

A Project Report On
ANDROID / IOS APPLICATION: COLLEGE MANAGEMNET APP

Submitted by

Mst. ABHISHEK RAI (SS16IF037)
Mst. AFFAN ANSARI (SS16IF033)
Mst. TAUSIF KHAN (SS16IF032)
Mst. MOMIN MOHD SHAHAB (SS16IF049)
Mst. NIKHIL CHAUBE (SS16IF023)

Project guide

Lect. Namrata Wankhade

In Partial fulfillment of the requirement for the Award of Diploma



INFORMATION TECHNOLOGY DEPARTMENT
GOVERNMENT POLYTECHNIC, MUMBAI
MUMBAI 400051
2019-2020

Certificate

This is to certify that the project entitled Android Application: College Management App being submitted by ABHISHEK RAI (SS16IF037), AFFAN ANSARI (SS16IF033), TAUSIF KHAN (SS16IF032), MOMIN MOHD SHAHAB (SS16IF049), NIKHIL CHAUBE (SS16IF023) to the department of Information Technology, Government Polytechnic, Mumbai for the Award of the Diploma in the academic year 2019-20.

Project Guide

(Lect. Namrata Wankhade)

Head of Department

(Dr. Rajesh A. Patil)

External Examiner

()

Principal

(Prof. S. D. Deshpande)

Acknowledgement

We would, firstly like to express our heartfelt gratitude towards our respected Principal Prof. S. D. Deshpande and our Head of Department Dr. Rajesh A. Patil for providing us immense facilities, guidance and never ending support.

The completion of any inter-disciplinary project depends upon co-operation and combined efforts of several sources of Knowledge. We take this opportunity to express our profound gratitude and deep regards to our guide Prof. Namrata Wankhade for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her, time to time shall carry us long way in the journey of life on which we are about to embark.

We are highly obliged to Lect. Namrata Wankhade , Lect. Deepali B. Gosavi and all the faculties at Government Polytechnic, Mumbai for the valuable information provided by them in their respective fields. We are grateful for their co-operation during the tenure of our project, which also paved the way for the completion of our task on time.

Our Special thanks to Mr. K. G. Karulkar, Mr. S. G. Jadhav, Mr. A. K. Hemade, Mr. P. B. Yadhav, Mr. J. N. Sagar and all the supporting staff who never left a single stone unturned for providing us with all possible resources, when we were in dire need of them.

Lastly, we thank almighty, our parents, family, friends and well wishers who always looked for the chance to help us in whatever means came forth and for their constant encouragement without which the project would not be a distinct reality.

Signature of Student

(Mst. ABHISHEK RAI)

(Mst. AFFAN ANSARI)

(Mst. TAUSIF KHAN)

(Mst. MOMIN MOHD SHAHAB)

(Mst. NIKHIL CHAUBE)

Preface

We take great opportunity to represent project report on mobile application so put some useful information regarding our project before readers. We have made sincere attempts and have tried to present this matter in precise and compact form, the language being as simple as possible. We are sure that the information contained this volume would certainly prove useful for better insight in the scope and the dimension of this project in this true prospective.

The perseverance and deep involvement shown by our group members made this vague task look very interesting and simple. We have designed this application project completely from scratch and we have not incorporated any ready made material from the Internet or any other sources to make our project seemingly more attractive and meaningful.

Abstract

The era of mobile technology opens the windows to the Android and IOS app. The websites are vanishing and the mobile phones are emerging. It's the time to change from conventional websites to apps, which has become the part of our daily routine. We are introducing "GPM.apk" the Android and IOS Application Software which would be a miniature of our college website. It works not only as a website, but also it can work as a small college management software. Our multipurpose program is considering the user as either an albertian or non-albertian, student or parent, faculties or office staffs individually. Project gives a total solution to everyone.

The application becomes also a Mobile version of our official website. It gives us more comfort and a better user interface. It acts as an overview about the campus to a non albertian. The college history, departments, workshops, faculties, library etc & major events conducted & to review them. Latest news and updates about college is got through notifications. Individual account can be created for students. It provides a college identity by admission code when you create an account.

Students can share ideas about projects, discussions. Department organizations can invite or make publicity to their events. Application forms can be taken print. It is also useful for the parents to communicate with the college. Students can interact with faculties directly. Latest news and updates about college is got through notifications. The android app is also a navigator : The application gathers your current location and shows the exact minimal route to the GOVERNMENT POLYTECHNIC MUMBAI, COLLEGE campus accessing the GPS.

SR NO	TOPICS	PAGE NO
1	Chapter 1 Introduction 1.1 Introduction 1.2 Motivation 1.3 Scope of the Work 1.4 Existing system vs proposed system	8
2	Chapter 2 System Requirements 2.1 Hardware Requirements 2.2 Software Requirements	12
3	Chapter 3 System Design 3.1 Flow Chart 3.2 Sequence Diagram 3.3 Data Flow Diagram 3.4 Use case Diagram 3.5 Prototype model	17
4	Chapter 4 4.1 Project Planning And Estimation	20
5	Chapter 5 Coding And GUI 5.1 Coding And GUI	30
6	Chapter 6 Testing 6.1 Unit Testing	93

	6.2 Integration Testing 6.3 System Testing 6.3.1 Recovery Testing 6.3.2 Security Testing 6.4 Performance Testing 6.5 Usability Testing 6.6 Compatibility Testing 6.7 Acceptance Testing 6.8 Dummy Test Cases	
7	Chapter 7 Future Scope	100
8	Chapter 8 Conclusion	102
9	Chapter 9 Reference	104

Chapter 1: Introduction

Chapter 1: Introduction

1.1 Introduction

The era of mobile technology opens the windows to the Android and IOS app. The websites are vanishing and the mobile phones are emerging. It's the time to change from conventional websites to apps, which has become the part of our daily routine. We are introducing "GPM.apk" the android and IOS Application Software which would be a miniature of our college website. It works not only as a website, but also it can work as a small college Management software. Our multipurpose program is considering the user as either an albertian or non-albertian, student or parent, faculties or office staffs individually. Project gives a total solution to everyone.

The application becomes also a Mobile version of our official website. It gives us more comfort and a better user interface. It acts as an overview about the campus to a non albertian. The college history, departments, workshops, faculties, library etc & major events conducted & to review them. Latest news and updates about college is got through notifications. Individual account can be created for students. It provides a college identity by admission code when you create an account.

Students can share ideas about projects, discussions. Department organizations can invite or make publicity to their events. Application forms can be taken print. It is also useful for the parents to communicate with the college. Students can interact with faculties directly. Latest news and updates about college is got through notifications. The app is also a navigator: The application gathers your current location and shows the exact minimal route to the GOVERNMENT POLYTECHNIC COLLEGE campus accessing the GPS.

1.2 Motivation

College management app is one of the important issues for most of the education institutes like school, College, classes, etc. So in our college app there are many features or we can say functionalities like, as in college office we see, there is huge crowd for getting the concession form so, we will provide functionality where user can provide information online and this will be verified by the concerned staff and then the student just have to go in office, and the office staff will take a print and give it to the student. So, in this way the time and the efforts of the staff as well as the students will be saved. To handle this functionality, we make use of our admin pannel which will have a staff login, from where he can obtain the data.

And also Displaying the student result on the app. In this functionality will allow easy seeing of results, instead of going to college and seeing the result the student can sit at home and see the result. In our project we will provide the result of the particular student when he enters his enrollment number. Whenever the results comes, an notification will be sent to the students phone and he can then view his result. The students will quickly get an update when the results come allowing faster access to the results.

And also Complain Portal in this functionality will help the students and faculties to give complain about the any aspect of the college. The concerned team will look into the matter and will take the necessary actions on it. This functionality will be improve the working of the institute. This helps in smooth functioning of the institute and every particular individual will be satisfied.

1.3 Scope of Work

The scope of the project is the system on which the application is installed , i.e the project is developed as a Android mobile application, and it will work for a particular institute. But later on the project can be modified to operate it by the students only.

The purpose of developing mobile application is to computerized As, in college office we see, there is huge crowd for getting the concession form so, we will provide functionality where user can provide information online and this will be verified by the concerned staff and then the student just have to go in office, and the office staff will take a print and give it to the student. So, in this way the time and the efforts of the staff as well as the students will be saved. To handle this functionality, we make use of our admin pannel which will have a staff login, from where he can obtain the data.

And also Displaying the student result on the app. In This functionality will allow easy seeing of results, instead of going to college and seeing the result the student can sit at home and see the result. In our project we will provide the result of the particular student when he enters his enrollment number. Whenever the results comes, an notification will be sent to the students phone and he can then view his result. The students will quickly get an update when the results come allowing faster access to the results.

And also Complain Portal in this functionality will help the students and faculties to give complain about the any aspect of the college. The concerned team will look into the matter and will take the necessary actions on it. This functionality will be improve the working of the institute. This helps in smooth functioning of the institute and every particular individual will be satisfied.

1.4 Existing system vs Proposed system

Since our college was not having an application for the college management we created college management app for a smooth functioning and smooth data flow in the organization. This college management application makes it easier to handle all the management related operations. Our motive was to create a proposed system that fulfill the needs of the institute. Our system not only reduces the complexity in the data collected but also it helps the students do their work successful and easily. This propose system includes the features like railway concession, feedback, and query portal. These modules help the system work successfully.

Chapter 2: System Requirement

Chapter 2: System Requirement

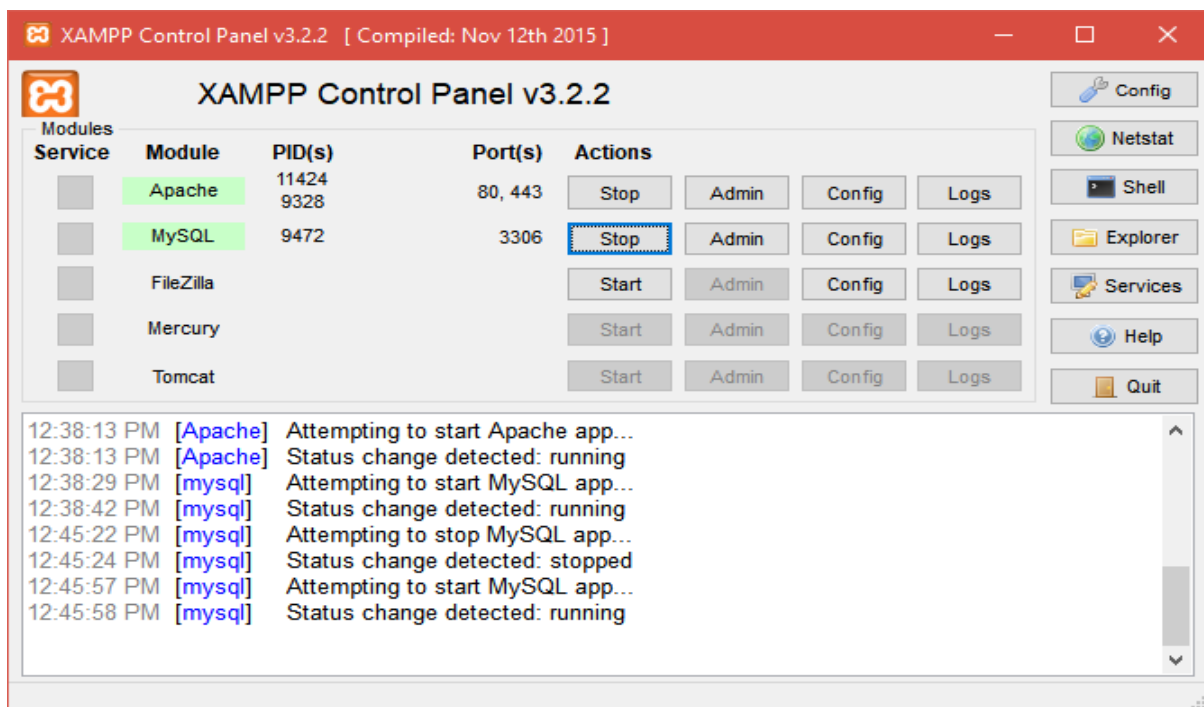
2.1 Software Requirements

2.1.1 ADMIN

2.1.1.1 XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply



on an operating system by a developer, with the advantage a number of common add-in applications such as Wordpress and Joomla! can also be installed with similar ease using Bitnami .

The term XAMPP is an apparent acronym. However, there is no official acronym expansion specified on the Apache Friends website. Their homepage header reads "XAMPP Apache + MariaDB + PHP + Perl", indicating that this abbreviation is MySQL was replaced with MariaDB on 2015-10-19 and beginning with XAMPP versions 5.5.30 and 5.6.14, effectively altering the meaning of the acronym.

While both letters P are de facto interchangeable, convention used at the Apache Friends website indicates that the first letter P is short for PHP and the latter letter P is short for Perl.

2.1.1.2 WEB browser

Google chrome (version number 6.0.472 and above)

Mozilla version number (36.0.4 and above)

Microsoft edge (20.10512 and above)

2.1.2 Android as Operating System

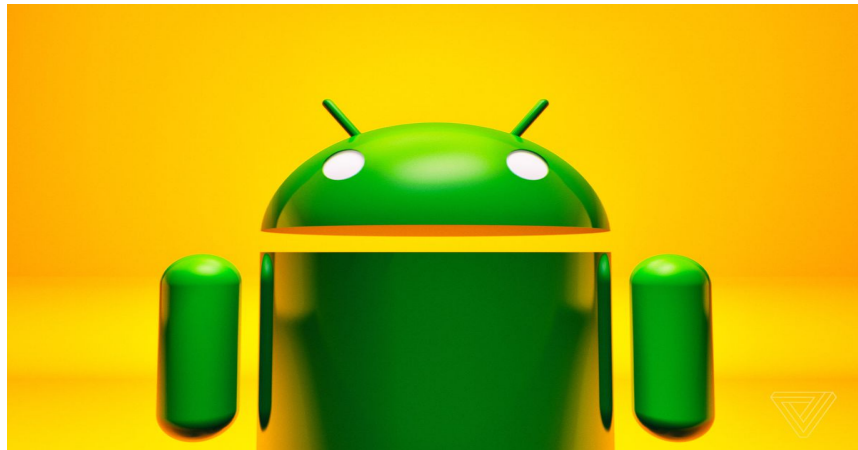


Figure 3.2: Android OS

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The operating system has since gone through multiple major releases, with the current version being 9 "Pie", released in August 2018. The core Android source code is known as Android Open Source Project (AOSP), and is primarily licensed under the Apache License.

Android version (lollipop 5.0 and above)

2.1.3 ios as Operating System

iOS (formerly iPhone OS) is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, iPad, and iPod Touch. It is the second most popular mobile operating system globally after Android.
ios version number(ios7 and above)

2.1.4 React native

React Native is a JavaScript framework for building native mobile apps. It uses the React framework and offers large amount of inbuilt components and APIs.
This tutorial is designed for JavaScript and React developers who aspire to learn mobile building skills. By following this course, you will expand your React and JavaScript knowledge, learn some concepts of functional programming, and prepare to enter the mobile world. Since JavaScript world is moving forward, we will keep up with it and use EC6 syntax in this tutorial. To be able to follow this tutorial, you should be familiar with React and have solid JavaScript knowledge. Even if you do not have previous experience with React, you should be able to follow it. In this tutorial, we will explain some fundamental React concepts.

2.1.5 React native app

The apps you are building with React Native aren't mobile web apps because React Native uses the same fundamental UI building blocks as regular iOS and Android apps. Instead of using Swift, Kotlin or Java, you are putting those building blocks together using JavaScript and React.

2.2 Hardware Requirements

Sr. no.	Devices	Ram
1.	Android	1gb
2.	ios	1gb
3.	Pc or laptop	1gb

2.3 Non-functional Requirements

The non-functional requirements are as follows:

- Convenience: application provides convenience of college related activities.
- Accessibility: access of class time-table, curriculum and college notices.
- Saves Time: faster reach to the content updated by college.
- Cheaper: application will be freely available.
- Mobility: application on mobile devices makes it easier to carry and light.

Chapter 3- Project Planning and Estimation

Chapter 3- Project Planning and Estimation

3.1 Planning-

Planning is the very initial phase of the project development. When we decided to make a project, we first planned everything that would take the project towards the success.

3.1.1 Our project plan included the following things-

When to start the project?- We started the project from September onwards.

We also assigned different functions to each person in the group. We started the project planning by collecting information from different sources. We went through different aspects of planning phase making the resources available.

Once a project is found to be feasible, software project managers undertake project planning. Project planning is undertaken and completed even before any development activity starts. Project planning consists of the following essential activities:

- Estimating the following attributes of the project:

Project size: What will be problem complexity in terms of the effort and time required to develop the product?

Cost: How much is it going to cost to develop the project?

Duration: How long is it going to take to complete development?

Effort: How much effort would be required?

The effectiveness of the subsequent planning activities is based on the accuracy of these estimations.

- Scheduling manpower and other resources
- Staff organization and staffing plans
- Risk identification, analysis, and abatement planning
- Miscellaneous plans such as quality assurance plan, configuration management plan, etc.

