

CHAPTER 1

INTRODUCTION

1.1 General

Mobile Application, also called as Mobile App, is a general term referring to programs developed to run on mobile devices such as smartphones and tablets. The environment typically associated with each mobile device is called mobile operating system such as Windows, Android, IOS etc.; these mobile applications can be downloaded and installed through various sources. The term “App” is shortened form of the term ‘Application Software’. With the advancement of technology the computational power within a computer has been made available within a smartphone letting users adapt to mobile technologies very easily. As a result most the software that was available in the computers had to be made available because of the adaptive nature of mobile technologies. With the penetration of technology in various fields the mobile apps are being made available for many fields. This has made our day to day activities available with the push of a button or swipe of a thumb. Owing to these advancements the development and deployment of mobile scaled applications is seemingly becoming popular. The mobile environment provides the necessary resources for these mobile apps to perform well. Thus any service that is available on computer can be made available within the mobile environment in the form of application. These mobile applications are similar to working of computer applications but are limited by the resources available within the mobile device. But with the availability of cloud computing some of the resources can be extended beyond capability of mobile devices. These mobile applications serve handy purposes as they are available anywhere anytime without much hassle. Owing recent trends the computational capacity within the mobile environment is surpassing that of computers. With the global rise of computational devices these mobile environments are being embedded into various other devices where in these mobile apps play a greater and widespread role in the fore coming era of technology.

1.2 Motivation

Our Interaction with generic software environments is continuously evolving. In a university level these software environments are common now days in the form of learning management systems or in the form interactive applications. Applications like Moodle, Capterra, and Litmos etc. are general software applications that can be customized as per the university and environment needs? But all these existing applications provide a general solution

to the problem of learning management where as they might lack some of the concrete features. Our study with the existing systems shows that the need for applications that support a learning management system without compromising the principles of dedicated network for social interaction within the university and common implications like portability, Accessibility etc., is increasing. Keeping in mind the issues that arise within a university even with the adoption of features of a learning management system, there is better scope for applications that adopt to socialization and connectivity without compromising the foundation principles of learning management system. Keeping these things in consideration, we designed an idea that could solve the very basic issues that need to be definitely addressed at a university level. Thus the Application **UNI-CON** abbreviation for University Connection is the application that we hope would solve the fundamental issues that are addressed.

1.3 Problem Description

There are many learning management applications that can be customized as per individual university needs but they are left blank at addressing many issues that universities face. Some of them are universal connectivity, accessibility anytime anywhere, socialization, Compatibility, Interaction etc., Applications that address all these needs are very few that too with limited accessibility. Some of them are available with some tagged price. In our current project we implement our idea of one such application that addresses most of these problems. We record the initial level observations that need to be readdressed. Our main objective is to develop a generic application that could address all these needs dynamically to the maximum extent.

1.4 Related works

There are several research papers and existing works that predate to 2010. Upon examining the works of various companies and research scholars we formed the base for our idea to be implemented. In one of the research papers The Frame Work Design of Mobile Learning Management System by J.Hemabala, the author described the educational opportunities of teaching in a real time wireless classroom using mobile devices. Conventional classroom learning has certain weaknesses. The author clearly presented an institutional survey from two hundred undergraduate students on the problems faced in conventional classrooms generally. The main purpose of her research is to enhance the conventional classroom teaching and learning approach, and the author contribution to this research study is to overcome the learning difficulties faced by students in International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 01– Issue 02, November 2012 www.ijcit.com 180 conventional classroom through the use of mobile devices in wireless classrooms. The survey was conducted with two hundred undergraduate students on the use of mobile devices in a wireless classroom from universities in Malaysia. The questionnaire were distributed to find the weaknesses of conventional learning and the

type of mobile learning applications that they would like to use in a classroom using a mobile device, and the five point Likert scale can be used for each question. From the survey results, specific mobile learning applications are being developed for students and instructors. These applications could be used on a Pocket PCs, notebook computers and mobile phones. The author also provided a variety of instructional application such as classroom, chat room, collaborative text editor, synchronization of power point slides, accessing on the courseware on a mobile device, sending and receiving feedback, and emails and accessing to remote computing resources. The author describes the architecture of GroupNet and demonstrates its functionality in the specific area of mobile learning. GroupNet uses handheld devices as the main hardware and GroupNet architecture consists of four layers. The function of the GroupNet kernel is the activity control, online status, files sharing, message passing and session recording. GroupNet concept is designed to support group tasks, thus the GroupNet application are also designed and implemented according to the nature of group tasks. GroupNet concept proposed in this study is more focused onto the design of mobile learning management system (m-LMS) which can better support mobile learning for a small group of learners with effective social interaction within proximity.

We also studied some of the existing applications that are a breakthrough in Mobile learning management systems. Moodle is the pioneer in learning management systems. In the recent years Moodle also developed mobile app to substantiate the support for the computer based mobile learning management system. This is similar to GroupNet. In Moodle 2.1, we start to see device type theme settings and the ability to set a mobile and tablet themes to cater for the different devices. This approach meant that access from these devices will not be hindered due to screen size or resolutions. Moodle would use the appropriate theme for accessibility without the need to install a separate app. Recognizing the need to cater for mobile access with the increasing number of smartphones and tablets sold, Moodle HQ launched the first official app in September 2011. This is different to the device targeted theming mentioned above.

A number of web-services were developed in Moodle 2.0. In Moodle 2.1, we see further expansion to these web-services leading to support and integration of My Moodle app, HQ's first app. Moodle Mobile App is a Moodle client based on HTML5 that connects to your existing Moodle server. The focus here is on making content available offline and access to Moodle's internal messaging system. As with My Moodle, it lets you upload files to your Private File area. Given the right permission on your device, it will access the Photo album to select an existing photo to upload, take live shots using the built-in camera app and record audio. You can access all the courses that you are sign up to as a student, check out the contents and interact with other course participants in these courses. Several such implementations and research papers focused on the framework to be enabled so as to make these applications that support mobile learning management

systems work great. Some of the studies show that these existent learning management systems are lacking in many aspects that are required by the students as well as the faculty. A recent survey by the educational society of America shows that students are more interested and willing in adopting such application over traditional methods as these applications are making the content sharing very easy. Besides ease of accessibility and availability these application also enhance the concept of knowledge sharing. Even with Moodle and other learning management systems available knowledge sharing and content management tend to be particularly very easier than traditional methods. With the latest advancements in these applications and the universal adaptability these Mobile based learning systems are gaining much prominence now a days.

1.4.1 System Requirements

Hardware Requirements: A smartphone with basic processing capabilities with camera and internet connectivity

Software Requirements:

- OS : Android operating system
- RAM: for better performance we recommend 512Mb
- Memory: Min of 80Mb disk space

CHAPTER 2

OVERVIEW OF THE PROPOSED WORK

\

There are many numbers of quality works by many authors that are done and are being currently done by others. Many of the product oriented organizations already deployed their products while some of them are open source while some of them are not. Many authors have proposed efficient theories regarding the framework to be used and the experiments conducted on these theories so as to check their efficacy. One such theory was by J.Hemabala

Ref: [<http://www.ijcit.com/archives/volume1/issue2/Paper010224.pdf>]

This paper is focused on frame work design of mobile learning management system which improves learner's knowledge, performance, achievements, problem solving skills and individual learning system. The frame work design consists of a content module, learning module and evaluation module with learner, teacher and knowledge interface. It supports online, offline, in campus and out campus learning activities. The content module can be divided into five authoring tools i.e. content development, content management, content distribution, content collaboration and content delivery. The wireless technology development leads to restructuring of conventional classroom learning process into mobile learning from group of network with collaborative learning.

Moodle is an organization that has pioneered the process of learning management system and it is completely open source. We are taking many references from Moodle as it has already developed mobile apps for learning management system.

Ref: [https://docs.moodle.org/26/en/mobile_app]

And other references that we have studied are

- [http://www.academia.edu/11306874/Integrating_Mobile_Learning_Management_System_Using_Activity_theory_In_Higher_Education]
- [<https://net.educause.edu/ir/library/pdf/ers1414.pdf>]
- [https://books.google.co.in/books?id=qWjIQm8iYJgC&pg=PA990&lpg=PA990&dq=search+mobile+lms&source=bl&ots=jX8kpbDiZ8&sig=mL_r2yAVoFR7xcWaBDvO9DGKJXo&hl=en&sa=X&ved=0CGQQ6AEwC2oVChMloPiEpoyPyQIVSgmOCh3GkAF8#v=onepage&q=research%20mobile%20lms&f=false]
- [<http://www.mlearn.org.za/CD/papers/Attewell.pdf>]

2.1 INTRODUCTION OF PROBLEM AND ITS RELATED CONCEPTS

With the advancement of technology, the power of computing and connectivity is very much owned in our hands in a Mobile device. The power of processing and accessibility are being ruled by these mobile environments. Making much use of its aspects the existing systems are mainly focused on enabling a wide spread learning management system. The current day system allows us for an active learning environment but they lack in accessibility. Some of the daily important tasks are not being addressed by these Learning Management systems. They are concerned with content management and resource sharing than other normal things which is making them a bit lacking to adopt

Mobile LMS or Mobile leaning management systems are the systems that use mobile platform as resources and function upon these mobile environments. They provide normal functionalities like other learning management systems such as Learning capability building by knowledge sharing and content management, resource sharing etc.,

Although these functionalities are limited by the environment, most of the current day mobiles are as fast as computers. Hence the concept of environment issues can be omitted. These mobile based learning management applications similar to that of the other applications and can be restrained to university or the organization level for the issues being addressed.

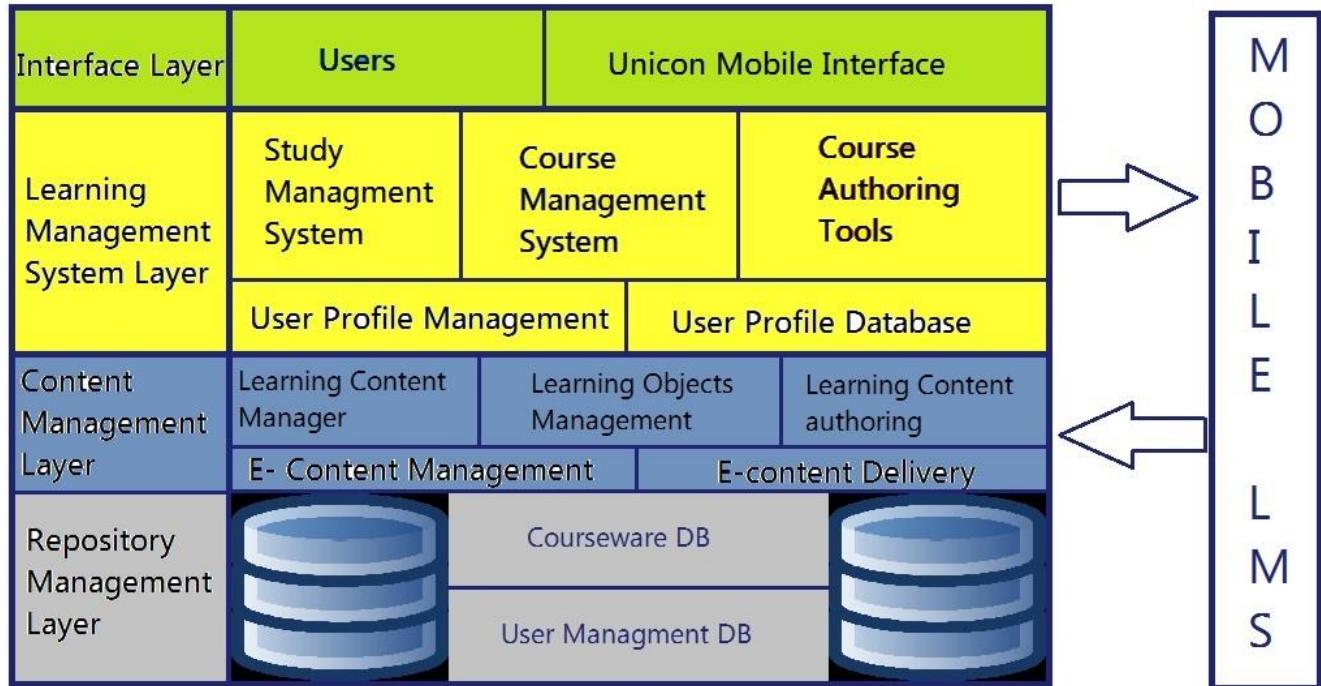
Related Concepts:

Mobile LMS: Mobile learning management service is a type of service that can be designed and made available on the mobile environment. It deals with the management of learning resources and application oriented services

Mobile environment: Mobile environment refers to the type of operating system that the mobile user is using. Essentially our application is targeted for android operating system users. Hence we are referring to android OS.

Learning Environment: The learning environment refers to a school or a college atmosphere where there will be faculty as instructors and students as learners. It follows typical University environment like posting tasks, submitting assignments, posting notes for learners etc.

2.2 Overview of the Proposed System

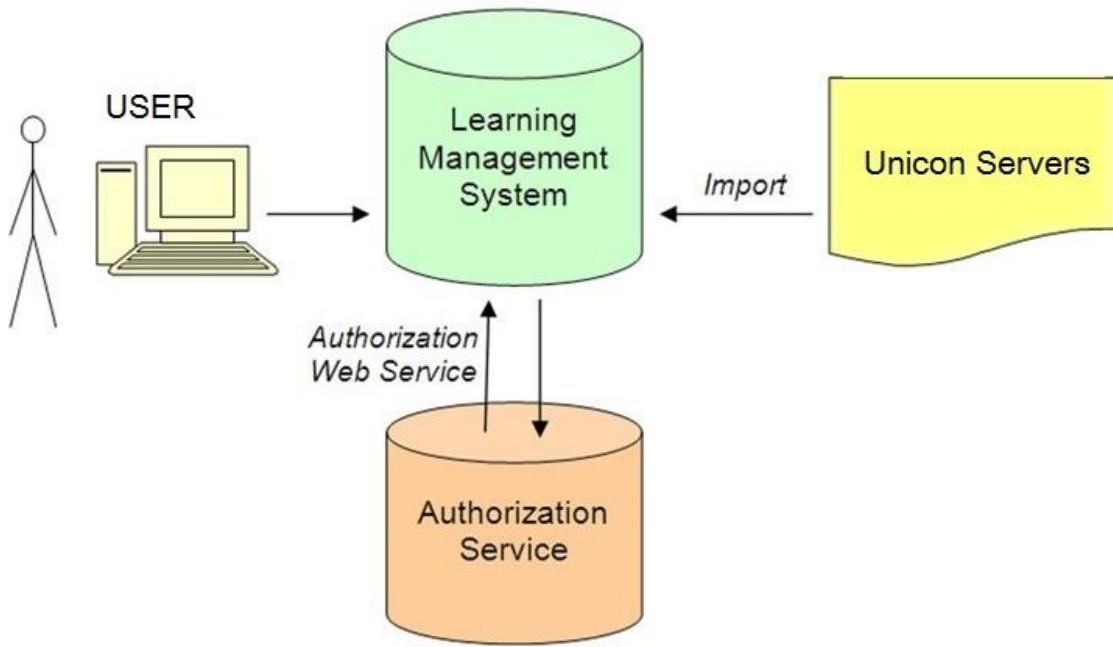


2.2.1 Layered Architecture Diagram

The Layered Architecture diagram represents various application and interface layers that are totally functional in the Unicon mobile application. From the very basic Repository layer which we are being serviced by parse web servers, Courseware database management and Users management is practically portable. These databases from parse are purely relational databases that make it very easy for the user to form relational tables. The content management layer which rests upon the database layer Comprises of the learning content management and learning objects management system which is parsed only after the learning content is authored by users.

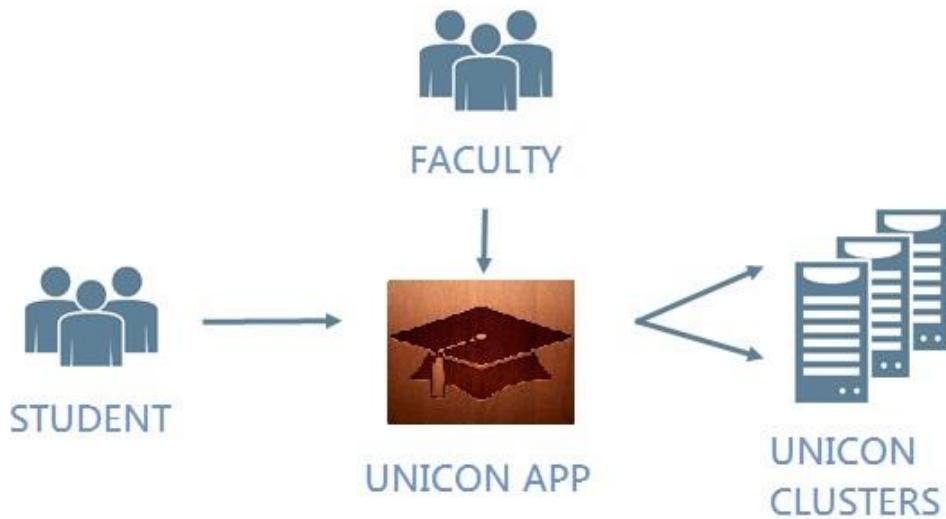
Upon this layer the Learning management system layer rests which consists of two main sub layers called user profile management and user profile database layer. This layer comprises of the key layers of LMS such as study management system and course management system associated with it. The course authoring tool is pure user oriented one and a promoter of the complete learning management system. The rules are set and played by these course authors. The

interface for all these objects and functionalities is provided by the interface layers which supports various users and individual functionalities for each of them. Even though all of these layers rest upon each other they are bind to the mobile Learning management system layers.



2.2.2 UNICON APP Architecture

The Typical architecture of the system Consists of authorization as key factor as the whole application uses role based authentication system with individual permission confirming to key actions and objects by individual users. All the data resides on the Unicon clusters. Hence the application should continuously interact with application services for the data to be imported and updated with the application web servers.



2.2.3 Basic Architecture

The UNICON clusters consist of clustered databases for user services. They store the information in parse databases. Due to the constraint of role based authentication system these clusters are restrained to minimalistic performance. For the successful running of the application the server clusters must continuously function as no local database is being used for storing the local variables and data. As a result the application totally functions on cloud servers and databases. These servers are provided by parse where the application libraries used in the application continuously interact with the clusters for data.

The application typically consists of parse servers and database at the back end and Android application interface at the front end. Service requests and user management are the typical functionalities. User role based authentication system is the key feature controlling the permission of read and writes into the database. The typical course breakdown and authoring determines the course breakdown and Mentoring by the faculty. These are key features that determine the performance of the application. Currently the servers are not redundant controlled but soon the server's capability will be matched for redundant data interference and control.

