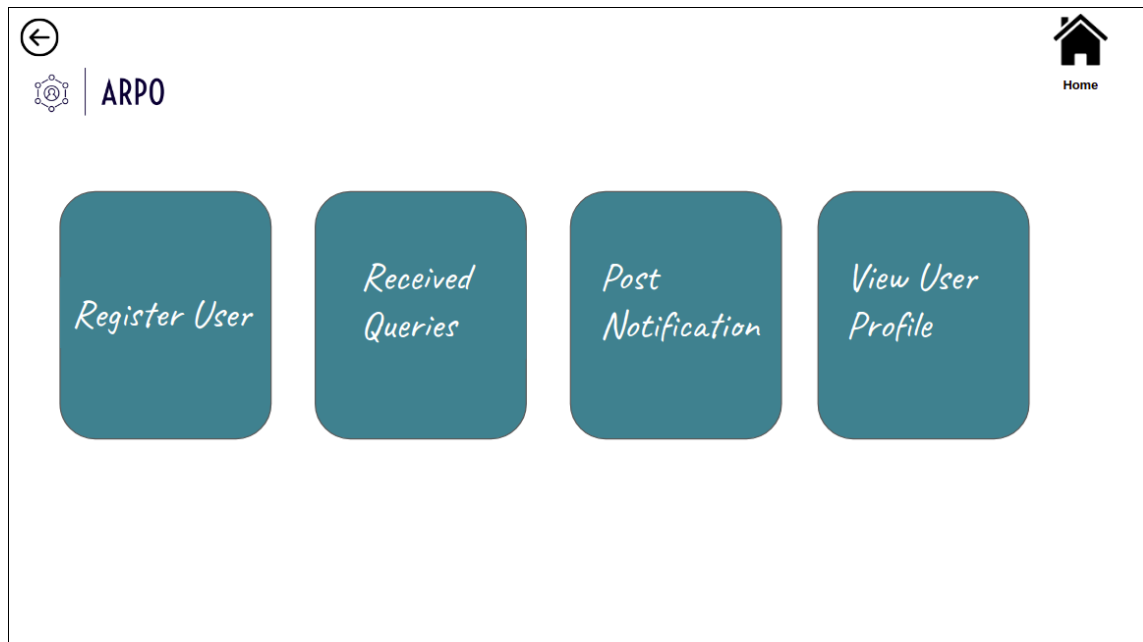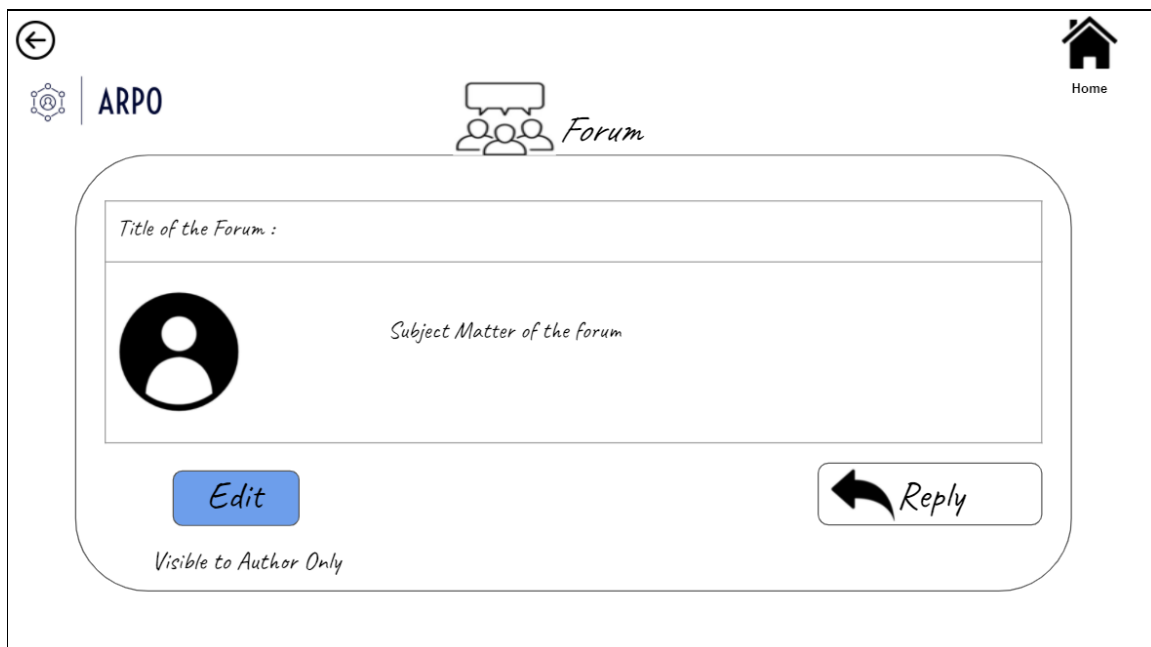
All these different functionalities of the technical staff members are shown below in their respective interfaces.

Register User Tab:
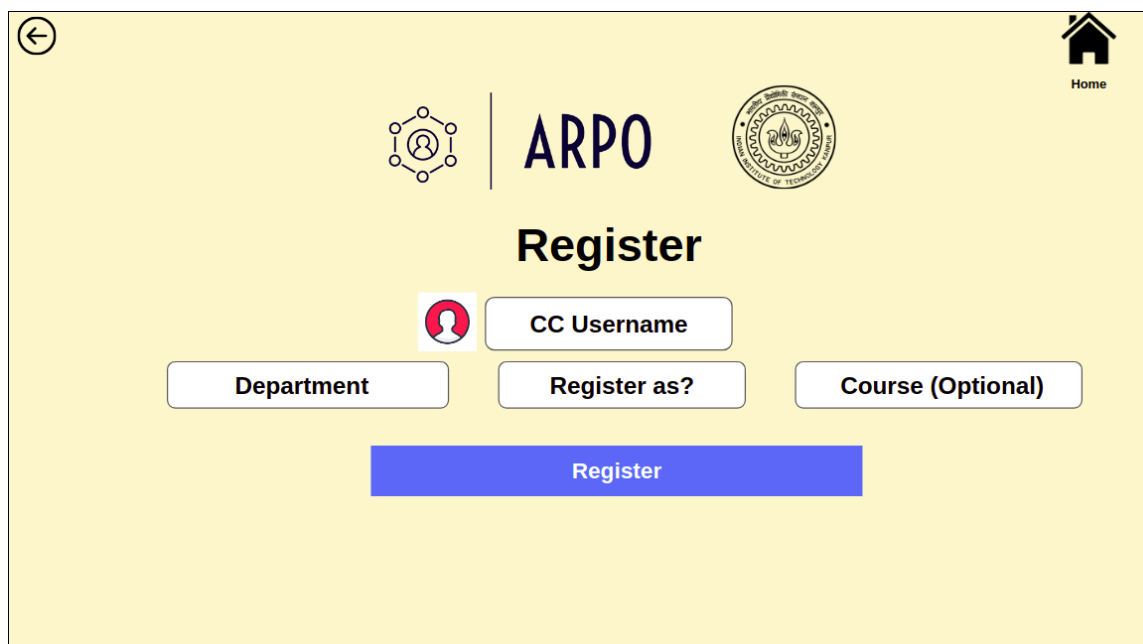
Here, TSM can register any new user on the platform, be it instructor or a student or admin member(course is an optional field for this reason).