3.2 Estimation is the process of finding an estimate, or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable.

Estimation determines how much money, effort, resources, and time it will take to build a specific system or product. Estimation is based on –

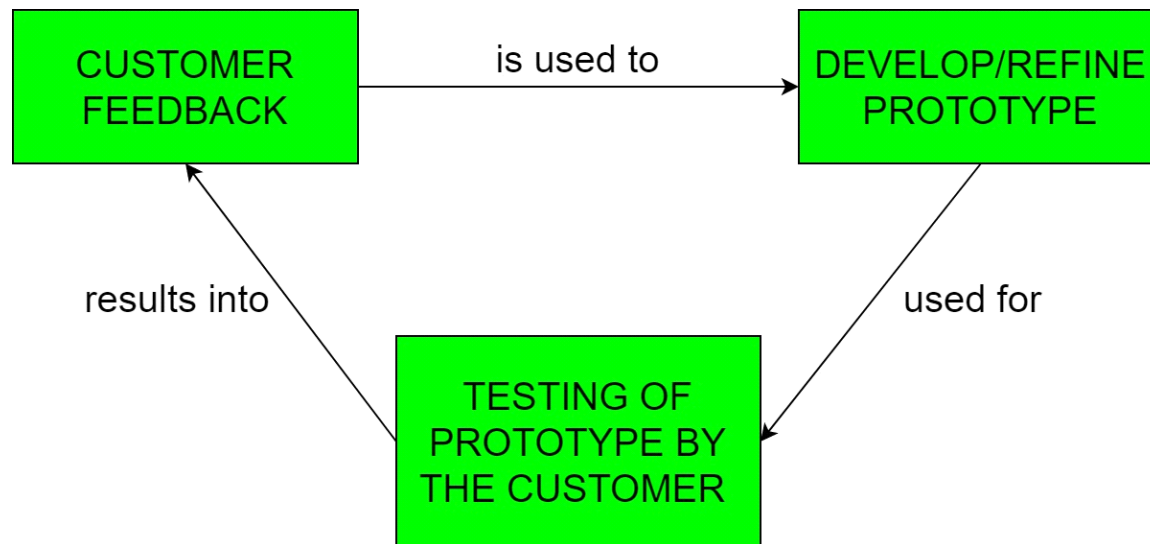
- Past Data/Past Experience
- Available Documents/Knowledge
- Assumptions
- Identified Risks

The four basic steps in Software Project Estimation are –

- Estimate the size of the development product.
- Estimate the effort in person-months or person-hours.
- Estimate the schedule in calendar months.
- Estimate the project cost in agreed currency.

3.3 Prototyping Model :-

Prototyping is defined as the process of developing a working replication of a product or system that has to be engineered. It offers a small scale facsimile of the end product and is used for obtaining customer feedback as described below:



The Prototyping Model is one of the most popularly used Software Development Life Cycle Models (SDLC models). This model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.

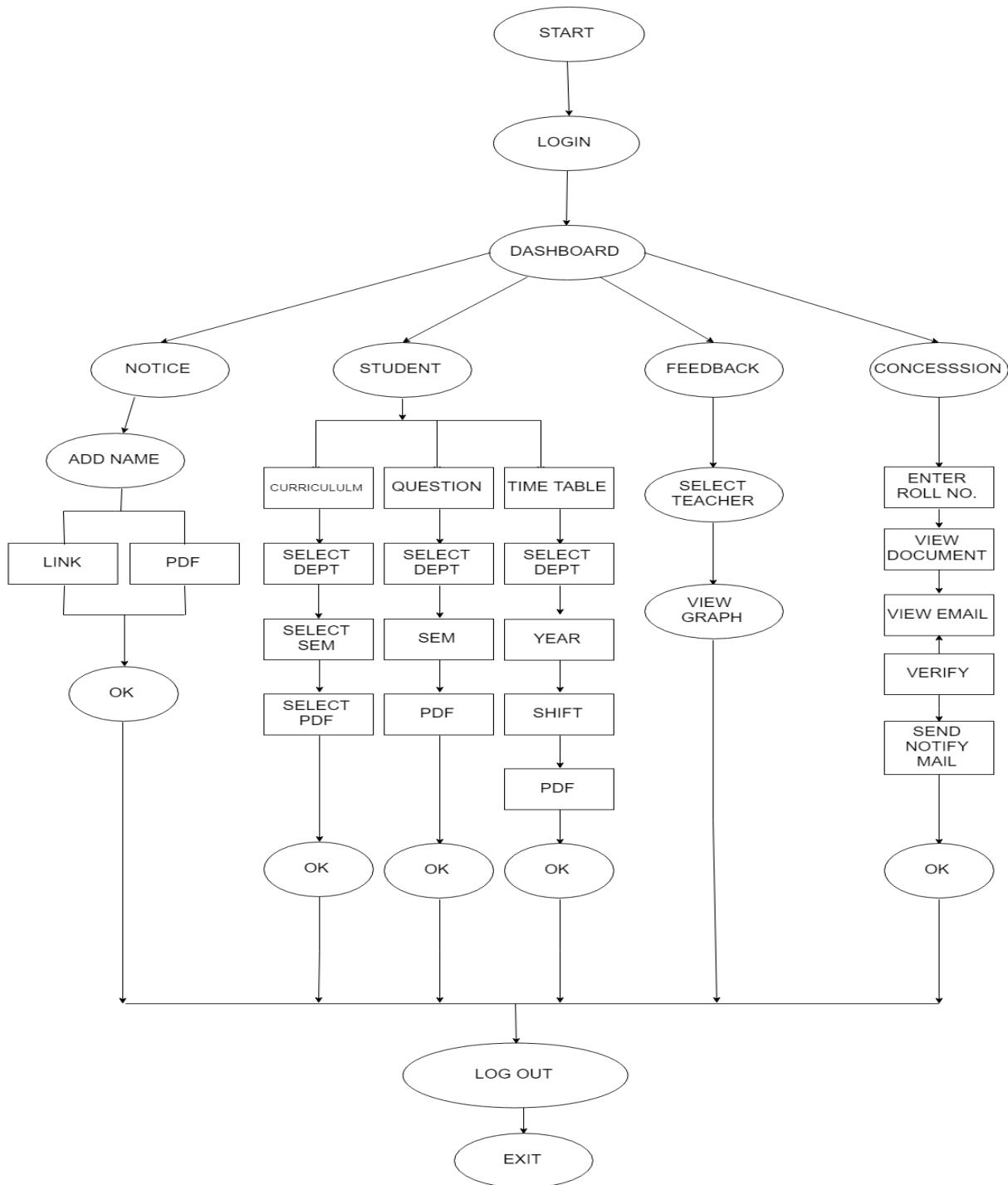
In this process model, the system is partially implemented before or during the analysis phase thereby giving the customers an opportunity to see the product early in the life cycle. The process starts by interviewing the customers and developing the incomplete high-level paper model. This document is used to build the initial prototype supporting only the basic functionality as desired by the customer. Once the customer figures out the problems, the prototype is further refined to eliminate them. The process continues till the user approves the prototype and finds the working model to be satisfactory.

