
Design Document

for

DigiSchool

Version 1.1

Prepared by

Group 6:

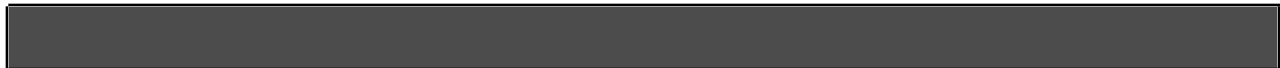
Group Name: Dementors

GUGULOTH VARUN	180272	varunguguloth@gmail.com
SUMIT NAGLE	170732	sumit.nagle@gmail.com
GAURAV KUMAR	180265	phogat102135@gmail.com
MOTA JITENDRA	180434	jitendrathierider23@gmail.com
AKASH DAYANAND CHAVAN	180054	akashchavan180054@gmail.com
DIKSHANT RAJ MEENA	180247	dkrajmeena27899@gmail.com
SHREYASI PRASAD	190824	student.shreyasi@gmail.com
YASHWANT TAILOR	180894	yasht88101@gmail.com
YEDDULA VINAY CHANDU	191181	chandu8042439@gmail.com
AMAN RAW	190108	amanraw4444@gmail.com

Course: CS253

Mentor TA: Shatroopa Saxena

Date: 15-02-2022



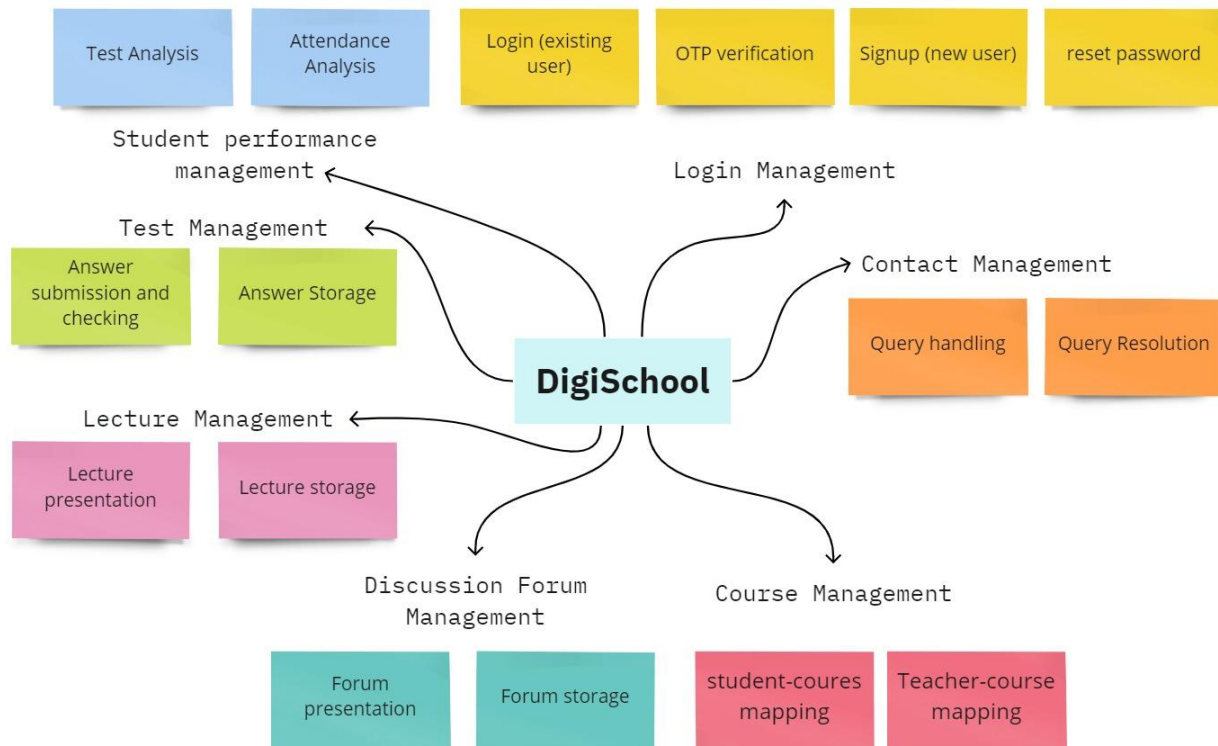
CONTENTS	II
REVISIONS	II
1 CONTEXT DESIGN	1
1.1 CONTEXT MODEL	1
1.2 HUMAN INTERFACE DESIGN	1
2 ARCHITECTURE DESIGN	2
3 OBJECT-ORIENTED DESIGN	3
3.1 USE CASE DIAGRAM	3
3.2 CLASS DIAGRAM	3
3.3 SEQUENCE DIAGRAM	3
3.4 STATE DIAGRAM	3
4 PROJECT PLAN	4
5 OTHER REQUIREMENTS	5
APPENDIX A - GROUP LOG	6