Received Queries Section :

View of Received Query Forum :

Post Notifications Section :

ARPO

Notification

Title of the Notification :

Subject Matter of the Notification

Post

Home

ADMIN'S INTERFACE:

Initial dashboard for Admins look like the following.

Admins can answer to received queries and post notifications from "manage" section which can be seen below.

Manage:
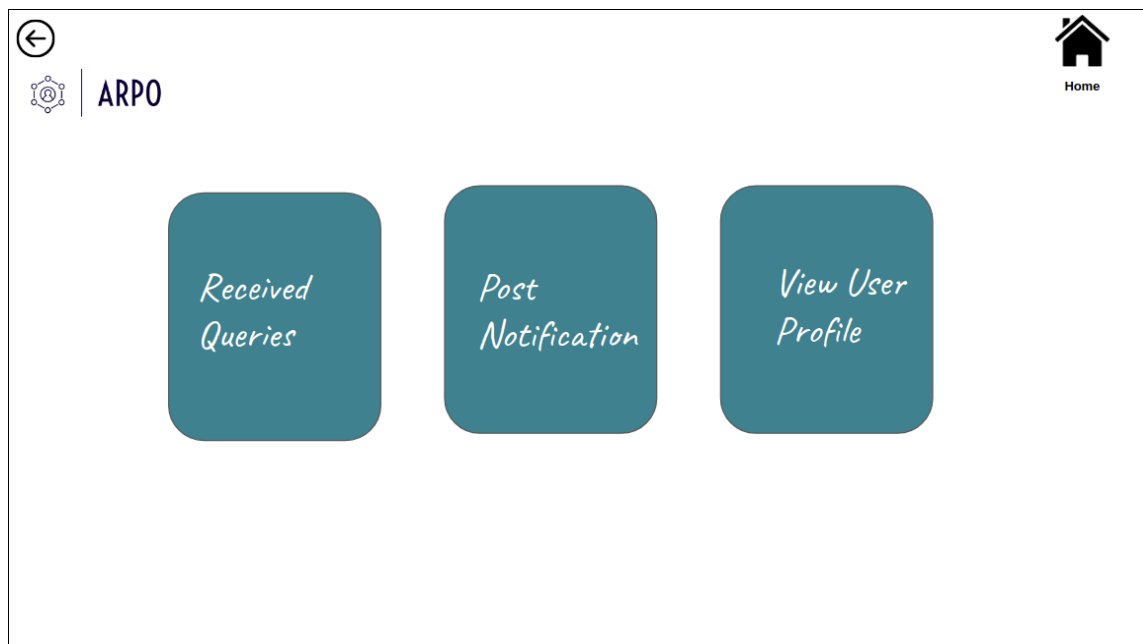
The interfaces for all these functionalities are shown below.

Received Queries Section :

View of Received Query Forum:

This is similar to that of TSM's interface.

Post Notifications Tab:

Technical Support Query:

Admins can also contact the technical staff for any technical queries and concerns like everyone else.

2 Architecture Design

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application.

The model manages the database. MySQL is the database management system used. It receives information about query resolution and users' profiles from the controller and gives updates about queries, notifications and users' profiles to the controller. It also gets information of refreshed requests from the view. The view defines the display, user interface. HTML, CSS, Javascript are used for the view. It updates information about posted queries, announcements, profile management and status updations to the controller and gets the query notifications, status updations, and user and profile management from the controller. The controller receives updates from the view and sends information to the model. Java is used for the controller. The view and the controller are directly connected to the browser.

Thus, this architecture is the best fit for the project to connect frontend, backend and database from end to end for various functionalities as well it allows the data to change independently of its representation and vice versa. The only disadvantage is that it may involve additional code or code complexity where the data interaction is simple and thus may take more time in debugging.
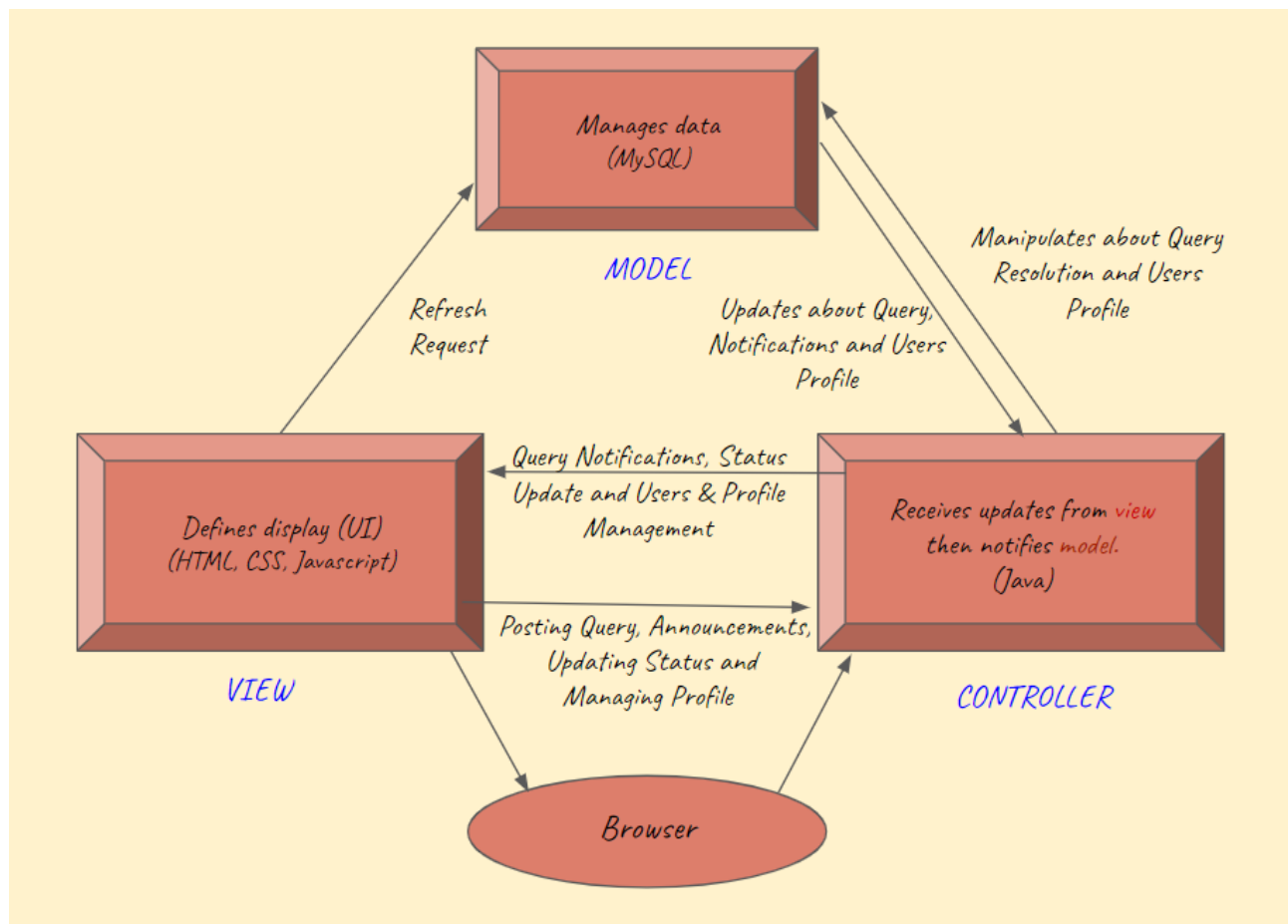
3 Object Oriented Design