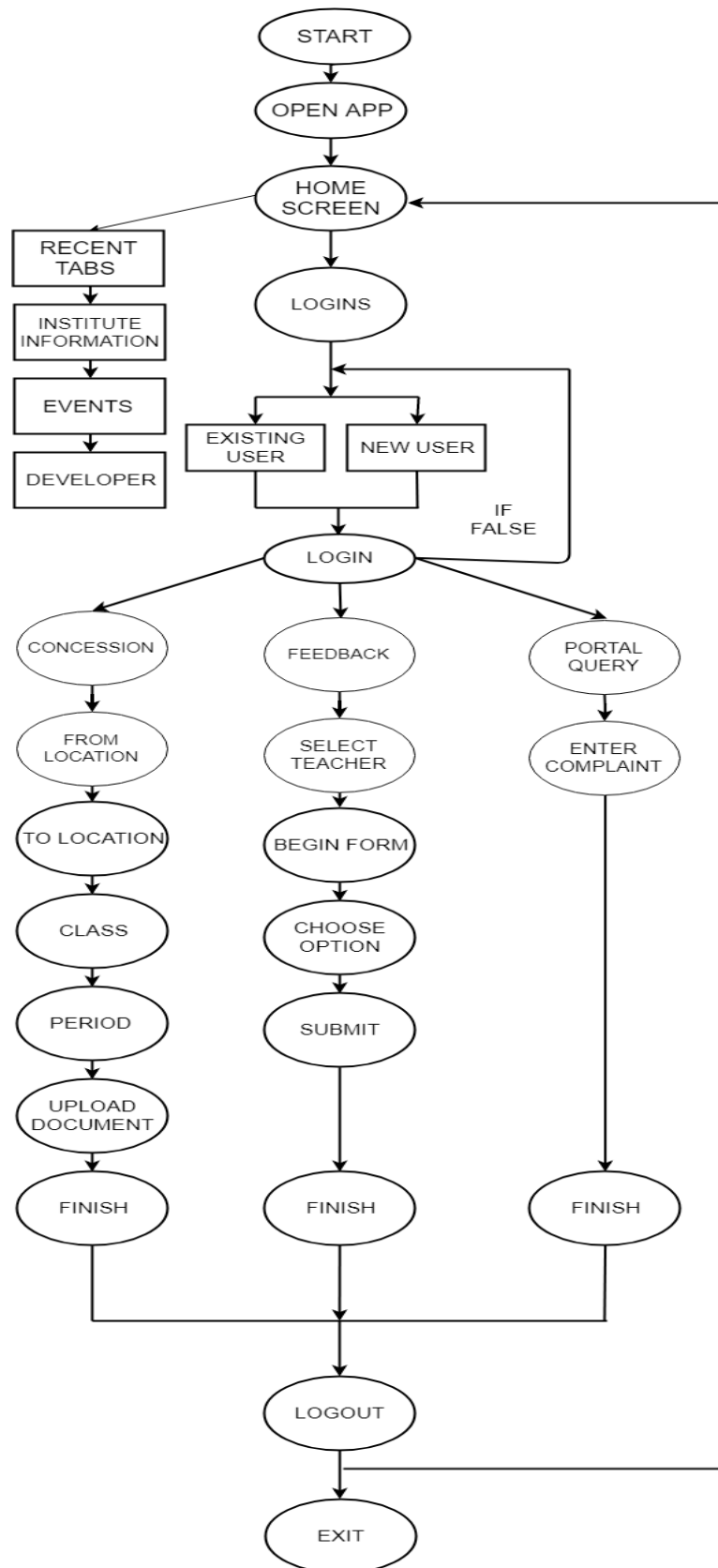
Chapter 4: System Design

Chapter 4: System Design

4.1 Flow Chart -

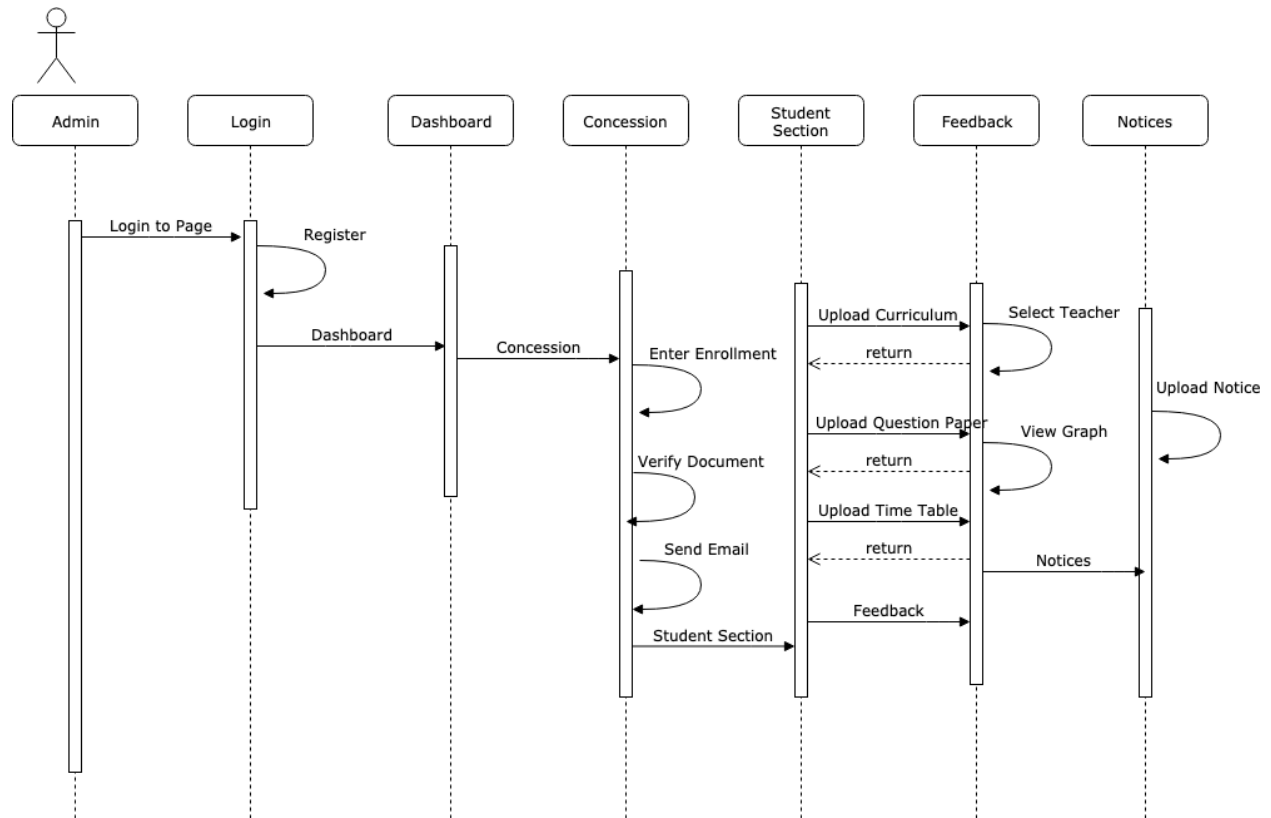
4.1.1 Admin Panel :-



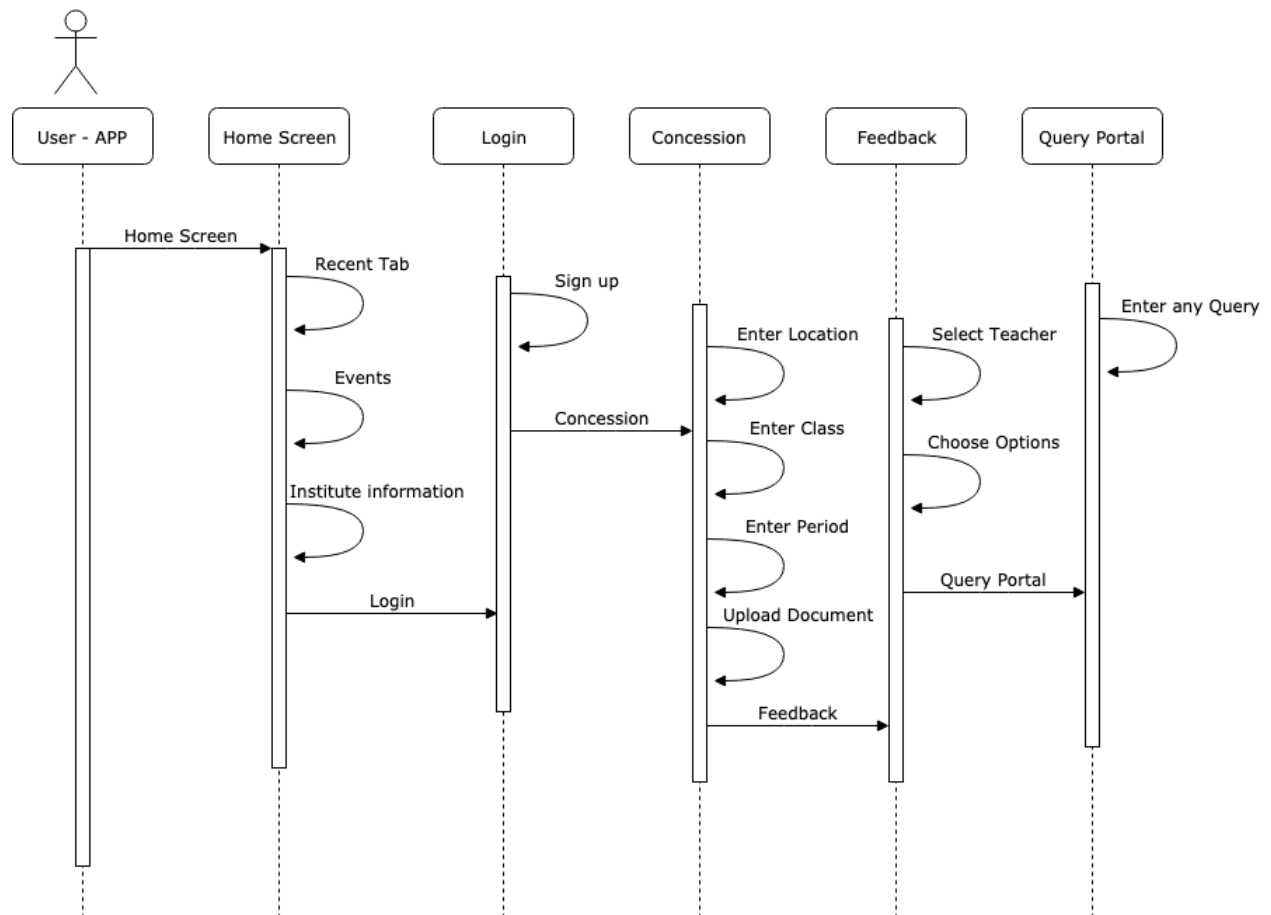
4.1.2 User Panel :-

4.2 Sequence Diagram :-

4.2.1 Admin Flow Diagram :-

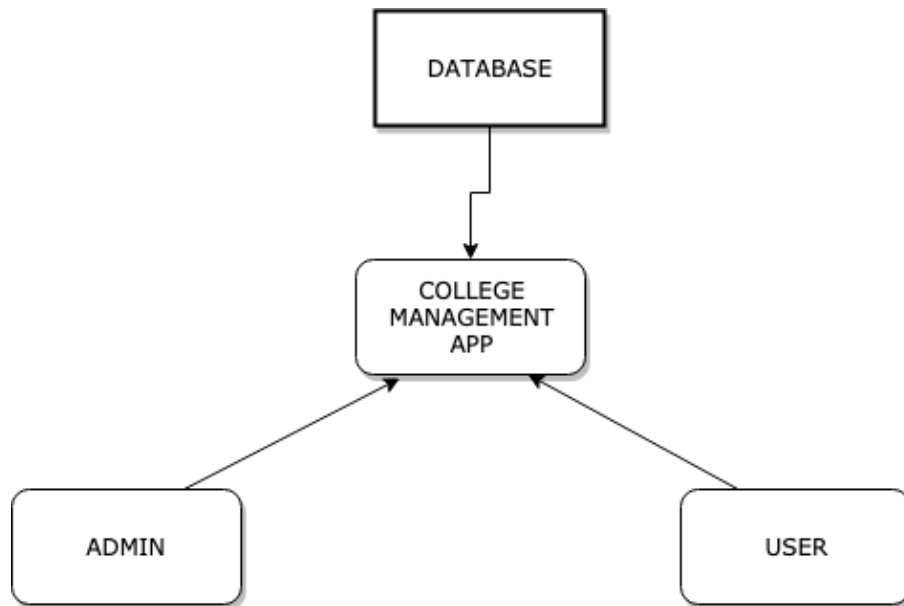


4.2.2 User Flow Diagram :-

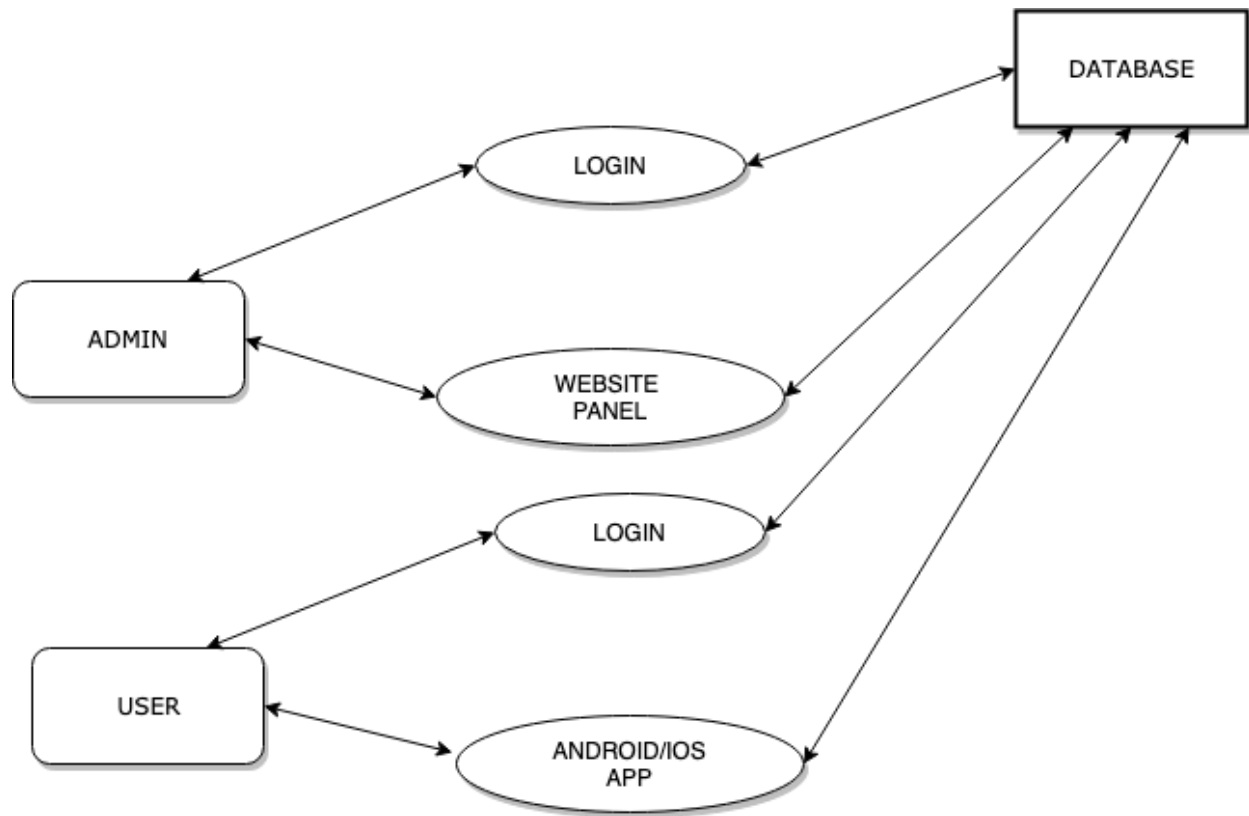


4.3 Data Flow Diagram : -

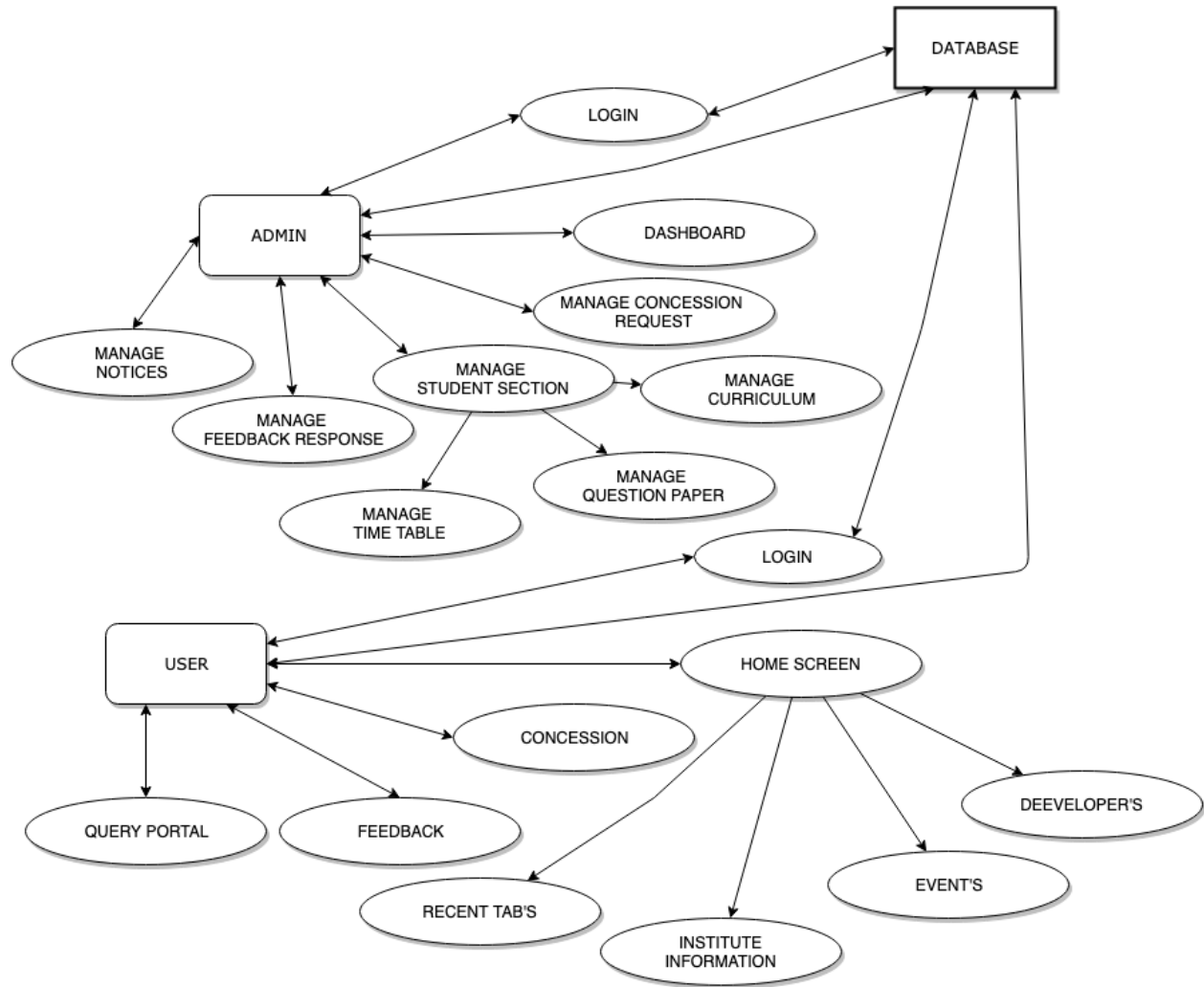
4.3.1 Level 0



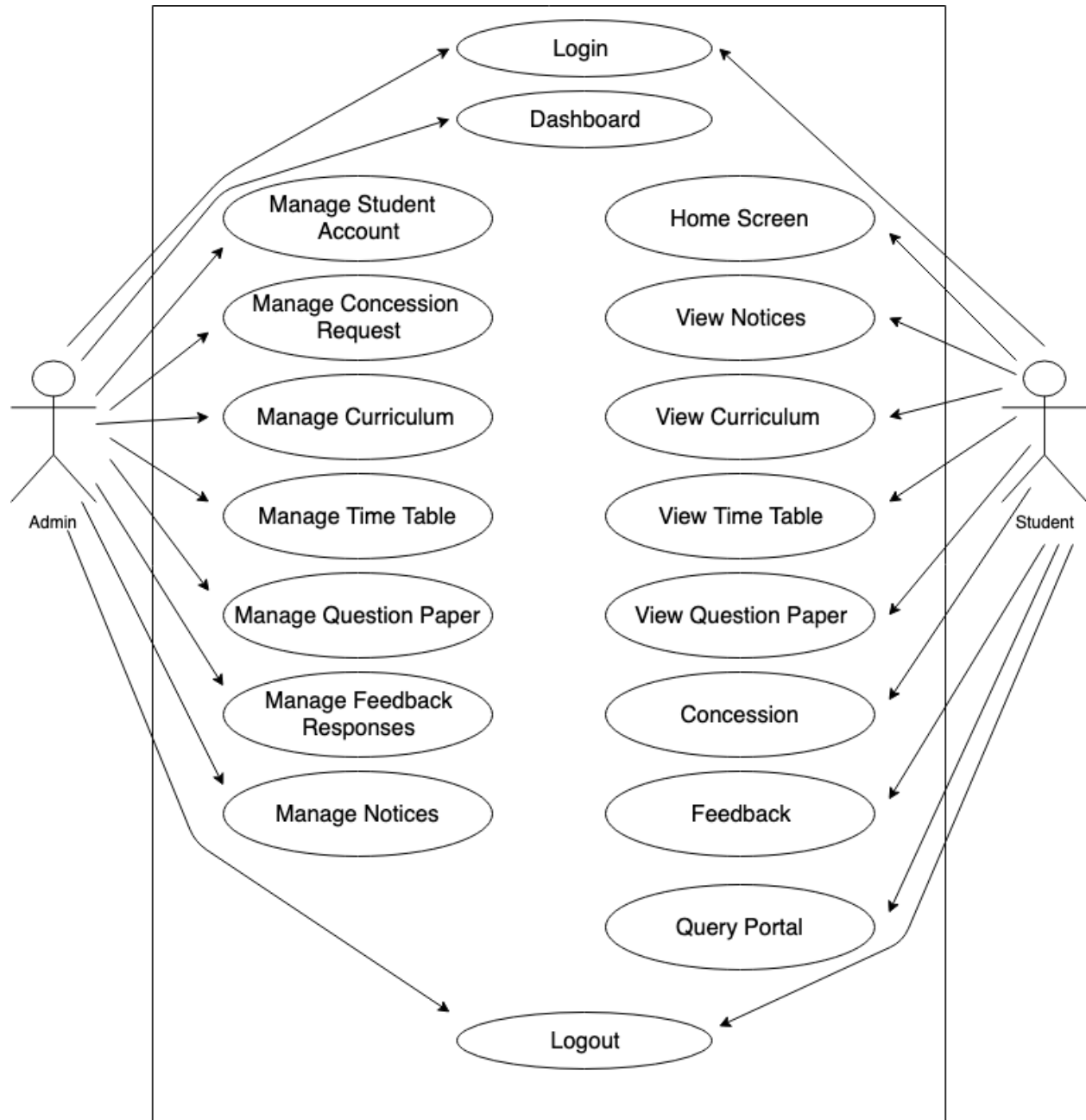
4.3.2 Level 1 : -



4.3.3 Level 2 :-



4.3.4 Level 3 :-



CHAPTER 5

CODING

Chapter 5: Coding

5.1 User-APP

5.1.1 Home Screen

```
import React, { Component } from 'react';
import {
  TouchableOpacity, Image, SafeAreaView, ScrollView,
  StyleSheet,
  View,
  // Text,
  Alert,
  TextInput,
  Platform,
  StatusBar,
  // ScrollView,
  // Image,
  Dimensions,
  Animated,
} from 'react-native';
import { Overlay } from 'react-native-elements'
import Gallery from 'react-native-image-gallery';

import Curriculum from '../student section/options/curr'
import AcadCal from '../student section/options/acadCal'
import ClassTT from '../student section/options/classTT'
import QuestPaper from '../student section/options/questPaper'
import Result from '../student section/options/result'
import PlacSer from '../centres/options/placSer'

import { Block, Card, Text, Icon, Label } from
'../components';
import * as theme from '../constants/theme'
import Icon1 from 'react-native-vector-icons/Ionicons'
import Category from './components/Explore/Category'
import HomeNew from './components/Explore/Home'
import Tag from './components/Explore/Tag'
const { height, width } = Dimensions.get('window')
const styles = StyleSheet.create({
  overview: {
    flex: 1,
    flexDirection: 'column',
    backgroundColor: theme.colors.white,
    marginBottom: 31,
    paddingBottom: 31
  }
})
```

```

    },
    margin: {
        marginHorizontal: 25,
    },
    driver: {
        marginBottom: 11,
    },
    avatar: {
        width: 48,
        height: 48,
        borderRadius: 24,
    }
});

class Home extends Component {
    static navigationOptions = {

        headerTitle: (
            <Block row middle><Text h4>Home</Text></Block>
        ),

    }
    constructor(props) {
        super(props)

        this.state = {
            isVissibleGallery: false,
            isVissiblresult: false,
            isVissibleClassTT: false,
            isVissibleCalender: false,
            isVissiblePlace: false,
            isVissibleQuestPaper: false,
            isVisibleGalleryTechNex: false,
            isVisibleGalleryFDBIT: false,
            isVisibleGalleryTechkriti: false,
            isVisibleGalleryKalpak: false

        }

    }

    componentWillMount() {

        this.scrollY = new Animated.Value(0)

        this.startHeaderHeight = 80
        this.endHeaderHeight = 50
        if (Platform.OS == 'android') {

```



```

        this.startHeaderHeight = 100 +
StatusBar.currentHeight
        this.endHeaderHeight = 70 + StatusBar.currentHeight
    }

    this.animatedHeaderHeight = this.scrollY.interpolate({
        inputRange: [0, 50],
        outputRange: [this.startHeaderHeight,
this.endHeaderHeight],
        extrapolate: 'clamp'
    })

    this.animatedOpacity =
this.animatedHeaderHeight.interpolate({
        inputRange: [this.endHeaderHeight,
this.startHeaderHeight],
        outputRange: [0, 1],
        extrapolate: 'clamp'
    })

    this.animatedTagTop =
this.animatedHeaderHeight.interpolate({
        inputRange: [this.endHeaderHeight,
this.startHeaderHeight],
        outputRange: [-30, 10],
        extrapolate: 'clamp'
    })

    this.animatedMarginTop =
this.animatedHeaderHeight.interpolate({
        inputRange: [this.endHeaderHeight,
this.startHeaderHeight],
        outputRange: [50, 30],
        extrapolate: 'clamp'
    })

}
render() {
    return (
        <SafeAreaView style={styles.overview}>
            <View style={{ flex: 1 }}>

                <ScrollView
                    scrollEventThrottle={16}
                    onScroll={Animated.event(
                        [
                            { nativeEvent: { contentOffset:
{ y: this.scrollY } } }
                        ]
                    )}
            </View>
        </SafeAreaView>
    )
}