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
version V.1.0	GUGULOTH VARUN SUMIT NAGLE GAURAV KUMAR MOTA JITENDRA AKASH CHAVAN SHREYASI PRASAD YASHWANT TAILOR	First Version (Version V.1.0): First Design Document/Draft for DigiSchool	15/02/22
version V.2.0	Guguloth Varun and Sumit Nagle	Added some more diagram such as state and class diagrams	06/04/22

1 Context Design

1.1 Context Model



1.2 Human Interface Design

Registration Page

At first the users must register to DigiSchool and give the required information. There will be radio buttons for different priviledge roles. The page roughly looks like this,

The mockup shows the registration page for Digi School. The page has a header with the site name and navigation links. The main content area features a 'Student SignUp' form with the following fields:

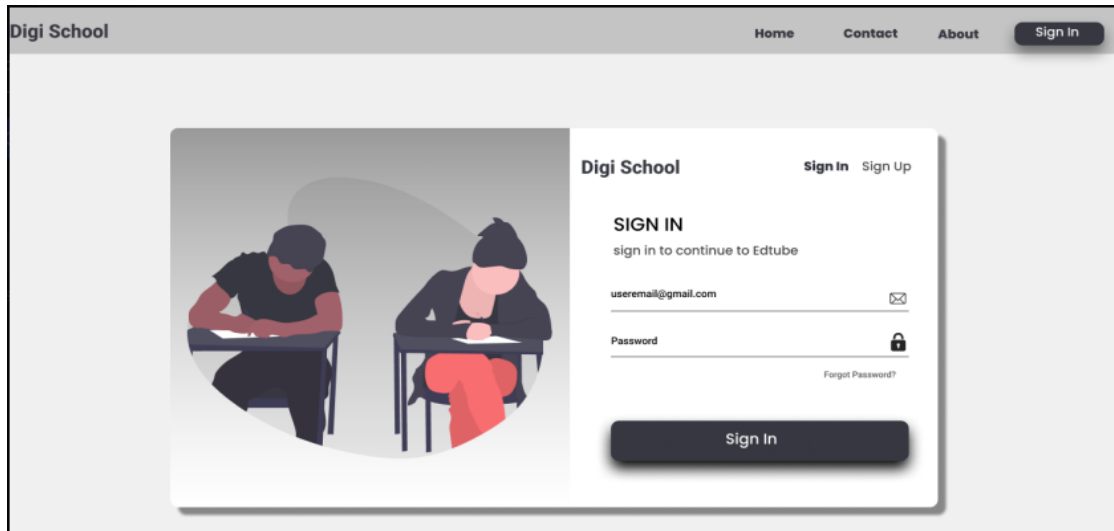
- First Name
- Last Name
- Email
- Class
- Section
- School
- Contact number
- Create Password
- Confirm Password

A 'Sign Up' button is located at the bottom of the form.