3.1 Use Case Diagrams
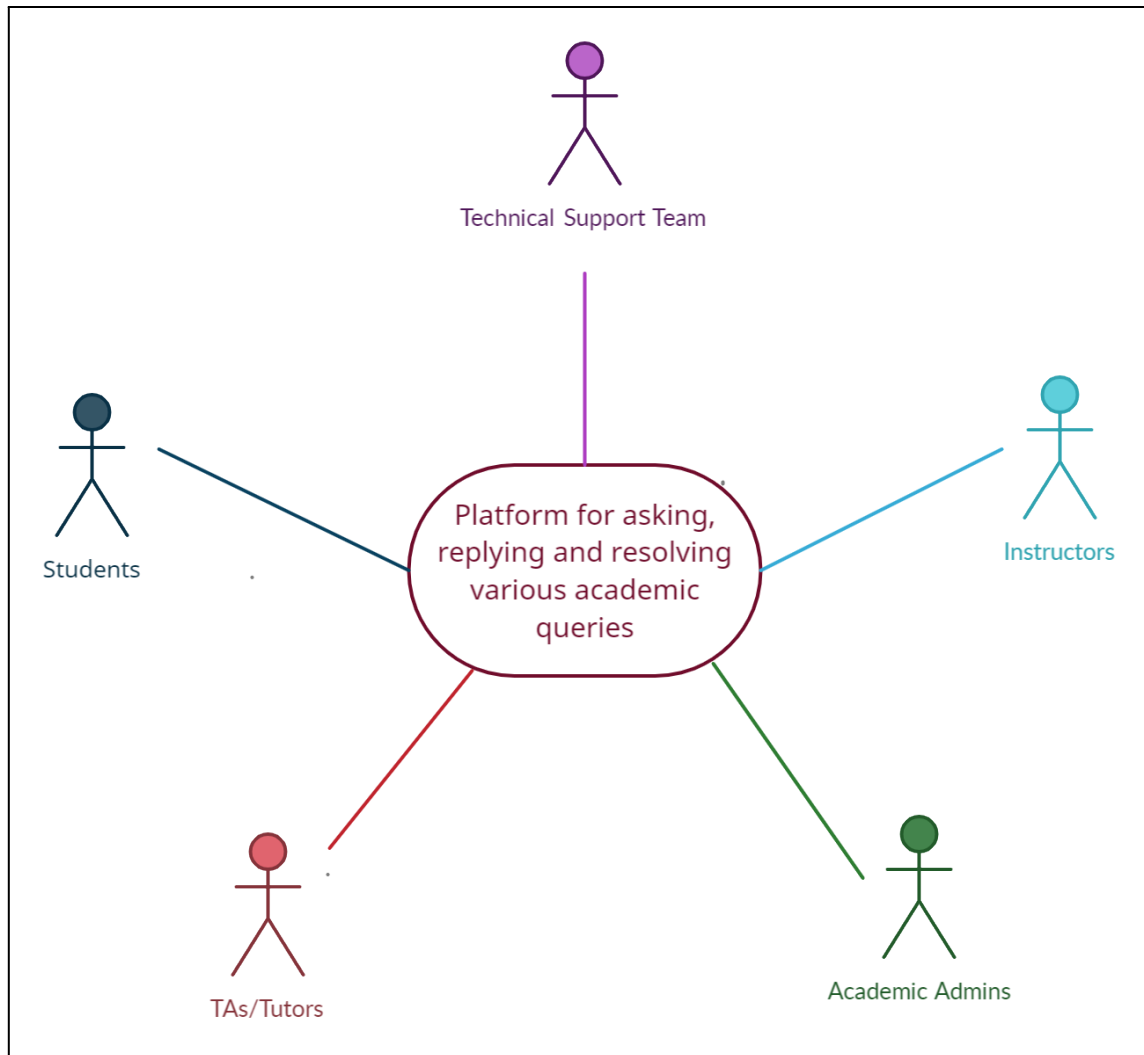
Use case diagram U0 encapsulating the entire system and all the five actors

The five actors involved are: Students, Instructors, TAs, Academic Affairs Admins, Technical Support Team.