```

```

      >
      <View style={{ flex: 1, backgroundColor:
'white', paddingTop: 20 }}>
        <Text style={{ fontSize: 24,
fontWeight: '700', paddingHorizontal: 20 }}>
          Recent Tab's
        </Text>

        <View style={{ height: 130,
marginTop: 20 }}>
          <ScrollView
            horizontal={true}
showsHorizontalScrollIndicator={false}
          >
            <Overlay
isVisible={this.state.isVissibleGallery}
              onBackdropPress={() =>
this.setState({ isVissibleGallery: false })}
windowBackgroundColor="rgba(255, 255, 255, .5)"
              //
overlayBackgroundColor="red"
              width={width * 0.85}
              height={height * 0.85}
            >
              <Curriclum />
            </Overlay>
          </Overlay>

isVisible={this.state.isVissibleCalender}
              onBackdropPress={() =>
this.setState({ isVissibleCalender: false })}
windowBackgroundColor="rgba(255, 255, 255, .5)"
              //
overlayBackgroundColor="red"
              width={width * 0.85}
              height={height * 0.85}
            >
              <AcadCal />
            </Overlay>
          </Overlay>

isVisible={this.state.isVissibleClassTT}
              onBackdropPress={() =>
this.setState({ isVissibleClassTT: false })}

```

```

windowBackgroundColor="rgba(255, 255, 255, .5)"
//
overlayBackgroundColor="red"
width={width * 0.85}
height={height * 0.85}
>
  <ClassTT />
</Overlay>
<Overlay

isVisible={this.state.isVissiblePlace}
onBackdropPress={() =>
this.setState({ isVissiblePlace: false })}

windowBackgroundColor="rgba(255, 255, 255, .5)"
//
overlayBackgroundColor="red"
width={width * 0.85}
height={height * 0.85}
>
  <PlacSer />
</Overlay>
<Overlay

isVisible={this.state.isVissibleQuestPaper}
onBackdropPress={() =>
this.setState({ isVissibleQuestPaper: false })}

windowBackgroundColor="rgba(255, 255, 255, .5)"
//
overlayBackgroundColor="red"
width={width * 0.85}
height={height * 0.85}
>
  <QuestPaper />
</Overlay>
<Overlay

isVisible={this.state.isVissiblresult}
onBackdropPress={() =>
this.setState({ isVissiblresult: false })}

windowBackgroundColor="rgba(255, 255, 255, .5)"
//
overlayBackgroundColor="red"
width={width * 0.85}
height={height * 0.85}

```

```

        >
        <Result />
    </Overlay>
    <TouchableOpacity
        onPress={() =>
this.setState({ isVisibleGallery: true })}
        >
        <Category imageUri={{
uri: 'http://192.168.43.64/GPM/images/curriculum.jpg' }}
            name="Curriculum"
        />
    </TouchableOpacity>
    <TouchableOpacity
        onPress={() =>
this.setState({ isVisibleCalender: true })}
        >
        <Category imageUri={{
uri: 'http://192.168.43.64/GPM/images/academic.jpg' }}
            name="Academic
Calendar"
        />
    </TouchableOpacity>
    <TouchableOpacity
        onPress={() =>
this.setState({ isVisibleresult: true })}
        >
        <Category imageUri={{
uri: 'http://192.168.43.64/GPM/images/result.jpg' }}
            name="Result"
        />
    </TouchableOpacity>
    <TouchableOpacity
        onPress={() =>
this.setState({ isVisibleClassTT: true })}
        >
        <Category imageUri={{
uri: 'http://192.168.43.64/GPM/images/class timetable.jpg' }}
            name="class time
table "
        />
    </TouchableOpacity>
    <TouchableOpacity
        onPress={() =>
this.setState({ isVisibleQuestPaper: true })}
        >
        <Category imageUri={{
uri: 'http://192.168.43.64/GPM/images/questionpaper.jpg' }}

```

```

name="Question Paper
"
        />
        </TouchableOpacity>
        <TouchableOpacity
            onPress={() =>
this.setState({ isVisiblePlace: true })}
        >
            <Category imageUri={{
uri: 'http://192.168.43.64/GPM/images/placement_services.jpg' }}
            name="placement
services "
            />
            </TouchableOpacity>
        </ScrollView>
    </View>
    <View style={{ marginTop: 40,
paddingHorizontal: 20 }}>
        <Text style={{ fontSize: 24,
fontWeight: '700' }}>
            Government Polytechnic
Mumbai Autonomous Institute
        </Text>
        <View style={{ width: width -
40, height: 200, marginTop: 20 }}>
            <Image
                style={{ flex: 1,
height: null, width: null, resizeMode: 'cover', borderRadius: 5,
borderWidth: 1, borderColor: '#dddddd' }}
                source={{ uri:
'http://192.168.43.64/GPM/images/img.jpg' }}
            />
        </View>
    </View>
    </View>
    </View>
    <View style={{ paddingVertical: 25,
marginBottom: 40 }}>
        <Card row middle
style={styles.margin}>
            <Block>
                <Text style={{ fontWeight:
'bold', fontSize: 24 }} paragraph color="black3">
                    Institute Information
                {'\n'}

```

</Text>

<Text paragraph

color="black3">

Establishment of Polytechnic at Elphinston Technical School, Dhobi Talav, Mumbai with 60 intake in Civil Engineering on 15th June, 1960. The Polytechnic acquired existing campus in May, 1985.{'\n'}

</Text>

<Text paragraph

color="black3">

Introduction of new diploma courses under State Government plan schemes viz. Electronics engineering and Instrumentation in 1988, Mechanical in 1989.{'\n'}

</Text>

<Text paragraph

color="black3">

Implementation of World Bank assisted projects in 1990–98. Major components were capacity expansion, quality and efficiency improvement. Under this started a new course, Computer Engineering in 1992, which was awarded academic autonomy in 1990 to design, develop, implement own curriculum and issue own diploma.{'\n'}

</Text>

<Text paragraph

color="black3">

Implementation of Canada India Industry Polytechnic Linkage Project (CIILP–2002 to 2005). The project focus was training of faculty and staff.{'\n'}

</Text>

<Text paragraph

color="black3">

The Polytechnic has been awarded Narsee Monjee Award for best performance in the year 1999 by ISTE.{'\n'}

</Text>

```

        </Block>
      </Card>
    </View>
    <View style={{ marginTop: 40,
marginBottom: 40 }}>
      <Text style={{ fontSize: 24,
fontWeight: '700', paddingHorizontal: 20 }}>
        Events
      </Text>
      <View style={{ paddingHorizontal:
20, paddingBottom: 20, marginTop: 20, marginBottom: 20,
flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-
between' }}>
        <Overlay
isVisible={this.state.isVisibleGalleryTechNex}
onBackdropPress={() =>
this.setState({ isVisibleGalleryTechNex: false })}
windowBackgroundColor="rgba(255, 255, 255, .5)"
//
overlayBackgroundColor="red"
width={width * 0.85}
height={height * 0.85}
>
      { /* <GalleryClick /> */}
      <Gallery
        style={{ flex: 1,
backgroundColor: 'black' }}
        images={[
          // { source:
require('yourApp/image.png'), dimensions: { width: 150, height:
150 } },
          { source: { uri:
'http://192.168.43.64/GPM/images/technex/1.JPG' } },
          { source: { uri:
'http://192.168.43.64/GPM/images/technex/2.JPG' } },
          { source: { uri:
'http://192.168.43.64/GPM/images/technex/3.JPG' } },
          { source: { uri:
'http://192.168.43.64/GPM/images/technex/4.JPG' } },
          { source: { uri:
'http://192.168.43.64/GPM/images/technex/5.JPG' } },
          { source: { uri:
'http://192.168.43.64/GPM/images/technex/7.JPG' } },
          { source: { uri:
'http://192.168.43.64/GPM/images/technex/6.JPG' } },

```

```

                                { source: { uri:
'http://192.168.43.64/GPM/images/technex/8.JPG' } }},
                                { source: { uri:
'http://192.168.43.64/GPM/images/technex/9.JPG' } }},

                                }}
                                />
                                </Overlay>
                                <Overlay

isVisible={this.state.isVisibleGalleryKalpak}
                                onBackdropPress={() =>
this.setState({ isVisibleGalleryKalpak: false })}

windowBackgroundColor="rgba(255, 255, 255, .5)"
                                //
overlayBackgroundColor="red"

                                width={width * 0.85}
                                height={height * 0.85}
                                >
                                { /* <GalleryClick /> */}
                                <Gallery
                                style={{ flex: 1,

backgroundColor: 'black' }}

                                images={[
                                // { source:
require('yourApp/image.png'), dimensions: { width: 150, height:
150 } },

                                { source: { uri:
'http://192.168.43.64/GPM/images/kalpak/1.JPG' } }},
                                { source: { uri:
'http://192.168.43.64/GPM/images/kalpak/2.JPG' } }},
                                { source: { uri:
'http://192.168.43.64/GPM/images/kalpak/3.JPG' } }},
                                { source: { uri:
'http://192.168.43.64/GPM/images/kalpak/4.JPG' } }},
                                { source: { uri:
'http://192.168.43.64/GPM/images/kalpak/5.JPG' } }},
                                { source: { uri:
'http://192.168.43.64/GPM/images/kalpak/7.JPG' } }},
                                { source: { uri:
'http://192.168.43.64/GPM/images/kalpak/6.JPG' } }},
                                { source: { uri:
'http://192.168.43.64/GPM/images/kalpak/8.JPG' } }},

                                }}
                                />

```



```

        </Overlay>
        <TouchableOpacity
          onPress={() =>
this.setState({ isVisibleGalleryTechNex: true })}

        >
          <HomeNew width={width}
            name="TechNex 2019"
            rating={5}

image='http://192.168.43.64/GPM/images/photo.jpg'
          />
        </TouchableOpacity>
        <TouchableOpacity
          onPress={() =>
this.setState({ isVisibleGalleryKalpak: true })}

        >
          <HomeNew width={width}
            name="Kalpak 2019"
            rating={4}

image='http://192.168.43.64/GPM/images/kalpak.JPG'
          />
        </TouchableOpacity>
      </View>
      <View style={{ paddingHorizontal:
20, paddingBottom: 20, marginTop: 20, marginBottom: 20,
flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-
between' }}>
        <Overlay

isVisible={this.state.isVisibleGalleryFDBIT}
          onBackdropPress={() =>
this.setState({ isVisibleGalleryFDBIT: false })}

windowBackgroundColor="rgba(255, 255, 255, .5)"
          //
overlayBackgroundColor="red"

          width={width * 0.85}
          height={height * 0.85}
        >
          {/* <GalleryClick /> */}
          <Gallery
            style={{ flex: 1,
backgroundColor: 'black' }}

            images={[

```

```

// { source:
require('yourApp/image.png'), dimensions: { width: 150, height:
150 } },
        { source: { uri:
'http://192.168.43.64/GPM/images/fdpit/1.jpeg' } },
        { source: { uri:
'http://192.168.43.64/GPM/images/fdpit/2.jpeg' } },
        { source: { uri:
'http://192.168.43.64/GPM/images/fdpit/3.jpeg' } },
        { source: { uri:
'http://192.168.43.64/GPM/images/fdpit/4.jpeg' } },

    ]}
  />
</Overlay>
<TouchableOpacity
  onPress={() =>
this.setState({ isVisibleGalleryFDBIT: true })}

  >

    <HomeNew width={width}
      name="Big Data & Hadoop
FDP"
      rating={5}
      image='http://192.168.43.64/GPM/images/fdp.jpg'

    />
  </TouchableOpacity>

  <HomeNew width={width}
    name="Konstruct 2019"
    rating={3}
    image='http://192.168.43.64/GPM/images/konstruct6.jpg'

  />
</View>
<View style={{ paddingHorizontal:
20, paddingBottom: 20, marginTop: 20, marginBottom: 20,
flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-
between' }}>

  <Overlay

    isVisible={this.state.isVisibleGalleryTechkriti}
    onBackdropPress={() =>
this.setState({ isVisibleGalleryTechkriti: false })}

```

```

windowBackgroundColor="rgba(255, 255, 255, .5)"
//
overlayBackgroundColor="red"
width={width * 0.85}
height={height * 0.85}
>
  { /* <GalleryClick /> */ }
  <Gallery
    style={{ flex: 1,
backgroundColor: 'black' }}
    images={[
      // { source:
require('yourApp/image.png'), dimensions: { width: 150, height:
150 } },
      { source: { uri:
'http://192.168.43.64/GPM/images/techkriti/1.JPG' } },
      { source: { uri:
'http://192.168.43.64/GPM/images/techkriti/2.JPG' } },
      { source: { uri:
'http://192.168.43.64/GPM/images/techkriti/3.JPG' } },
      { source: { uri:
'http://192.168.43.64/GPM/images/techkriti/4.JPG' } },
      { source: { uri:
'http://192.168.43.64/GPM/images/techkriti/5.JPG' } },

    ]}
  />
</Overlay>
<HomeNew width={width}
  name="Mechnova 2019"
  rating={3}

image='http://192.168.43.64/GPM/images/mech.jpg'

/>
<TouchableOpacity
  onPress={() =>
this.setState({ isVisibleGalleryTechkriti: true })}

>

  <HomeNew width={width}
    name="TechKriti 2018"
    rating={4}

image='http://192.168.43.64/GPM/images/techkriti.JPG'

```

```

        />
        </TouchableOpacity>
    </View>
    <View style={{ paddingHorizontal:
20, paddingBottom: 20, marginTop: 20, marginBottom: 20,
flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-
between' }}>

        <HomeNew width={width}
            name="TechKnow 2019"
            rating={4}

image='http://192.168.43.64/GPM/images/techknow.jpeg'
        />

    </View>
</View>
<View style={{ paddingVertical: 25,
marginBottom: 40 }}>

    <Card
        title="Team Memebers"
        style={[styles.margin, {
marginTop: 18 }]}
    >
        <Block style={styles.driver}>
            <TouchableOpacity
activeOpacity={0.8}
                onPress={() =>
this.props.navigation.navigate('St')}
            >
                <Block row center>
                    <Block>
                        <Image
style={styles.avatar}
                            source={{
uri: 'http://192.168.43.64/GPM/images/team.png' }}
                        />
                    </Block>
                    <Block flex={2}>
                        <Text h4>AFAAN
ANSARI</Text>
                        { /* <Text
paragraph color="gray">Chevrolet Bolt</Text> */
                    </Block>
                    <Block>

```

```

                                </Block>
                                </Block>
                                </TouchableOpacity>
                                </Block>
                                <Block style={styles.driver}>
                                  <TouchableOpacity
activeOpacity={0.8}>
                                <Block row center>
                                  <Block>
                                    <Image
style={styles.avatar}
                                source={{
uri: 'http://192.168.43.64/GPM/images/team.png' }}
                                  />
                                </Block>
                                <Block flex={2}>
                                  <Text
h4>ABHISHEK RAI</Text>
                                { /* <Text
paragraph color="gray">Tesla Model X</Text> */
                                </Block>
                                <Block>
                                  </Block>
                                  </Block>
                                  </TouchableOpacity>
                                </Block>
                                <Block style={styles.driver}>
                                  <TouchableOpacity
activeOpacity={0.8}>
                                <Block row center>
                                  <Block>
                                    <Image
style={styles.avatar}
                                source={{
uri: 'http://192.168.43.64/GPM/images/team.png' }}
                                  />
                                </Block>
                                <Block flex={2}>
                                  <Text h4>NIKHIL
CHAUBE</Text>
                                { /* <Text
paragraph color="gray">Volvo Intellisafe</Text> */
                                </Block>
                                <Block>
                                  </Block>

```

```

        </Block>
        </TouchableOpacity>
    </Block>
    <Block style={styles.driver}>
        <TouchableOpacity
activeOpacity={0.8}>

            <Block row center>
                <Block>
                    <Image

style={styles.avatar}
                        source={{
uri: 'http://192.168.43.64/GPM/images/team.png' }}
                        />
                    </Block>
                    <Block flex={2}>
                        <Text h4>SHAHAB
MOMIN</Text>

                        { /* <Text
paragraph color="gray">Volvo Intellisafe</Text> */ }
                    </Block>
                    <Block>

                        </Block>
                    </Block>
                </TouchableOpacity>
            </Block>
    <Block style={styles.driver}>
        <TouchableOpacity
activeOpacity={0.8}>

            <Block row center>
                <Block>
                    <Image

style={styles.avatar}
                        source={{
uri: 'http://192.168.43.64/GPM/images/team.png' }}
                        />
                    </Block>
                    <Block flex={2}>
                        <Text h4>TAUSIF
KHAN</Text>

                        { /* <Text
paragraph color="gray">Volvo Intellisafe</Text> */ }
                    </Block>
                    <Block>

                        </Block>
                </Block>
    </Block>