CHAPTER 3

ANALYSIS AND DESIGN

3.1 Brief Introduction

The UNICON application features top level minimalistic design system for cohesion between modules. As a result all the modules are clearly abstracted in design to promote the cohesion between modules. All these modules act individually in terms of data service or interface but they can be integrated into single application. This hat makes the application unique in terms of service. Thus the application can also be made to provide individual modules as a service in the form of API's or service URL's. Apart from these factors the design is kept pretty much simple so as to afford maximum users with a minimum complexity. All the design elements are standard components of the application that form crucial parts to serve the user. The minimalistic design is what drives the application in terms of performance. The main focus of the application is to solve the Real time problems encountered. Hence simple design and refactoring are part of the planning game from the initial stages.

The various benefits associated with these minimalistic design features are simplicity in application with the scope further enhancements. Since the process taken is incremental the cohesive functions of the design are pretty much useful in various aspects.

3.2 Requirements Analysis

3.2.1.1 Functional requirements

- FR1 : The Users must be able to register and login
- FR2 : The users depending on their roles should have functional accessibility
- FR3: The Data should be continuously accessible and should be auto updated
- FR4 : The users should able to perform update action to credentials deepening on permissions
- FR5 : The users Should be able to respond to the posts and Messages
- FR6: The user should be able to view the posts and messages within permissive limits
- FR7: The Users must be able register and add to the courses offered
- FR8 : The users must be provided the facility of creating or responding to requests and posts

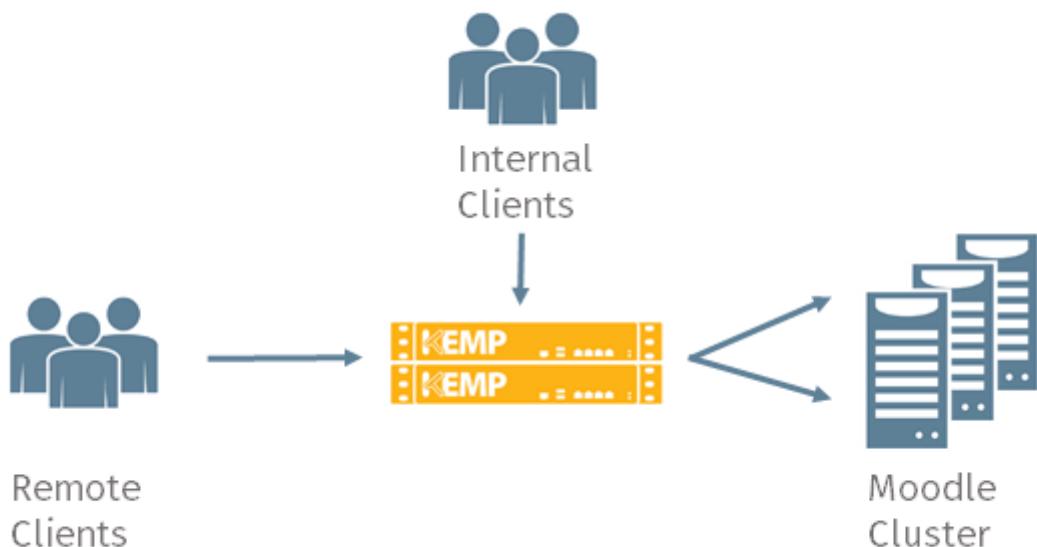
3.2.1.2 Domain Requirements

- MBaaS
- Android 4.1 above

3.2.1.1 Non-Functional requirements

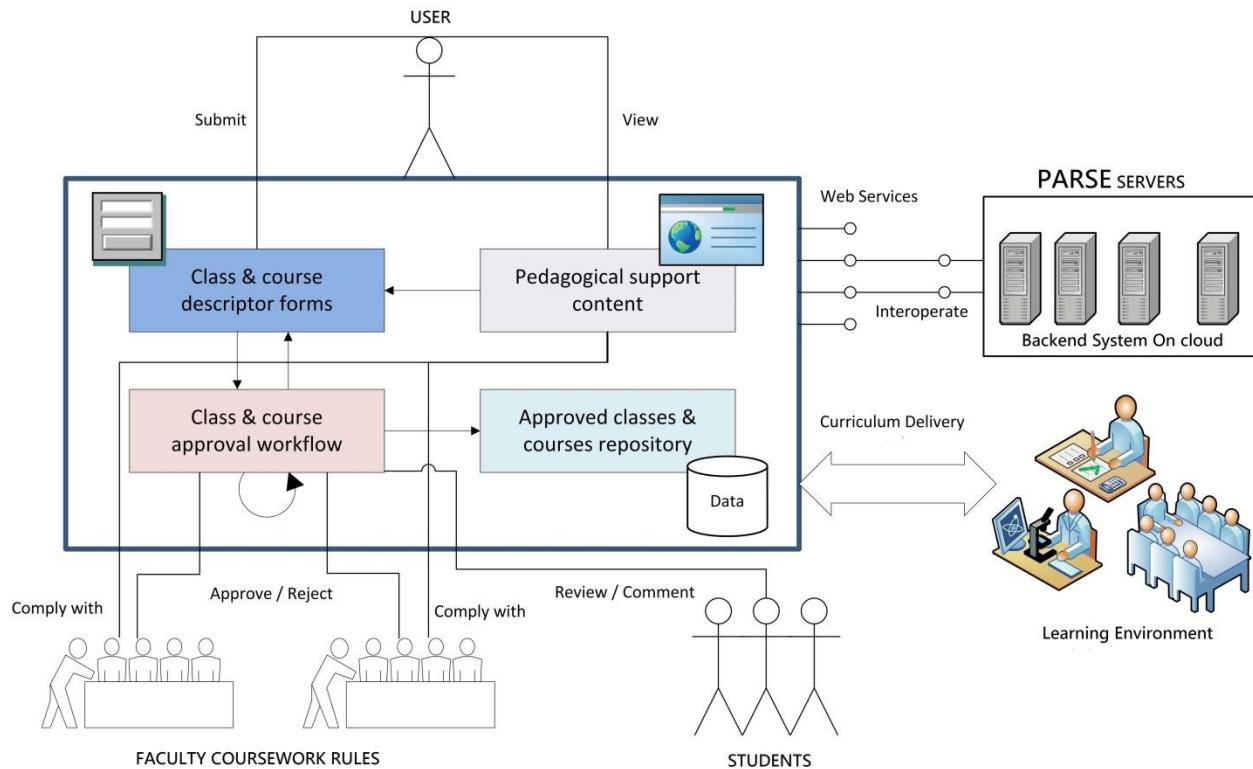
- Security
- Performance
- User Friendly UI
- Maintainability
- Scalability
- Usability
- Availability

3.3 Detailed Design



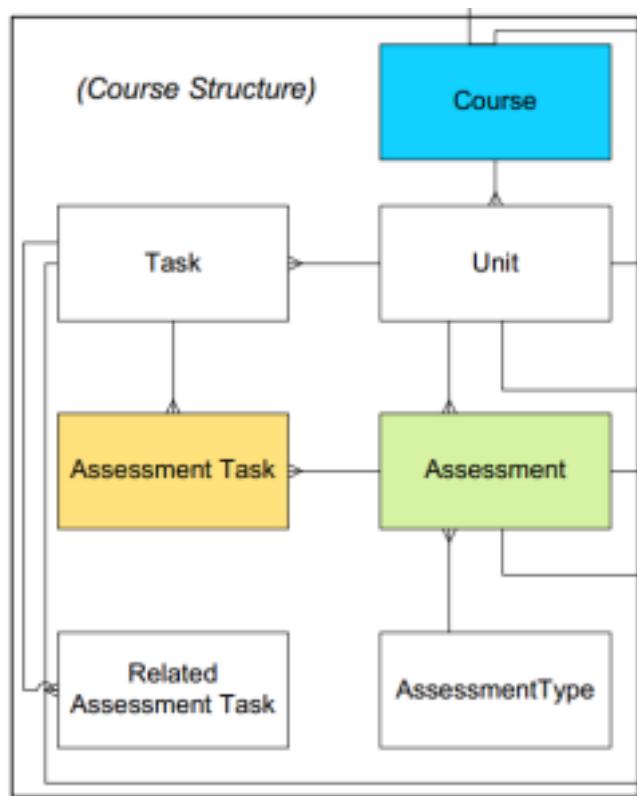
3.3.1 The Basic Architecture for UNICON

The system architecture typically is represented by the functions and integration within the modules and in between the modules. The cohesive functionality can be seen within the application structure modules.



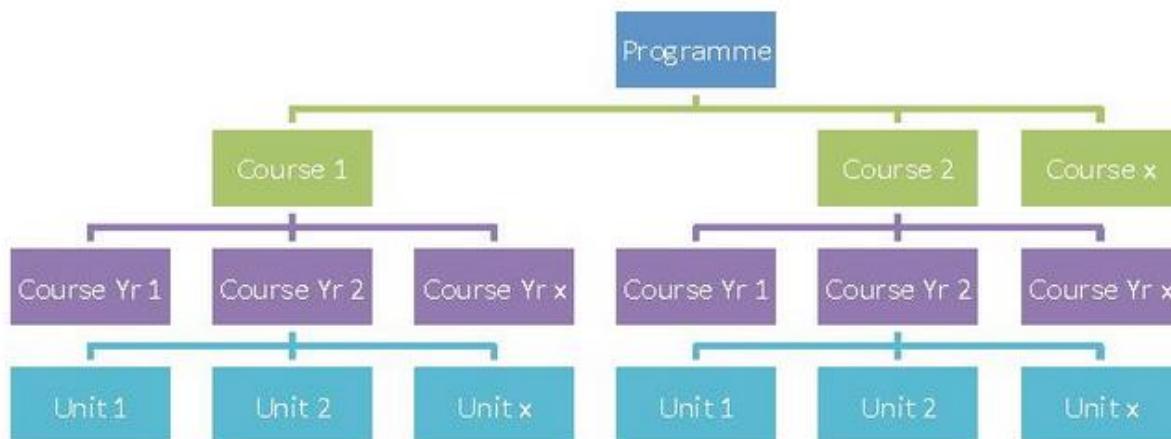
3.3.2 UNICON Design

The Unicon architecture typically represented within the diagram is composed of typical functionality that the application addresses. The various functional sub modules are the supporting types for the main functions that lay within the design architecture. The course authoring system acts as continuous monitoring system for the application. The central core always revolves around the typical functionalities of the system and the features of the system. The course objects serve as core features of the system this also comprises of the course structure and course initialization of the system within different roles. The key areas are the structuring and building blocks of the application where the standards that need to be achieved from the application side are decoded and proved to be applicable with various factors



3.3.3 Course Structuring

The Course structuring is a key feature in the building of the application, the entire content authoring is base for the course structuring in the design provided.

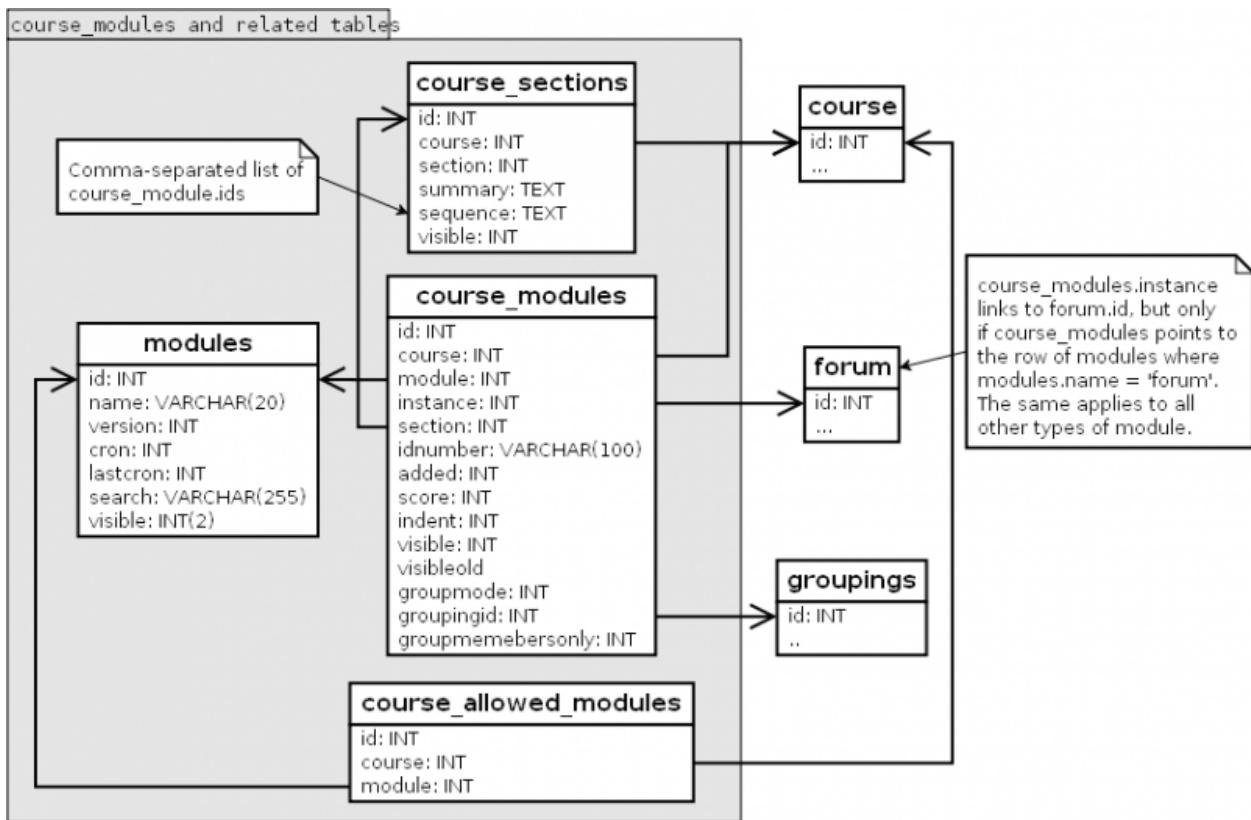


3.3.4 Content Authoring

The database design consists of two typical Parts:

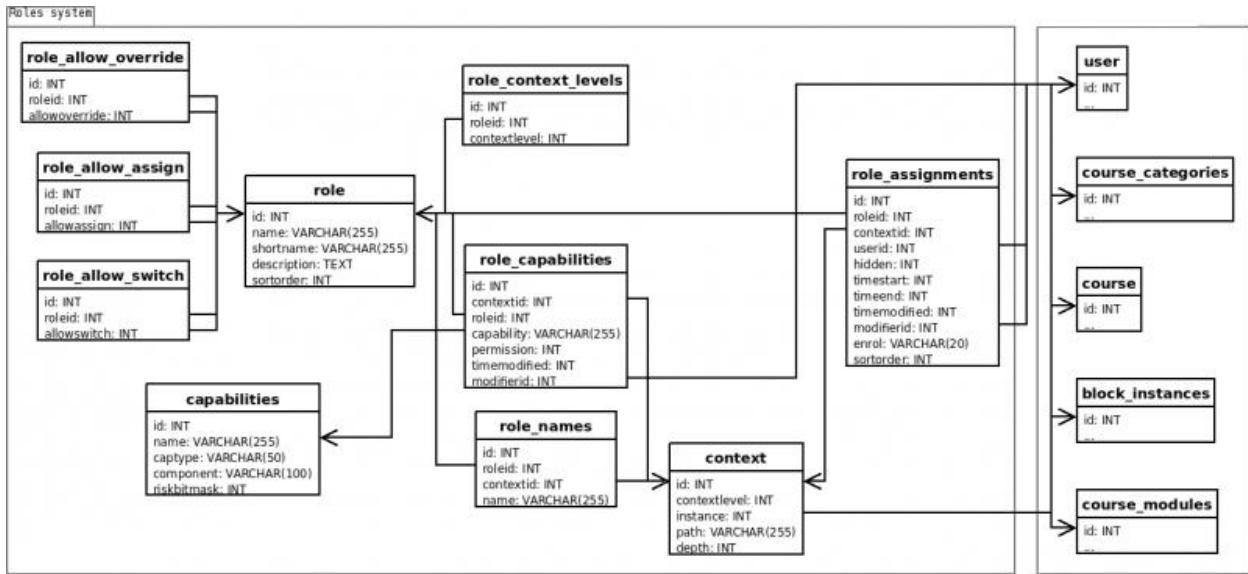
1. Modules Database Structure

2. Roles Database Structure



3.3.5 Modules Database

Module database typically addresses the database concerns of the system from module related aspects. From the perspective of relative modules and submodules all the issues are clearly defined by the relations of the database structure



3.3.6 ROLES database Structure

3.4 Module Description

Module: 1

Module Name: PROFILE

Profile is the module which consists of the personal information about the student and the faculty depending on their access/ login. The sub modules present in profile are

- View My Profile
- Edit My Profile
- Change Password
- Delete My Account

Sub Module: 1.1

Sub Module Name: View My Profile

It provides the overview of profile for both Student and the Faculty which includes the details like Name of the Student/Faculty, Email Id of the Student/Faculty, University Id of Student/Faculty, Profile picture of the Student/Faculty and their respective Roles.

Sub Module: 1.2

Sub Module Name: Edit My Profile

It provides an opportunity to update the details of the faculty/ student. The information that can be edited is Name, Email Id and the Profile Picture.

Sub Module: 1.3

Sub Module Name: Change Password

It facilitates the Student and the Faculty to change their password. By entering their Old Password, New Password and by Confirming their New Password Student/Faculty can update their New Password.

Sub Module: 1.4

Sub Module Name: Delete My Account

By Clicking on Delete My Account option both the Student and the Faculty are asked to confirm whether they really want to delete their account. If they select Yes then the account this be deleted else the account remains same as before.

Module: 2

Module Name: COURSES

This module provides information regarding the list of courses registered, list of pending courses, and list of students registered under a particular course. The Sub Modules present in the Courses module are

- Create Course
- Register for a New Course / Join Course
- My Courses
- Unregister the Course
- Delete the Course

Sub Module: 2.1

Sub Module Name: Create Course

Access Permissions: Faculty

Faculty can create a course through which students get added into it. The details required for creating the course are Course Name, Description regarding that particular course, Class Time which should include start time and end time, Class Days and they should also generate the Course Code by clicking on generate unique course code. By filling these details correctly, the faculty can be able to create course.

Sub Module: 2.2

Sub Module Name: Register for a New Course / Join Course

Access Permissions: Student

The Student can register for a course by searching the course using the course name. The list of courses related to that search content will be displayed. Student has to click on the desired course then it is required to select Yes/No when they are asked whether they are willing to register for that particular course. If the student selects Yes then the Request will be sent to the respective Faculty for approval else the request will be discarded.

Sub Module: 2.3

Sub Module Name: My Courses

Access Permissions: Student

It gives the list of courses registered by the student. On Clicking on the particular Course they could be able to view the information like Course Name, Description, Class Schedule, Class

Schedule, Professor and Professor Email Id. It also displays the posts regarding the particular course and the list of participants.

Access Permissions: Faculty

It gives the list of courses created by the student. On Clicking on the particular Course they could be able to view the information like Course Name, Description, Class Schedule, Class Schedule, Professor and Professor Email Id. It displays the posts regarding the particular course and the list of participants. The Faculty can also be able to create posts regarding that particular course.