Remarks: *The above use case is differentiated into three prime use cases: U1, U2 and U3 on the basis of functionalities involved in the system.*

Use Case #1 (U1)
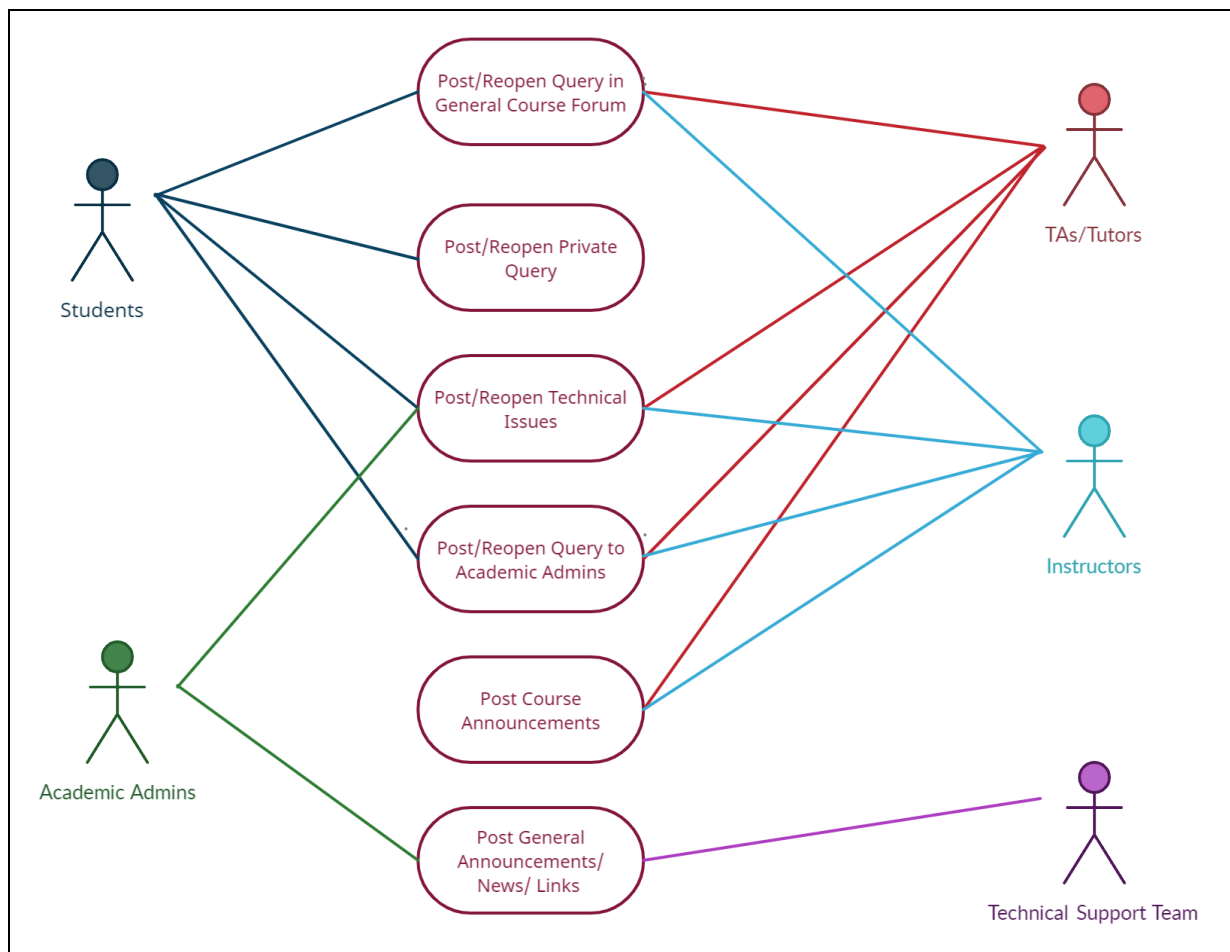
Use case diagram U1 encapsulating the query directing system

Purpose: Sending the queries and posting relevant information by all the concerned actors.

Requirements Traceability: Students' access to course forums, Instructors' access to course forum and registered students' course profile, TAs/Tutors' access as granted by Instructors, Academic Admins have access to students and instructors list and Technical Support Team have access to all users IDs'.

Priority: High. This is kept at high priority as the prime goal of this platform is to handle each query or announcement with utmost priority inorder in an efficient manner.

Preconditions: Each actor is a registered member of the institute and is logged into the platform with a unique ID.

Post conditions: Each query has reached the designated actor for further process in exact form.

Actors: Students, Instructors, TAs/Tutors, Academic Admins, Technical Support Team.

Exceptions: The concerned query is sent to the wrong actor or major technical issues like unable to login or send messages should be handled via emails.

Includes: None.

Notes/Issues:

(1) For unresolved queries, the query sent is visible to the actor and its status is kept in progress until addressed.
(2) Instructors have the authority to manage TAs/Tutors.

Use Case #2 (U2)
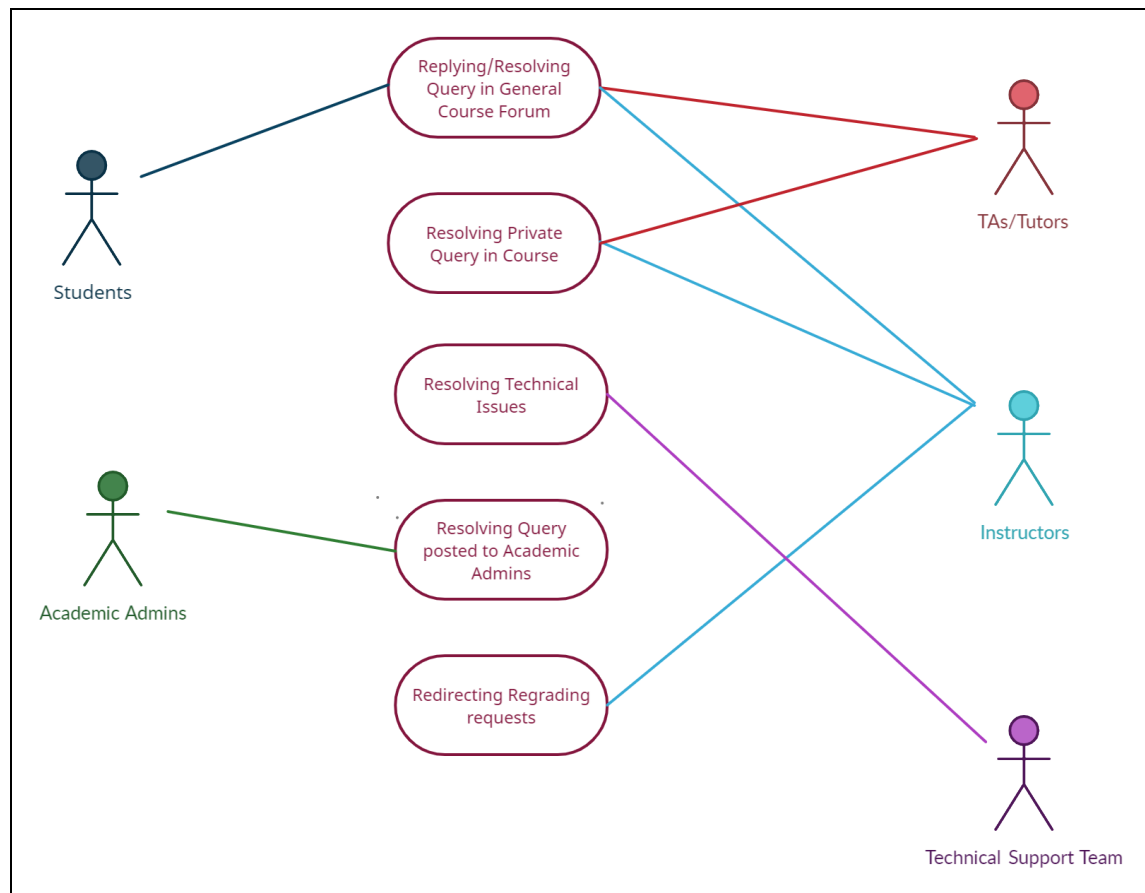
Use case diagram U2 encapsulating the query replying and resolving system