```

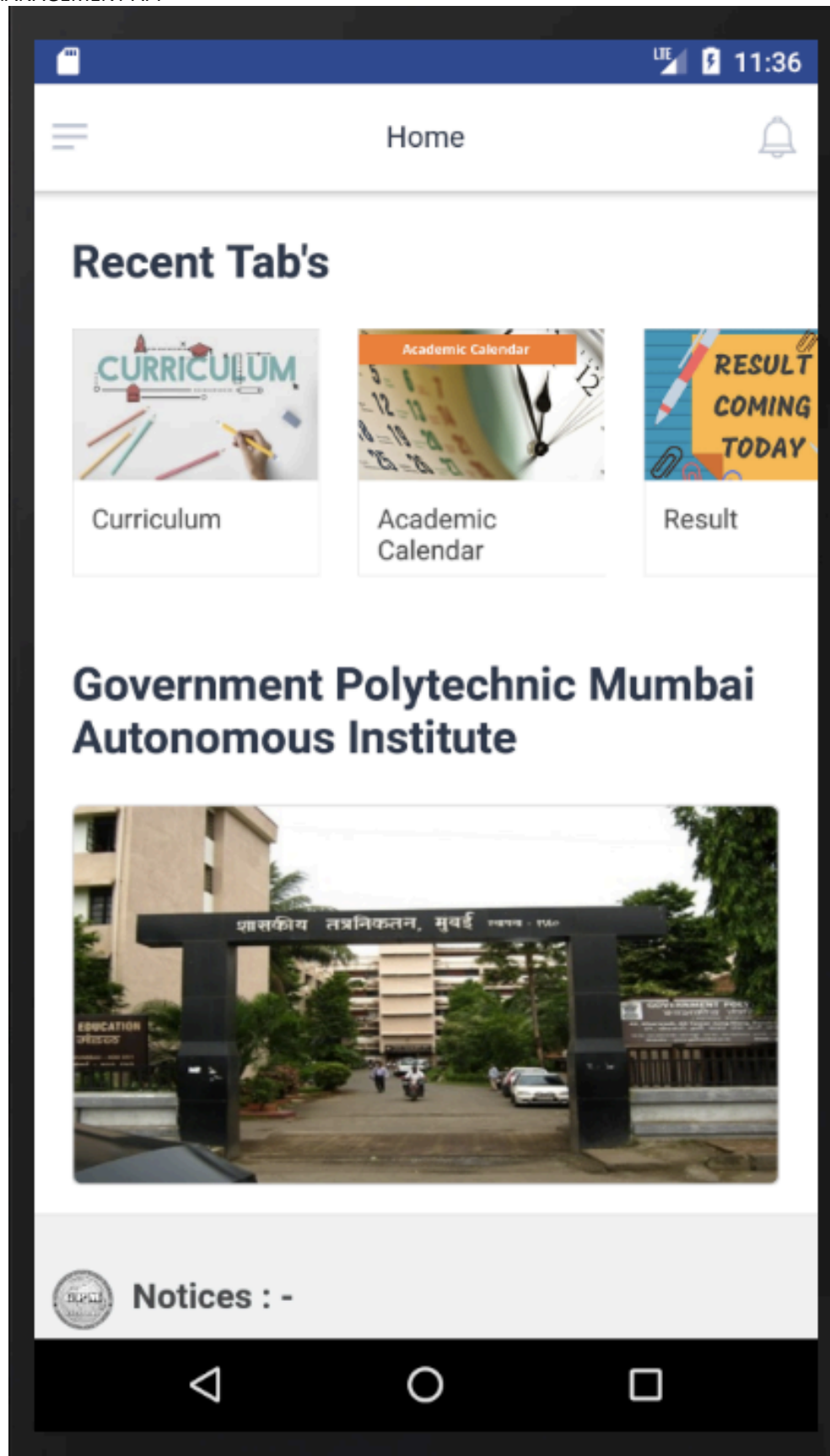
```
                                </Block>
                                </TouchableOpacity>
                                </Block>
                                </Card>

                                </View>
                                </ScrollView>

                                </View>

                                </SafeAreaView>
                                )
                                }
                                }
                                }

export default Home;
```



5.1.2 About US

```
import { StyleSheet, Text, View, TouchableOpacity,
Alert,Dimensions } from 'react-native';
import { Block, Card, Icon, Label } from '../components';
import { Overlay, Button } from 'react-native-elements'
// import Button from '../components/animated button/button'
let width = Dimensions.get('window').width;
let height = Dimensions.get('window').height;
import PrincipalDesk from './options/principleDesk'
import AboutInst from './options/aboutInst'
import MissionVission from './options/missionVission'
import OrganStruc from './options/organStruc'
import GovernBod from './options/governBod'
import Rti from './options/rti'
import CitiChar from './options/citiChar'
import ManDis from './options/manDis'
import Infra from './options/infra'

export default class AboutUS extends React.Component {
  static navigationOptions = {

    headerTitle: (
      <Block row middle><Text h4>About US</Text></Block>
    )
  }
  constructor(props) {
    super(props)
    this.state = {
      isVisiblePrincipalDesk: false,
      isVisibleAboutInst: false,
      isVisibleMissionVission: false,
      isVisibleOrganStruc: false,
      isVisibleGovernBod: false,
      isVisibleRti: false,
      isVisibleCitiChar: false,
      isVisibleManDis: false,
      isVisibleInfra: false,

    }
  }

  render() {
    const { style, full, opacity, children, ...props } =
this.props;
```

```

return (
  <View>
    <Overlay
      isVisible={this.state.isVisiblePrincipalDesk}
      onBackdropPress={() => this.setState({
isVisiblePrincipalDesk: false })}
      windowBackgroundColor="rgba(255, 255, 255, .5)"
      // overlayBackgroundColor="red"
      width={width * 0.85}
      height={height * 0.85}
      fullScreen={true}
    >
      <PrincipalDesk />
    </Overlay>
    {/* */}
    <Overlay
      isVisible={this.state.isVisibleAboutInst}
      onBackdropPress={() => this.setState({
isVisibleAboutInst: false })}
      windowBackgroundColor="rgba(255, 255, 255, .5)"
      // overlayBackgroundColor="red"
      width={width * 0.85}
      height={height * 0.85}
      fullScreen={true}

    >
      <AboutInst />
    </Overlay>
    {/* */}
    <Overlay
      isVisible={this.state.isVisibleMissionVission}
      onBackdropPress={() => this.setState({
isVisibleMissionVission: false })}
      windowBackgroundColor="rgba(255, 255, 255, .5)"
      // overlayBackgroundColor="red"
      width={width * 0.85}
      height={height * 0.85}
      fullScreen={true}

    >
      <MissionVission />
    </Overlay>
    {/* */}
    <Overlay
      isVisible={this.state.isVisibleOrganStruc}
      onBackdropPress={() => this.setState({
isVisibleOrganStruc: false })}

```

```

        windowBackgroundColor="rgba(255, 255, 255, .5)"
        // overlayBackgroundColor="red"
        width={width * 0.85}
        height={height * 0.85}
        fullScreen={true}

    >
      <OrganStruc />
    </Overlay>
    { /* */ }
    <Overlay
      isVisible={this.state.isVisibleGovernBod}
      onBackdropPress={() => this.setState({
isVisibleGovernBod: false })}
      windowBackgroundColor="rgba(255, 255, 255, .5)"
      // overlayBackgroundColor="red"
      width={width * 0.85}
      height={height * 0.85}
      fullScreen={true}

    >
      <GovernBod />
    </Overlay>
    { /* */ }
    <Overlay
      isVisible={this.state.isVisibleRti}
      onBackdropPress={() => this.setState({ isVisibleRti:
false })}
      windowBackgroundColor="rgba(255, 255, 255, .5)"
      // overlayBackgroundColor="red"
      width={width * 0.85}
      height={height * 0.85}
      fullScreen={true}

    >
      <Rti />
    </Overlay>
    { /* */ }
    <Overlay
      isVisible={this.state.isVisibleCitiChar}
      onBackdropPress={() => this.setState({
isVisibleCitiChar: false })}
      windowBackgroundColor="rgba(255, 255, 255, .5)"
      // overlayBackgroundColor="red"
      width={width * 0.85}
      height={height * 0.85}
      fullScreen={true}

```

```

    >
      <CitiChar />
    </Overlay>
    {/* */}
    <Overlay
      isVisible={this.state.isVisibleManDis}
      onBackdropPress={() => this.setState({
isVisibleManDis: false })}
      windowBackgroundColor="rgba(255, 255, 255, .5)"
      // overlayBackgroundColor="red"
      width={width * 0.85}
      height={height * 0.85}
      fullScreen={true}

    >
      <ManDis />
    </Overlay>
    {/* */}
    <Overlay
      isVisible={this.state.isVisibleInfra}
      onBackdropPress={() => this.setState({ isVisibleInfra:
false })}
      windowBackgroundColor="rgba(255, 255, 255, .5)"
      // overlayBackgroundColor="red"
      width={width * 0.85}
      height={height * 0.85}
      fullScreen={true}

    >
      <Infra />
    </Overlay>
    {/* */}

    <View>
    <View style={{ flexDirection: 'row' }}
    >
      <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
        <Button
          title="Principal desk"
          type="outline"
          onPress={() => this.setState({
isVisiblePrincipalDesk: true })}
        />
      </View>
      <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
        <Button

```

```

        title="About Institute"
        type="outline"
        onPress={() => this.setState({ isVisibleAboutInst:
true })}}
      />
    </View>

    </View>

    <View style={{ flexDirection: 'row' }}
    >
      <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
        <Button
          title="Mission and Vision"
          type="outline"
          onPress={() => this.setState({
isVisibleMissionVission: true })}}
        />
      </View>
      <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
        <Button
          title="Organizing Structure"
          type="outline"
          onPress={() => this.setState({
isVisibleOrganStruc: true })}}
        />
      </View>

    </View>
    <View style={{ flexDirection: 'row' }}
    >
      <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
        <Button
          title="Governing Body"
          type="outline"
          onPress={() => this.setState({ isVisibleGovernBod:
true })}}
        />
      </View>
      <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
        <Button
          title="RTI Act"

```

```

        type="outline"
        onPress={() => this.setState({ isVisibleRti: true
    }}}
        />
    </View>

    </View>
    <View style={{ flexDirection: 'row' }}
    >
        <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
            <Button
                title="Citizen Charter"
                type="outline"
                onPress={() => this.setState({ isVisibleCitiChar:
true }}}
            />
        </View>
        <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
            <Button
                title="Mandatory Disclosure"
                type="outline"
                onPress={() => this.setState({ isVisibleManDis:
true }}}
            />
        </View>

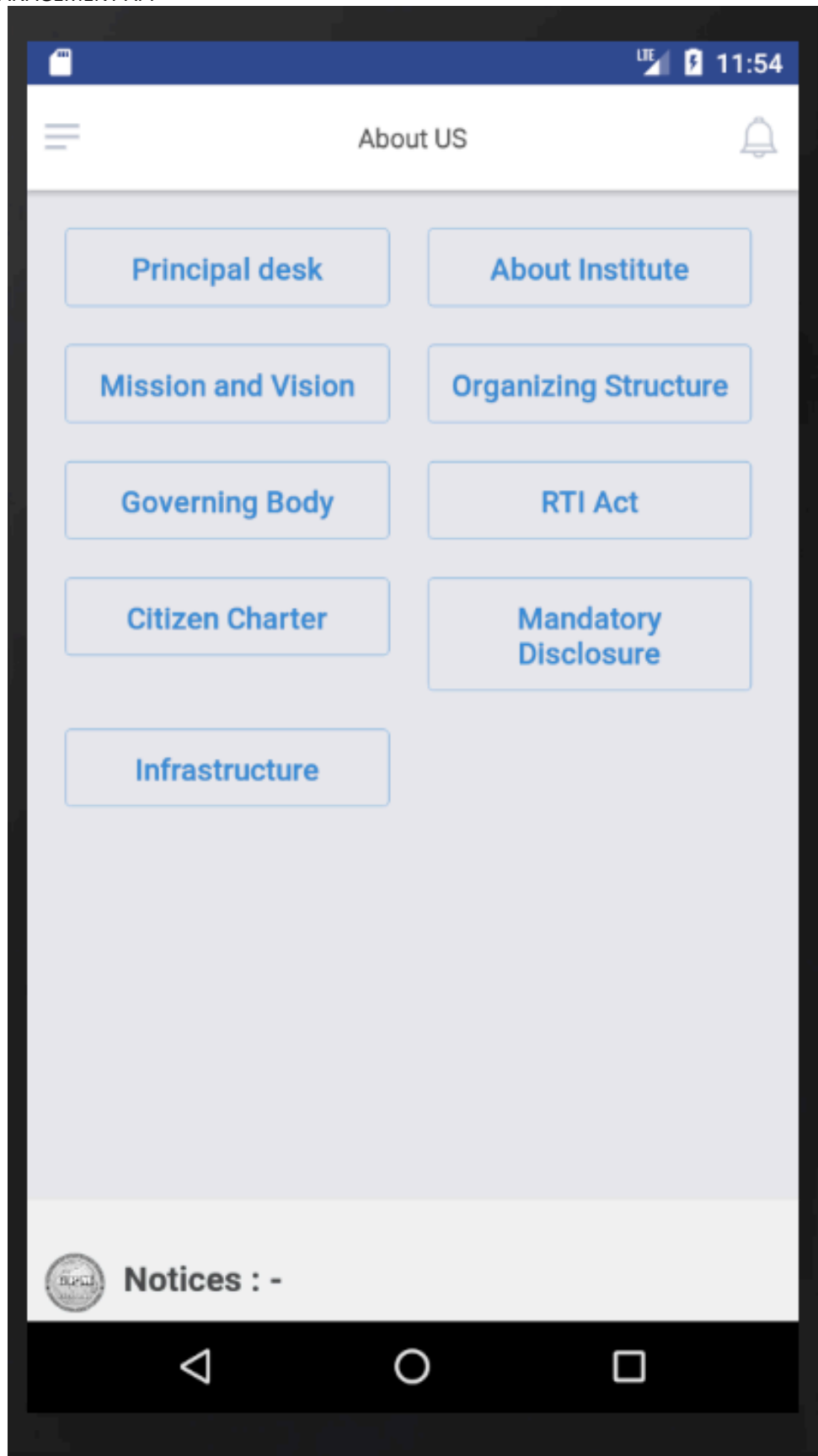
    </View>
    <View style={{ flexDirection: 'row' }}
    >
        <View style={{ width: width * 0.47, paddingLeft: width
* 0.05, paddingTop: width * 0.05 }}>
            <Button
                title="Infrastructure"
                type="outline"
                onPress={() => this.setState({ isVisibleInfra:
true }}}
            />
        </View>

    </View>
    </View>

    </View>
    );
}

```

```
}  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    backgroundColor: '#fff',  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
});
```



5.1.3 Index file (main screen module combined file)

```
import React, { Component } from "react";
import {
  View,
  Text,
  StyleSheet,
  Dimensions,
  Animated,
  PanResponder,
  ScrollView,
  Image,
  Slider
} from "react-native";
import IconFont from 'react-native-vector-icons/FontAwesome'
import Routes from '../routes/routes'
import Ln from '../notices/ln'
import Sc from '../notices/sc'
import Cep from '../notices/cep'
import Tender from '../notices/tender'

const SCREEN_HEIGHT = Dimensions.get('window').height
const SCREEN_WIDTH = Dimensions.get('window').width
import { createDrawerNavigator, createAppContainer,
createMaterialTopTabNavigator, createStackNavigator,
createSwitchNavigator } from 'react-navigation';
import Icon from 'react-native-vector-icons/MaterialIcons'
const Notice = createMaterialTopTabNavigator({
  Ln: {
    screen: Ln,
    navigationOptions: ({ navigation }) => ({
      title: 'Latest News',
      tabBarIcon: ({ tintColor }) => (
        <IconFont name='gitlab' color={tintColor} size={24} />
      )
    }),
  },
  Sc: {
    screen: Sc,
    navigationOptions: ({ navigation }) => ({
      title: 'Student Corner',
      tabBarIcon: ({ tintColor }) => (
        <IconFont name='gitlab' color={tintColor} size={24} />
      )
    }),
  },
},
```

```

Cep: {
  screen: Cep,
  navigationOptions: ({ navigation }) => ({
    title: 'CEP News',
    tabBarIcon: ({ tint_color }) => (
      <IconFont name='gitlab' color={tint_color} size={24} />
    )
  }),
},
Tender: {
  screen: Tender,
  navigationOptions: ({ navigation }) => ({
    title: 'Tender News',
    tabBarIcon: ({ tint_color }) => (
      <IconFont name='gitlab' color={tint_color} size={24} />
    )
  }),
},
},
{
  tabBarOptions: {
    labelStyle: {
      fontSize: 12,
      fontWeight : 'bold'
    },
    tabStyle: {
      paddingBottom: SCREEN_HEIGHT * 0.2
    },
    style: {
      backgroundColor: '#F0F3F4',
    },
    activeTintColor: '#E74C3C',
    showIcon: true,
    tabBarPosition: 'bottom'
  },
  tabBarPosition: 'bottom',
  lazy: true
}
)
export default class Footer extends Component {
  state = {
    isScrollEnabled: false
  }
  componentWillMount() {

```

```

this.scrollTopOffset = 0

this.animation = new Animated.ValueXY({ x: 0, y:
SCREEN_HEIGHT - 90 })

this.panResponder = PanResponder.create({

  onMoveShouldSetPanResponder: (evt, gestureState) => {

    if ((this.state.isScrollEnabled && this.scrollTopOffset <=
0 && gestureState.dy > 0) || !this.state.isScrollEnabled &&
gestureState.dy < 0) {
      return true
    } else {
      return false
    }
  },
  onPanResponderGrant: (evt, gestureState) => {
    this.animation.extractOffset()
  },
  onPanResponderMove: (evt, gestureState) => {

    this.animation.setValue({ x: 0, y: gestureState.dy })
  },
  onPanResponderRelease: (evt, gestureState) => {

    if (gestureState.moveY > SCREEN_HEIGHT - 120) {
      Animated.spring(this.animation.y, {
        toValue: 0,
        tension: 1
      }).start()
    }
    else if (gestureState.moveY < 120) {
      Animated.spring(this.animation.y, {
        toValue: 0,
        tension: 1
      }).start()
    }
    else if (gestureState.dy < 0) {
      this.setState({ isScrollEnabled: true })

      Animated.spring(this.animation.y, {
        toValue: -SCREEN_HEIGHT + 120,
        tension: 1
      }).start()
    }
    else if (gestureState.dy > 0) {
      this.setState({ isScrollEnabled: false })
    }
  }
})