Sub Module: 2.4

Sub Module Name: Unregister the Course

Access Permissions: Student

The Student is provided with an option to withdraw the registered courses by clicking on Unregister the course and confirming their request of withdrawal by selecting Yes/No.

Sub Module: 2.5

Sub Module Name: Delete the Course

Access Permissions: Faculty

The Faculty can delete the course from the list of courses created by them.

Module: 3

Module Name: AUTHENTICATION

This module provides the access permissions to the Faculty and the Student. The Sub Modules available in this module are

- 3.1 Registration
- 3.2 Login

Sub Module: 3.1

Sub Module Name: Registration

Access Permissions: Student, Faculty

Only registered Students and Faculty are allowed to access this application. Student/ Faculty can register by providing the information like Name, Email Id, University Id, Password, Role and Profile picture (optional).

Sub Module: 3.2

Sub Module Name: Login.

Access Permissions: Student, Faculty

Student/ Faculty can be able to login into the application using their registered details like University Id and Password.

Module: 4

Module Name: TASKS

Both the Student and the Faculty are involved in this module. The work assigned to the students is given by the faculty in the form of tasks which are to be submitted by the students within the particular Time Span. The Sub Modules Present in it is as follows

- 4.1 Create Task
- 4.2 View Task
- 4.3 Submit Task

Sub Module: 4.1

Sub Module Name: Create Task

Access Permissions: Faculty

Faculty can create a task for the student of that particular course by mentioning the Title, Context, adding Image (optional) and by selecting the Category. Category can be any of the following

- Announcement
- Submission Request
- Material Post
- Forum Thread

For the above Submission Deadline, Material File has to be added depending on the type of category.

Sub Module: 4.2

Sub Module Name: View Task

Access Permissions: Student, Faculty

Both the faculty and the student can view the posts regarding the Tasks which include information like Title, Description, Category, and Faculty Name. Submission Deadline and Material File can also be viewed depending upon the category.

Sub Module: 4.3

Sub Module Name: Submit Task

Access Permissions: Student

The student can submit the assigned task posted by the faculty of respective subject. Student would be able to upload the file as the submission and also comment on it.

Module: 5

Module Name: FORUM

It contains all the notifications and updates with respect to all the Occasions within the University. They include updates on Assignments, Fests, Events, Holidays, and Guest Lecture etc. The sub modules present in it are

- 5.1 Create Notification
- 5.2 View Notification

Sub Module: 5.1

Sub Module Name: Create Notification

Faculty would be able to create notification depending upon the necessity. It is a mode of communication between them. The Faculty provides the details like Title, Description and select the category and submit that particular post.

Sub Module: 5.2

Sub Module Name: View Notification

All the posts regarding events, occasions, holidays, assignments etc. will be notified to the students as well as the faculty. Student/ Faculty can view them on the home page. It includes the Title and Description of that particular event.

CHAPTER 4

IMPLEMENTATION

4.1 Tools Used

Due the cross functional nature of service required by the application we have to use the by the services provided by some service providers. As the system functionalities completely rely on the data clusters from the servers we have chosen cloud to be the appropriate environment for the data parsing of the application. Developing the application for mobile environment poses various restrictions for the development. But certain alternative cognitions have been used to cover the tracks. The application is made for android environment. He3nce certain cognitive functionalities that the environment provides are highlighted in the application development that serve to be key features of Interface control.

The key tools Used in the Process of Application Development are:

Creately: For Designs



Creately is diagramming and design software operated by Cinergix, Pty Ltd. It is a cloud-based diagram tool built on Adobe's Flex/Flash technologies and provides a visual communication platform for virtual teams. It can be used to create info-graphics, flowcharts, Gantt charts, organizational charts, website wireframes, UML designs, mind maps, circuit board designs, doodle art and many other diagram types. It features a drag-and-drop WYSIWYG interface and easy to use collaboration capabilities. Their main product is a browser-based application with support for all modern web browsers like Firefox, Safari, Chrome and Opera browser. For people who prefer offline diagramming they provide a desktop software which runs on Microsoft Windows, Mac OS and Linux.

Parse: Backend (Servers and Database)



Parse is a cloud app platform that enables users to add a scalable and powerful backend to launch a full-featured app. Parse is the cloud app platform for Windows 8, Windows Phone 8, iOS, Android, JavaScript, and OS X. With Parse, you can add a scalable and powerful backend in minutes and launch a full-featured mobile or web app in record time without ever worrying about server management. Parse offers push notifications, social integration, data storage, and the ability to add rich custom logic to your app's backend with Cloud Code

Android Studio (Front End Integration)



Android Studio is an integrated development environment (IDE) for developing for the Android platform. It was announced on May 16, 2013 at the Google I/O conference by Google's Product

Manager, Katherine Chou. Android Studio is freely available under the Apache License 2.0. Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Based on Jet Brains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

4.2 Methodology

4.2.1 Modules and Design algorithms

Module: 1

Module Name: PROFILE

Profile is the module which consists of the personal information about the student and the faculty depending on their access/ login. The sub modules present in profile are

- View My Profile
- Edit My Profile
- Change Password
- Delete My Account

Sub Module: 1.1

Sub Module Name: View My Profile

Sub Module: 1.2

Sub Module Name: Edit My Profile

Sub Module: 1.3

Sub Module Name: Change Password

Sub Module: 1.4

Sub Module Name: Delete My Account

Algorithm:

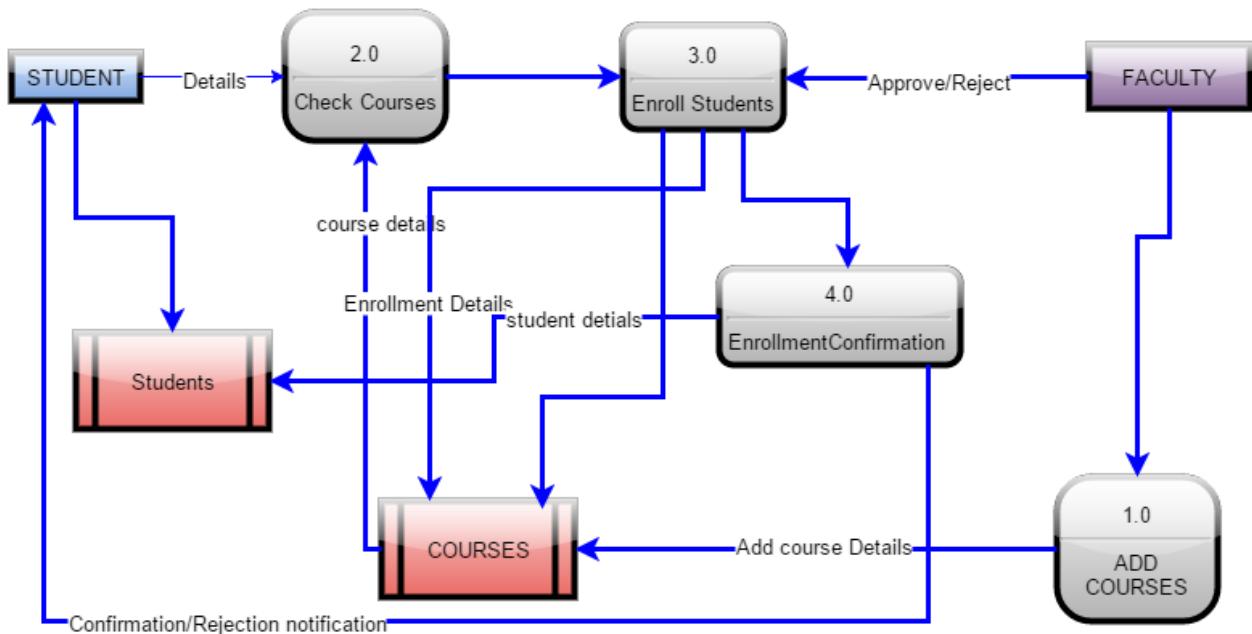
1. The application returns users UID key for data retrieval from users table
2. Once the UID key is proper permissions and access retrieval starts
3. The access retrieval permits users to submit the data or retrieve the data

4. UID collapses after the process

Module: 2

Module Name: COURSES

- Create Course
- Register for a New Course / Join Course
- My Courses
- Unregister the Course
- Delete the Course



4.2.1.1 Data Flow - Course Module

Algorithm:

1. Initiate the process by returning UID
2. The UID checks for permission access
3. Depending on permission of roles the Server Allows data retrieval
4. Submitting data prompts UID to be submitted
5. UID key dumped when logout

Module: 3

Module Name: AUTHENTICATION

- 3.1 Registration
- 3.2 Login

Sub Module: 3.1

Sub Module Name: Registration

Sub Module: 3.2

Sub Module Name: Login.

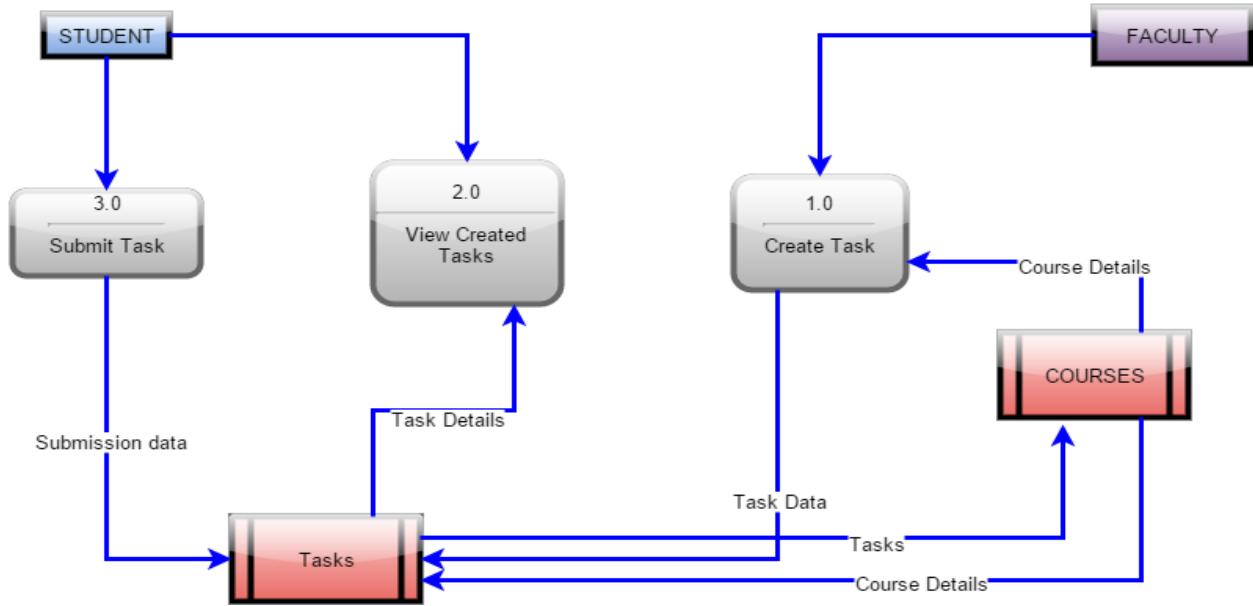
Algorithm:

1. Enter Authentication credentials
2. The application communicates with the server
3. If credentials are true → server returns UID
4. UID serves as key for role based authentication system
5. Provide UID for retrieving data

Module: 4

Module Name: TASKS

- 4.1 Create Task
- 4.2 View Task
- 4.3 Submit Task



4.2.1.2 Task Design Flow

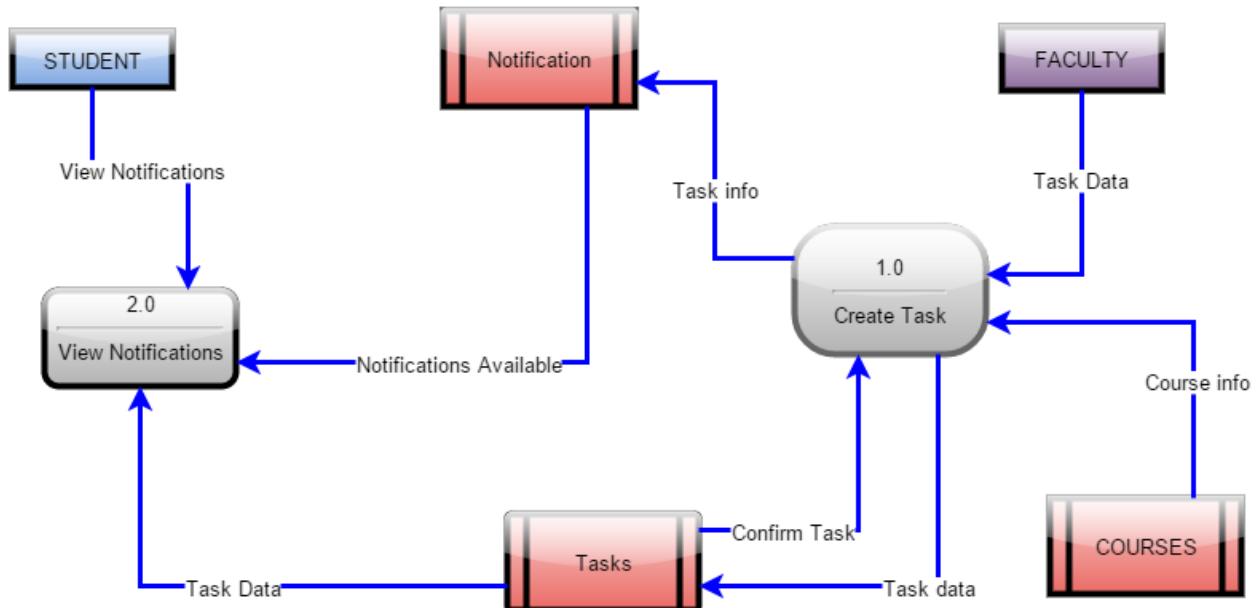
Algorithm:

1. 1. User returns the UID key to server
2. The server authenticates the UID key
3. Session variable initiate
4. User gets access permission for data retrieval
5. Retrieved data can be updated by submitting with UID key
6. UID key collapses after the retrieval

Module: 5

Module Name: FORUM

- 5.1 Create Notification
- 5.2 View Notification



4.2.1.3 Notification Data Flow

Algorithm:

1. User passes the UID key for notification retrieval
2. The table associated with the authentic UID key are retrieved
3. Data load into the columns of posts and notifications
4. UID collapses

4.2.2 Pseudo Code

The code for each module is represented in the form of activity. Hence for each module code has initiated parts

```
//Home Screen Loader
private CharSequence mTitle;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        getWindow().setNavigationBarColor(getResources().getColor(R.color.color_dark));
        getWindow().setStatusBarColor(getResources().getColor(R.color.color_dark));
    }
}

//Menu Initiator and Navigator
public void onNavigationDrawerItemSelected(int position) {
    Fragment selection = null ;
    if(position>2&&!studentFlag){
        position++;
    }
    switch(position){
        case 0 : selection = new homeFragment();
        mTitle="Home";
        break;
        case 1 : selection = new profileFragment();
        mTitle="Profile Settings";
        break;
        case 2 : selection = new coursesFragment();
        mTitle="Courses";
        break;
        case 3 : selection = new submissionsFragment();
        mTitle="Submission Posts";
        break;
        case 4 : selection = new forumFragment();
        mTitle="Forum";
        break;
        case 5 : selection = new settingsFragment();
```

```

        mTitle="App Settings";
        break;
    case 6 : selection = new supportFragment();
        mTitle="Dev Support";
        break;
    }
FragmentManager fragmentManager = getFragmentManager();
fragmentManager.beginTransaction()
.replace(R.id.container, selection)
.commit();
}

```

Home Fragment :

```

public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    final ListView lv1 = (ListView) getActivity().findViewById(R.id.listView2);
    final ListView lv2 = (ListView) getActivity().findViewById(R.id.listView3);
    final ArrayList<String> ann = new ArrayList<>(),events = new ArrayList<>();
    ParseQuery<ParseObject> query = ParseQuery.getQuery("Events");
    query.findInBackground(new FindCallback<ParseObject>() {
        @Override
        public void done(List<ParseObject> list, ParseException e) {
            if(list.size()>0){
                for(ParseObject po:list){
                    if(po.getString("type").equals("Ann"))
                        ann.add(po.getString("info"));
                    else events.add(po.getString("info"));
                }
                ArrayAdapter<String> adapter1 = new ArrayAdapter<>(getActivity(),
                android.R.layout.simple_list_item_1,ann);
                lv1.setAdapter(adapter1);
                ArrayAdapter<String> adapter2 = new ArrayAdapter<>(getActivity(),
                android.R.layout.simple_list_item_1,events);
                lv2.setAdapter(adapter2);
            }
        }
    });
}

```

Profile Fragment

```
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    //getActivity().setTitle("Profile Settings");
    List<String> options = new ArrayList<>();
    options.add("View My Profile");
    options.add("Edit My Profile");
    options.add("Change Password");
    options.add("Settings");
    options.add("Delete My Account");

    ListView lv = (ListView) getView().findViewById(R.id.listView4);
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(getActivity(),
    android.R.layout.simple_list_item_1,options);
    adapter.setNotifyOnChange(true);
    lv.setAdapter(adapter);

    lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            Intent intent;
            switch(position){
                case 0 :
                    intent = new Intent(getActivity(),profileActivity.class);
                    intent.putExtra("edit",false);
                    intent.putExtra("user",ParseUser.getCurrentUser().get("username").toString());
                    startActivity(intent);
                    break;
                case 1 :
                    intent = new Intent(getActivity(),profileActivity.class);
                    intent.putExtra("edit", true);
                    intent.putExtra("user",ParseUser.getCurrentUser().get("username").toString());
                    startActivity(intent);
                    break;
                case 2 :
                    intent = new Intent(getActivity(),change_passwordActivity.class);
                    startActivity(intent);
            }
        }
    });
}
```

```

        break;
    case 3 :
getActivity().getFragmentManager().beginTransaction().replace(R.id.container,new
settingsFragmet()).commit();
        break;
    case 4 :
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("Are You Sure ??");
        builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                try {ParseUser.getCurrentUser().delete();
                    ParseUser.logOut();
                    Intent in = new Intent(getActivity(),LogIn_SignUp.class);
                    startActivity(in);
                    getActivity().finish();
                } catch (ParseException e) {
                    e.printStackTrace();
                }
            }
        })
    
```

Submission Fragment :

```

public submissionsFragment() {
    // Required empty public constructor
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_submissions, container, false);
}

@Override
public void onActivityCreated(Bundle savedInstanceState) {