Login Page

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

(Admin wont require this. Moreover, for admin we implement django's default admin app)



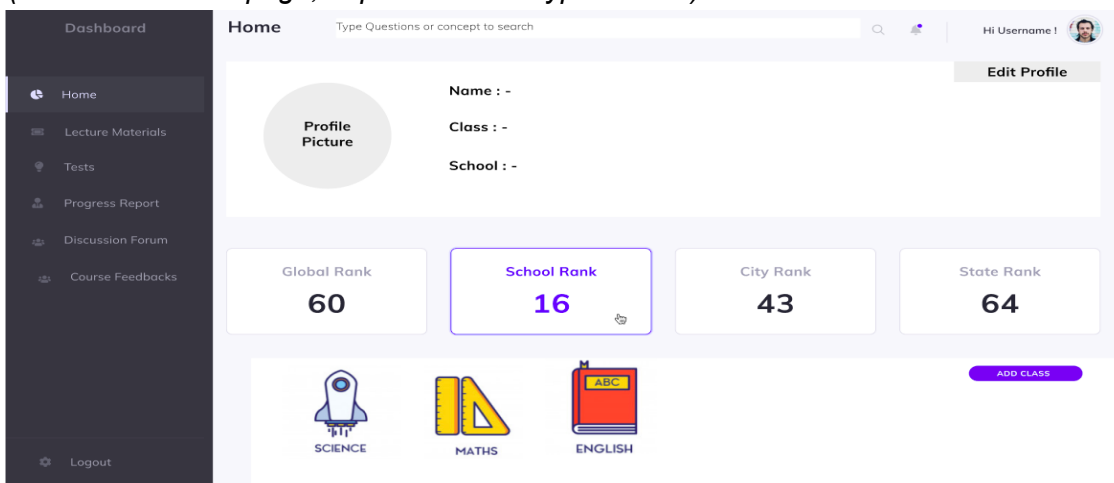
At the first login, it will ask for **otp verification**, which is sent on the register email address (in case of student) or it will ask for **unique code**, which is given to each teacher but their institution (in case of teacher)

Any further login, wont ask for the same.

After login, user is directed to their dynamic profile/dashboard page.

Profile/Dashboard Page

After a user logs into the platform, he/she enters his/her "Homepage" or "Dashboard" as shown below (some sections of page, depends on the type of user).



Students can change and edit the profile details from the edit profile tab. However they are only allowed to do it once. If clicked (after edited) the page will simply show the edited details and no edit option will be there.

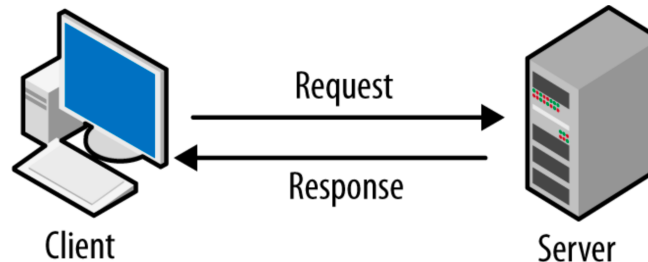
The screenshot shows the 'Edit Profile' form. On the left is a dark sidebar with a 'Dashboard' menu and a list of options: Home, Lecture Materials, Tests, Progress Report, Discussion Forum, Course Feedbacks, and Logout. The main content area has a header with 'Home' and a search bar. Below the header is a large grey oval labeled 'Edit Profile Picture'. To the right of the oval are input fields for 'Name', 'Class', 'School', 'Blood Type', and 'Contact Number'. Below these are fields for 'Father's Name', 'Mother's Name', 'Father's Contact Number', 'Mother's Contact Number', 'Father's Occupation', and 'Mother's Occupation'. A green 'SAVE' button is at the bottom right.

In the Tests tab the students can view/answer the tests and can view scores for assessed tests. Teachers can even create/edit tests and can check the submitted tests. Viewing a specific test, opens the instructions/score and answer section (based on end date). (This same concept is applied in case of lectures tab with only lecture specific things).

The screenshot shows the 'Select Subject' form. The sidebar is the same as in the previous screenshot, but the 'Test' option is highlighted. The main content area has a header with 'Home->Test' and a search bar. Below the header is a purple 'UPGRADE CLASS' button. The 'Select Subject' section features three icons: a rocket for 'SCIENCE', a ruler and triangle for 'MATHS', and a book for 'ENGLISH'.