```

```

        Animated.spring(this.animation.y, {
          toValue: SCREEN_HEIGHT - 120,
          tension: 1
        }).start()
      }
    }

  })
}

render() {

  const animatedHeight = {
    transform: this.animation.getTranslateTransform()
  }

  animatedImageHeight = this.animation.y.interpolate({
    inputRange: [0, SCREEN_HEIGHT - 90],
    outputRange: [200, 32],
    extrapolate: "clamp"
  })
  animatedSongTitleOpacity = this.animation.y.interpolate({
    inputRange: [0, SCREEN_HEIGHT - 500, SCREEN_HEIGHT - 90],
    outputRange: [0, 0, 1],
    extrapolate: "clamp"
  })
  animatedImageMarginLeft = this.animation.y.interpolate({
    inputRange: [0, SCREEN_HEIGHT - 90],
    outputRange: [SCREEN_WIDTH / 2 - 100, 10],
    extrapolate: "clamp"
  })
  animatedHeaderHeight = this.animation.y.interpolate({
    inputRange: [0, SCREEN_HEIGHT - 90],
    outputRange: [SCREEN_HEIGHT / 2, 90],
    extrapolate: "clamp"
  })
  animatedSongDetailsOpacity = this.animation.y.interpolate({
    inputRange: [0, SCREEN_HEIGHT - 500, SCREEN_HEIGHT - 90],
    outputRange: [1, 0, 0],
    extrapolate: "clamp"
  })
  animatedBackgroundColor = this.animation.y.interpolate({
    inputRange: [0, SCREEN_HEIGHT - 90],
    outputRange: ['rgba(0,0,0,0.5)', 'white'],
    extrapolate: "clamp"
  })
  return (

```

```

    <Animated.View style={{ flex: 1, backgroundColor:
animatedBackgroundColor }}>
      <Routes />
      <Animated.View
        {... this.panResponder.panHandlers}
        style={[animatedHeight, { position: 'absolute', left:
0, right: 0, zIndex: 10, backgroundColor: '#F0F3F4', height:
SCREEN_HEIGHT }]}
      >
        <ScrollView
          scrollEnabled={this.state.isScrollEnabled}
          scrollEventThrottle={16}
          onScroll={event => {
            this.scrollOffset =
event.nativeEvent.contentOffset.y
          }}
        >
          <Animated.View
            style={{ height: animatedHeaderHeight,
borderTopWidth: 1, borderTopColor: '#ebe5e5', flexDirection:
'row', alignItems: 'center' }}
          >
            <View style={{ flex: 4, flexDirection: 'row',
alignItems: 'center' }}>
              <Animated.View style={{ height:
animatedImageHeight, width: animatedImageHeight, marginLeft:
animatedImageMarginLeft }}>
                <Image style={{ flex: 1, width: null, height:
null, paddingBottom: 10 }}
                  source={require('./icon.png')}
                />
              </Animated.View>
              <Animated.Text style={{ opacity:
animatedSongTitleOpacity, fontSize: 18, fontWeight: 'bold',
paddingLeft: 10, paddingBottom: 5 }}>
                Notices : -
              </Animated.Text>
            </View>

          </Animated.View>

          <Animated.View style={{ height:
animatedHeaderHeight, opacity: animatedSongDetailsOpacity }}>

            <View

```

```
        style={{ flex: 2, flexDirection: 'row',
alignItems: 'center', justifyContent: 'center' }}
        >
        <Notice />

        </View>
        <View
        >

        </View>
        </Animated.View>

        </ScrollView>
        </Animated.View>

        </Animated.View>
    );
}
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center'
  }
});
```

5.1.4 Navigator file

```
import { createDrawerNavigator, createAppContainer,
createStackNavigator,
createSwitchNavigator } from 'react-navigation';
import React, { Component } from 'react'
import { Container, Header, Content } from 'native-base';
import { Text, Button, TouchableOpacity } from 'react-native'
import firebase from 'firebase'

import SideMenu from './sidemenu'

import home from '../menu/home/home'
import aboutUS from '../menu/about us/aboutUS'
import admission from '../menu/admission/admission'
import alumini from '../menu/alumini/alumini'
import centres from '../menu/centres/centres'
import contact from '../menu/contact/contact'
import department from '../menu/department/department'
import gallery from '../menu/gallery/gallery'
import studentSec from '../menu/student section/studentSec'

import { Block, Card, Icon, Label } from '../components';

import Login from '../login/login'
import Signup from '../login/signup'

import Feature from '../features/feature'
const homeNav = createStackNavigator({
  home : {
    screen : home
  },
  aboutUS : {
    screen : aboutUS
  },
  admission : {
    screen :admission
  },
  department : {
    screen : department
  },
  studentSec : {
    screen : studentSec
  },
  centres : {
    screen : centres
  },
},
```

```

    gallery : {
      screen : gallery
    },
    alumini : {
      screen : alumini
    },
    contact : {
      screen : contact
    }
  }, {
    // headerMode : 'none'
    navigationOptions: ({ navigation }) => ({
      headerLeftContainerStyle: {
        paddingLeft: 34,

      },
      headerRightContainerStyle: {
        paddingRight: 24
      },
      headerLeft: (
        <TouchableOpacity style={{marginLeft : 10}}
onPress={() => navigation.toggleDrawer()} ><Icon menu
/></TouchableOpacity>
      ),
      headerRight: (
        <TouchableOpacity style={{marginRight : 10}}
onPress={() => navigation.navigate('Login')} ><Icon notification
/></TouchableOpacity>
      ),
    })
  })

const Dr = createDrawerNavigator({
  home: {
    screen: homeNav
  },

}, {

  contentComponent: SideMenu,
  drawerWidth: 300
});

```



```

const St2 = createStackNavigator({
  Feature : {screen : Feature}

}, {
  navigationOptions: ({ navigation }) => ({
    headerLeftContainerStyle: {
      paddingLeft: 34,

    },
    headerRightContainerStyle: {
      paddingRight: 24
    },

    headerRight: (
      <TouchableOpacity style={{ marginRight: 10 }}
onPress={() => {
      firebase.auth().signOut().then(function () {
        // Sign-out successful.
        this.props.navigation.navigate('home')
        console.log('correct logout')

      }).catch(function (error) {
        // An error happened.
        console.log(error)
      })
      navigation.navigate('Dr')
    } }><Icon notification /></TouchableOpacity>
    ),
    headerTitle: (
      <Block row middle><Text
h4>Features</Text></Block>
    ),
  }),
})

const St = createStackNavigator({
  Login: { screen: Login
  },
  Signup: { screen: Signup },
  St2 : St2
}, {
  initialRouteName: 'Login',
  navigationOptions: ({ navigation }) => ({
    }),

```

```
        headerMode: 'none'
      })
const Sw = createSwitchNavigator({
  Dr: Dr,
  St: St,
})
export default Sw
1.
```

5.1.5 Login

```
import React, { Component } from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  View,
  Image,
  Dimensions,
  TextInput,
  Button,
  TouchableOpacity,
  ImageBackground,
  Alert,
  ToastAndroid
} from 'react-native';
import firebase from 'firebase'; // 4.8.1

const { width, height } = Dimensions.get("window");

const background = require("./login1_bg.png");
const mark = require("./login1_mark.png");
const lockIcon = require("./login1_lock.png");
const personIcon = require("./login1_person.png");
const backIcon = require("./back.png");
firebase.initializeApp({

  apiKey: "AIzaSyA20QpG8lowe8FYk7DF6PXRbiHACwRbiJU",
  authDomain: "chatapp-16f24.firebaseio.com",
  databaseURL: "https://chatapp-16f24.firebaseio.com",
  projectId: "chatapp-16f24",
  storageBucket: "chatapp-16f24.appspot.com",
  messagingSenderId: "559355718425"
});
let json = []
export default class Login extends Component {
  static navigationOptions = {
    headerMode: 'none'
  }
  constructor(props) {
    super(props)
    this.state = {
      email: '',
      password: '',
      data: [],
```

```

    }
  }

  login = async () => {
    const { email, password } = this.state
    try {

      await
      firebase.auth().signInWithEmailAndPassword(email,
      password).then(async function (user) {
        console.log(user)
        let response = await
        fetch("http://192.168.43.64/GPM/user_login.php", {
          method: 'POST',
          headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json',
          },
          body: JSON.stringify({

            email: email,

            password: password

          })
        })
        json = await response.json();
        // this.setState({ data: json });
        // console.log(this.state.data[0])
        console.log('response enroll', json[0].enroll)
        console.log('response enroll', json[0].name)
        console.log('response enroll', json[0].email)

        console.log('response enroll', json[0])

        await ToastAndroid.showWithGravityAndOffset(
          `welcome ${email}`,
          ToastAndroid.LONG,
          ToastAndroid.BOTTOM,
          25,
          50,
        );

      })
    }
  }

```

```

        await this.props.navigation.navigate('St2'
        , {
            Enroll: json[0].enroll,
            Name: json[0].name,
            Email: json[0].email,
            year: json[0].year,
            dept: json[0].dept,
            shift: json[0].shift,
            mobile: json[0].mobile,
            address: json[0].address,
            gender: json[0].gender,
            dob: json[0].dob,
        }
    );
}

catch (error) {
    console.log(error.toString())
    Alert.alert(error.toString())
}

}

render() {
    return (
        <View style={styles.container}>
            <ImageBackground source={background}
style={[[styles.container, styles.bg]]}
            resizeMode="cover">
                <View style={styles.headerContainer}>

                    <View style={styles.headerIconView}>
                        <TouchableOpacity
style={styles.headerBackButtonView} onPress={() =>
this.props.navigation.navigate('Dr')}>
                            <Image
                                source={backIcon}
style={styles.backButtonIcon}
                                resizeMode="contain"
                            />
                        </TouchableOpacity>
                    </View>

```

```

        </View>
        <View style={styles.markWrap}>
            <Image source={mark} style={styles.mark}
resizeMode="contain" />
        </View>
        <View style={styles.wrapper}>

            <View style={styles.inputWrap}>
                <View style={styles.iconWrap}>
                    <Image source={personIcon}
style={styles.icon} resizeMode="contain" />
                </View>
                <TextInput
                    placeholder="Email ID"
                    placeholderTextColor="#FFF"
                    onChangeText={email =>
this.setState({ email })}
                    style={ [styles.input,
styles.whiteFont]}
                />
            </View>
            <View style={styles.inputWrap}>
                <View style={styles.iconWrap}>
                    <Image source={lockIcon}
style={styles.icon} resizeMode="contain" />
                </View>
                <TextInput
                    placeholderTextColor="#FFF"
                    placeholder="Password"
                    onChangeText={password =>
this.setState({ password })}
                    style={ [styles.input,
styles.whiteFont]}
                    secureTextEntry
                />
            </View>
            <TouchableOpacity activeOpacity={.5}>
                <View>
                    <Text
style={styles.forgotPasswordText}>Forgot Password?</Text>
                </View>
            </TouchableOpacity>
            <TouchableOpacity style={{ alignContent:
'center', alignItems: "center", justifyContent: "center", }}
activeOpacity={.5}
                onPress={() => this.login()}
            >
                <View style={styles.button}>

```

```

        <Text
style={styles.buttonText}>Sign In</Text>
      </View>
    </TouchableOpacity>
  </View>

  <View style={[styles.container, {
paddingBottom: 10, marginBottom: 10 }]}>
    <View style={styles.signupWrap}>
      <Text
style={styles.accountText}>Don't have an account?</Text>
      <TouchableOpacity
activeOpacity={.5}>
        <View>
          <Text
style={styles.signupLinkText} onPress={() =>
this.props.navigation.navigate('Signup')}>Sign Up</Text>
        </View>
      </TouchableOpacity>
    </View>
  </ImageBackground>
</View>
);
}
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
  markWrap: {
    flex: 1,
    paddingVertical: 30,
  },
  mark: {
    width: null,
    height: null,
    flex: 1,
  },
  background: {
    width,
    height,
  },
  wrapper: {
    paddingVertical: 30,
  },
  inputWrap: {

```

```

        flexDirection: "row",
        marginVertical: 10,
        height: 50,
        borderBottomWidth: 1,
        borderBottomColor: "#CCC"
    },
    iconWrap: {
        paddingHorizontal: 7,
        alignItems: "center",
        justifyContent: "center",
    },
    icon: {
        height: 20,
        width: 20,
    },
    input: {
        flex: 1,
        paddingHorizontal: 10,
    },
    button: {
        backgroundColor: "#FF3366",
        paddingVertical: 20,
        alignItems: "center",
        justifyContent: "center",
        marginTop: 30,
        width: width * 0.5,
        alignContent: "center",
        borderRadius: 40
    },
    buttonText: {
        color: "#FFF",
        fontSize: 18,
    },
    forgotPasswordText: {
        color: "#D8D8D8",
        backgroundColor: "transparent",
        textAlign: "right",
        paddingRight: 15,
    },
    signupWrap: {
        backgroundColor: "transparent",
        flexDirection: "row",
        alignItems: "center",
        justifyContent: "center",
    },
    accountText: {
        color: "#D8D8D8"
    }

```

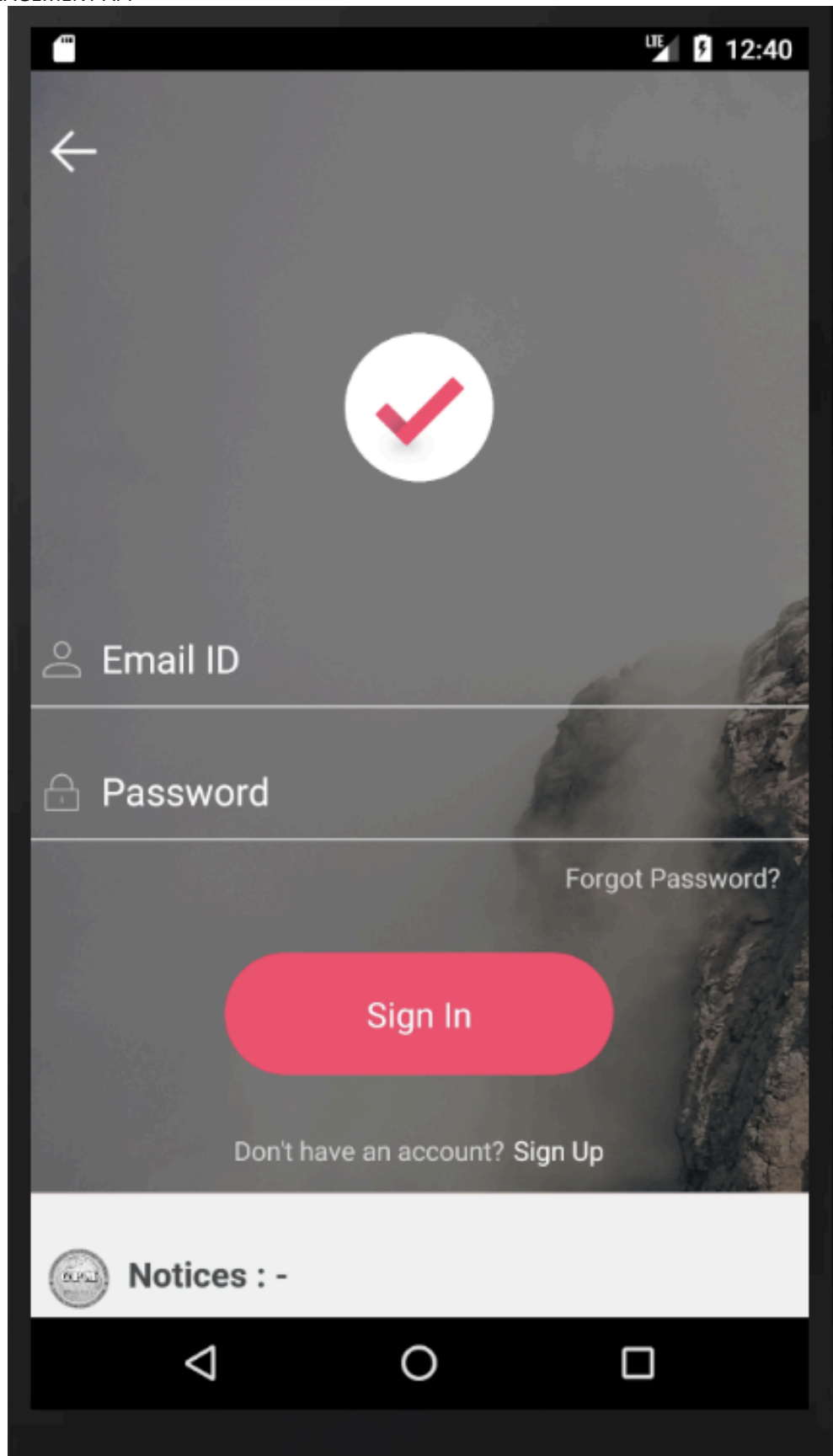


```
    },
    signupLinkText: {
      color: "#FFF",
      marginLeft: 5,
    },
    container: {
      flex: 1,
    },
    bg: {
      paddingTop: 30,
      width: null,
      height: null
    },
    headerContainer: {
      flex: 1,
    },
    inputsContainer: {
      flex: 3,
      marginTop: 50,
    },
    footerContainer: {
      flex: 1
    },
    headerIconView: {
      marginLeft: 10,
      backgroundColor: 'transparent'
    },
    headerBackButtonView: {
      width: 25,
      height: 25,
    },
    backButtonIcon: {
      width: 25,
      height: 25
    },
    headerTitleView: {
      backgroundColor: 'transparent',
      marginTop: 25,
      marginLeft: 25,
    },
    titleViewText: {
      fontSize: 40,
      color: '#fff',
    },
    inputs: {
      paddingVertical: 20,
    },
    inputContainer: {
```

```

        borderWidth: 1,
        borderBottomColor: '#CCC',
        borderColor: 'transparent',
        flexDirection: 'row',
        height: 75,
      },
      iconContainer: {
        paddingHorizontal: 15,
        justifyContent: 'center',
        alignItems: 'center',
      },
      inputIcon: {
        width: 30,
        height: 30,
      },
      input: {
        flex: 1,
        fontSize: 20,
      },
      signup: {
        backgroundColor: '#FF3366',
        paddingVertical: 25,
        alignItems: 'center',
        justifyContent: 'center',
        marginBottom: 15,
      },
      signin: {
        justifyContent: 'center',
        alignItems: 'center',
        backgroundColor: 'transparent',
      },
      greyFont: {
        color: '#D8D8D8'
      },
      whiteFont: {
        color: '#FFF'
      }
    }
  })