```

```

super.onActivityResult(savedInstanceState);
posts = new ArrayList<>();
final ListView lv = (ListView) getView().findViewById(R.id.listView8);
ParseQuery<ParseObject> query = ParseQuery.getQuery("Posts");
query.whereEqualTo("category", "Submission Request");
query.orderByDescending("createdAt");
query.include("author");
query.findInBackground(new FindCallback<ParseObject>() {
    @Override
    public void done(List<ParseObject> list, ParseException e) {
        postAdapter adapter = new postAdapter(getActivity(), R.layout.row_layout_post, list);
        adapter.setNotifyOnChange(true);
        posts = list;
        lv.setAdapter(adapter);
    }
});
lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Intent intent = new Intent(getActivity(), postActivity.class);
        intent.putExtra("id", posts.get(position).getInt("post_id"));
        intent.putExtra("courseCode", 0);
        startActivity(intent);
    }
});
}
}

```

Forum Fragment

```

public forumFragment() {
    // Required empty public constructor
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

}

```

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_forum, container, false);
}

@Override
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    posts = new ArrayList<>();
    final ListView lv = (ListView) getView().findViewById(R.id.listView8);
    ParseQuery<ParseObject> query = ParseQuery.getQuery("Posts");
    query.whereEqualTo("category", "Forum Thread");
    query.orderByDescending("createdAt");
    query.include("author");
    query.findInBackground(new FindCallback<ParseObject>() {
        @Override
        public void done(List<ParseObject> list, ParseException e) {
            postAdapter adapter = new postAdapter(getActivity(), R.layout.row_layout_post, list);
            adapter.setNotifyOnChange(true);
            posts = list;
            lv.setAdapter(adapter);
        }
    });
    lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
            Intent intent = new Intent(getActivity(), postActivity.class);
            intent.putExtra("id", posts.get(position).getInt("post_id"));
            intent.putExtra("courseCode", 0);
            startActivity(intent);
        }
    });
}

```

COURSE Activity

```
protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_courses);
final ActionBar actionBar = getSupportActionBar();
actionBar.setDisplayShowHomeEnabled(true);
actionBar.setIcon(R.mipmap.ic_launcher);
if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    getWindow().setNavigationBarColor(getResources().getColor(R.color.color_dark));
    getWindow().setStatusBarColor(getResources().getColor(R.color.color_dark));
}
findViewById(R.id.pager).setBackgroundColor(getResources().getColor(R.color.color_light));
// Set up the action bar.
actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

// Create the adapter that will return a fragment for each of the three
// primary sections of the activity.
mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());

// Set up the ViewPager with the sections adapter.
mViewPager = (ViewPager) findViewById(R.id.pager);
mViewPager.setAdapter(mSectionsPagerAdapter);

// When swiping between different sections, select the corresponding
// tab. We can also use ActionBar.Tab#select() to do this if we have
// a reference to the Tab.
mViewPager.setOnPageChangeListener(new ViewPager.SimpleOnPageChangeListener() {
    @Override
    public void onPageSelected(int position) {
        actionBar.setSelectedNavigationItem(position);
    }
});

// For each of the sections in the app, add a tab to the action bar.
for (int i = 0; i < mSectionsPagerAdapter.getCount(); i++) {
    // Create a tab with text corresponding to the page title defined by
    // the adapter. Also specify this Activity object, which implements
    // the TabListener interface, as the callback (listener) for when
    // this tab is selected.
    actionBar.addTab(
        actionBar.newTab()
}

```

```

        .setText(mSectionsPagerAdapter.getPageTitle(i))
        .setTabListener(this));
    }
}

@Override
public void onTabSelected(ActionBar.Tab tab, FragmentTransaction fragmentTransaction) {
    // When the given tab is selected, switch to the corresponding page in
    // the ViewPager.
    mViewPager.setCurrentItem(tab.getPosition());
}

@Override
public void onTabUnselected(ActionBar.Tab tab, FragmentTransaction fragmentTransaction)
{
}

@Override
public void onTabReselected(ActionBar.Tab tab, FragmentTransaction fragmentTransaction)
{
}

/**
 * A {@link FragmentPagerAdapter} that returns a fragment corresponding to
 * one of the sections/tabs/pages.
 */
public class SectionsPagerAdapter extends FragmentPagerAdapter {

    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        Fragment selection = null ;
        switch(position){
            case 0 : selection = new
courseView_Fragment(getIntent().getIntExtra("courseCode",0));
            break;
            case 1 : selection = new posts_Fragment(getIntent().getIntExtra("courseCode",0));
        }
        return selection;
    }
}

```

```
        break;
    case 2 : selection = new
participants_Fragment(getIntent().getIntExtra("courseCode",0));
        break;
    }
    return selection;
}