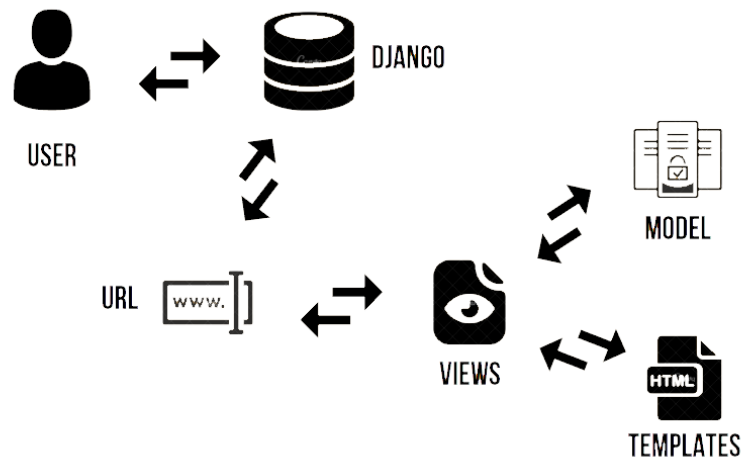
2 Architecture Design

Our web application two major software architectures, *client-server model* (for request - response between users and our web application) and *MVT model* (used by django) for all the backend handling of request and generating response.



Here a HTTP request (stateless) is sent to the server (such as asking for pages, submitting some information to server) and the server on validation of request, do the desired actions and then returns an HTTP response. This same happens again and again for the whole duration, when a user is interaction with digiSchool.

At low-level implementation it involves things like, socket connection, TCP handshake and so on.