```



5.1.6 Sign Up

```
import React, { Component } from 'react'
import {
  AppRegistry,
  StyleSheet,
  Text,
  View,
  Image,
  TextInput,
  TouchableOpacity,
  ImageBackground,
  Dimensions,
  ScrollView,
  Picker,
  Modal,
  Button,
  Alert,
  ToastAndroid
} from 'react-native'
const { width, height } = Dimensions.get("window");
// import LocationPicker from 'react-native-location-picker';
import firebase from 'firebase'; // 4.8.1

const background = require("./signup_bg.png");
const backIcon = require("./back.png");
const personIcon = require("./signup_person.png");
const lockIcon = require("./signup_lock.png");
const emailIcon = require("./signup_email.png");
const birthdayIcon = require("./signup_birthday.png");
const dropIcon = require("./dropdown.png");

export default class Signup extends Component {

  constructor(props) {
    super(props)

    this.state = {
      modalVisible: false,
      enroll: '',
      name: '',
      email: '',
      password: '',
      year: '',
      dept: '',
      shift: '',
      mobile: '',
    }
  }
}
```

```

        address: '',
        dob : '',
        gender : ''

    }
}

pres = () => {
    const { enroll, name, email, password, year, dept, shift,
mobile, address ,dob,gender} = this.state

    console.log(enroll)
    console.log(name)
    console.log(email)
    console.log(password)
    console.log(year)
    console.log(dept)
    console.log(shift)
    console.log(mobile)
    console.log(address)

    fetch('http://192.168.43.64/user_registration.php', {
        method: 'POST',
        headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({

            enroll: enroll,

            name: name,
            email: email,
            password: password,
            year: year,
            dept: dept,
            shift: shift,
            mobile: mobile,
            address: address,
            dob : dob,
            gender : gender,

        })

    }).then((response) => response.json())
        .then((responseJson) => {

```

```

        firebase.auth().createUserWithEmailAndPassword(email,
password).catch(function(error) {
    // Handle Errors here.
    var errorCode = error.code;
    var errorMessage = error.message;
    // ...
});
// Showing response message coming from server after
inserting records.
// Alert.alert(resposnseJson);
ToastAndroid.showWithGravityAndOffset(
    responseJson,
    ToastAndroid.LONG,
    ToastAndroid.BOTTOM,
    25,
    50,
);
}).catch((error) => {
    console.error(error);
});

}
render() {
    return (
        <View style={styles.container}>
            <ImageBackground
                source={background}
                style={[styles.container, styles.bg]}
                resizeMode="cover"
            >
                <ScrollView>
                    <View style={styles.headerContainer}>

                        <View style={styles.headerIconView}>
                            <TouchableOpacity
                                style={styles.headerBackButtonView} onPress={() =>
                                this.props.navigation.goBack()}>
                                <Image
                                    source={backIcon}
                                    style={styles.backButtonIcon}
                                    resizeMode="contain"
                                />
                            </TouchableOpacity>
                        </View>
                        <View style={styles.headerTitleView}>
                            <Text style={styles.titleViewText}>Sign
Up</Text>
                        </View>
                    </View>
                </ScrollView>
            </ImageBackground>
        </View>
    );
}

```

```

</View>

<View style={styles.inputsContainer}>

  <View style={styles.inputContainer}>
    <View style={styles.iconContainer}>
      <Image
        source={personIcon}
        style={styles.inputIcon}
        resizeMode="contain"
      />
    </View>
    <TextInput
      style={[styles.input, styles.whiteFont]}
      placeholder="Enrollment No."
      placeholderTextColor="#FFF"
      onChangeText={enroll => this.setState({ enroll
    }}}

      underlineColorAndroid='transparent'
    />
  </View>

  <View style={styles.inputContainer}>
    <View style={styles.iconContainer}>
      <Image
        source={personIcon}
        style={styles.inputIcon}
        resizeMode="contain"
      />
    </View>
    <TextInput
      style={[styles.input, styles.whiteFont]}
      placeholder="Name"
      placeholderTextColor="#FFF"
      underlineColorAndroid='transparent'
      onChangeText={name => this.setState({ name })}

    />
  </View>

  <View style={styles.inputContainer}>
    <View style={styles.iconContainer}>
      <Image
        source={emailIcon}
        style={styles.inputIcon}
        resizeMode="contain"

```

```

        />
      </View>
      <TextInput
        style={[styles.input, styles.whiteFont]}
        placeholder="Email"
        placeholderTextColor="#FFF"
        onChangeText={email => this.setState({ email
    }}}

    />
  </View>

  <View style={styles.inputContainer}>
    <View style={styles.iconContainer}>
      <Image
        source={lockIcon}
        style={styles.inputIcon}
        resizeMode="contain"
      />
    </View>
    <TextInput
      secureTextEntry={true}
      style={[styles.input, styles.whiteFont]}
      placeholder="Password"
      placeholderTextColor="#FFF"
      onChangeText={password => this.setState({
password }}}

    />
  </View>

  <View style={styles.inputContainer}>
    <View style={styles.iconContainer}>
      <Image
        source={dropIcon}
        style={styles.inputIcon}
        resizeMode="contain"
      />
    </View>

    <Picker
      selectedValue={this.state.year}
      style={[styles.input, styles.whiteFont, {
color: '#FFF', paddingTop: 13, marginTop: 12 }]}
      onValueChange={(itemValue, itemIndex) =>
this.setState({ year: itemValue })}
      prompt='Year'
    >

```



```

        <Picker.Item label="I Year" value="I Year" />
        <Picker.Item label="II Year" value="II Year"
/>
        <Picker.Item label="III Year" value="III Year"
/>
    </Picker>
</View>

<View style={styles.inputContainer}>
  <View style={styles.iconContainer}>
    <Image
      source={dropIcon}
      style={styles.inputIcon}
      resizeMode="contain"
    />
  </View>

  <Picker
    selectedValue={this.state.dept}
    style={[styles.input, styles.whiteFont, {
color: '#FFF', paddingTop: 13, marginTop: 12 }]}
    onValueChange={({itemValue, itemIndex) =>
this.setState({ dept: itemValue })}
    prompt='Department'
  >

      <Picker.Item label="Civil Engineering"
value="Civil Engineering" />
      <Picker.Item label="Computer Engineering"
value="Computer Engineering" />
      <Picker.Item label="Electrical Engineering"
value="Electrical Engineering" />
      <Picker.Item label="Electronics Engineering"
value="Electronics Engineering" />
      <Picker.Item label="Information Technology"
value="Information Technology" />
      <Picker.Item label="Instrumentation
Engineering" value="Instrumentation Engineering" />
      <Picker.Item label="Rubber Technology"
value="Rubber Technology" />
      <Picker.Item label="Leather Technology"
value="Leather Technology" />
      <Picker.Item label="Mechanical Engineering"
value="Mechanical Engineering" />

    </Picker>

```

```

</View>

<View style={styles.inputContainer}>
  <View style={styles.iconContainer}>
    <Image
      source={dropIcon}
      style={styles.inputIcon}
      resizeMode="contain"
    />
  </View>

  <Picker
    selectedValue={this.state.shift}
    style={[styles.input, styles.whiteFont, {
color: '#FFF', paddingTop: 13, marginTop: 12 }]}
    onValueChange={({itemValue, itemIndex) =>
this.setState({ shift: itemValue })}
    prompt='Shift'
  >

    <Picker.Item label="I Shift" value="I Shift"
/>
    <Picker.Item label="II Shift" value="II Shift"
/>

  </Picker>
</View>

<View style={styles.inputContainer}>
  <View style={styles.iconContainer}>
    <Image
      source={lockIcon}
      style={styles.inputIcon}
      resizeMode="contain"
    />
  </View>
  <TextInput
    style={[styles.input, styles.whiteFont]}
    placeholder="Mobile No."
    placeholderTextColor="#FFF"
    onChangeText={mobile => this.setState({ mobile
}}}

  />
</View>

<View style={styles.inputContainer}>
  <View style={styles.iconContainer}>
    <Image

```

```

        source={lockIcon}
        style={styles.inputIcon}
        resizeMode="contain"
      />
    </View>
    <TextInput
      style={[styles.input, styles.whiteFont]}
      placeholder="Address"
      placeholderTextColor="#FFF"
      onChangeText={address => this.setState({
address  })}

      />
    </View>
    <View style={styles.inputContainer}>
      <View style={styles.iconContainer}>
        <Image
          source={lockIcon}
          style={styles.inputIcon}
          resizeMode="contain"
        />
      </View>
      <TextInput
        style={[styles.input, styles.whiteFont]}
        placeholder="Date of Birth"
        placeholderTextColor="#FFF"
        onChangeText={dob => this.setState({ dob })}

      />
    </View>

    </View>
    <View style={styles.inputContainer}>
      <View style={styles.iconContainer}>
        <Image
          source={dropIcon}
          style={styles.inputIcon}
          resizeMode="contain"
        />
      </View>

      <Picker
        selectedValue={this.state.gender}
        style={[styles.input, styles.whiteFont, {
color: '#FFF', paddingTop: 13, marginTop: 12 }]}
        onChange={(itemValue, itemIndex) =>
this.setState({ gender: itemValue })}
        prompt='Gender'

```

```

        >
        <Picker.Item label="Male" value="Male" />
        <Picker.Item label="Female" value="Female" />
        <Picker.Item label="Other" value="Other" />

      </Picker>
    </View>

    <View style={styles.footerContainer}>

      <TouchableOpacity style={{ alignContent: 'center',
alignItems: "center", justifyContent: "center", marginTop: 10 }}
      // onPress={() => this.setState({ modalVisible:
true })}
      onPress={() => this.pres()}
    >
      <View style={styles.signup}>
        <Text style={styles.whiteFont}>Join</Text>
      </View>
    </TouchableOpacity>

    <TouchableOpacity onPress={() =>
this.props.navigation.navigate('Login')} style={{ paddingBottom:
50, marginBottom: 50 }}>
      <View style={styles.signin}>
        <Text style={styles.greyFont}>Already have an
account?<Text style={styles.whiteFont}> Sign In</Text></Text>
      </View>
    </TouchableOpacity>
  </View>
</ScrollView>
</ImageBackground>
</View>
);
}
}

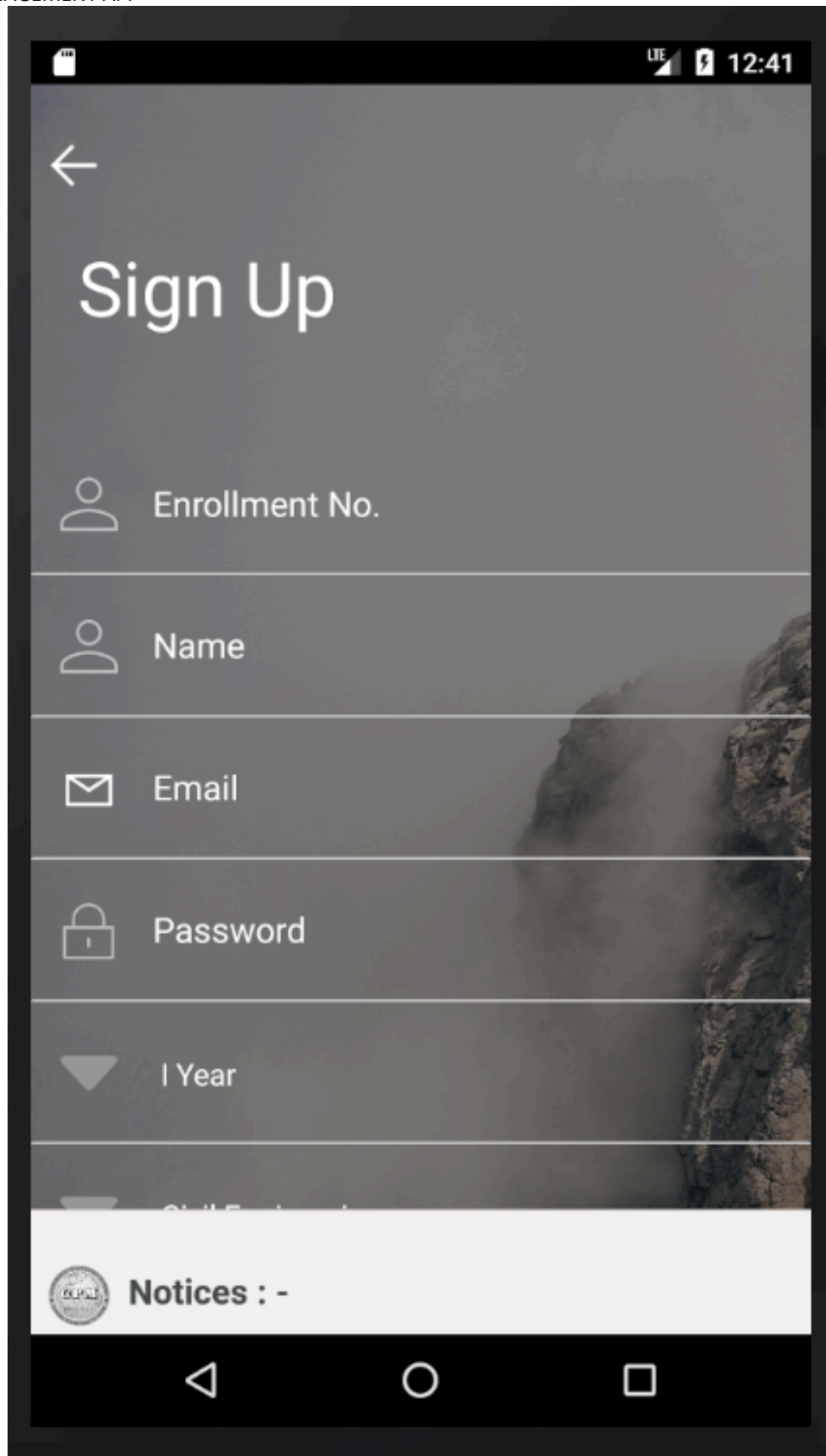
let styles = StyleSheet.create({
  container: {
    flex: 1,
  },
  bg: {
    paddingTop: 30,
    width: null,
    height: null
  },
  headerContainer: {

```

```
    flex: 1,
  },
  inputsContainer: {
    flex: 3,
    marginTop: 50,
  },
  footerContainer: {
    flex: 1
  },
  headerIconView: {
    marginLeft: 10,
    backgroundColor: 'transparent'
  },
  headerBackButtonView: {
    width: 25,
    height: 25,
  },
  backButtonIcon: {
    width: 25,
    height: 25
  },
  headerTitleView: {
    backgroundColor: 'transparent',
    marginTop: 25,
    marginLeft: 25,
  },
  titleViewText: {
    fontSize: 40,
    color: '#fff',
  },
  inputs: {
    paddingVertical: 20,
  },
  inputContainer: {
    borderWidth: 1,
    borderBottomColor: '#CCC',
    borderColor: 'transparent',
    flexDirection: 'row',
    height: 75,
  },
  iconContainer: {
    paddingHorizontal: 15,
    justifyContent: 'center',
    alignItems: 'center',
  },
  inputIcon: {
    width: 30,
    height: 30,
```

```
    },
    input: {
      flex: 1,
      fontSize: 20,
    },
    signup: {
      backgroundColor: '#FF3366',
      paddingVertical: 25,
      alignItems: 'center',
      justifyContent: 'center',
      marginBottom: 15,
      marginTop: 15,


      width: width * 0.5,
      alignContent: "center",
      borderRadius: 40
    },
    signin: {
      justifyContent: 'center',
      alignItems: 'center',
      backgroundColor: 'transparent',
    },
    greyFont: {
      color: '#D8D8D8'
    },
    whiteFont: {
      color: '#FFF',
      fontSize: 18,
    }
  }
})
```





The image shows a mobile application interface for signing up. At the top, there is a status bar with a signal icon, 'LTE', a battery icon, and the time '12:41'. Below this is a dark header with a white back arrow on the left and the title 'Sign Up' in large white font. The main area has a background image of a rocky cliff. It contains five input fields, each with an icon on the left and a label on the right: a person icon for 'Enrollment No.', a person icon for 'Name', an envelope icon for 'Email', a lock icon for 'Password', and a dropdown arrow for 'I Year'. At the bottom, there is a white bar with a circular logo on the left and the text 'Notices : -' on the right. The very bottom is a black navigation bar with three white icons: a triangle, a circle, and a square.