Purpose: Resolving the queries initiated to and delivered to all the concerned actors.

Requirements Traceability: Students' access to course forums, Instructors' access to course forum and registered students' course profile, TAs/Tutors' access as granted by Instructors, Academic Admins have access to students and instructors list and Technical Support Team have access to all users IDs'.

Priority: High. This is kept at high priority as the prime goal of this platform is to handle each query (here, resolving queries) with utmost priority inorder to provide an efficient system to clarify any query.

Preconditions: Each actor is a registered member of the institute and is logged into the platform with a unique ID and the initiated query has reached the designated actor in exact form for further response.

Post conditions: Each resolved query is maintained on the portal.

Actors: Students, Instructors, TAs/Tutors, Academic Admins, Technical Support Team.

Exceptions: There may be a delay in response from an actor or a query replied or redirected to the wrong actor.

Includes: Use Case #1 (U1).

Notes/Issues:

(1) All resolved queries are maintained and status is updated as closed.
(2) Instructors have the authority to manage TAs/Tutors.

Use Case #3 (U3)
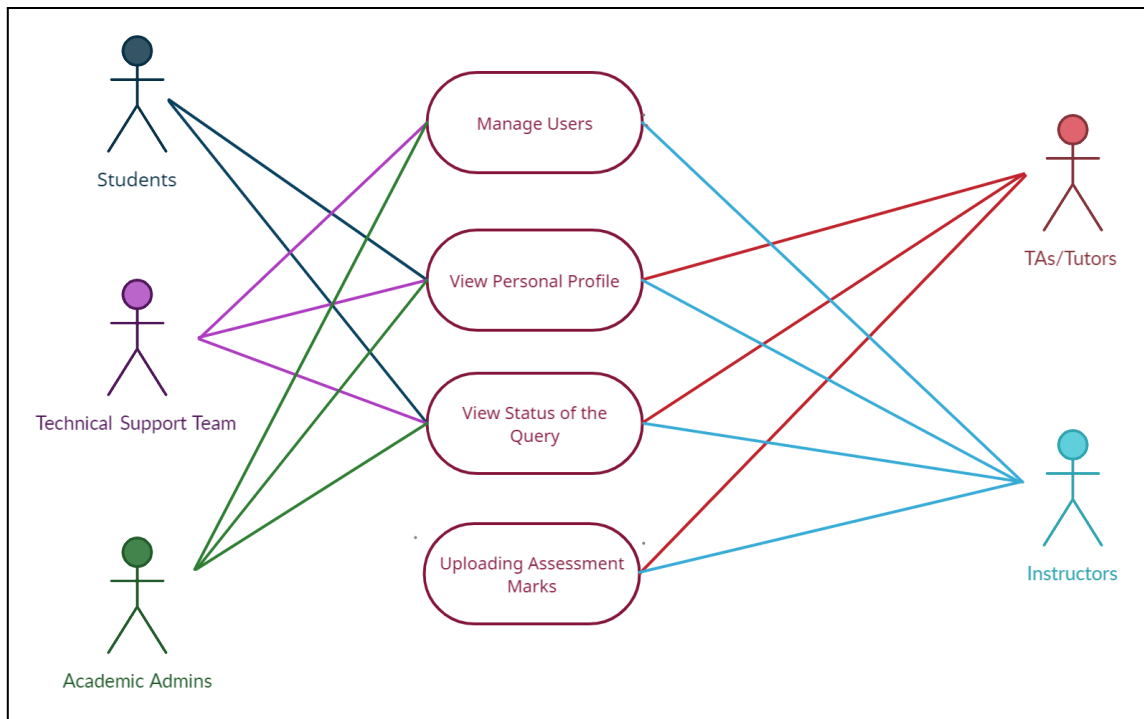
Use case diagram U3 encapsulating the handling of technicalities in actors' interfaces

Purpose: Providing all other functionalities other than posting information, sending or replying queries.

Requirements Traceability: Students' access to course forums, Instructors' access to course forum and registered students' course profile, TAs/Tutors' access as granted by Instructors, Academic Admins have access to students and instructors list and Technical Support Team have access to all users IDs'.

Priority: Medium. The proper handling of the platform without any technical issue enables each actor to perform its duty effectively.

Preconditions: Each actor is a registered member of the institute and is logged into the platform with a unique ID.

Post conditions: Not Applicable.

Actors: Students, Instructors, TAs/Tutors, Academic Admins, Technical Support Team.

Exceptions: None.

Includes: None.

Notes/Issues:

(1) All unresolved queries' status is "in progress", while destination actors can update the status accordingly as they progress and all resolved queries are marked as accepted/ closed.
(2) Instructors have the authority to manage TAs/Tutors.