@Override
public int getCount() {
    // Show 3 total pages.
    return 3;
}

@Override
public CharSequence getPageTitle(int position) {
    Locale l = Locale.getDefault();
    switch (position) {
        case 0:
            return getString(R.string.title_1).toUpperCase(l);
        case 1:
            return getString(R.string.title_2).toUpperCase(l);
        case 2:
            return getString(R.string.title_3).toUpperCase(l);
    }
    return null;
}
}
```

4.3 Implementation

The GUI (Graphical User Interface) is cardinal android interface as the current support is only for android application. The application features a light weight UI (User Interface) unlike the other existing applications like Moodle. The user interface typically accommodates principle interfaces embedded within the system. It does not rule out the scope for further enhancements or developments. The Menu and other dashboard features respond to accommodate any changes that need to be taken care of. The GUI consists of standard android components of interface.

The application user simple interface to communicate with the user. This includes simple menu options and interface elements. The whole GUI component consists of very few interface elements. The objects returned by parse servers are handled by the interface elements so as to create an easy environment for the user. We have kept in our efforts to make the User interface as simple as possible.

The Application icon is used to represent the study with a symbol of university study or graduation.



4.3.1 UNICON-Icon

On clicking the icon the application gets connected to the Parse servers. Continuous internet connection must be maintained as the Parse servers Provide MBaaS cloud service Uninterruped network connection must be maintained in order to retrieve or upload data to the Unicon clusters. Even though in case of interrupted connection the interface allows the user for a reconnection from time to time. There are no such session expirations as the application completely resides on the user machine. The application runs using a minimal memory footprint. The user interface is simply designed to meet the needs of the user. The application responds the interface elements in a simple manner that android accommodate the environment for interface elements.

After launching the application the splash screen that appears on the screen can be used to display the logo of the college fort additional customizations. For now we just left the space blank without any customizations.

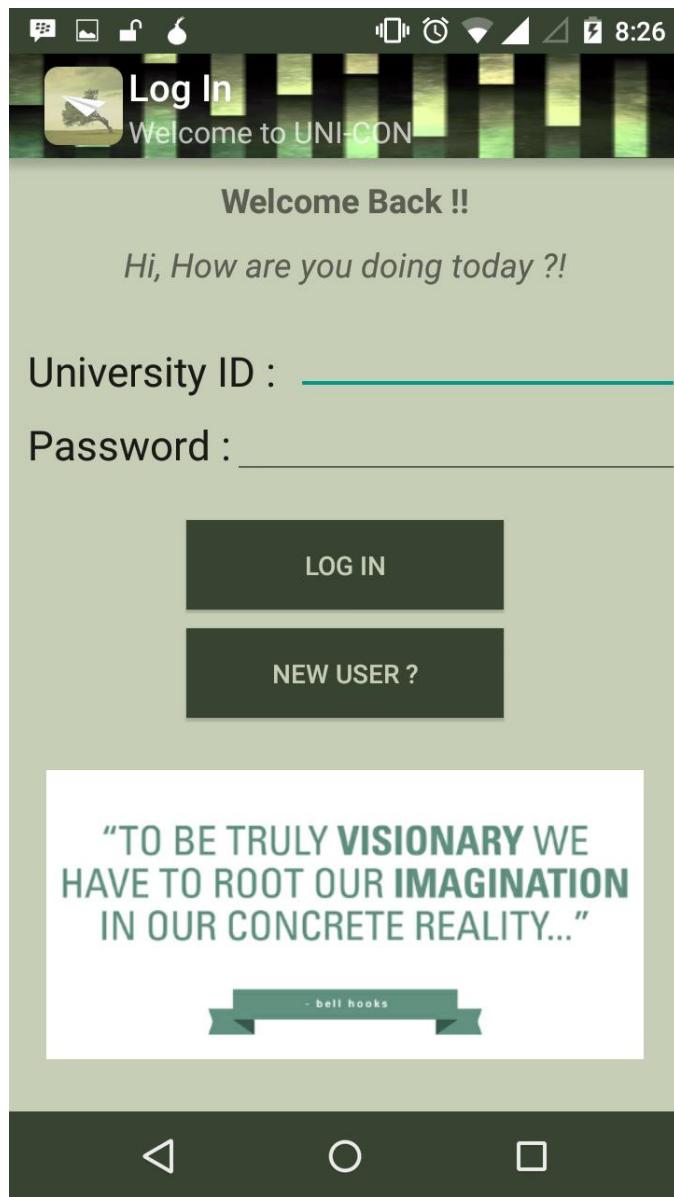


4.3.2 Splash Screen

This screen can be customized for the user interface according to the user community like school or college. This screen helps the interface as background connections like network establishment and connection with the servers take place. In order to conceal the background tasks or actions that gets triggered once the application starts weave used the splash screen functionality

The splash screen functionality is part of android interface and can be modified for the user's needs and this is the initial screen that users see.

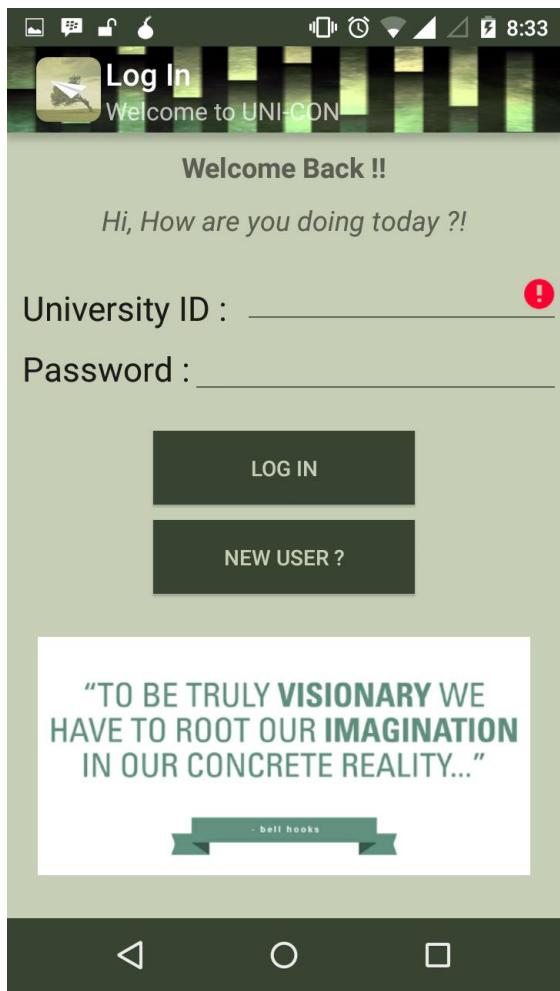
The next screen shows the Authentication module which consists of Login and Register options



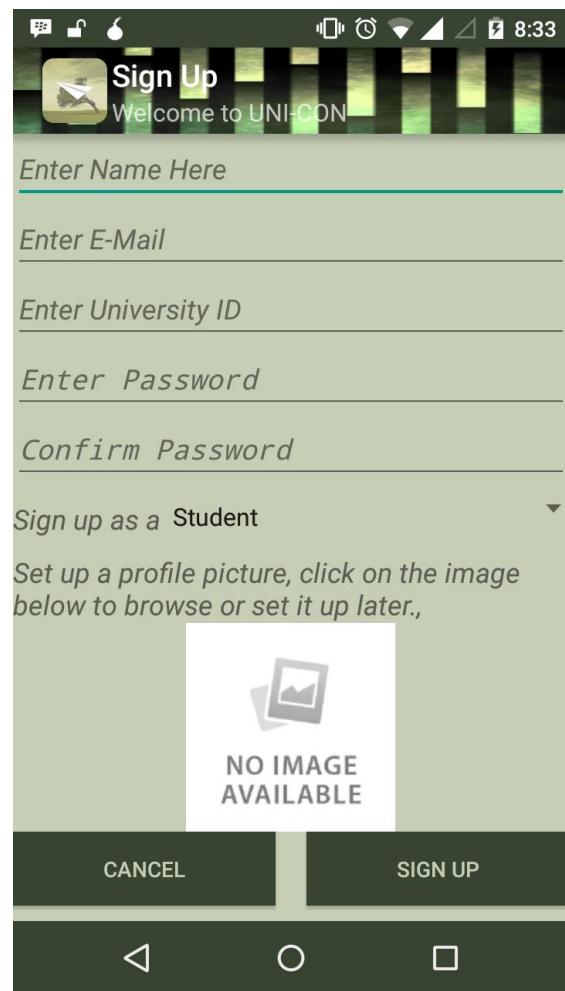
4.3.3 Authentication Screen

The authentication screen gives the user the options to login. This is role based login and the connection with the parse servers and UNICON clusters intiates only after the authentication module. The authentication module consists of two parts

- Registration
- Login



4.3.4 LOGIN

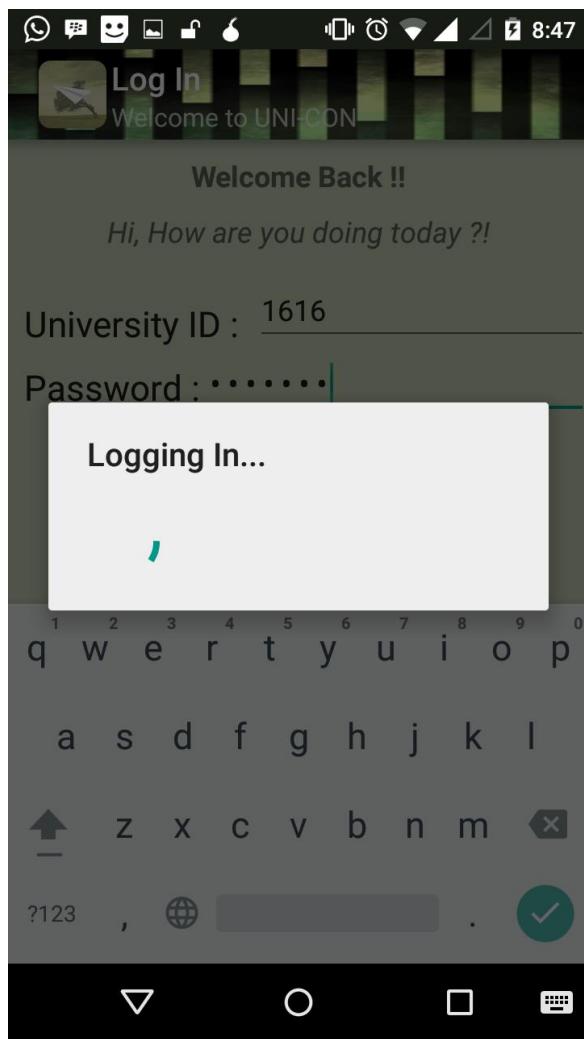


4.3.5 REGISTRATION

The login and registration are a part of authentication module. The User has to register himself if he is a first time user and had to provide his login credentials in the login screen to access the services.

For the registration Purpose the user has to enter valid mail address and University ID provided by the university. The same university will be login Id for the future communication and references. Once the user enters the credentials required he/she is registered user. The user can then login to avail the Service depending on their role.

Since it is a role based authentication system the user has to supply the input of role as a student or as a faculty. This is very important in the authentication process as the entire system is role based authentication system and the permission to access or rewrite the data depends on the permissions obtained by the user. Once the user registers and is logged in he will be directed to the Home screen or dash board screen.



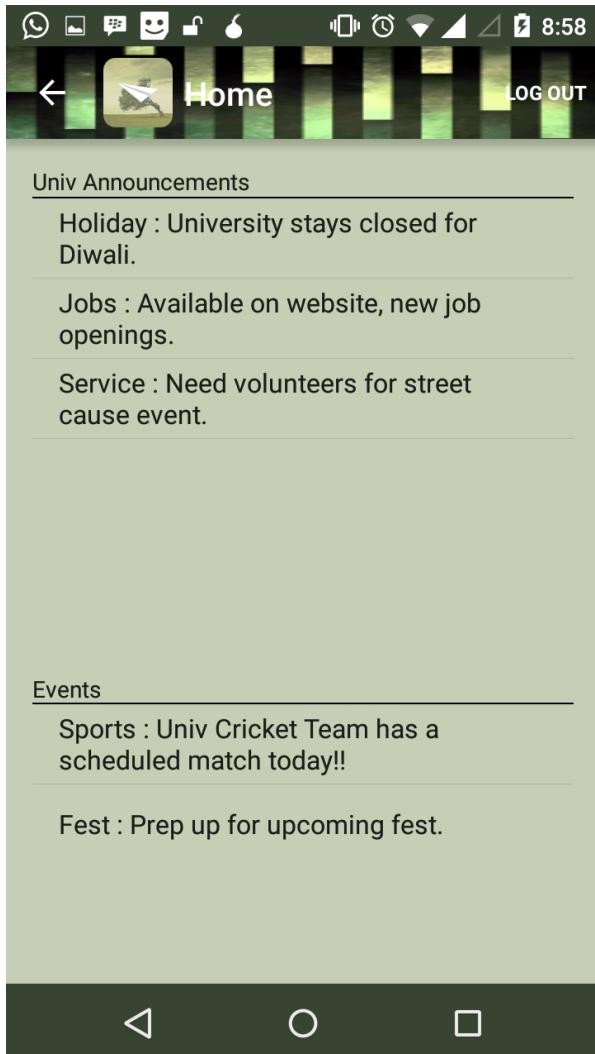
4.3.6 LOGGING IN Screen



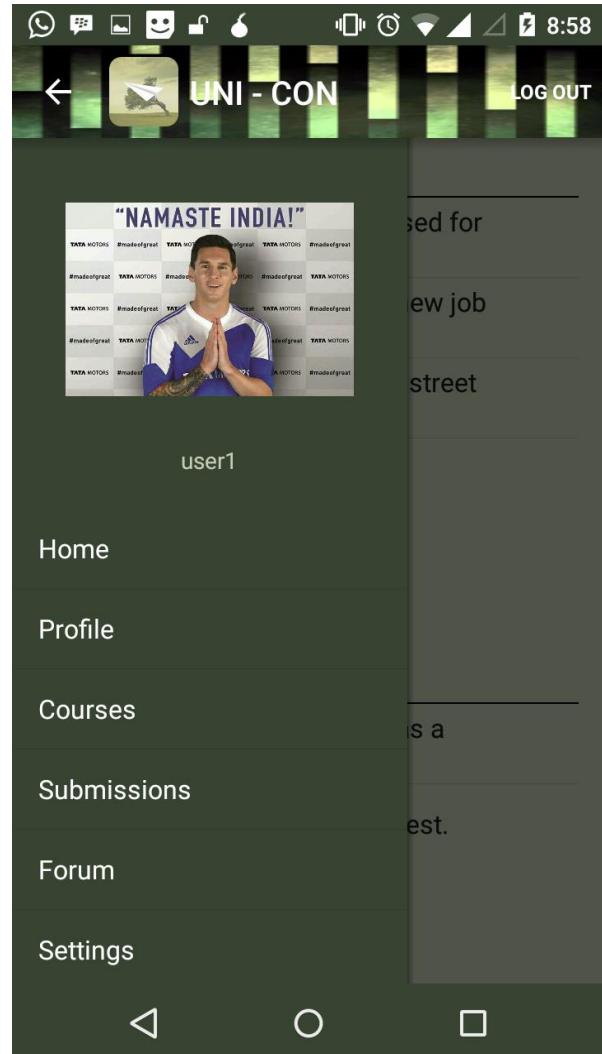
