
MyKJSSC Mobile Application

1906854

T.Y. B.Sc. IT (Semester V)

Academic Year 2019-2020



K. J. Somaiya College of Science and Commerce
Autonomous, Affiliated to University of Mumbai
Vidyavihar, Mumbai - 400 077



DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project documentation entitled, " **MyKJSSC Mobile Application**", of **Atharva. Pradhan** bearing Seat. No: 1906854 is submitted as the practical work in subject of Software Project Management of T.Y. B.Sc. IT Semester V for the academic year 2019-2020.

In charge

Coordinator

External Examiner

Date:

College Seal

Table of Contents

Sr. No.	Particular	Page No.
1	Chapter 1: Introduction	3
1.1	Background	4
1.2	Objectives	4
1.3	Purpose, Scope and Applicability	4
2	Chapter 2: Survey of Technologies	6
3	Chapter 3: Requirements and Analysis	10
3.1	Problem Definition	11
3.2	Requirements Specification	11
3.3	Project SDLC model	12
3.4	Scheduling	13
3.5	Software and Hardware Requirements	13
3.6	Preliminary Product Description	13
3.7	Conceptual Models	14
	3.7.1 Use case Diagram	14
	3.7.2 Data Flow Diagram	16
	3.7.3 E-R Diagram	18
4	Chapter 4: System Design	19
4.1	Significant Modules in project	20
4.2	Flow-charts or Algorithms for project execution flow	20
4.3	User Interface Design screenshots	22
4.4	Security Issues	25
4.5	Testing Process	25
4.6	Data Dictionary	25
5	Chapter 5: Future Enhancements	27

Chapter 1: Introduction

1.1. Background:

Many colleges have their own websites that support large screen devices such as PC's and Laptops only. When they are opened in mobile devices, they either lack in design or resolution support. The notices posted in the website are jumbled too and not properly sorted.

To solve this issue for mobile devices and app named MyKJSSC app will be developed. The MyKJSSC app will be a mobile application available to android user. The app will provide access to all sorts of notices with easy navigation that are displayed on the college website in one click. The app will allow the users to download the particular notices and view them on the go. It is not only for the students but for the Professors too. Professors can upload notices on the app from anywhere anytime which will be also reflected on the website.

1.2. Objectives:

The main objective of this project is to provide ease to students while accessing their college notices and save time. The MyKJSSC app is expected to:

- Provide a user-friendly interface to access their account and view notices.
- Function in simple and initiative manner.

1.3. Purpose, Scope and Applicability:

This project is under taken to help students in finding and downloading the notices in proper and well organised format as compared to the current website that they are using to view notices in their mobile devices.

1.3.1. Purpose:

This project is under taken to help students in finding and downloading the notices in proper and well organised format as compared to the current website that they are using to view notices in their mobile devices.

1.3.2. Scope

This project will solve the issues of viewing and downloading the notices in mobile devices and will give the professors flexibility of upload notices through their mobile devices with and ease.

1.3.3. Applicability

With the help of MyKJSSC app students and professors can view and download any notices on the go without any hesitation or without facing the problems that they encounter while surfing on the website.

This application also gives professors flexibility to post any notices on the go, rather than visiting the data center and uploading the notice.

The notices uploaded through the mobile device will also be reflected on the website.

1.4. Project Products

The following products will be handed will the Project:

- Training Manual: Will be shown on Start as the User Registers and will be an in-app module.
- Mobile Applications: Final product that will run on both Android OS and iOS.

Chapter 2: Survey of Technologies

For this project we are using “Flutter” Platform which uses “Dart” languages. Apps developed on Flutter are light weight and run on Android OS and iOS simultaneously. The competitors of Flutter are Kiwi which uses Python, Android Kotlin/Java, Swift.

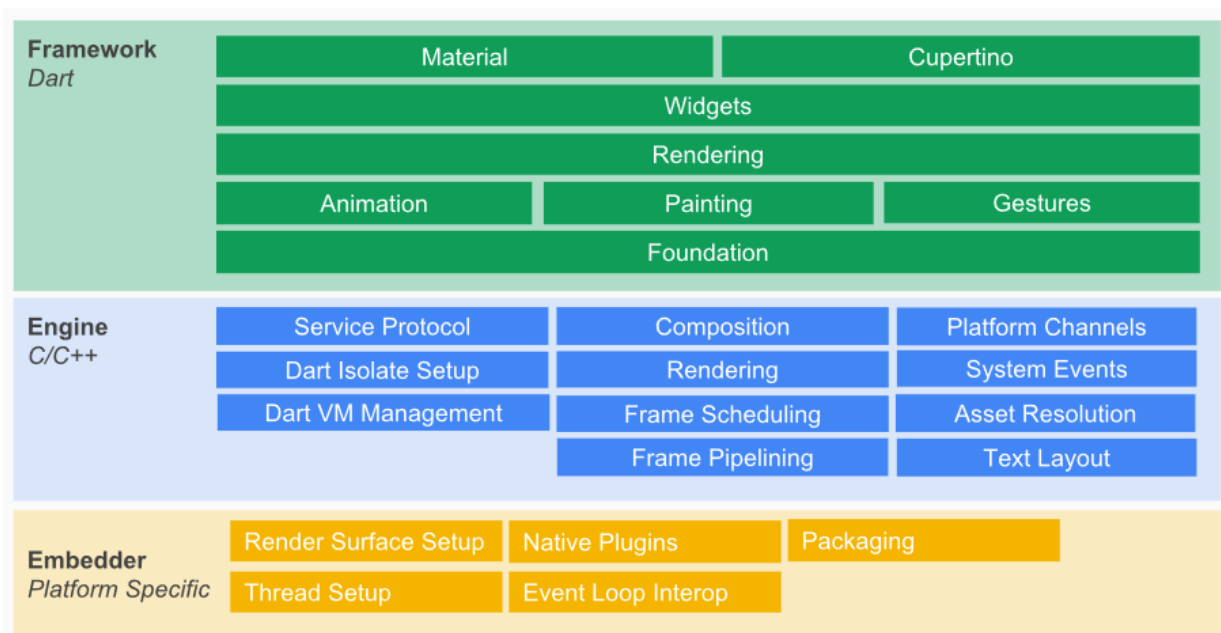
The libraries of Kiwi are not as developed as that of others so the UI’s made on it are not good and pleasing to others, on the other hand Kotlin/Java can only be used to make applications that run on Android OS and this increases the work of developer as he needs to again develop code for other platforms. Same goes in case of Swift it only supports iOS development.