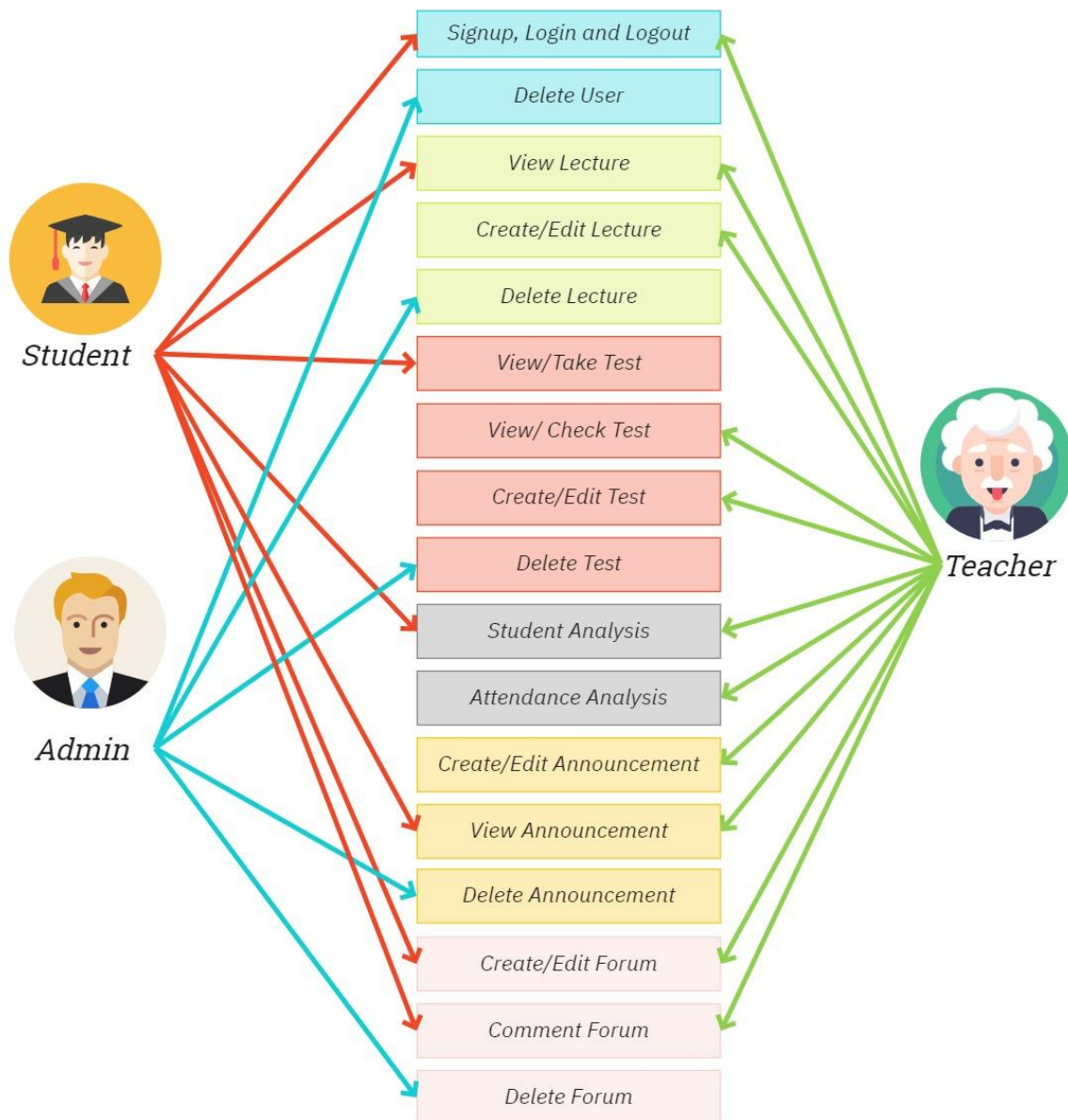


Django (the backend framework that we are using) works using *MVT models*, where *Template* is responsible for all the frontend stuffs, *Models* is responsible for data handling (such as read/write/update) and *Views* works as a *controller* (in MVC model) which manipulates and interacts with *Model* and *Template*, and finally returns the response.

3 Object Oriented Design

3.1 Use Case Diagram

We have three user category/privilege levels, each have a sets of features and available functionality.

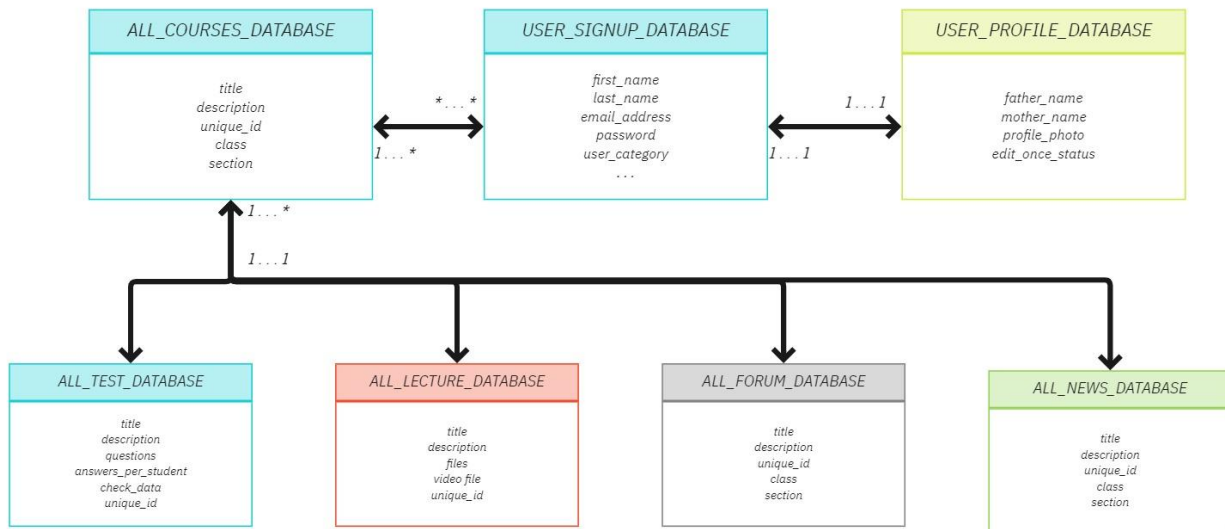


NOTE: Deletion Action is kept only to admin, so that repetitive creation and deletion of important elements (test, lectures, forums) are avoided by the users.