3.2 Class Diagrams
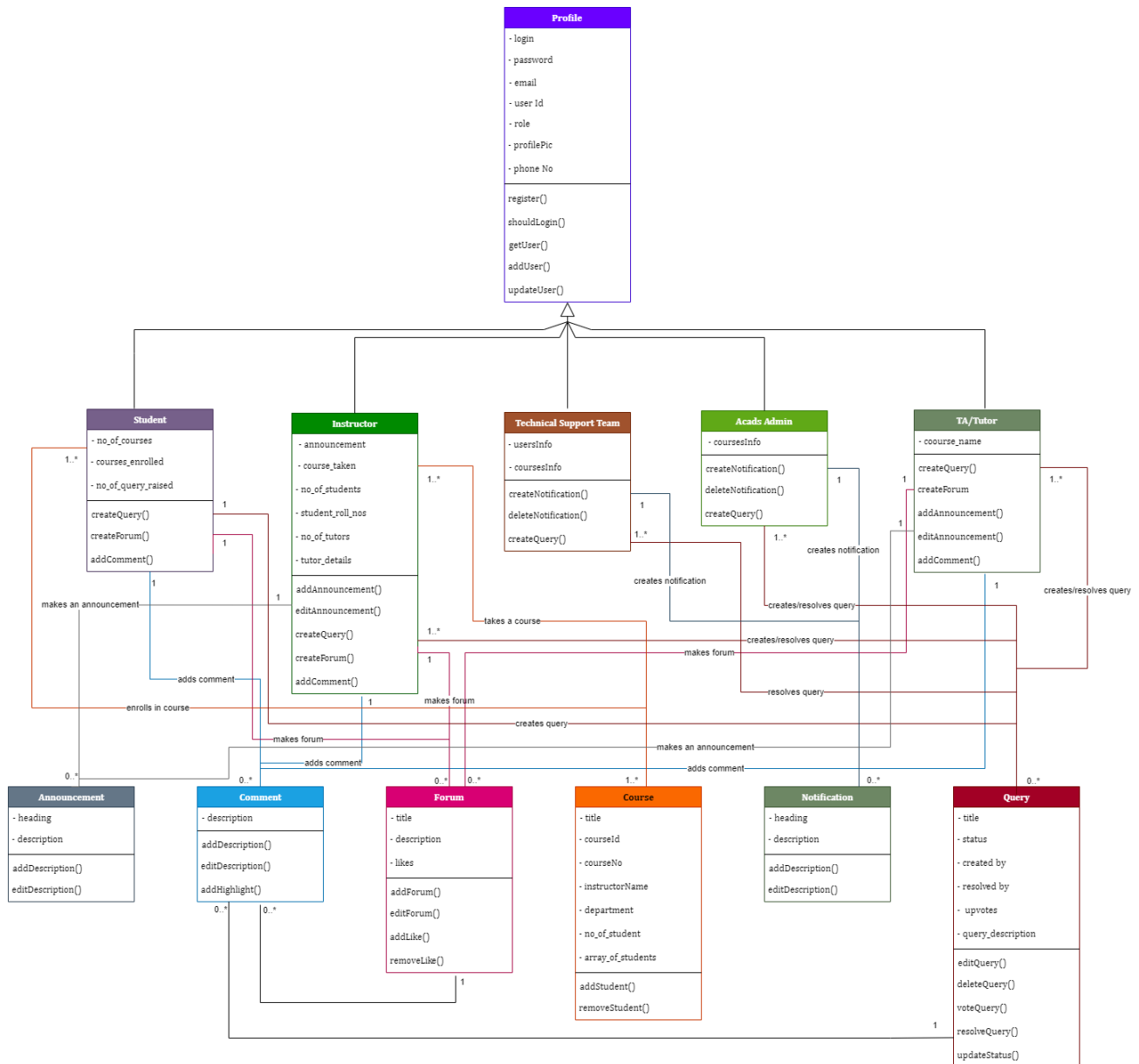
Class Diagram depicting relation between various classes

The above class diagram represents the relations between various classes and their different functionalities.

3.3 Sequence Diagrams
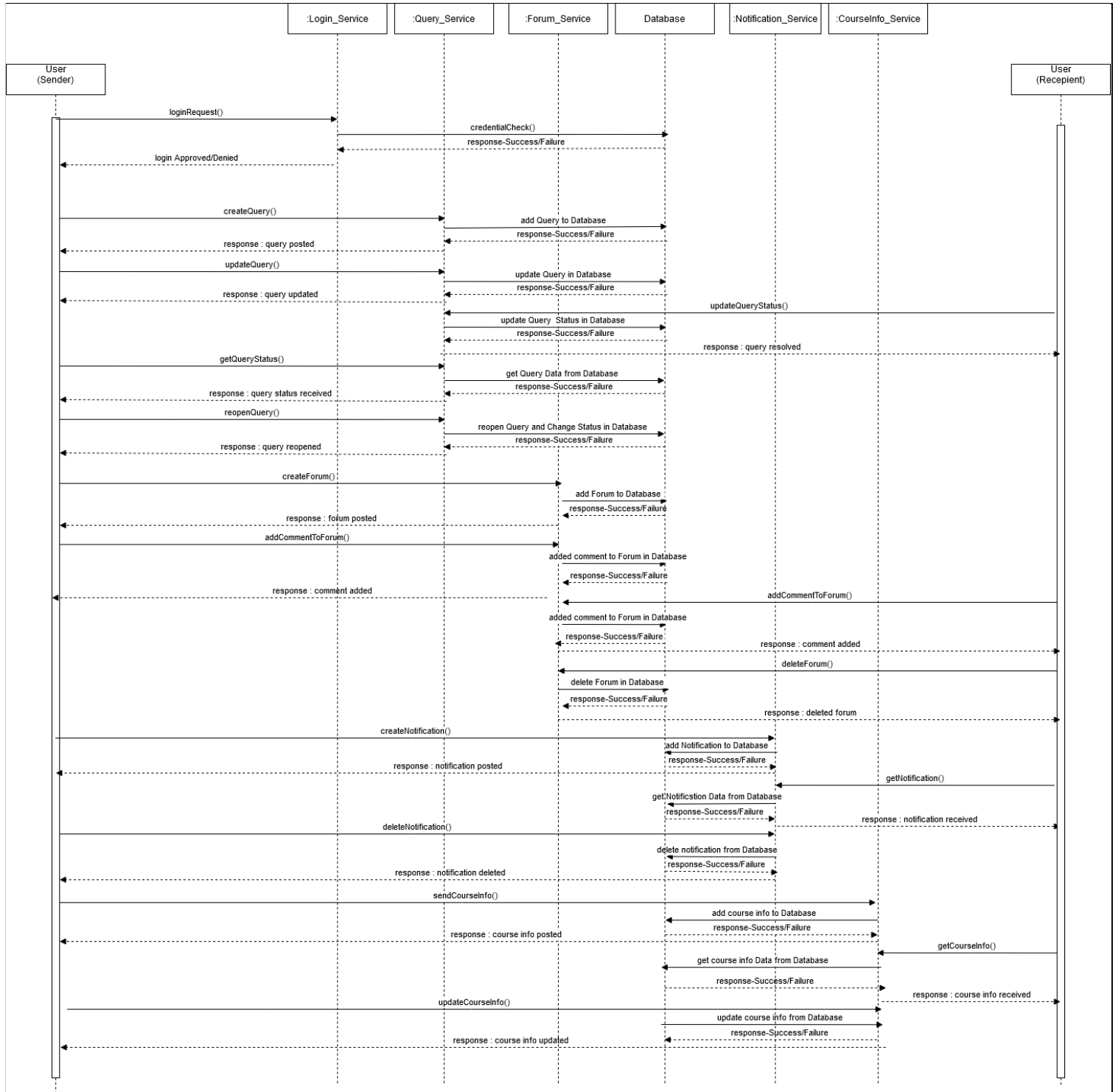
The above diagram has the user(sender) which is the user who will be adding entities to the portal, and the user(receiver) is the user who will be receiving the entities from the portal.