The benefits of Flutter are:

- **Extremely Fast App Development:** With features like *Hot-reload*, code changes in Flutter are reflected as soon as the alterations are made. The *Hot-reload* usually doesn’t take more than milliseconds, which in turn, helps developers to maintain high speeds and dynamicity for mobile app development. Flutter widgets assist in the creation of native interfaces within few minutes.
- **Faster Running of Applications:** Flutter apps provide smooth and seamless scrolling experiences while in use, without much hangs or cuts.
- **Reduced Efforts of Testing:** Since Flutter apps are cross-platform, the testers do not always require to run the same set of tests on different platforms, such as iOS and Android, for the same app. Cross-platform application development enables applications to run on multiple platforms by writing the programming code only once (follows the WORA concept). This saves the time of the testing team. Thus, Flutter is used extensively by every mobile app developer, Utah, for faster testing, delivery and deployment of mobile applications.
- **Access of Native Features:** It is extremely simple to set-up Flutter. Application developers can easily have access to native features of low-level machines where Flutter is initiated. Since it is cross-platform in nature, codes written in existing Swift, Java, Objective-C, etc. can be reused again and again.
- **Excellent User Interfaces:** With built-in design-centric widgets, high-end development tools, advanced APIs, scrolling and navigation features, etc., Flutter helps in the creation of stunning and expressive user interfaces.
- **Reactive Framework:** With reactive framework, the developers do not need to update UI contents manually. Once the variables are updated, the UI changes will be visible automatically.

- **Good for MVP:** Flutter is good for developing MVP (Minimum Viable Product) apps due to its speedy development process and cross-platform nature. The cross-platform concept also aids in reducing costs when apps are developed with Flutter. Many modern undertakings and SMEs are thus employing flutter to increase development speeds and get the maximum outputs at lower costs.

The Flutter framework is organized into a series of layers, with each layer building upon the previous layer.



The upper layers of the framework are used more frequently than the lower layers.

The goal of this design is to help you do more with less code. For example, the Material layer is built by composing basic widgets from the widgets layer, and the widgets layer itself is built by orchestrating lower-level objects from the rendering layer.

The layers offer many options for building apps. Choose a customized approach to unlock the full expressive power of the framework, or use building blocks from the widgets layer, or mix and match. You can compose the ready-made widgets Flutter provides, or create your own custom widgets using the same tools and techniques that the Flutter team used to build the framework.

Nothing is hidden from you. You reap the productivity benefits of a high-level, unified widget concept, without sacrificing the ability to dive as deeply as you wish into the lower layers.

Chapter 3:

Requirements and Analysis

3.1. Problem Definition

MyKJSSC is a mobile application which allows Somaiya ns to access various types of notices like General, Scholarship, Admission, Regular and Exam Timetable, Results.

It allows the Professors to upload notices from the app which will be reflected on the website also which save the time of visiting the data center or waiting for another professor to upload it.

This application sorts all the notices and presents to the user in neat and clean format

3.2. Requirement Specification

Login Module:

- User name
- Password

Registration Module:

- College name
- Email ID
- Full Name
- Contact no.
- Email Verification
- New Password
- Confirm Password

Notice Module

- Notice Name
- Date of Upload
- Uploader's name
- Time Limit
- Tags

3.3. Project SDLC Model

For this project I am selecting incremental model.

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.

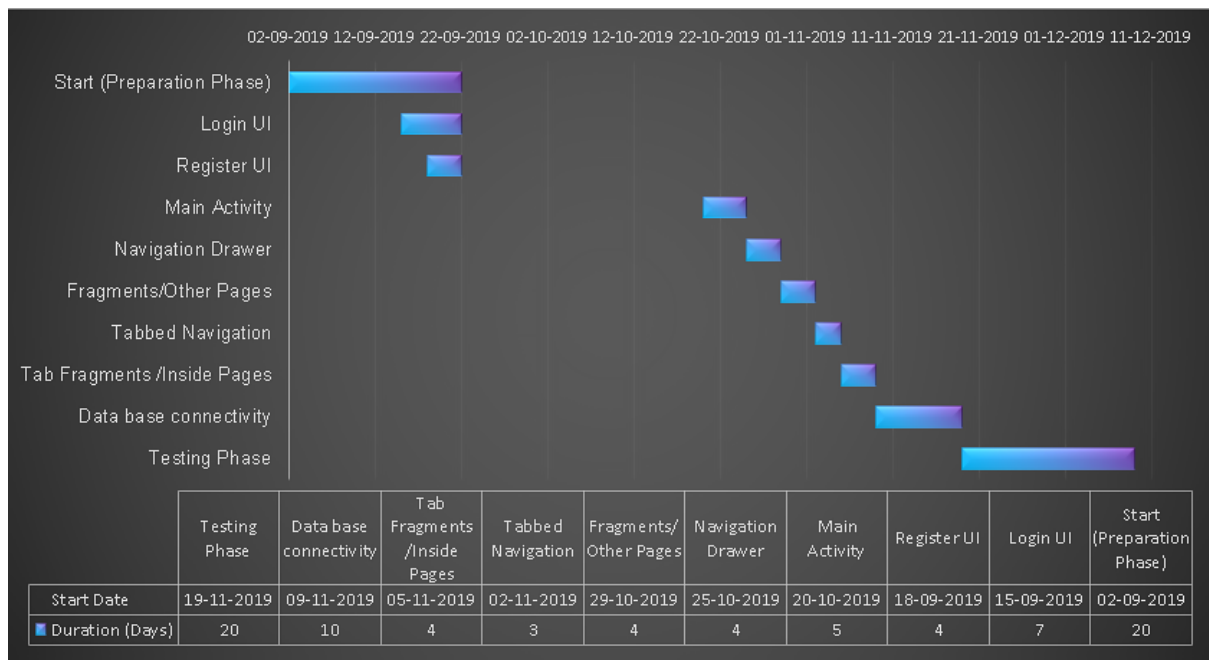
Each iteration passes through the **requirements, design, coding and testing phases**. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.

The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment.

In the current project a base application will be provided to the users as beta testing their feedbacks will take on weekly basis and the modifications will be done immediately until the project reaches the state where it should be released as stable product.

Each module is designed and tested before implementing it in the main application activity and changes are made immediately as the bug is discovered.

3.4. Planning and Scheduling



3.5. Software and Hardware Requirements:

Client-Side Requirements:

- Require a smart phone with either Android OS or iOS.
- Require Internet Connection.
- Android version 5.1.0+ and iOS version 6+.

Developer Side Requirements:

- PC/Laptop with min Ram of 4gb.
- Local Host or Web hosting.
- Emulator or physical device with Android version 5.1.0 or higher and iOS version 6 or higher.