←


Sign Up


 Enrollment No.

 Name

 Email

 Password

 I Year

 Notices : -

5.1.7 Features

```
import React from 'react';
import { StyleSheet, Text, View, TouchableOpacity, Dimensions }
from 'react-native';
// import prop
import { Block, Card, Icon, Label } from '../components';
import { Overlay, Button } from 'react-native-elements'
let width = Dimensions.get('window').width;
let height = Dimensions.get('window').height;
import Concession from './options/concession'
import Feedbck from './options/feedbck'
import Portal from './options/portal'
import firebase from 'firebase'

export default class Gallery extends React.Component {

  constructor(props) {
    super(props)
    this.state = {
      isVisibleConcession: false,
      isVisibleFeedback: false,
      isVisiblePortal: false,
    }
  }

  logout = () => {
    this.props.navigation.navigate('Login')

    firebase.auth().signOut().then(function () {
      // Sign-out successful.

      console.log('correct logout')

    }).catch(function (error) {
      // An error happened.
      console.log(error)
    });
  }

  render() {
    return (
      <View>

        <Overlay
          isVisible={this.state.isVisibleConcession}

```



```

        onBackdropPress={() => this.setState({
isVisibleConcession: false })}
        windowBackgroundColor="rgba(255, 255, 255, .5)"
        // overlayBackgroundColor="red"
        width={width * 0.85}
        height={height * 0.85}
    >
    <Concession
        name={this.props.navigation.state.params.Name}
        email={this.props.navigation.state.params.Email}
        enroll={this.props.navigation.state.params.Enroll}
        year={this.props.navigation.state.params.year}
        dept={this.props.navigation.state.params.dept}
        shift={this.props.navigation.state.params.shift}
        mobile={this.props.navigation.state.params.mobile}
        address={this.props.navigation.state.params.address}
        gender={this.props.navigation.state.params.gender}
        dob={this.props.navigation.state.params.dob}
    />
</Overlay>
{ /* */ }
<Overlay
    isVisible={this.state.isVisibleFeedback}
    onBackdropPress={() => this.setState({
isVisibleFeedback: false })}
    windowBackgroundColor="rgba(255, 255, 255, .5)"
    // overlayBackgroundColor="red"
    width={width * 0.85}
    height={height * 0.85}
    >
    <Feedbck
        name={this.props.navigation.state.params.Name}
        email={this.props.navigation.state.params.Email}
        enroll={this.props.navigation.state.params.Enroll}
    />
</Overlay>
{ /* */ }
<Overlay
    isVisible={this.state.isVisiblePortal}
    onBackdropPress={() => this.setState({
isVisiblePortal: false })}
    windowBackgroundColor="rgba(255, 255, 255, .5)"
    // overlayBackgroundColor="red"
    width={width * 0.85}
    height={height * 0.85}
    fullScreen={true}
    >
    <Portal

```

```

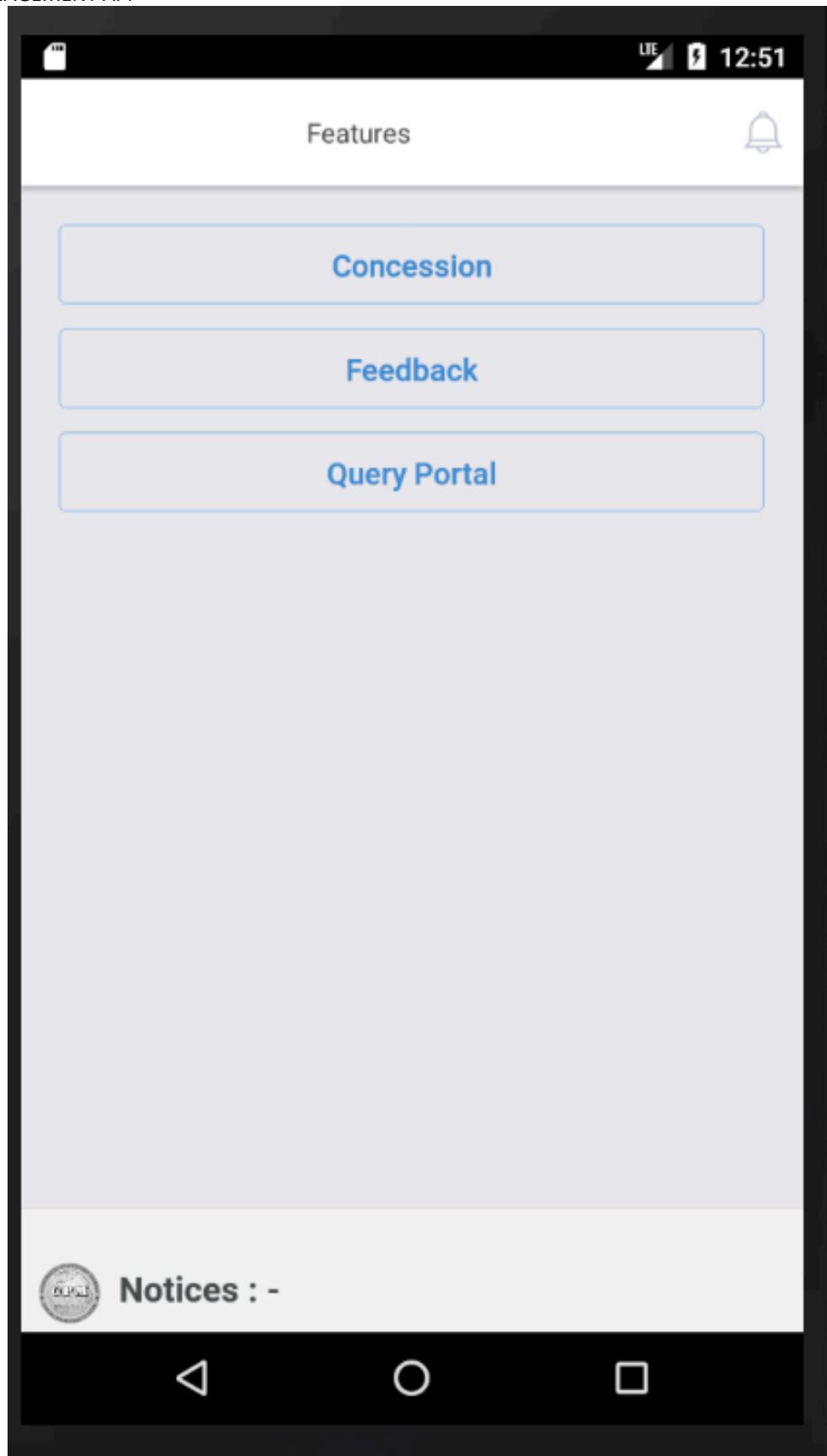
name={this.props.navigation.state.params.Name}

    />
  </Overlay>
  <View>
    <View style={{ flexDirection: 'column' }}
    >
      <View style={{ paddingLeft: width * 0.05,
paddingRight: width * 0.05, paddingTop: width * 0.05, height:
height * 0.05, marginBottom: width * 0.05, }}>
        <Button
          title="Concession"
          type="outline"
          onPress={() => this.setState({
isVisibleConcession: true })})
        />
      </View>
      <View style={{ paddingLeft: width * 0.05,
paddingRight: width * 0.05, paddingTop: width * 0.05, height:
height * 0.05, marginBottom: width * 0.05, }}>
        <Button
          title="Feedback"
          type="outline"
          onPress={() => this.setState({
isVisibleFeedback: true })})
        />
      </View>
      <View style={{ paddingLeft: width * 0.05,
paddingRight: width * 0.05, paddingTop: width * 0.05, height:
height * 0.05, marginBottom: width * 0.05, }}>
        <Button
          title="Query Portal"
          type="outline"
          onPress={() => this.setState({ isVisiblePortal:
true })})
        />
      </View>
    </View>
  </View>
);
}
}

const styles = StyleSheet.create({

```

```
container: {  
  flex: 1,  
  backgroundColor: '#fff',  
  alignItems: 'center',  
  justifyContent: 'center',  
},  
});
```



SSS

CHAPTER 6

TESTING

Chapter 6: Testing

6.1 Unit Testing –

A Unit is a smallest testable portion of system or application which can be compiled, liked, loaded, and executed. This kind of testing helps to test each module separately.

The aim is to test each part of the software by separating it. It checks that component are fulfilling functionalities or not. This kind of testing is performed by developers.

6.1.1 How we have done-

In our project there are many modules(units), first we created units of basic functionalities, feedback , railway concession and complaint portal.

After creating the modules, we tested each modules separately which is nothing but unit testing. The main objective was to see whether all the desired functionalities are met or not. This type of testing is very easy and useful when are application has lot of units. Testing every unit is beneficial because we concentrate on each and every unit which also improves the reliability, efficiency, security etc.

The purpose is to validate that each unit of the software performs as designed.

Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.

After testing each individual units in the program the next step is to do integration. Unit testing is always done before integration testing.

6.2 Integration Testing-

Integration means combining. For Example, In this testing phase, different software modules are combined and tested as a group to make sure that integrated system is ready for system testing.

Integrating testing checks the data flow from one module to other modules. This kind of testing is performed by testers.

Although each software module is unit tested, defects still exist for various reasons like

- A Module, in general, is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration Testing becomes necessary to verify the software modules work in unity.
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous
- Inadequate exception handling could cause issues.

6.2.1 How we have done?-

After testing the individual units our now job is to combine all the units and test them together.

We combined all the units of basic functionality, railway concession, teacher's feedback and complaint portal. After combining the units , we tested them to see if they are producing any defects or errors.

Integration testing gives an idea about how our system will look like, it uncovers more errors than the unit testing.

We have done integration testing with the sandwich approach which is the combination of both the top down and bottom up approach.

6.3 System Testing-

System Testing is the testing of a complete and fully integrated software product. Usually, software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

6.3.1 How we have done?-

- We tested the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. This is also called End to End testing scenario.
- Verify thorough testing of every input in the application to check for desired outputs.
- Testing of the user's experience with the application.

6.3.2 While doing system testing we did-

6.3.2.1 Recovery testing-

If anything happens to the system, is the system able to recover from the failure like if we are downloading an pdf from our app, n suddenly you come out of the app, so when you will return to the app, will the downloading start again from the point where it stopped. We tested all the possible failures from which we can recover. We tested our railway concession feature, that is while uploading the documents, if suddenly we come out of the app, the uploading should not stop. We have also kept backup for our application so that if anything happens to it we can restore it.

6.3.2.2 Security testing-

Security testing attempts to verify that the protection mechanisms built in the system will protect it from improper penetration.

We tested our application against all the possible security measures. Like while we login into the system, and after that when we logout and click the back button it should provide the user with the login page again rather than providing again the logged-in application.

6.4 Performance testing-

We did performance testing to ensure that the output of the application is given within the time. Our main aim was to measure the Response time, Throughput and utilization. During performance testing we had make sure that all the functions of the system is giving their respective time easily and within less time.

6.5 Usability testing-

We have done usability testing by give our application to small set of target end users so that they expose usability defects. During usability testing we ensured that our application can be used easily by the users, it provides flexibility in handling controls and the our system meets its desired objectives.

6.6 Compatibility Testing-

Since our application is an react-native based application it is compatible on both the mobile devices of android and iOS. During compatibility testing we ensured that our application works on different hardware, operating systems, network environments or mobile devices. We also ensured that our application is backward and forward compatible with the previous and new versions that will be coming into the market.

6.7 Acceptance Testing-

It is a phase of software development in which we tested our software by making it available to the intended audience. We took the experiences of the early users and made the final changes before showing to our guide. It allows the users to ensure that their needs are satisfied. We did both the types of testing that is the alpha and beta. In the alpha testing we tested the application at our site and in the beta testing we provided the beta version to our classmates and juniors to use and give the feedback.

6.8Dummy Test Cases –

SR NO.	Objective	Steps	Input	Expected output	Actual output	status
1	Verify that user is able to navigate through all the products across different categories	Open the app and check.	open the app.	yes	yes	pass
2	Verify that the app logo is clearly visible	Open the app and check.	open the app.	yes	yes	pass
3	Verify that category pages have relevant data listed specific to the category	Open the app and check.	open the app.	yes	yes	pass
4	Verify that app's railway concession functionality works fine.	Open the app and login into the app and verify the railway concession functionality.	Upload documents.	yes	yes	pass
5	Verify that app's feedback form functionality works fine.	Open the app and login into the app and verify the feedback form functionality.	give feedback.	yes	yes	pass
6	Verify that app's query portal functionality works fine	Open the app and login into the app and verify the query portal functionality.	Ask queries.	yes	yes	pass
7	Verify that user can upload documents for verification of concession documents.	Open the app and login into the app and verify whether you can upload documents.	Upload documents.	yes	yes	pass
8	If invalid information is given, show error message " please re-enter ".	Open the app and enter any wrong information and check it.	Give wrong input.	yes	yes	pass
9	Check the validations on Email and password.	Open the app and check the validations on the login and sign-up page.	Write info and see the validations.	yes	yes	pass
10	Check that duplicate password creation is restricted.	Open the app and try to create two passwords.	Create two passwords.	yes	yes	pass
11	Verify the page has both submit and cancel/reset buttons at the end.	Open the app and check whether every pages has it.	open the app.	yes	yes	pass
12	Check if the applications different pages load faster.	Open the app and go through every page and check whether it loads faster.	open the app.	yes	yes	pass
13	Verify that Enter/Tab key works as a substitute for the Sign in button.	Open the app and check.	open the app.	yes	yes	pass
14	Verify that clicking on browser back button after successful login should not take User to log out mode	Open the app and login and then press back button and check that you should be logged out of the app.	Login	yes	yes	pass
15	Verify that validation message gets displayed in case user leaves username or password field as blank.	Open the app and test the app by entering the username.	Enter the username.	yes	yes	pass

COLLEGE MANAGEMENT APP

16	Verify if there is checkbox with label "remember password" in the login page	Open the app and see the remember password field has on the login page and sign up page.	open the app.	yes	yes	pass
17	Verify the timeout of the login session	Open the app and verify the timeout time.	open the app.	yes	yes	pass
18	Test if there is option for logout then user logout successfully	Open the app and see whether logout option is their or not.	open the app.	yes	yes	pass
19	Verify that there is a limit on the total number of unsuccessful login attempts	Open the app and try to login again and again.	Login again and again.	yes	yes	fail
20	Display the successful submission message "successfully completed".	Open the app and do some transactions successfully.	Try uploading documents.	yes	yes	pass

CHAPTER 7

Future Scope

Chapter 7: Future Scope

7.1 Future Scope of College application

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add printing facility to the railway concession in future.
- We can give more advance application for college including more facilities
- We will host the application on online servers to make it accessible worldwide
- Create the master and slave database structure to reduce the overload of the database queries
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of college and course. Also, as it can be seen that's now-a-days the players are versatile, i.e. so there is scope for introducing a method to maintain the College application. Enhancements can be done to maintain all the key feature of application such as railway concession, faculty feedback and query portal.

We have left the all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them. In the last we would like to thanks all the person involved in the development of the application directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process.

CHAPTER 8

Conclusion

Chapter 8: Conclusion

Conclusion

This app is to enhance the effective handling of information flow and to manage different features and functionalities in the app. This app will provide notifications time to time and enhance the teaching scheme and save the time and efforts of the college staff. Our system not only reduces the complexity in the data collected but also it helps the students do their work successful and easily. This propose system includes the features like railway concession, feedback, and query portal. These modules help the system work successfully.

CHAPTER 9

References

Chapter 9: References

References-

1. <https://facebook.github.io/react-native/>
2. <https://reactnavigation.org/>
3. <https://nativebase.io/>
4. <https://www.npmjs.com/package/firebase>
5. <https://github.com/Elyx0/react-native-document-picker>
6. <https://github.com/react-native-training/react-native-elements>
7. <https://github.com/itinance/react-native-fs>
8. <https://github.com/FaridSafi/react-native-gifted-chat>
9. <https://github.com/archriss/react-native-image-gallery>
10. <https://www.npmjs.com/package/react-native-pdf>
11. <https://www.npmjs.com/package/react-native-radio-form>
12. <https://github.com/crazycodeboy/react-native-splash-screen>
13. <https://www.npmjs.com/package/react-native-star-rating>
14. <https://akveo.github.io/react-native-ui-kitten/#/home>
15. <https://www.npmjs.com/package/react-native-table-component>
16. <https://github.com/oblador/react-native-vector-icons>
17. <https://github.com/wkh237/react-native-fetch-blob>
18. <https://firebase.google.com/docs/?authuser=0>
19. <https://github.com/affan00733/GPM-app-new-ui.git>
20. <https://github.com/affan00733/GPM-Admin-panel-PHP.git>
21. <https://github.com/affan00733/GPM-app-server-.git>