NOTE: **Student** category have very restrictive actions available.

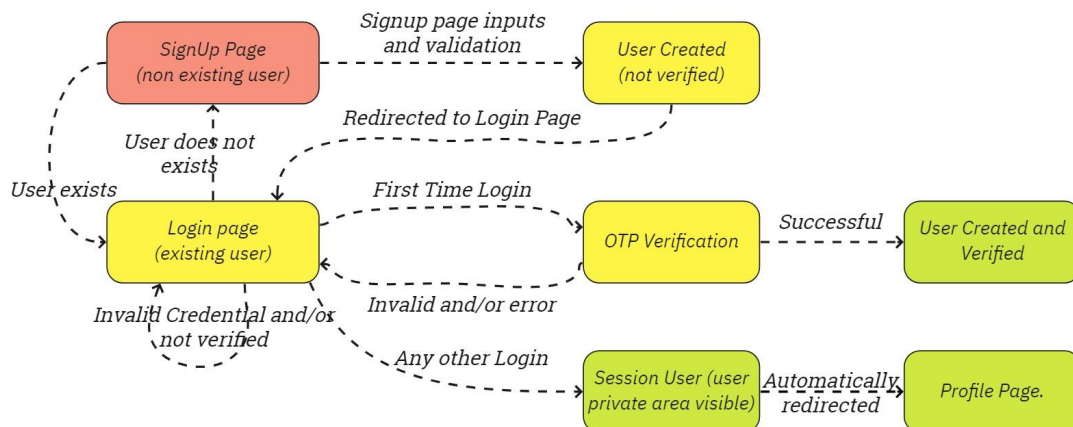
3.2 Class Diagrams

We have used classes only in databases, as due to the architecture of django, all the controller are functions define in views and Templates are simple files.