3.4 State Diagrams
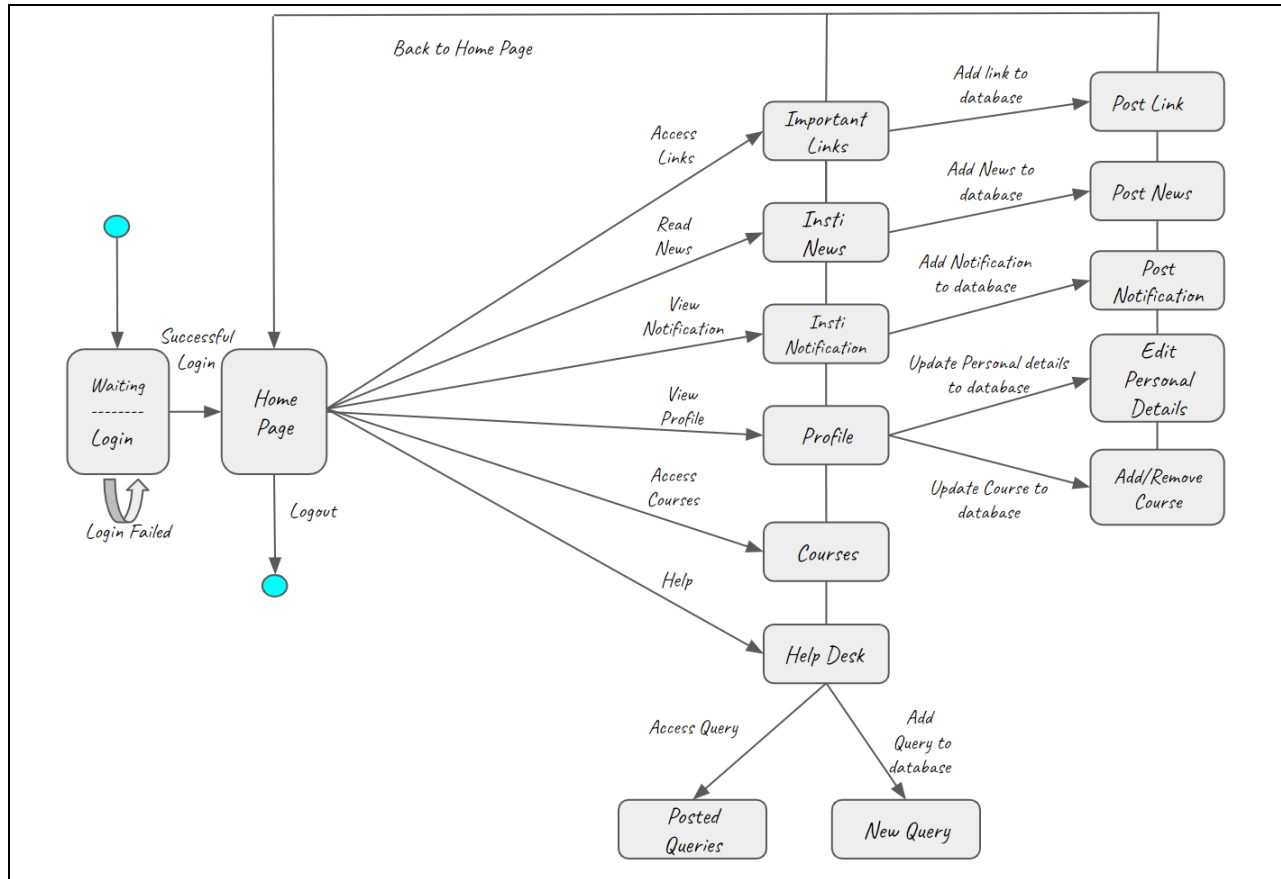
The above state diagram contains the main states which are self explanatory as shown in the figure. Moreover, the above figure reflects the superset of the state diagram and different users have access to different functionalities as already explained in the above sections of the document. The intrinsic details of state diagram for the courses is as shown below for enhance clarity:
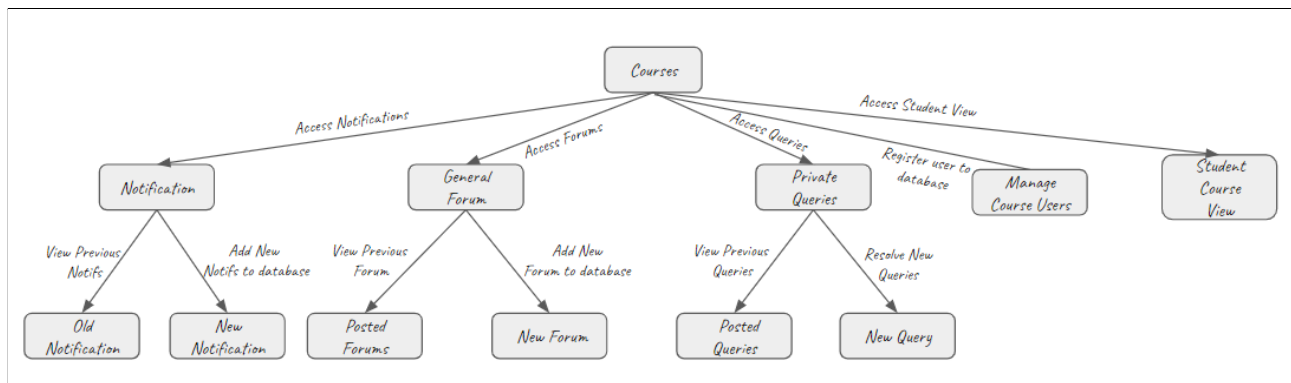
4 Project Plan

Please find the entire chart here

:https://docs.google.com/spreadsheets/d/1EfZU59jnNtYmMWM4QsKjD48BcwreTBrlbGRVUPBMDYE/edit#gid=0