4.3.7 DASHBOARD Screen

Once the user Logs in he can avail all the services from the application as a legitimate user. The Dash board consists of Basic notifications, a Home button and a Logout Button. All the University announcements and the Event Notifications will be visible on the home screen. The Home button gives access to the Navigation Drawer or aside menu that allows users to perform various functions. The Screens of the functions will be initiated once the user chooses the option he wants

The navigation drawer enables user to easily get the menu and choose an option and then he can hide the Manu giving user the space for dashboard. The navigation drawer is a design element for interface provided by the android system



4.3.8 Home Screen

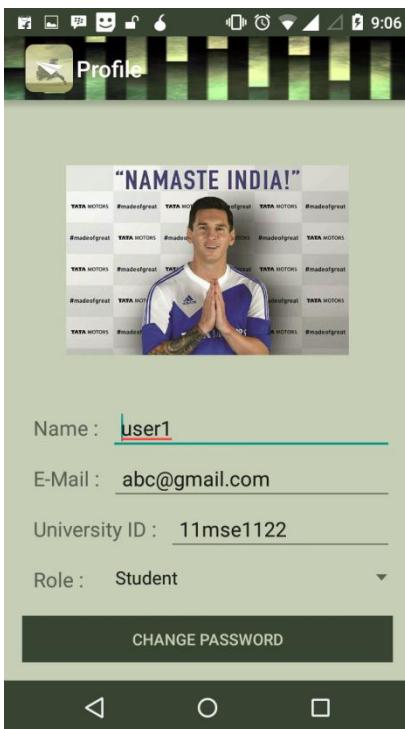
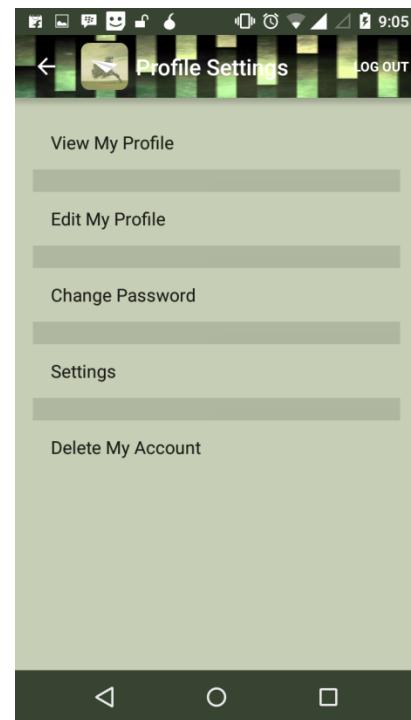
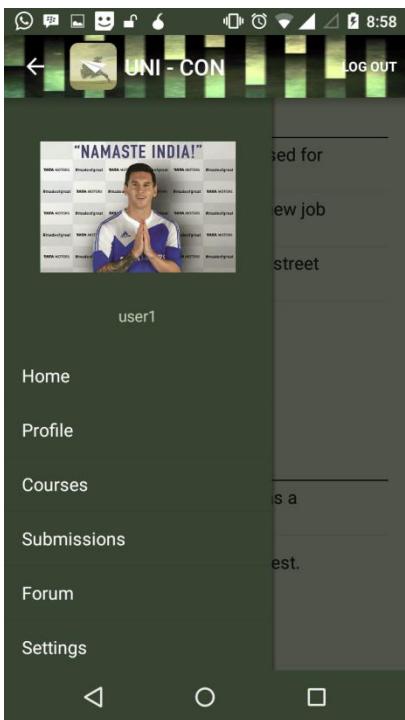


4.3.9 Navigation Menu

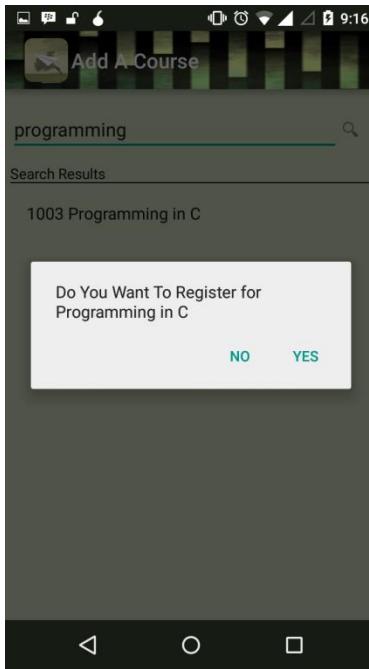
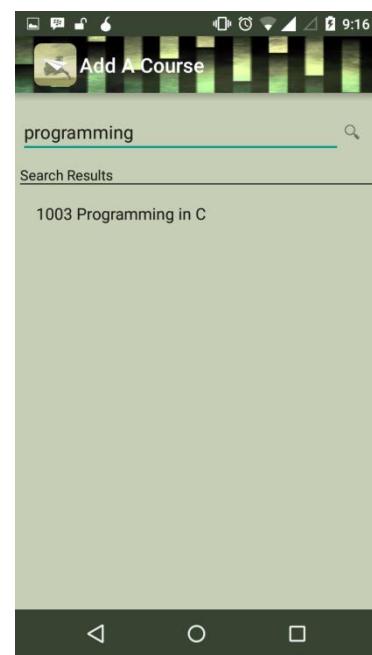
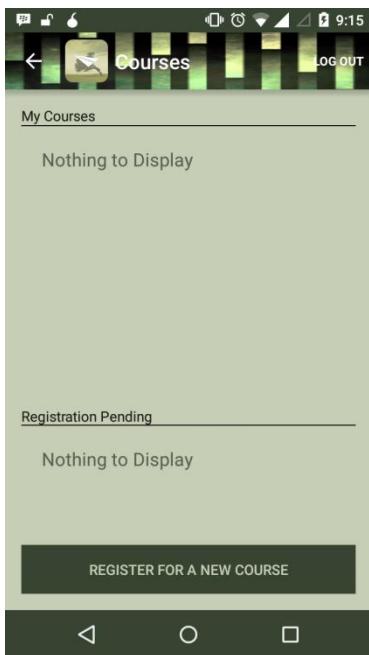
Depending on the tie of the user the authentication system, system provides options for the user. The options normally include the following screens

- Home
- Profile
- Courses
- Submissions
- Forum
- Settings
- App Support

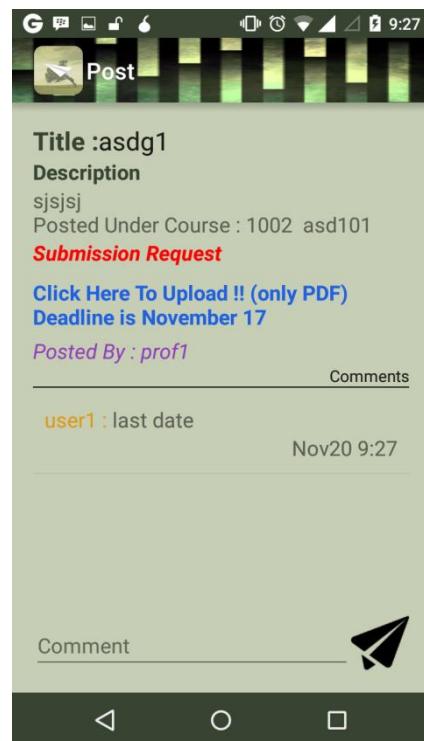
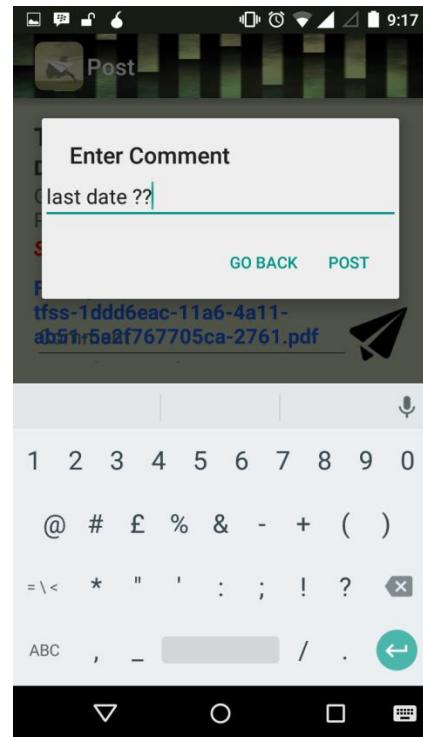
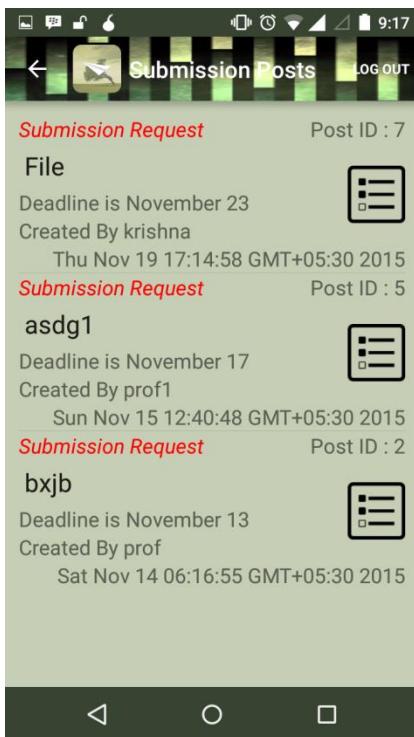
Screens for profile option:



Courses screens: (Sample for student)



Submissions: (Student Permissions)



4.4 Unit Test Cases

Project Name : UNICON								
Project Id	UC101							
Test Author:	Vamsi Krishna							
Date of Creation	9/11/2015							
Tester :	Vamsika							
Pre Condition : Install the UNICON apk File and uninurerrupted Net Connection								
Test Scenario: LOGIN MODULE								
Sl.No	Test Case Name	TestCase ID	Test Case Description	Step Num	Step Description	Expected Result	Actual Result	Pass/ Fail
1	Login	Login001	Validate UniversityID and Password	1	Enter UniversityID and password and click login button UserName:11MSE1027 Password: whamsy	Login should be successful	Login Successful	Pass
				2	Enter UniversityID and click login button UniversityID:11MSE1027	Login should not be successful	An error message "Enter password" should be displayed	Pass
				3	Enter Password and click login button Password:whamsy	Login should not be successful	An error message "Enter UniversityID" should be displayed	Pass
				4	Enter UniversityID and password and click login button UserName:11MSE1027 Password: asdf	Login should not be successful	An error message "Login Failed, Invalid UniversityID or password" should be displayed	Pass
Test Scenario: SIGNUP MODULE								
Sl.No	Test Case Name	TestCase ID	Test Case Description	Step Num	Step Description	Expected Result	Actual Result	Pass/ Fail
2	SignUp	SignUp001	Registration of a new user	1	Enter Name, Email, UniversityID, Password, Confirm Password, SignUp as, Upload Pic and Click on SignUp button.	SignUp should be successful	SignUp Successful	Pass
				2	Enter Name, Email, UniversityID, Password, Confirm Password, Upload Pic and Click on SignUp button.	SignUp should not be successfu	An error message "Fields Should not be empty" should be displayed	Pass
				3	Enter Name, Email, UniversityID, Password, SignUp as, Upload Pic and Click on SignUp button.	SignUp should not be successfu	An error message "Fields Should not be empty" should be displayed	Pass
				4	Enter Name, Email, UniversityID, Confirm Password, SignUp as, Upload Pic and Click on SignUp button.	SignUp should not be successfu	An error message "Fields Should not be empty" should be displayed	Pass
				5	Enter Name, Email, Password, Confirm Password, SignUp as, Upload Pic and Click on SignUp button.	SignUp should not be successfu	An error message "Fields Should not be empty" should be displayed	Pass
				6	Enter Name, UniversityID, Password, Confirm Password, SignUp as, Upload Pic and Click on SignUp button.	SignUp should not be successfu	An error message "Fields Should not be empty" should be displayed	Pass
				7	Enter Email, UniversityID, Password, Confirm Password, SignUp as, Upload Pic and Click on SignUp button.	SignUp should not be successful	An error message "Fields Should not be empty" should be displayed	Pass

Test Scenario: COURSE MODULE					
Sl.No	Test Case Name	TestCase ID	Test Case Description	Step Num	Step Description
3	Courses	Course001	Add a Course	1	Enter CourseName, Description, Class Time, Class Days, Class Address , Generate unique course code/num and click Create Course
				2	Enter CourseName, Description, Class Time, Class Days, Class Address , and click Create Course
				3	Enter CourseName, Description, Class Time, Class Days, Class Address , Generate unique course code/num and click Create Course Class Time: 11:00 To 11:00