3.6. Preliminary Product Description:

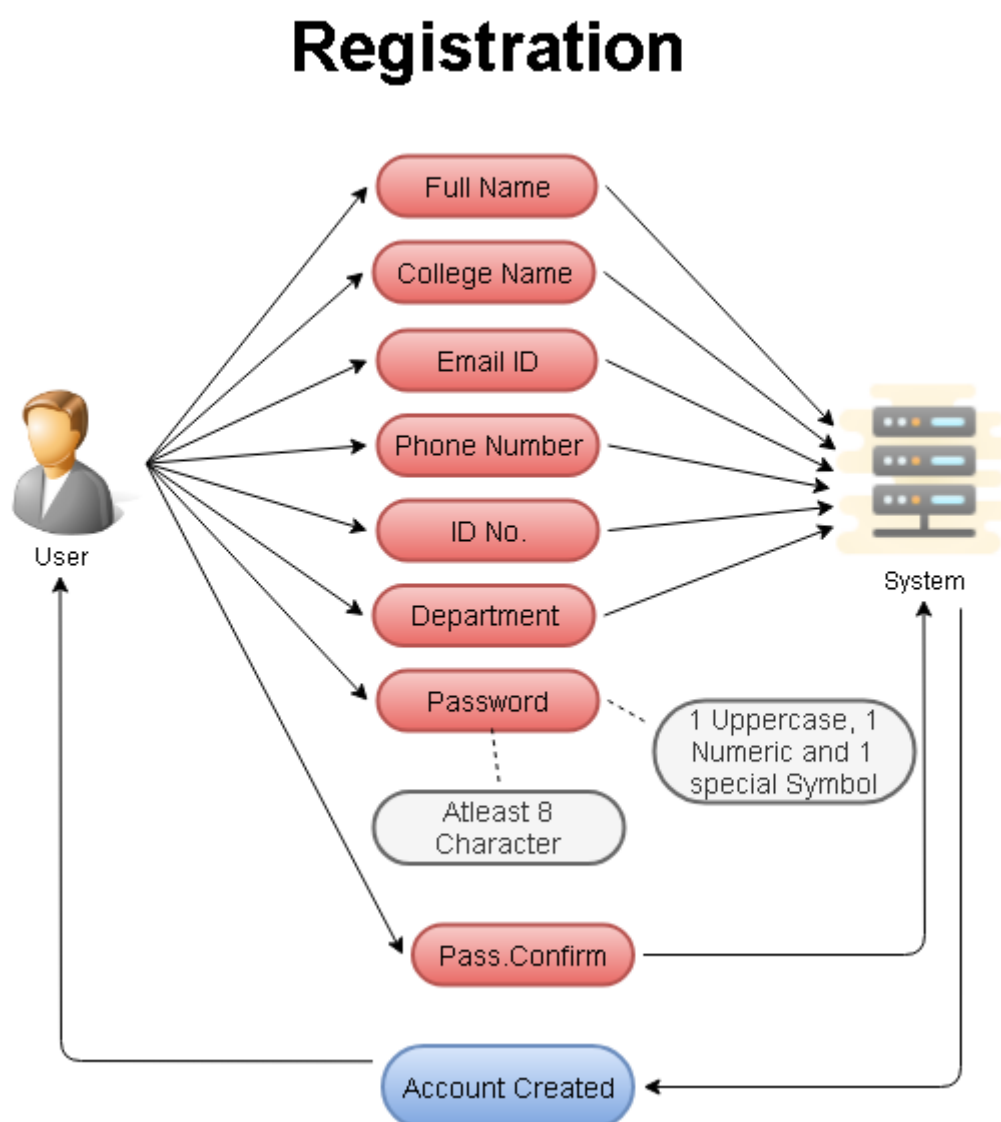
It is going to have 1 login screen and 1 registration screen. Once the user logs in successfully he will be automatically logged in each time he opens the app. The app will have auto sync after every 5 minutes. App will have Notice board section consisting of 3 tabs namely: General, Admissions, Scholarship.

It will have Results, Timetable and Profile section. Results will have further 3 sections i.e. Regular, ATKT and Revaluation and Timetable will have 2 Regular and Exams timetable.

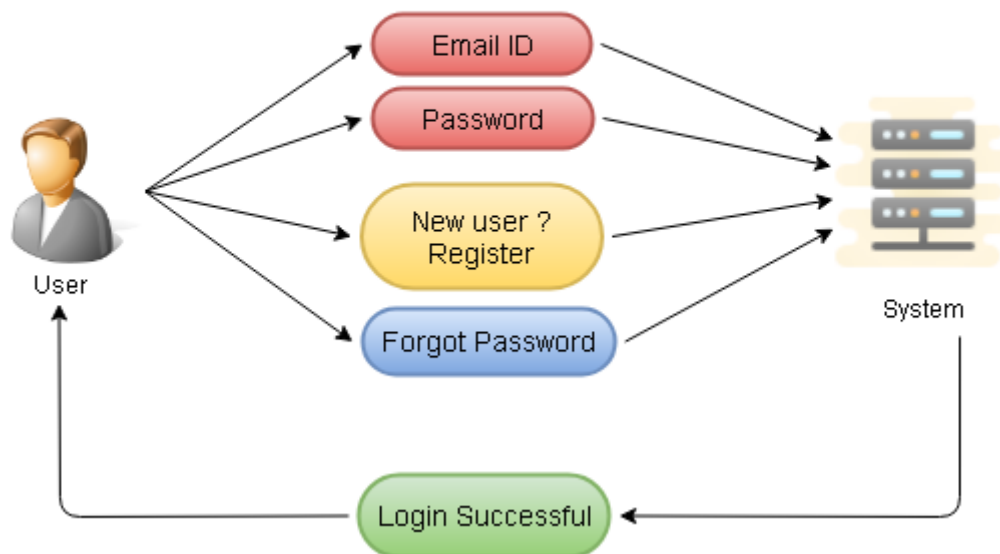
The Profile section will have all the details of the student which will be inevitable. The app will have settings tab so set the font and skin of app and a share button to share the app.