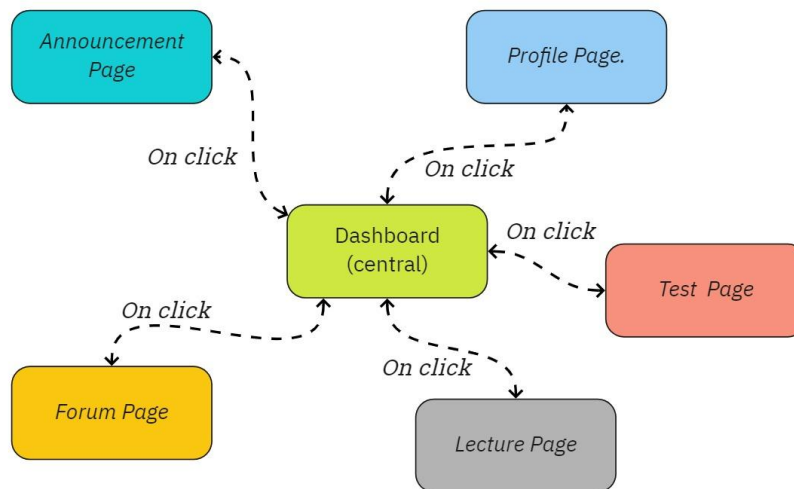


3.3 State Diagrams

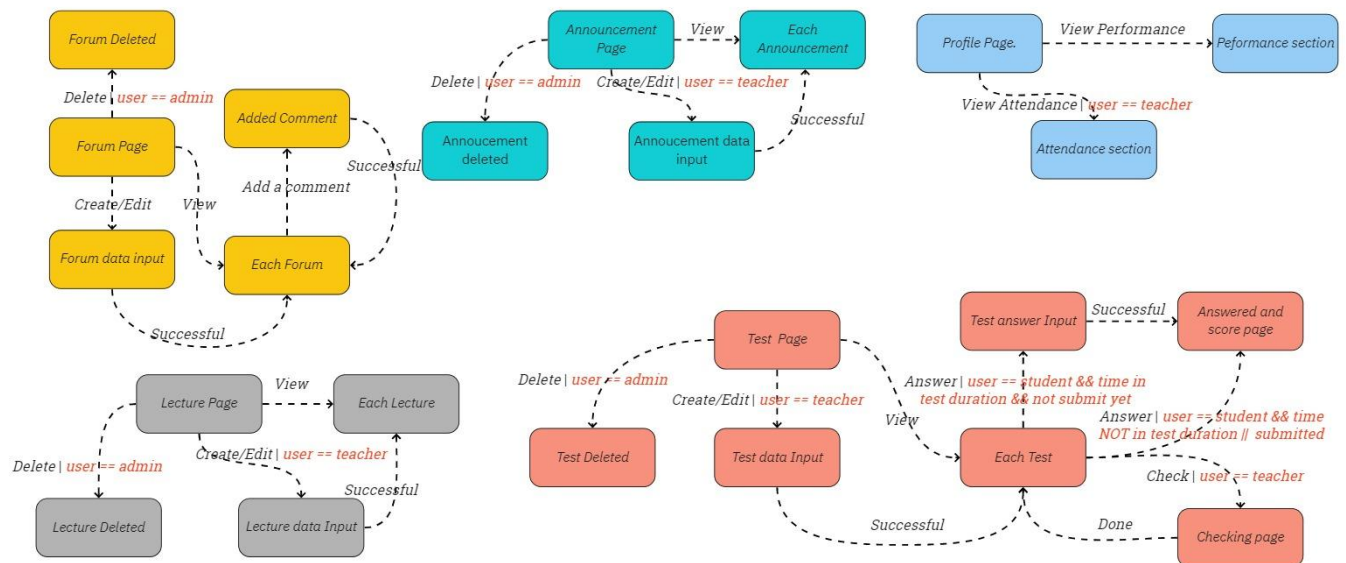
This state diagram shows the transition happened when a user singup, verify itself and login successfully.



This state diagram shows the central connection between different functionality of our web application



This state diagram shows the transition between functionality for each web app's function,



4 Project Plan

The entire sheet can be accessed here:

<https://docs.google.com/spreadsheets/d/1oqR7O5VSPu5boL2b62y-MojfHK5gnMr0AOQnsOYpIDU/edit?usp=sharing>

Function Focused	Description	Time Duration
Signup/Login/Logout with authentication	will implement all the dedication non-functional requirement such as otp verification, password encryption and so on.	0. 5 Week
Forum	implement database, create all view functions mapping to each functionality of forum. data handling and verification+handling user privilege handling	1 Week
Lecture	implement database, create all view functions mapping to each functionality of lecture. data handling and verification +handling user privilege handling	1 Week
Announcement and Profile	implement database, create all view functions mapping to each functionality of announcement and profile. data handling and verification. creating backend fuction, which do performance analysis and graph generation.	1 Week
Test	implement database, create all view functions mapping to each functionality of test. data handling and verification. Storing of answer of all student in a dedicated storage and provide viewing of these answer to teacher and store the answer and score in JSON file.	2 Weeks

5 Other Requirements

<This section is Optional. Please provide any other details that are suitable for being included in the design document .>

Appendix A - Group Log

We had discussions everyday with the team members through whatsapp group/calls and zoom meetings. The team members were also in direct touch with the assigned TA through official Whatsapp Group.

Meeting Minutes	Agenda
Feb 4, 2022 09:30 PM-10:00 PM	Group chat regarding the design.
Feb 5, 2022 04:00 PM-04:45 PM	Group Meet for discussion about various intricacies of the design document.
Feb 6, 2022 10:30 PM-11:30 PM	Group Meet regarding more features that could be added..
Feb 7, 2022 1:00 PM-03:30 PM	Group Meet for discussion on the context diagram and architecture design specifically.
Feb 8, 2022 1:00 PM-1:30 PM	Group Meet for discussion on the class diagram specifically
Feb 10, 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 11, 2022 05:00 PM-05:40 PM	Group meet regarding work distribution
Feb 12, 2022 12:00 PM-12:30 PM	Group Meet for the discussion on the architecture design
Feb 14, 2022 09:00 PM-11:30 PM	Group Meet to change few UI images for better understanding of the feature
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 for final reviewing