				4	Enter Description, Class Time, Class Days, Class Address , and click Create Course
				5	Enter CourseName, Description, Class Time, Class Days, Generate unique course code/num and click Create Course
				6	Enter CourseName, Description, Class Time, Class Days, Class Address , Generate unique course code/num and click Create Course Class Time: 11:00 To 10:00
		Course002	Approval of course	1	Click on Add Student to course
		Course003	Register for a Course	1	Search for course using course name and click Yes to register Course: Requirements
				2	Search for course using course code Course code: 1003

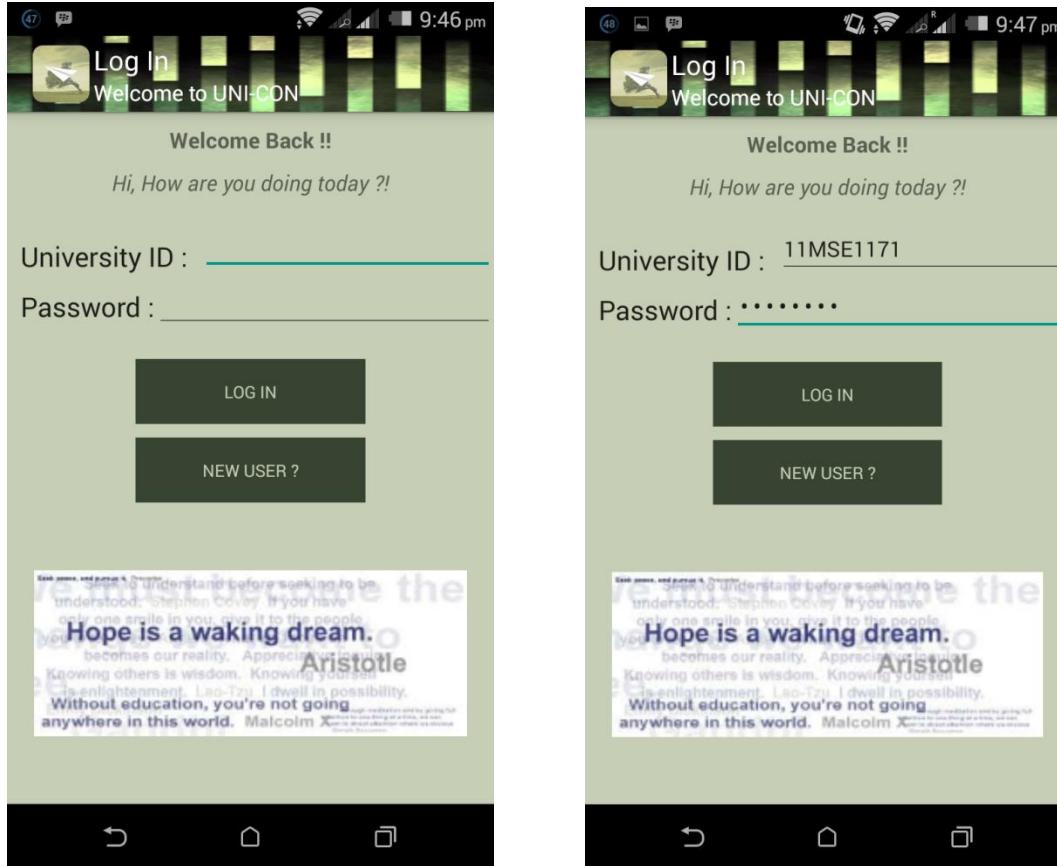
Test Scenario: PROFILE MODULE

Sl.No	Test Case Name	TestCase ID	Test Case Description	Step Num	Step Description
4	Profile	Profile001	Change Password	1	Enter old password, new password and confirm password and click save changes OldPassword: 12345 New password: whamsy confirm password: whamsy
				2	Enter old password, new password and confirm password and click save changes Old Password: 123 New password: whamsy confirm password: whamsy
				3	Enter old password, new password and confirm password and click save changes OldPassword: 12345 confirm password: whamsy
				4	Enter old password, new password and confirm password and click save changes OldPassword: 12345 New password: whamsy
		Profile002	Delete Account	1	Click on delete my account and select YES
				2	Click on delete my account and select NO
		Profile003	Edit Profile	1	Change existing Name and Click save Changes
				2	Change existing Email and Click save Changes

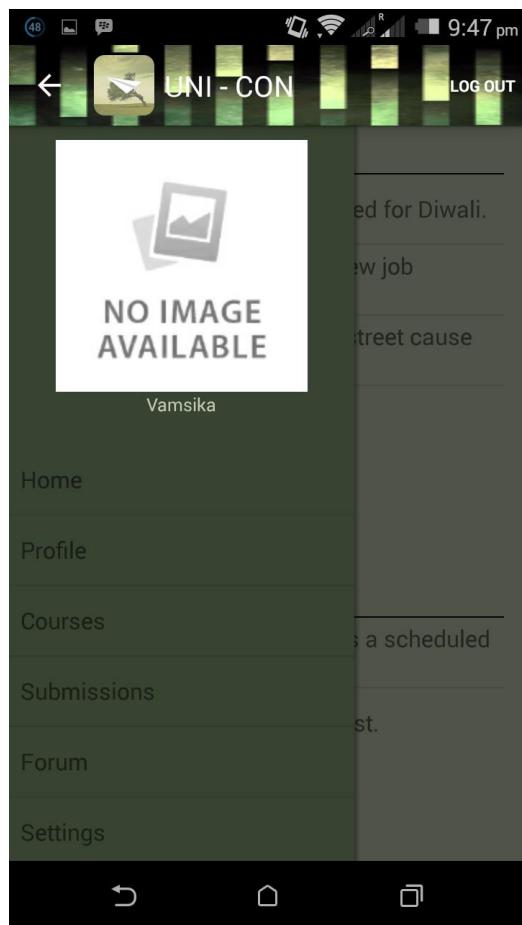
TestCase Scenario: TASK MODULE								
Sl.No	Test Case Name	TestCase ID	Test Case Description	Step Num	Step Description	Expected Result	Actual Result	Pass/ Fail
5	Task	Task001	Create Task	1	Enter Assignment Name,Time,Task ,Class and click on create Task	Task should be Created	Task Created	Pass
				2	Enter Time,Task ,Class and click on create Task	Task should not be Created	An error message "Filed's mustn't be empty" should be displayed	Pass
				3	Enter Assignment Name, Task ,Class and click on create Task	Task should not be Created	An error message "Filed's mustn't be empty" should be displayed	Pass
				4	Enter Assignment Name,Time,Class and click on create Task	Task should not be Created	An error message "Filed's mustn't be empty" should be displayed	Pass
				5	Enter Assignment Name,Time,Task and click on create Task	Task should not be Created	An error message "Filed's mustn't be empty" should be displayed	Pass
		Task002	Submit Task	1	Enter Assignment Name,upload file and submit	Task should be Submitted	Task Submitted	Pass
				1	Edit Assignment Name and Click on Save Changes	Task should be Modified	Task Modified	Pass
				2	Edit Time and Click on Save Changes	Task should be Modified	Task Modified	Pass
				3	Edit Task and Click on Save Changes	Task should be Modified	Task Modified	Pass
				4	Edit Class and Click on Save Changes	Task should be Modified	Task Modified	Pass
		Task004	View Task	1	Click on the task to view it	Task should be opened	Task Opened	Pass
Test Scenario: FORUM MODULE								
Sl.No	Test Case Name	TestCase ID	Test Case Description	Step Num	Step Description	Expected Result	Actual Result	Pass/ Fail
6	Forum	Forum001	View Notification	1	Click on the notification to view it	Notification should be opened	Notification viewed	Pass
				1	Enter Title,Description and select Category to create Notification	Notification should be created	Notification created	Pass
				2	Enter Title,Description to create	Notification should not be created	An error message "Filed's mustn't be empty" should be displayed	Pass
				3	Enter Title and select Category to create Notification	Notification should not be created	An error message "Filed's mustn't be empty" should be displayed	Pass
				4	Enter Description and select Category to create Notification	Notification should be created	Notification created	Pass

5.1 Results

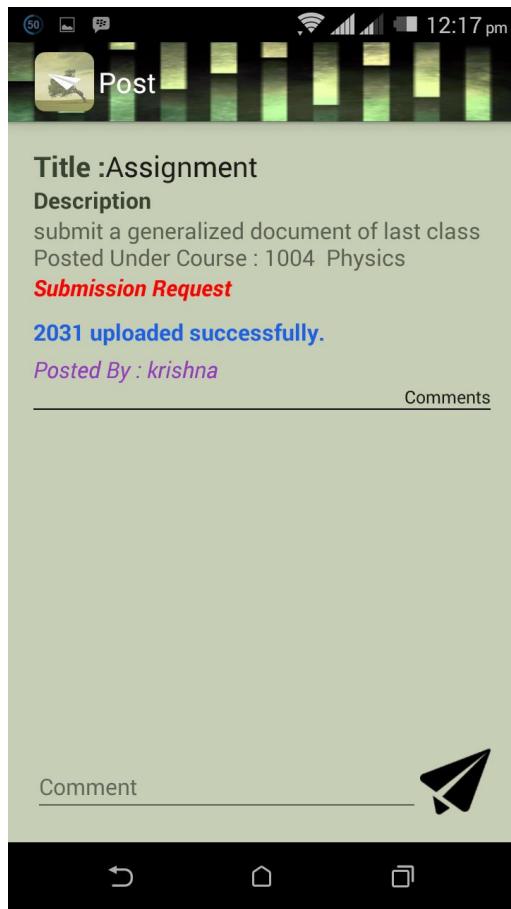
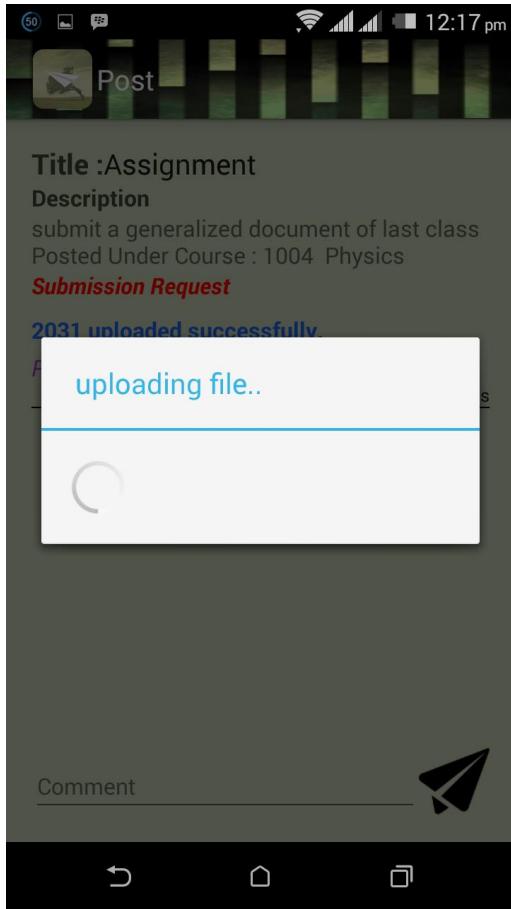
Student User



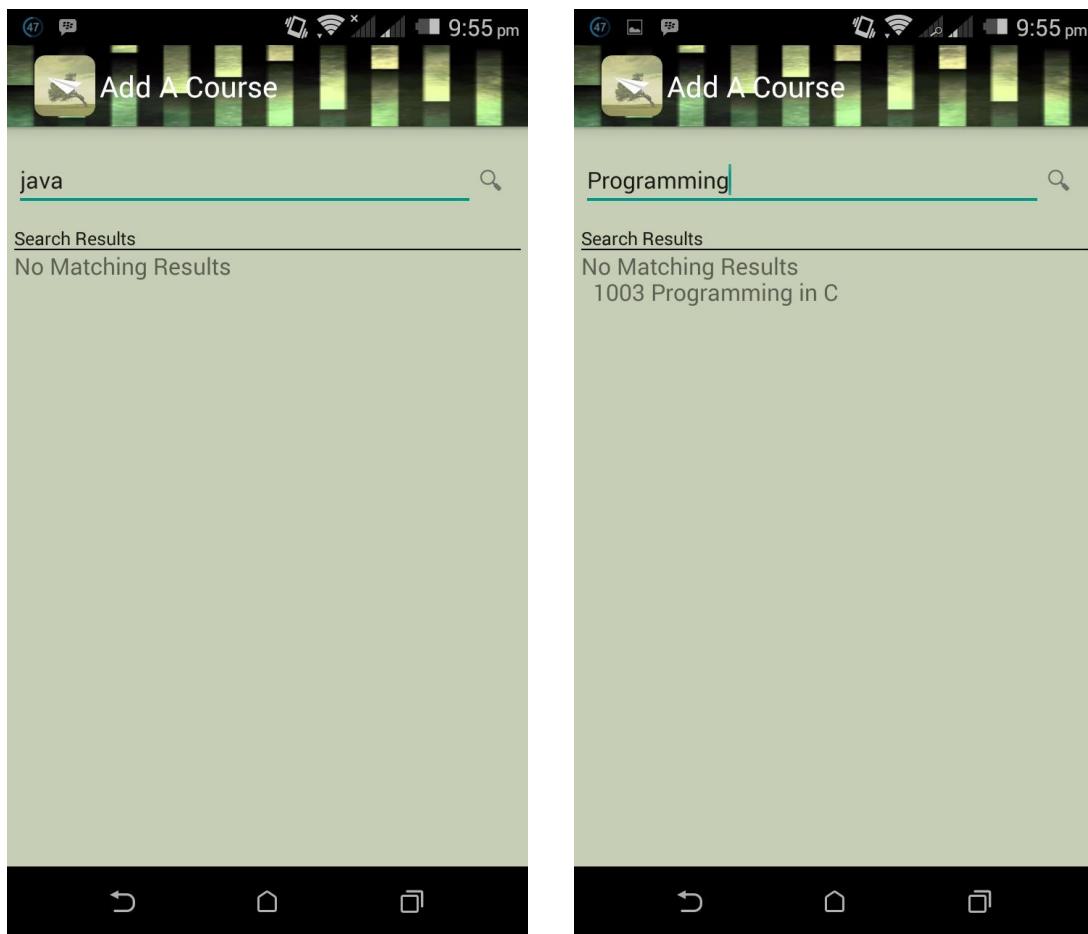
The login page consists of the information such as University Id and password. By entering the registered University Id and password correctly student will be able to login to the application.



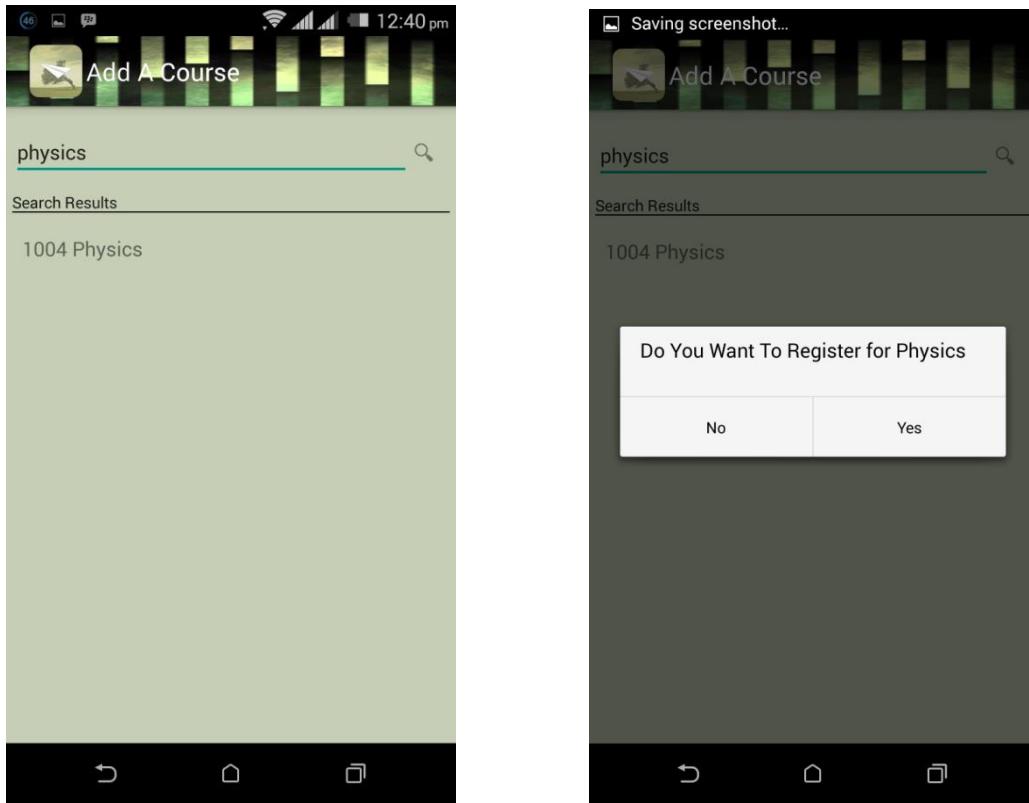
Once a student is successfully logged into the application he/she would be able to view the updates on home page and other modules will be displayed.



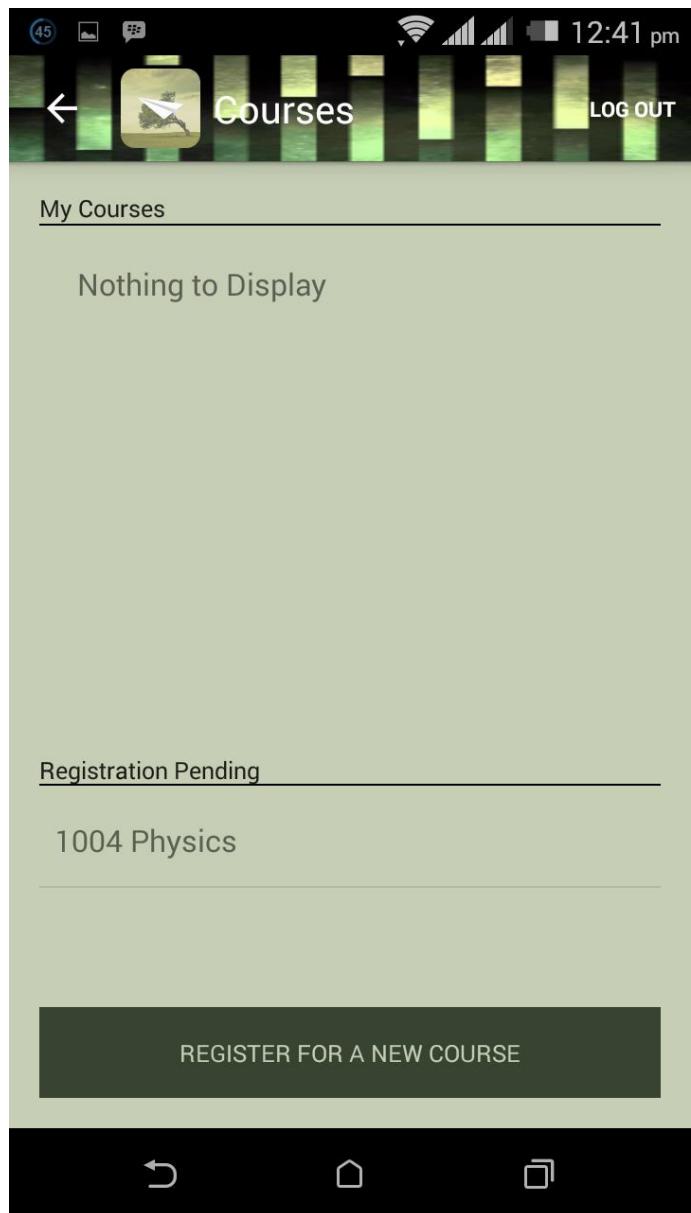
On clicking the upload link the student will be able to upload the file and it will be processed as shown in the image.



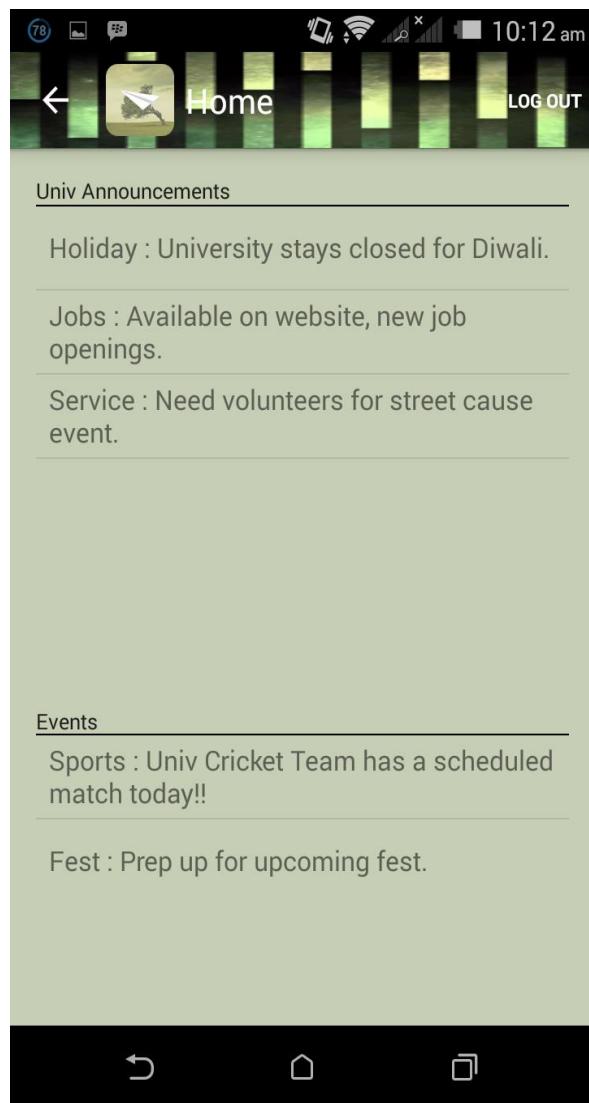
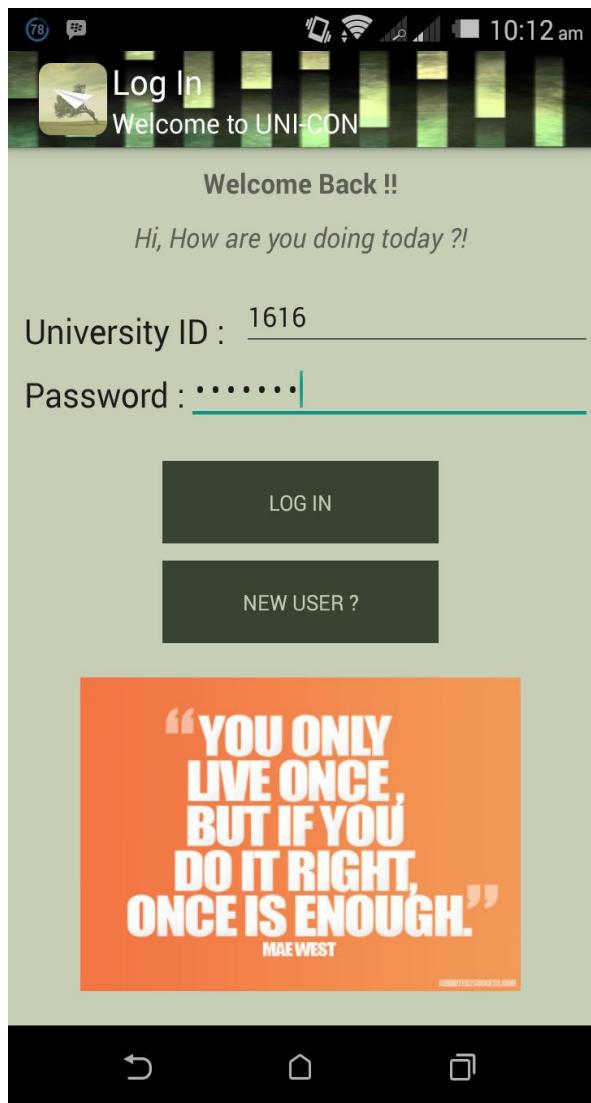