3.7. Conceptual Models:

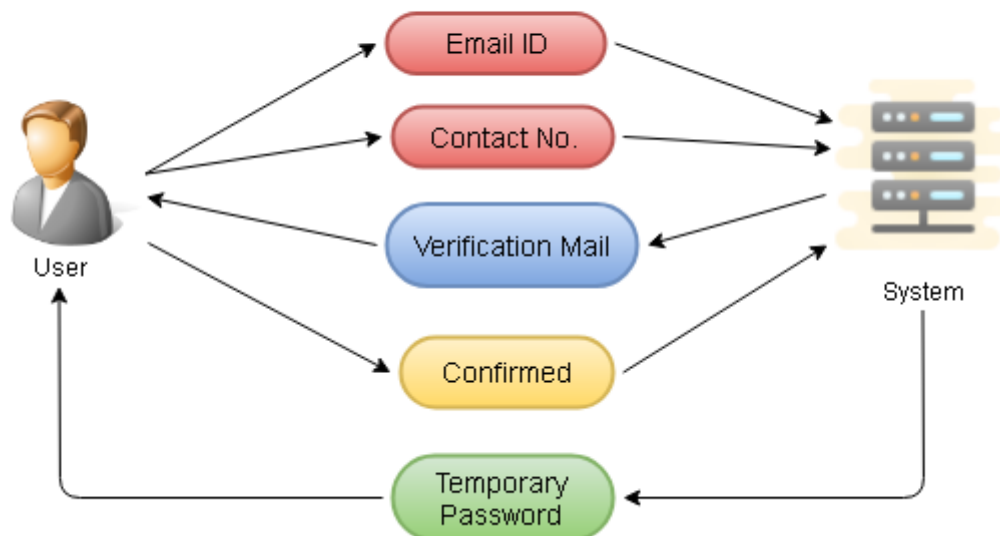
3.7.1. Use case Diagram



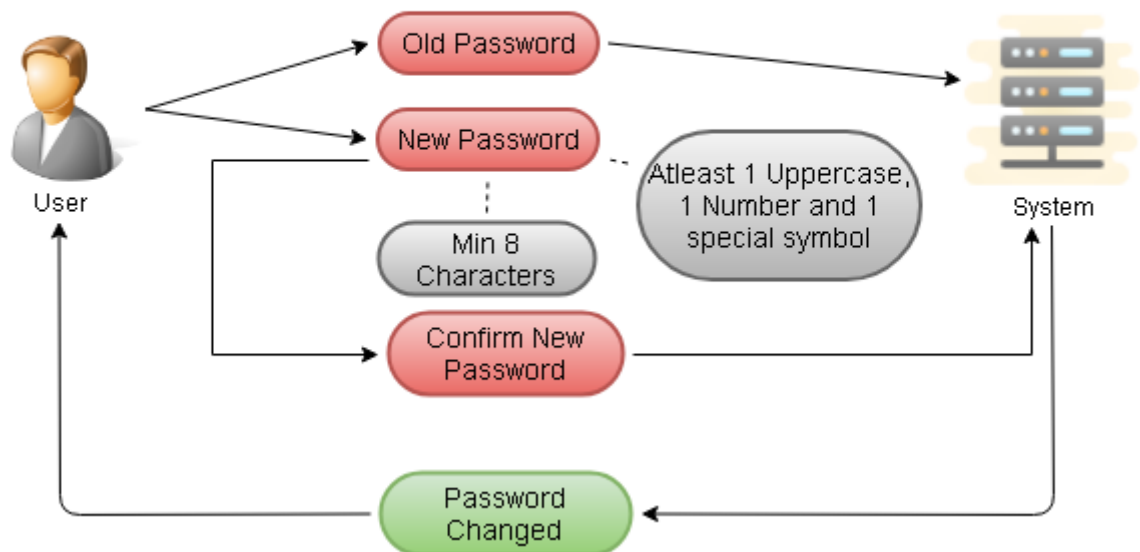
Login



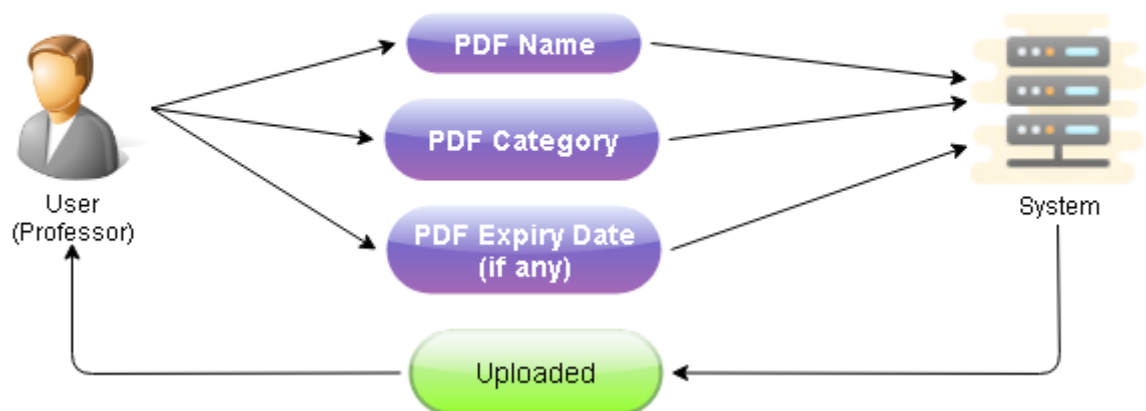
Forgot Password



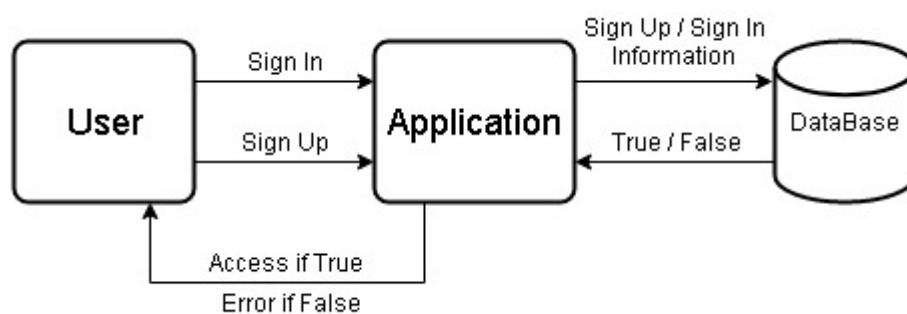
Change Password

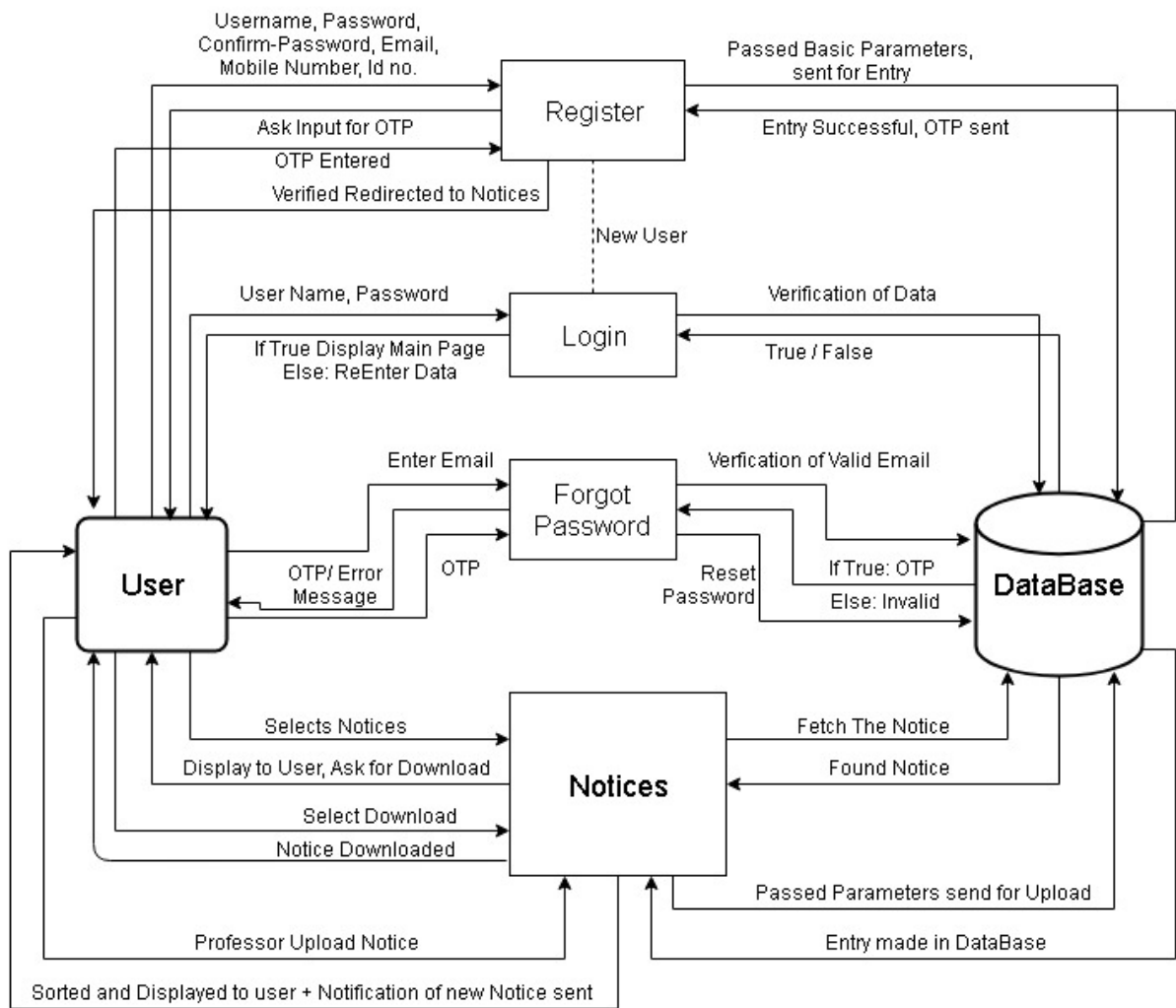


Upload Notices

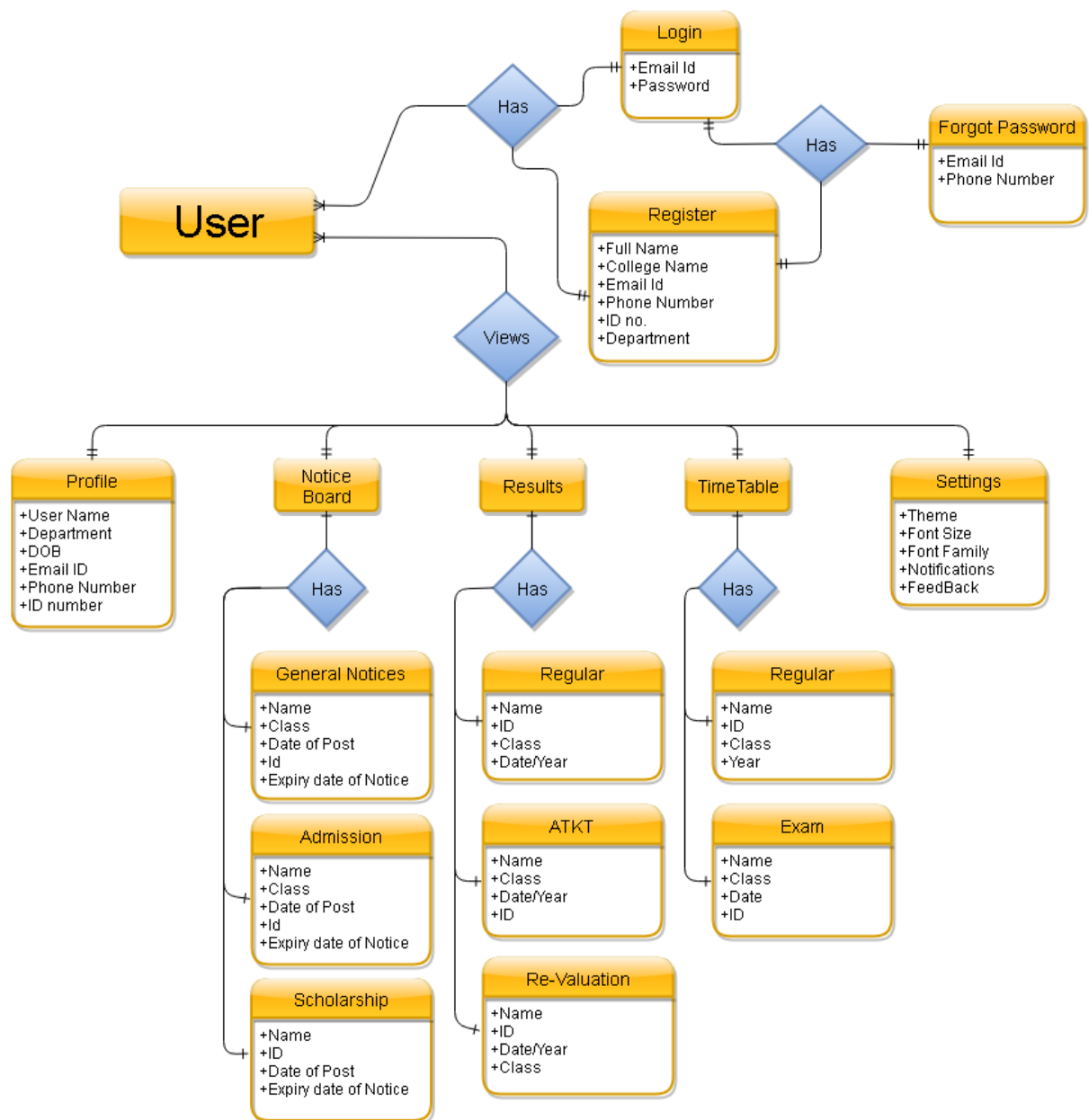


3.7.2. Data Flow Diagram





3.7.3. E-R Diagram



Chapter 4:

System Design

4.1. Significant Modules in project:

Significant Modules of the Project are

- Sorting of Notices: The notices are sorted as per date/month format and with the help of tags which define what type of notice is it like general, scholarship, exam.
- Upload of Document for Professors: This module lets professors upload notices to the website on the go and it will be reflected on the website also.