Serial No.	Tasks	Assigned To
1	**Research & Analysis**	Entire Team
1.1	Project Scope Discussion	Entire Team
1.2	Stakeholder Discussion	Entire Team
1.3	Requirement Discussion	Entire Team
2	**Design**	Entire Team
2.1	Product Functionality Design	Lakshay, Piyush, Shubhan, Jahnavi, Pranav
2.2	User Interface Mock Up/Human Interface Design	Akansh, Keshav, Atharv
2.3	Architecture Design	Akansh, Pranav, Jahnavi
2.4	Object Oriented Design	Manoj, Akshan, Lakshay, Piyush, Atharv, Keshav, Shubhan, Akansh
2.4.1	Use Case Diagram Design	Manoj, Akshan
2.4.2	Class Diagram Design	Manoj, Akshan
2.4.3	Sequence Diagram Design	Lakshay, Piyush
2.4.4	State Diagram Design	Atharv, Akansh, Keshav, Shubhan
2.5	Project Plan Design	Lakshay, Piyush

Serial No.	Tasks	Assigned To
3	**Development**	Entire Team
3.1	Develop Frontend Modules:	Manoj, Pranav, Akansh, Shubhan, Keshav, Lakshay
3.1.1	Login/Registration Interface	Manoj, Pranav, Shubhan
3.1.2	Main Dashboard Interface (User Specific)	Lakshay, Akansh, Keshav
3.1.3	Course Home Interface	Manoj, Pranav, Shubhan
3.1.4	Course Profile Interface	Manoj, Pranav, Shubhan
3.1.5	Notifications Interface	Lakshay, Akansh, Keshav
3.1.6	Query/Forum/Private Query Interface	Manoj, Pranav, Shubhan
3.1.7	Add Query Interface	Manoj, Pranav, Shubhan
3.1.8	Technical Support Query Interface	Lakshay, Akansh, Keshav
3.2	Developing Databases Models	Keshav, Akansh, Akshan
3.2.1	Developing the Database	Keshav, Akansh, Akshan
3.2.2	Filling Mock Data into Database	Keshav, Akansh, Akshan
3.3	Develop Backend Modules:	Atharv, Lakshay, Piyush, Akshan, Manoj, Jahnavi, Pranav
3.3.1	Developing Entity Classes	Atharv, Lakshay, Piyush, Akshan, Manoj, Jahnavi
3.3.1.1	Profile Class	Piyush, Lakshay, Atharv
3.3.1.2	Student SubClass	Piyush, Lakshay, Atharv
3.3.1.3	Instructor SubClass	Piyush, Lakshay, Atharv
3.3.1.4	Techincal Support Team SubClass	Piyush, Lakshay, Atharv
3.3.1.5	Acad Admin SubClass	Piyush, Lakshay, Atharv
3.3.1.6	TA/Tutor SubClass	Piyush, Lakshay, Atharv
3.3.1.7	Announcement Class	Piyush, Lakshay, Atharv
3.3.1.8	Comment Class	Manoj, Akshan, Jahnavi
3.3.1.9	Forum Class	Piyush, Lakshay, Atharv
3.3.1.10	Course Class	Manoj, Akshan, Jahnavi
3.3.1.11	Notification Class	Manoj, Akshan, Jahnavi
3.3.1.12	Query Class	Manoj, Akshan, Jahnavi
3.3.2	CRUD Query Service	Manoj, Akshan, Jahnavi
3.3.3	CRUD Notification Service	Manoj, Akshan, Jahnavi
3.3.4	CRUD Forum Service	Piyush, Lakshay, Atharv
3.4	Integrating FrontEnd And Backend Modules	Manoj, Akshan, Piyush, Atharv
3.4.1	Integrating API's with Mock Database	Manoj, Akshan, Piyush, Atharv
3.4.2	Integrating API's with Frontend	Manoj, Akshan, Piyush, Atharv
3.7	Integration Testing	Jahnavi, Pranav
3.8	Unit Testing	Jahnavi, Pranav

Serial No.	Tasks	Assigned To
4	**Testing/Issue Resolving**	Entire Team
4.1	System Testing	Shubhan, Jahnavi, Pranav, Atharv
4.2	Document Issues Found	Shubhan, Jahnavi, Pranav, Atharv
4.3	Correct Issues Found	FrontEnd and Backend Teams
5	**Feedback Analysis**	Entire Team
5.1	Addressing Beta Feedback	Entire Team
5.2	Making Updates based on Feedback	Entire Team
	* CRUD - Create Read Update Delete	

5 Other Requirements

The future scope for this project includes the integration with Pingala or Hello IITK Portal, thus enabling the formation of a single robust platform with user-friendly interactive interface for all authorized CC users. The assurance of these factors would lead to having a common integrated design/architecture for the same.

Appendix A - Group Log

There has been a continuous interaction among the team members in formal meets as well as informally through whatsapp group/calls. The team members were also in direct touch with the assigned TA through official Whatsapp Group and various zoom meetings for almost daily feedback and suggestions.

The tabular detail of some official group meets are indicted as follows:

Meeting Minutes	Agenda
Feb 4, 2022 04:30 PM-05:00 PM	Group Meet, discussed bringing on the design document.
Feb 6, 2022 04:00 PM-04:45 PM	Meet with the assigned TA for discussion about various intricacies of the design document.
Feb 6, 2022 10:30 PM-11:30 PM	Group Meet regarding discussion on feedback from the assigned TA for various sections of the design document and the work distribution.
Feb 9, 2022 3:00 PM-03:30 PM	Group Meet for discussion on the context diagram and architecture design specifically.
Feb 11, 2022 10:00 PM-10:30 PM	Group Meet for discussion on the class diagram specifically
Feb 12, 2022 05:40 PM-06:30 PM	Group Meet in which each member gave a short presentation about the work assigned and proposed the related reviews.
Feb 13, 2022 05:00 PM-05:40 PM	Meet with the assigned TA for reviewing and asking queries related to the work done.
Feb 14, 2022 12:00 PM-12:30 PM	Group Meet for the discussion on the architecture design for the feedback provided by the assigned TA in the last meet.
Feb 14, 2022 09:00 PM-11:30 PM	Group Meet for the discussion on section 3 as per the feedback provided by the assigned TA
Feb 15, 2022 01:30 PM-02:30 PM	Group Meet to review the work done and to make final updates for this version
Feb 15, 2022 07:30 PM-08:10 PM	Meet with the assigned TA for reviewing