4.2. Flow-charts or Algorithms for project execution flow:

Login Page Algorithm:

1. On Start app redirects to Login page
2. If User has Logged In earlier:
 - a. Redirect to Main Activity Page
 - b. Go to Step 6
3. Else:
 - a. User inputs Login Credentials.
 - b. Verification of input with database.
 - c. If True:
 - Redirect to Main Activity Page
 - Go to Step 6
 - d. Else:
 - Go to Step 3.a
4. If new User Redirect to Registration Page
5. If forgot Password redirect user to Forgot Password Activity
6. End

Registration Page Algorithm:

1. Take Inputs from the User.
2. If inputs satisfy basic parameters send data to database.
3. Else Go to Step 1.
4. After entering data to database send OTP to user on email.
5. Display Input field for user to enter OTP.
6. If OTP on email == User Input, Redirect to Login Page.
7. Else Go to step 5.
8. End

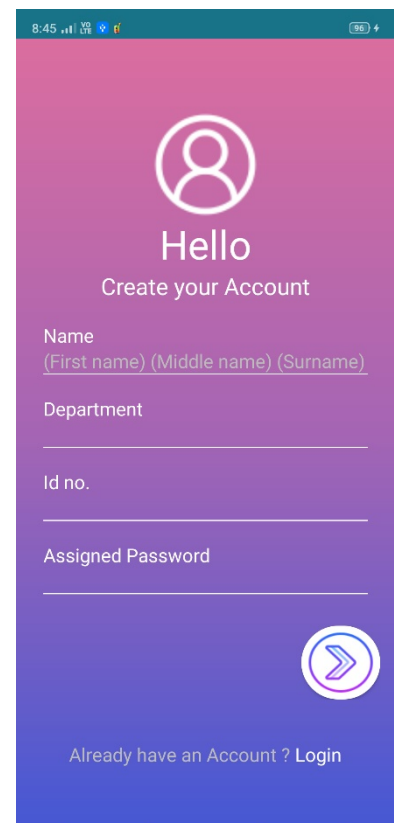
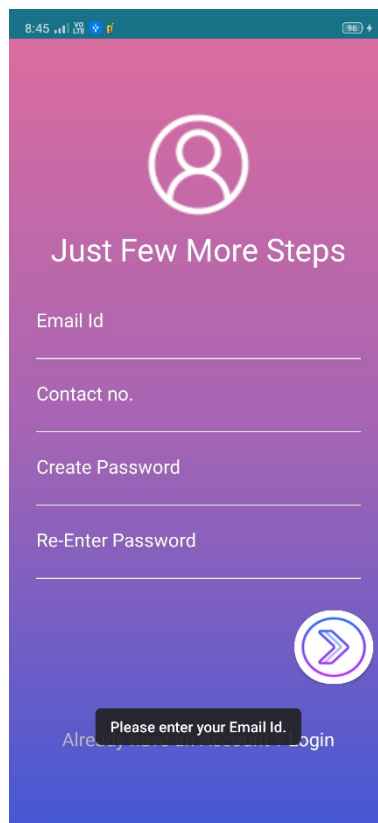
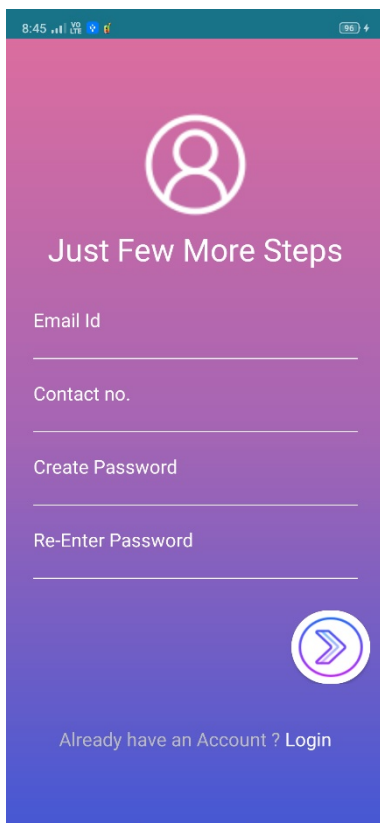
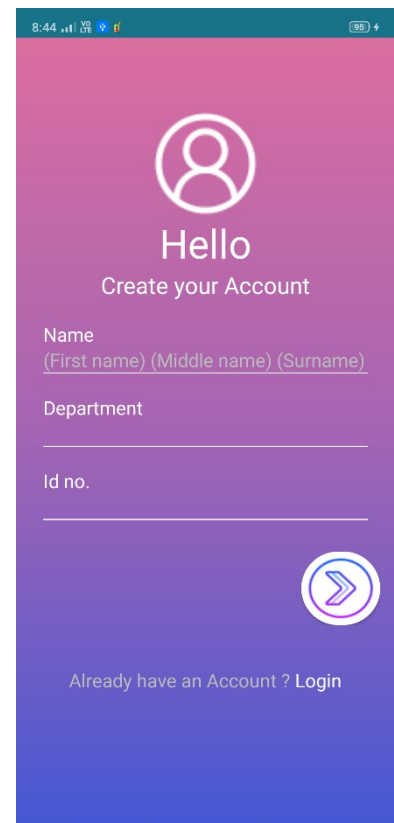
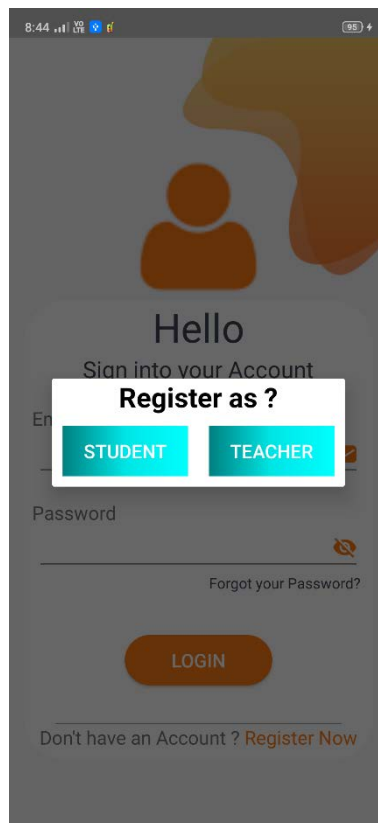
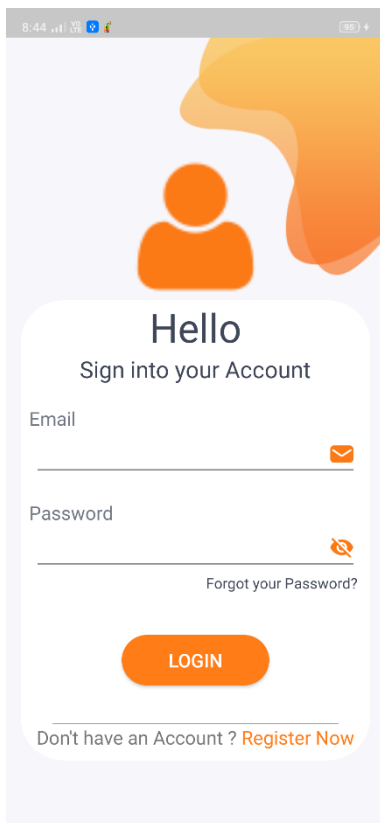
Forgot Password Algorithm:

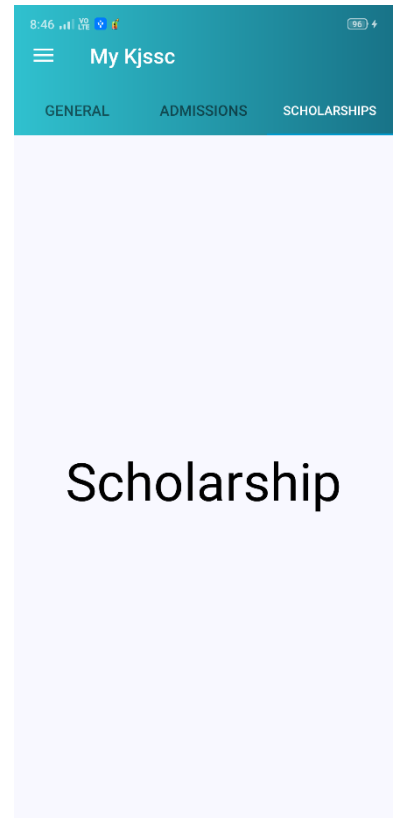
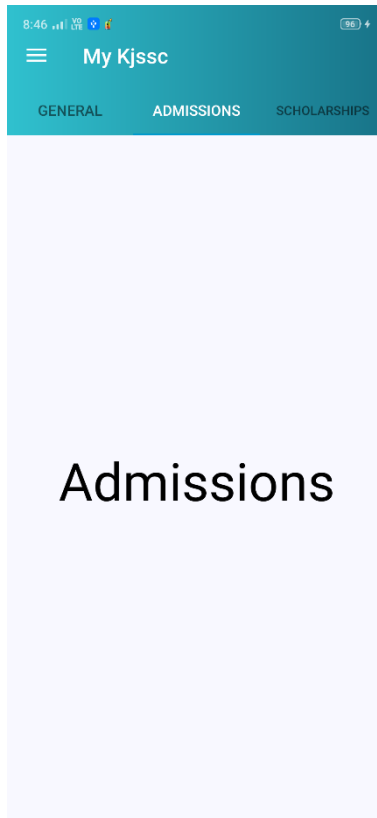
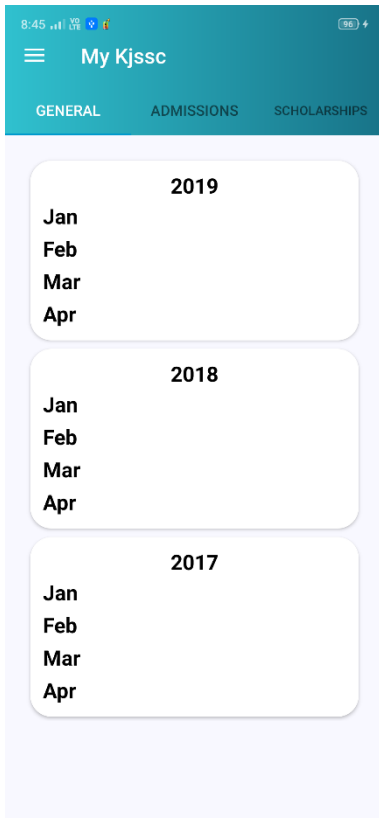
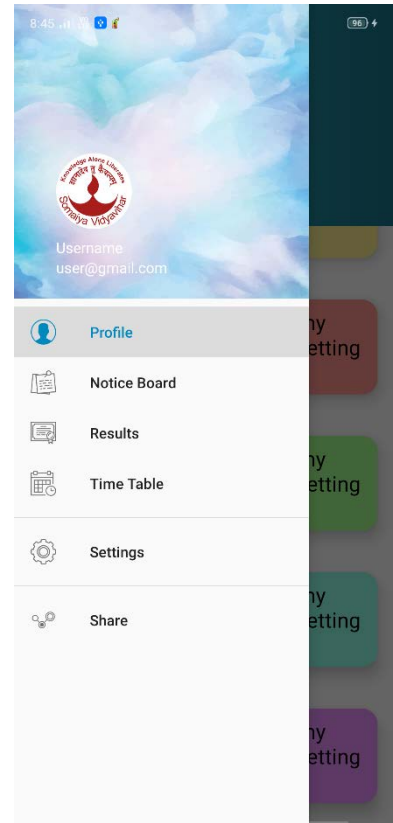
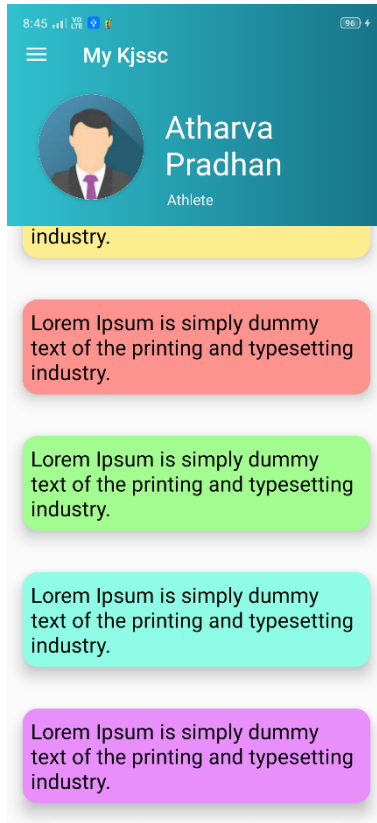
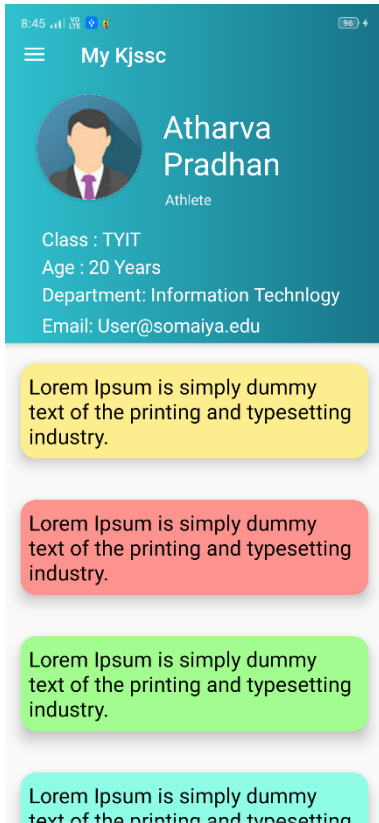
1. User enters Email id.
2. OTP sent to user email id.
3. Input field displayed to user to take OTP Input.
4. If OTP==User input:
 - a. Let user change password.
 - b. Verification of password and Confirm password.
 - c. If match found redirect to Login Page.
 - d. Else Go to Step 4.a and display message password did not match.
5. Else Go to Step 3.
6. End

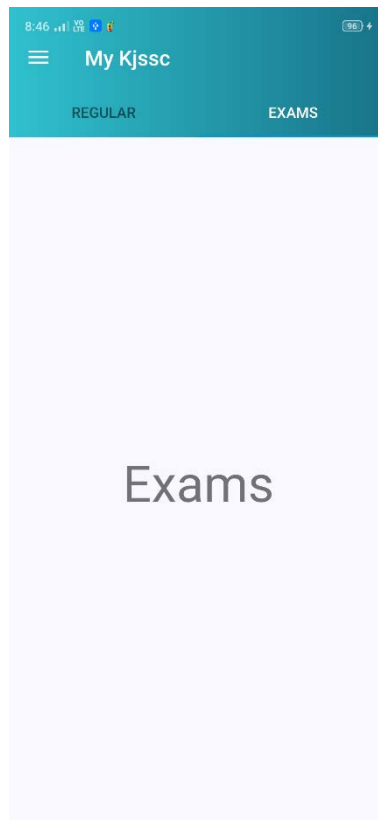
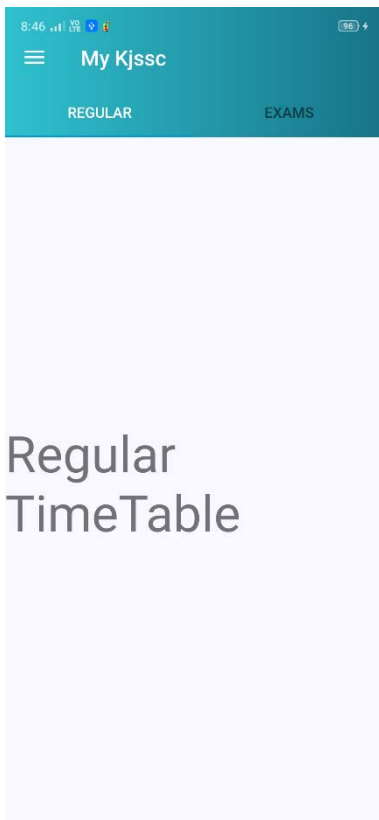
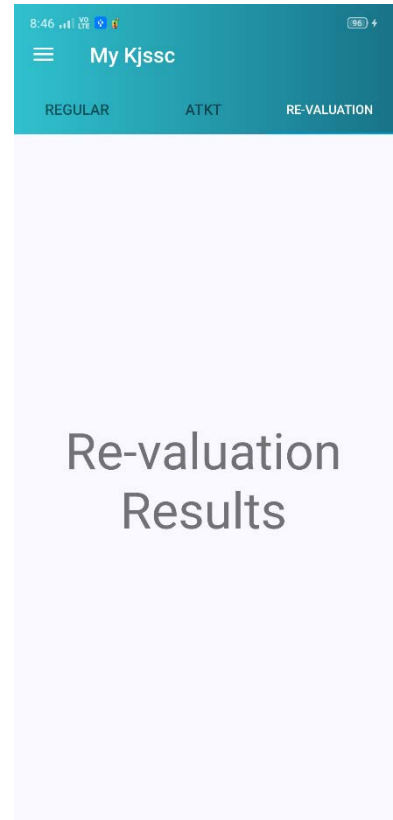
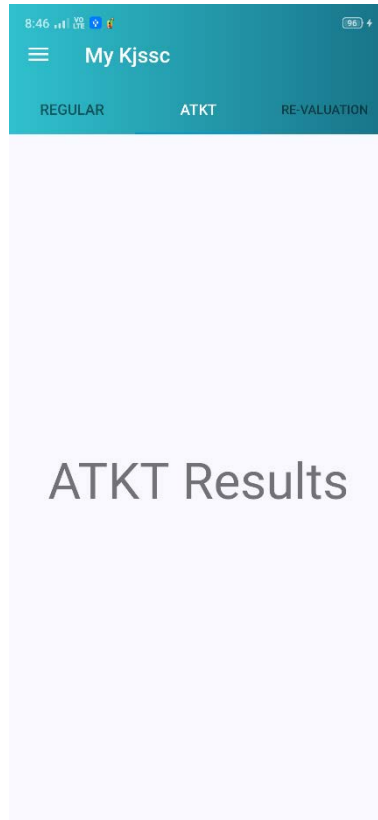
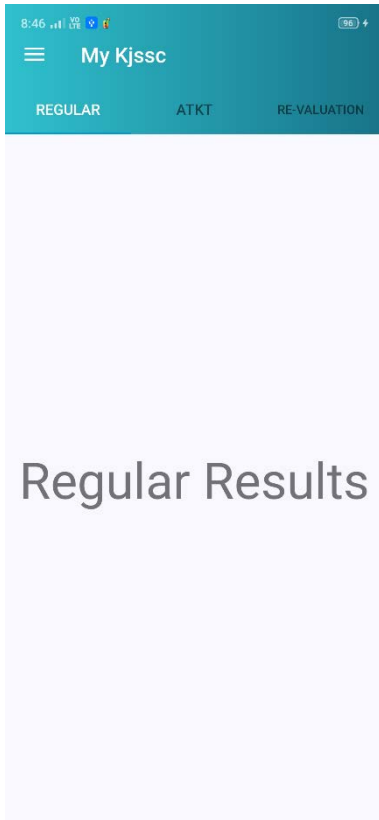
Sorting of Notices and Upload of Notices Algorithm:

1. Professor selects Notices to Upload.
2. After selection of Notices Various tags shown to the professor like general, admission, exam, result.
3. Professor need to select any one of the Tag for notice compulsorily and optional attribute of expiry date displayed
4. If Input satisfy parameter Notice Uploaded.
5. Else Go to Step 3.
6. After Upload of Notice Search for Notice Tag.
7. If Notice Tag == Result, display item in general notice
8. Else if Tag==Admission, display item in admission notices.
9. Continued for all Tags, End.

4.3. User Interface Design screenshots:







4.4. Security Issues:

The security issues that will be faced from the app will be hacking or cracking password to ensure that does not happen passwords will be sent to the database in encrypted format and an OTP will be sent back to the respective device. The notices can be opened from the app itself or from the file-browser also as they are the general notices.

4.5. Testing Process:

The testing of the UI was done as follows

Case 1: Testing of App Drawer:

- Opening of app drawer on every page.
- Clicking all the items and opening all the activities that the app drawer contains.
- Changing the orientation of the screen and checking whether the page get refreshed or the app drawer de-selects the activity.

Case 2: Tabbed Activities Checking:

- All tabbed activities on the main activity were opened and response was checked.
- Orientation Changing was done and tab switching was checked, passed all the parameters Successfully.

Case 3: Login and Registration:

- Validation of the Input was done at client side.
- Testing of whether the app lets user in if fields kept empty was done (if left empty entry prohibited).

4.6. Data Dictionary:

User:

Column Name	Data Type	Size	Constraint
Name	Varchar	20	Not null
Email	Varchar	20	Unique
Contact_Number	Varchar	12	Unique
Password	Varchar	16	Not null
ID_Number	Varchar	12	Primary key, Unique
DOB	Varchar	10	Not null

Notices:

Column Name	Data Type	Size	Constraint
Notice_Name	Varchar	20	Not Null
Tag	Varchar	20	Not Null
Expiry Date	Varchar	10	
Class	Varchar	10	
Uploaded_By	Varchar	20	Not Null

Chapter 5:

Future Enhancements

This project is scalable and can be extended further and can be implemented to all the Somaiya institutions by creating separate table of their data and registering the students through college level. The application can also be made available to other institutes too. The UI will be kept updated throughout the contract period and various add-ons can be added to this project which the students can also create during their project duration