Course is the another module present in the student profile with which the students requests for a course and by getting the approval from the respective faculty ,he/she gets access to that course. As shown in the image he /she will search for the course.



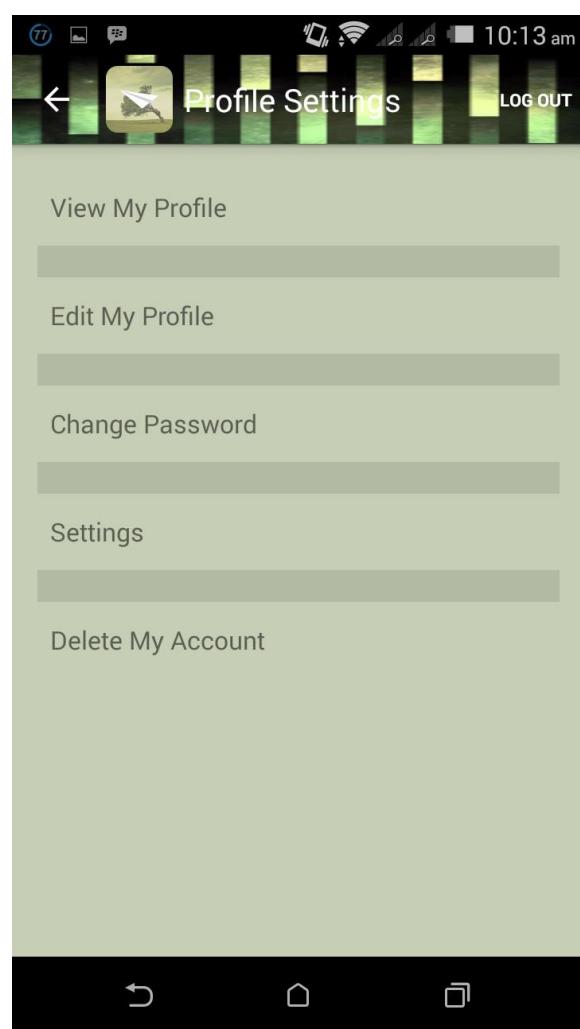
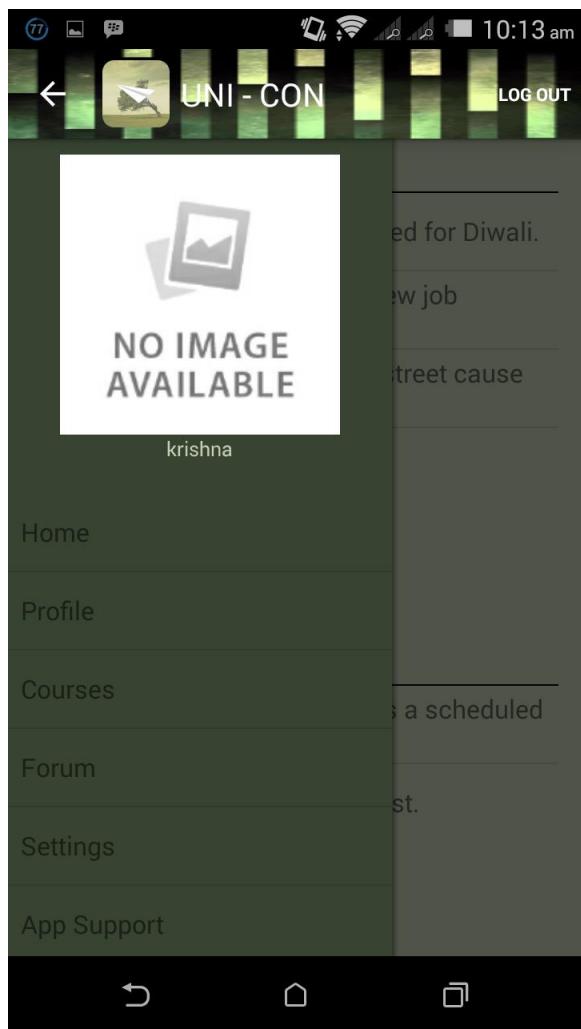
In this way the student search for the course and register themselves by selecting yes. Now the request will be sent to the particular faculty.



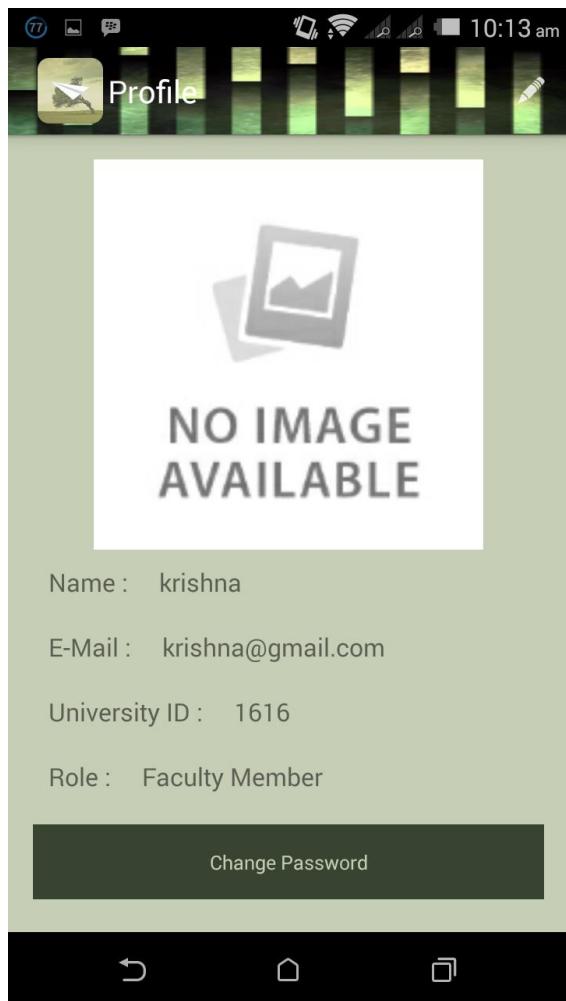
Till the faculty approves the request it will be under pending state. Once it is approved it will be in the list of registered courses.



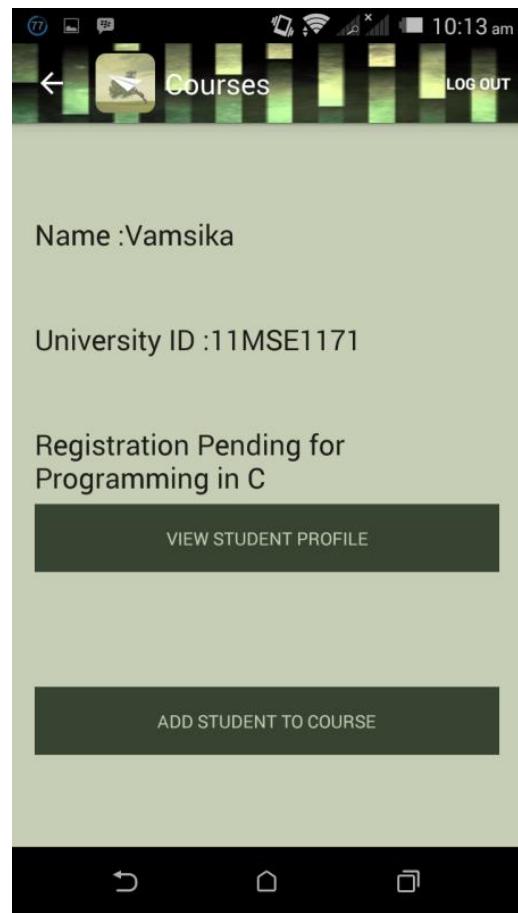
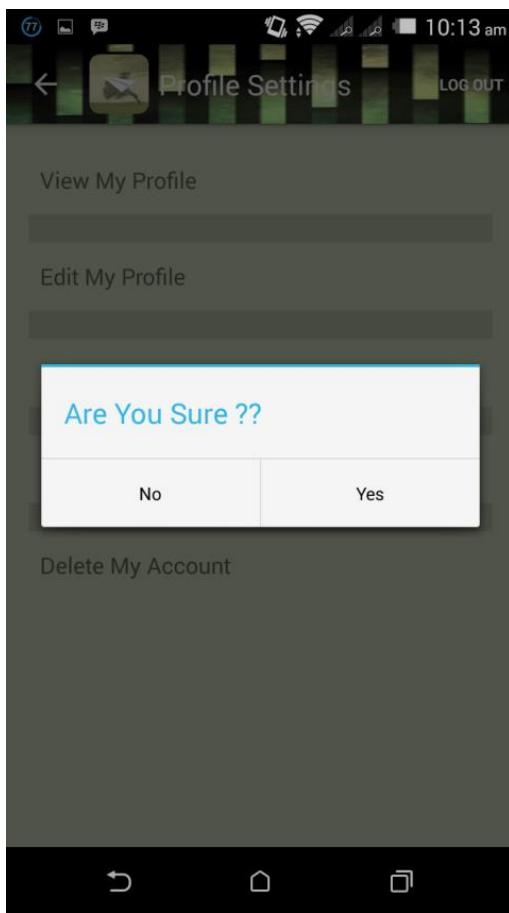
The Faculty will be able to login into the application with the registered credentials. After the successful login he/she will be directed to the home page this displays the university announcements, Events and other Updates.



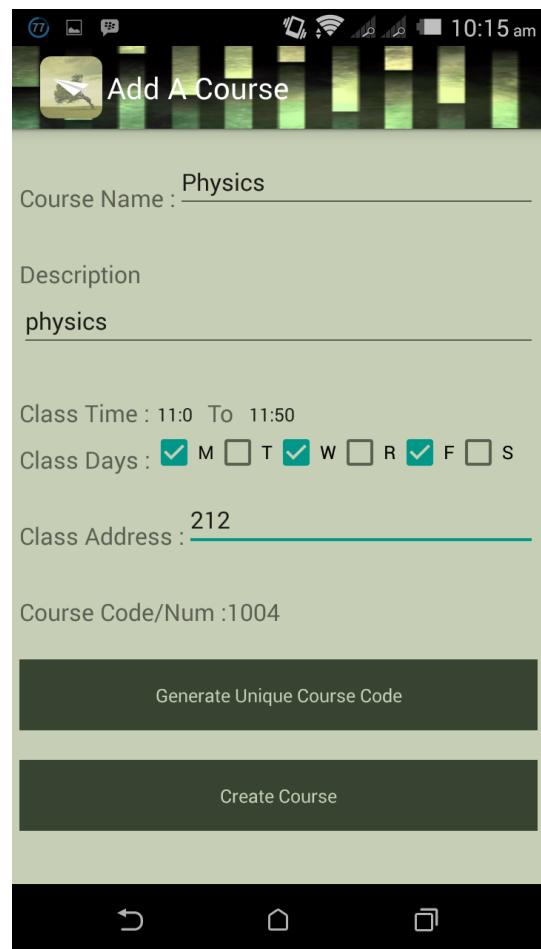
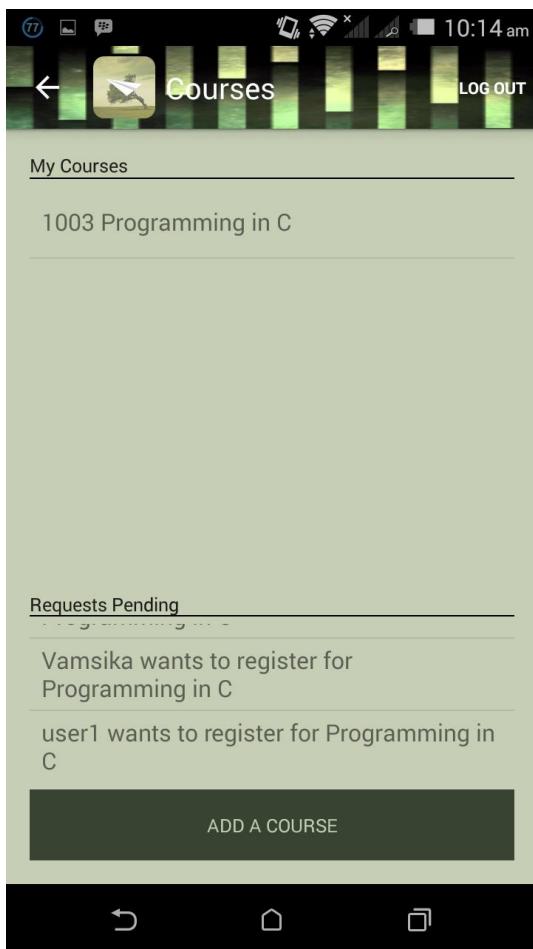
It displays the overview of the profile and the modules present in it. Profile Module consists of the sub modules shown above. By selecting each of the sub modules the faculty would be able to view the functionality



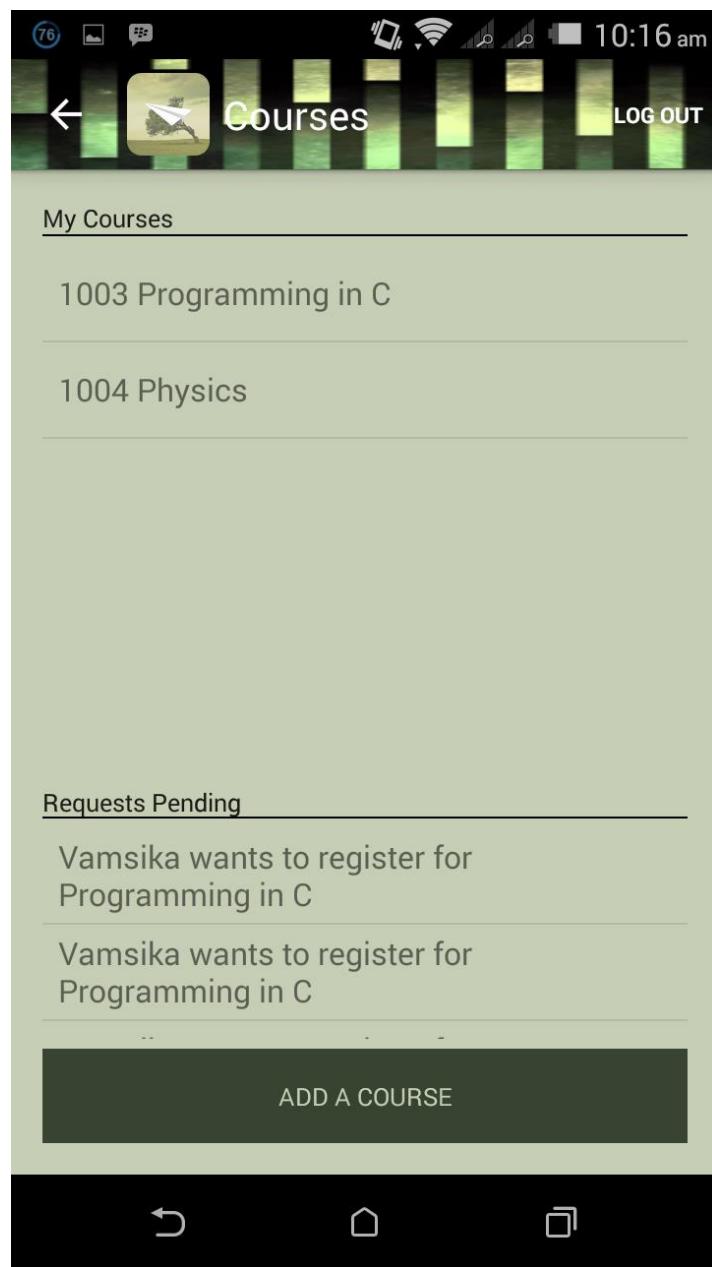
These are the other sub modules of the Profile module. Faculty can view the profile and update his/her password for the existing ones. By entering the old password, new password and confirming the new password the faculty can update their password. They can also edit their profile by selecting edit profile option



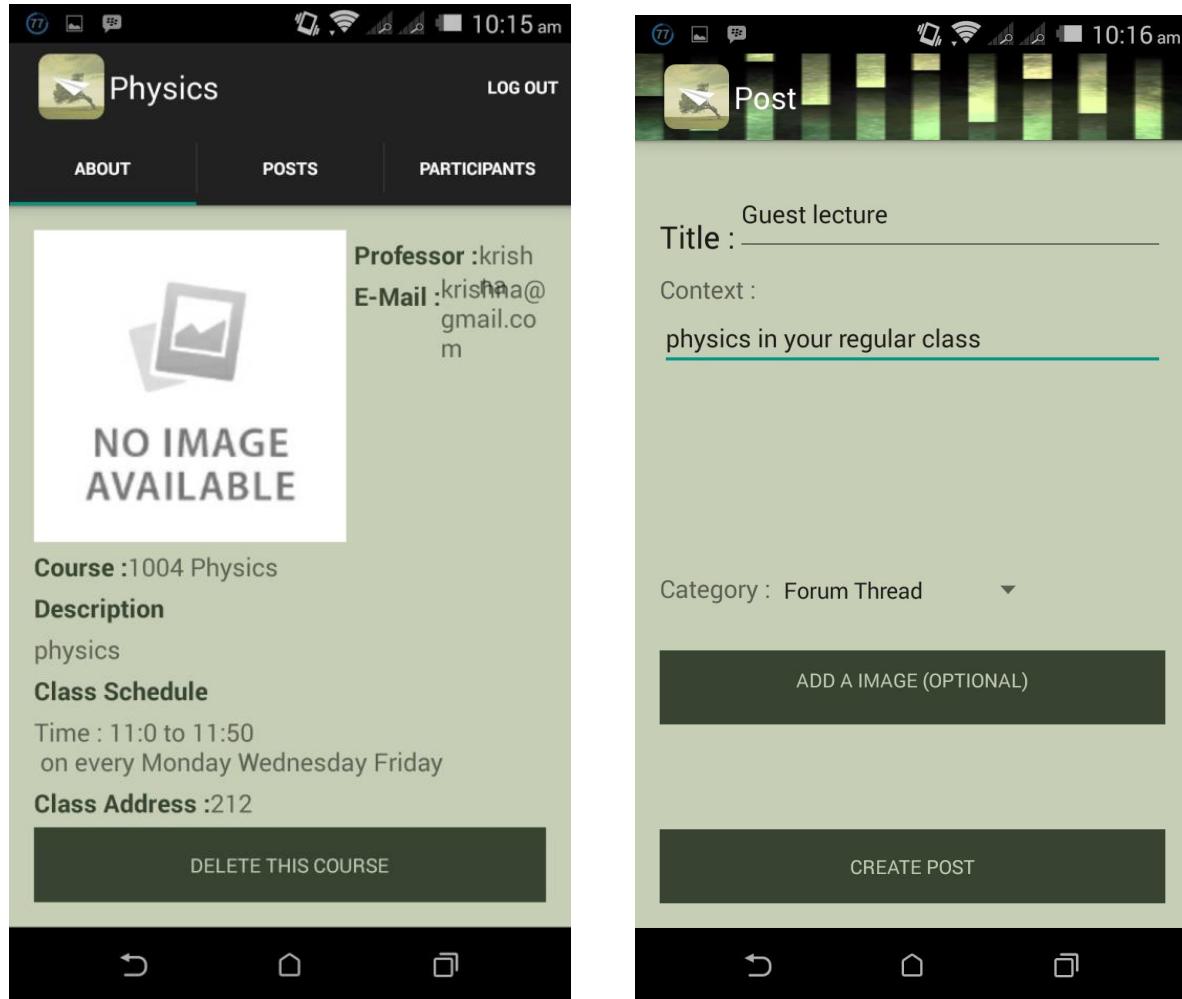
Faculty can delete their profile by clicking the delete my account option. Course is another sub module. When a course request is sent by the student to the professor the professor can view the student follows which displays the details of student.



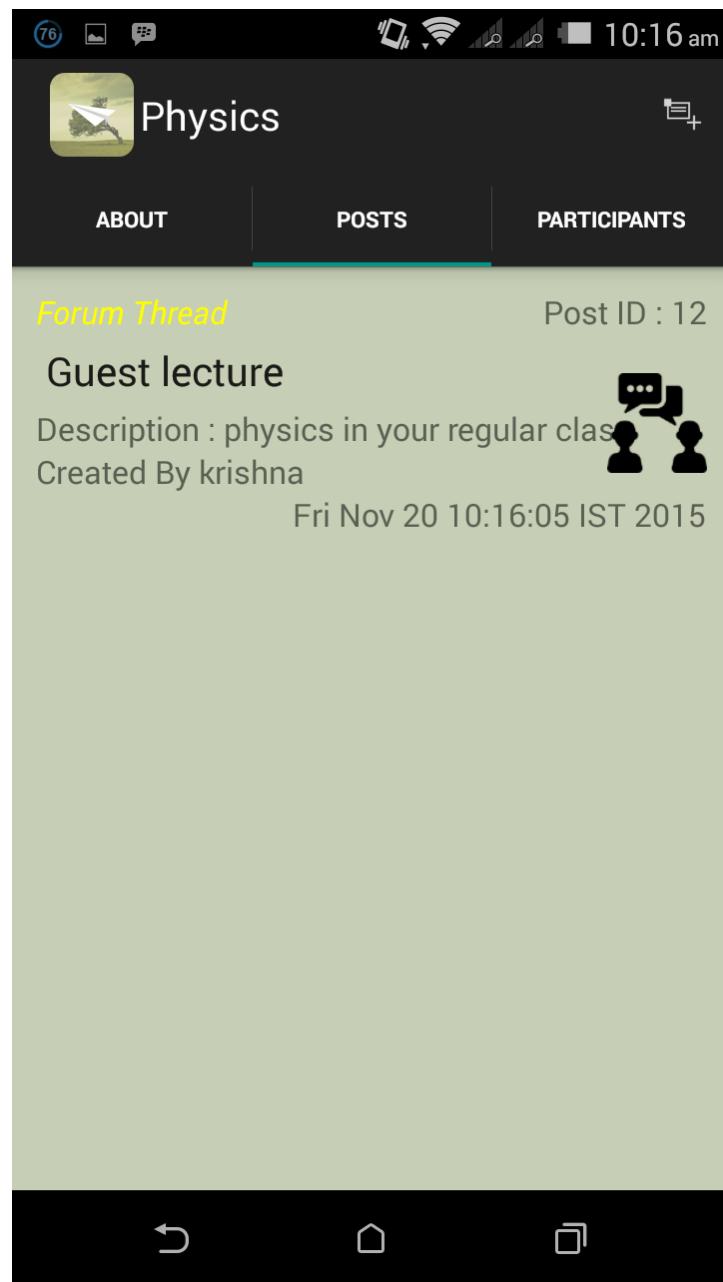
The list of courses created by the faculty and the pending requests sent by the students will be displayed here. Student can add a course by filling the form with the details shown in the picture.



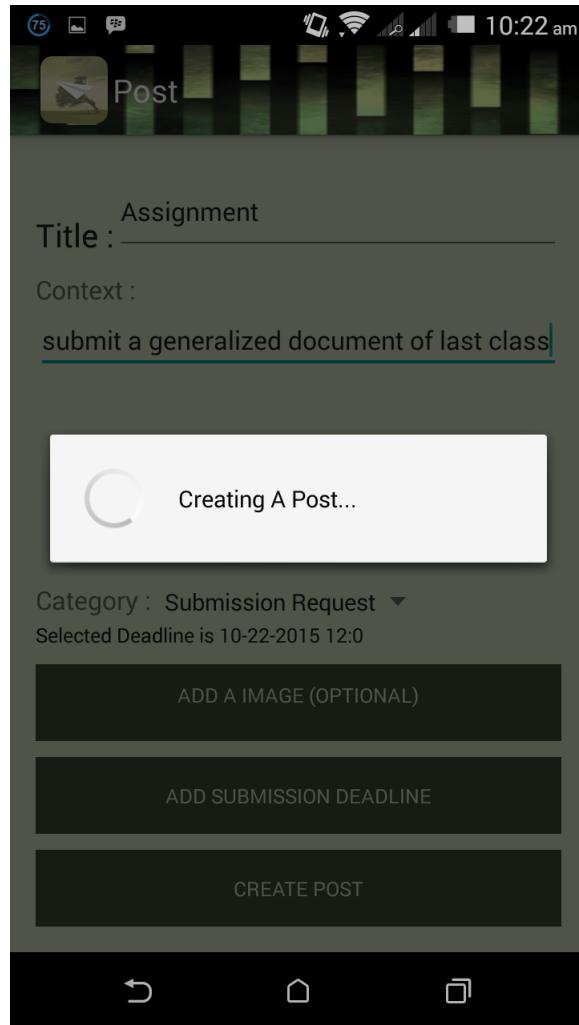
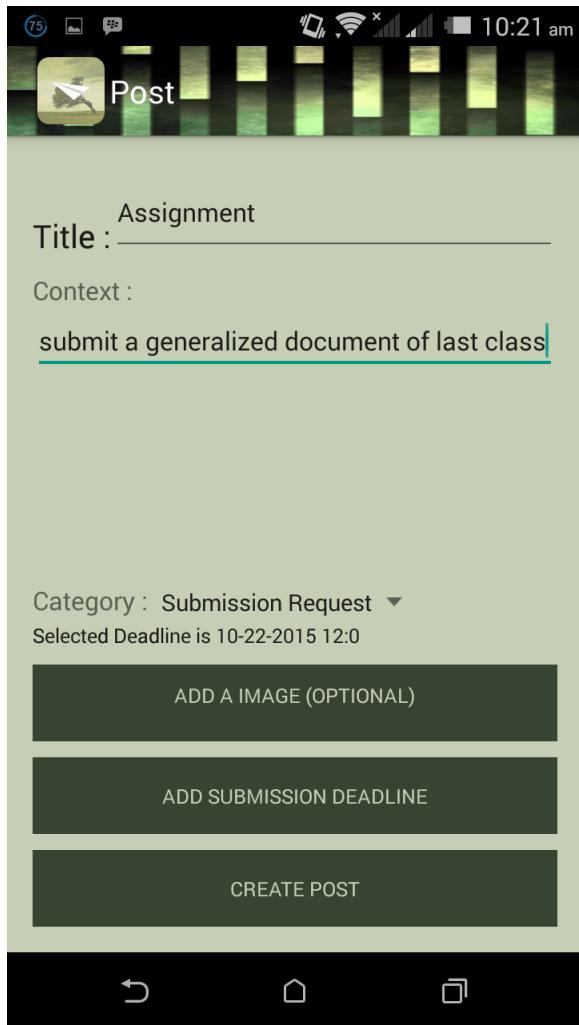
In this way the faculty can add a course .



The faculty can post various updates regarding class or any other events by selecting the course and creating the post in that particular subject. In the above image the faculty is informing about the guest lecture and the category is forum thread.



A forum thread is created as above with which all the students registered under that subject can view the post.

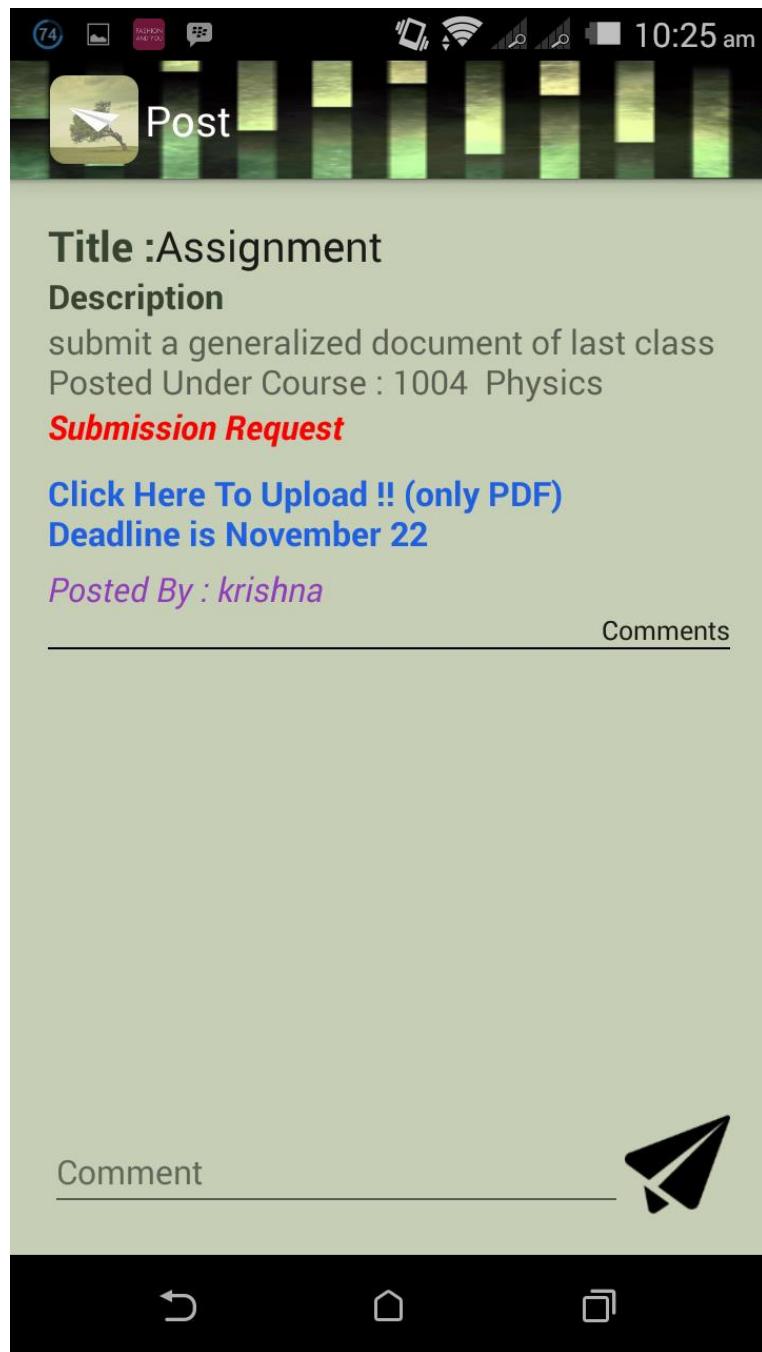


Other Category in creating a post is Submission request. It provides a link for the students to upload their file. It will be created as shown in the images.

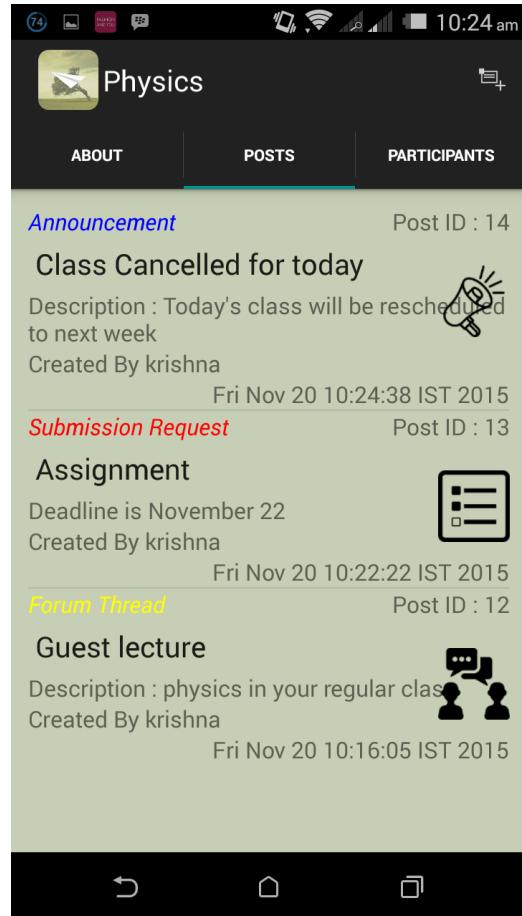
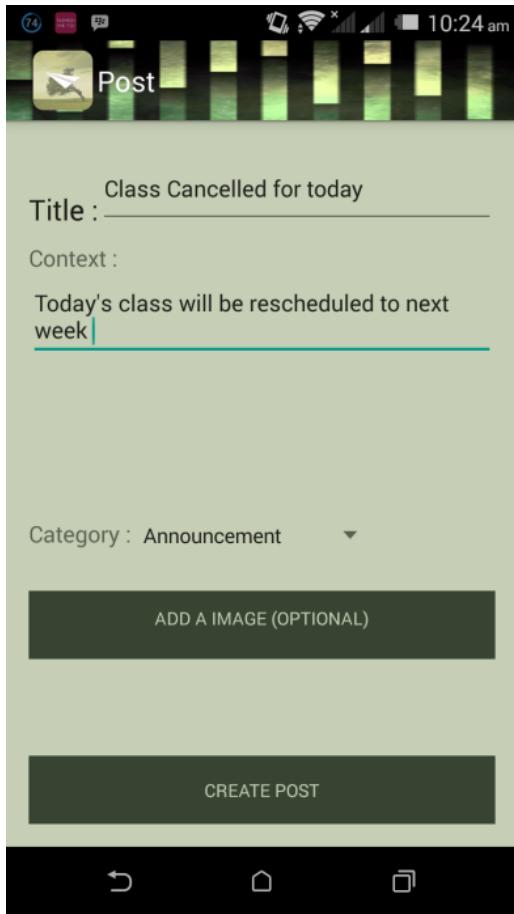
The screenshot shows a BlackBerry smartphone interface with a Facebook group feed for 'Physics'. At the top, there are standard status icons for battery level, signal strength, and time (10:22 am). Below the header, the group name 'Physics' is displayed next to a profile picture of a tree. A '+' icon is located in the top right corner of the header.

The main content area has three tabs: 'ABOUT', 'POSTS', and 'PARTICIPANTS'. The 'POSTS' tab is currently selected, indicated by a green underline. The first post is titled 'Submission Request' in red text, with 'Post ID : 13' in grey text to its right. It contains the text 'Assignment' and 'Deadline is November 22'. To the right of this text is a square icon containing three horizontal lines. Below this is the creation date 'Fri Nov 20 10:22:22 IST 2015'. The second post is titled 'Forum Thread' in yellow text, with 'Post ID : 12' in grey text to its right. It contains the text 'Guest lecture' and 'Description : physics in your regular clas'. To the right of this text is a square icon containing two stylized human figures with speech bubbles above them. Below this is the creation date 'Fri Nov 20 10:16:05 IST 2015'. At the bottom of the screen, there is a black navigation bar with three white icons: a circular arrow, a house, and a square.

After creating the submission request it will be appeared for the student as well as the faculty as shown above.



Students will be provided with a link to upload their file. By clicking on the “Click To Upload”, student can upload their file in the form of a pdf and submit it.

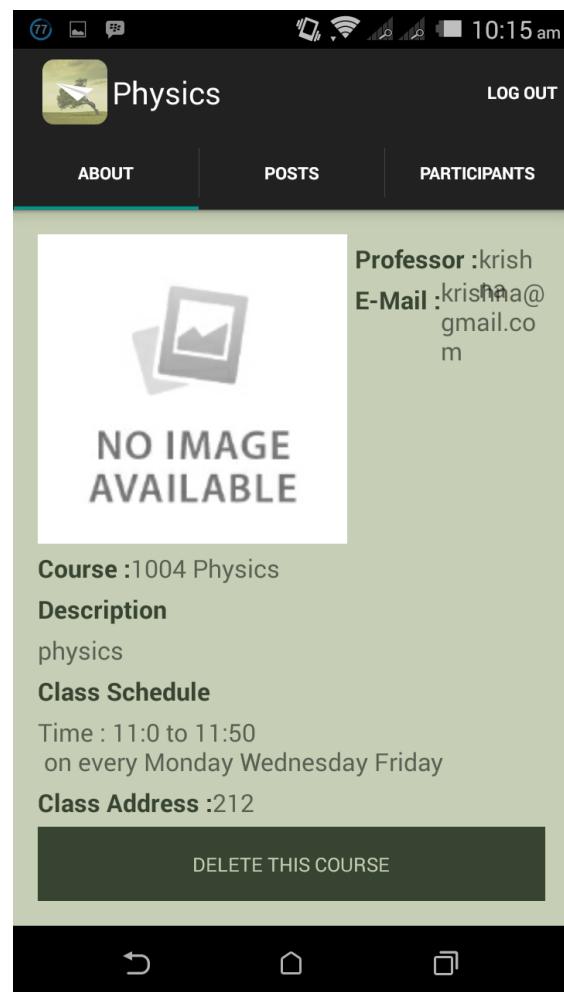
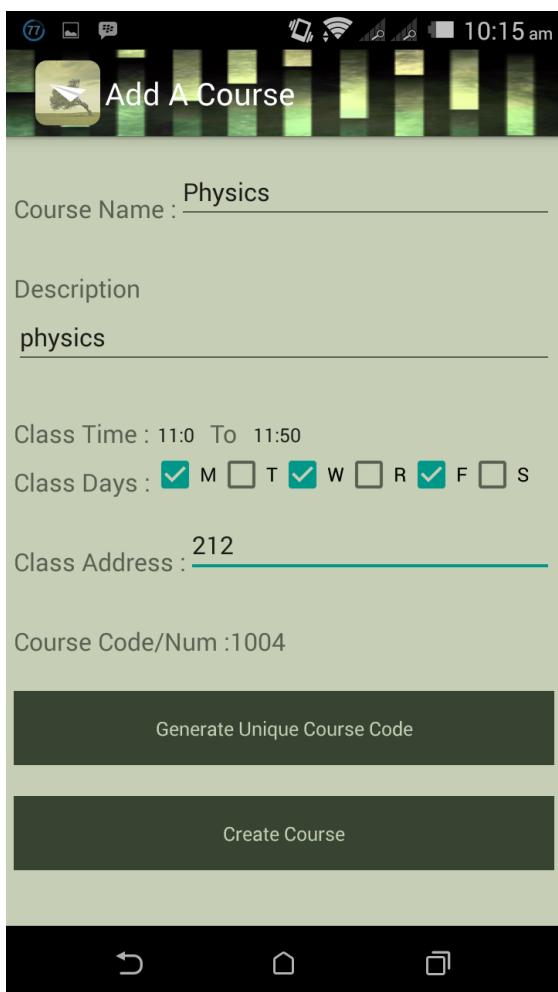


Announcements like cancellation of class can also be posted by the faculty by selecting the category as Announcement, with which student can be able to get notified.

5.2 Performance Analysis

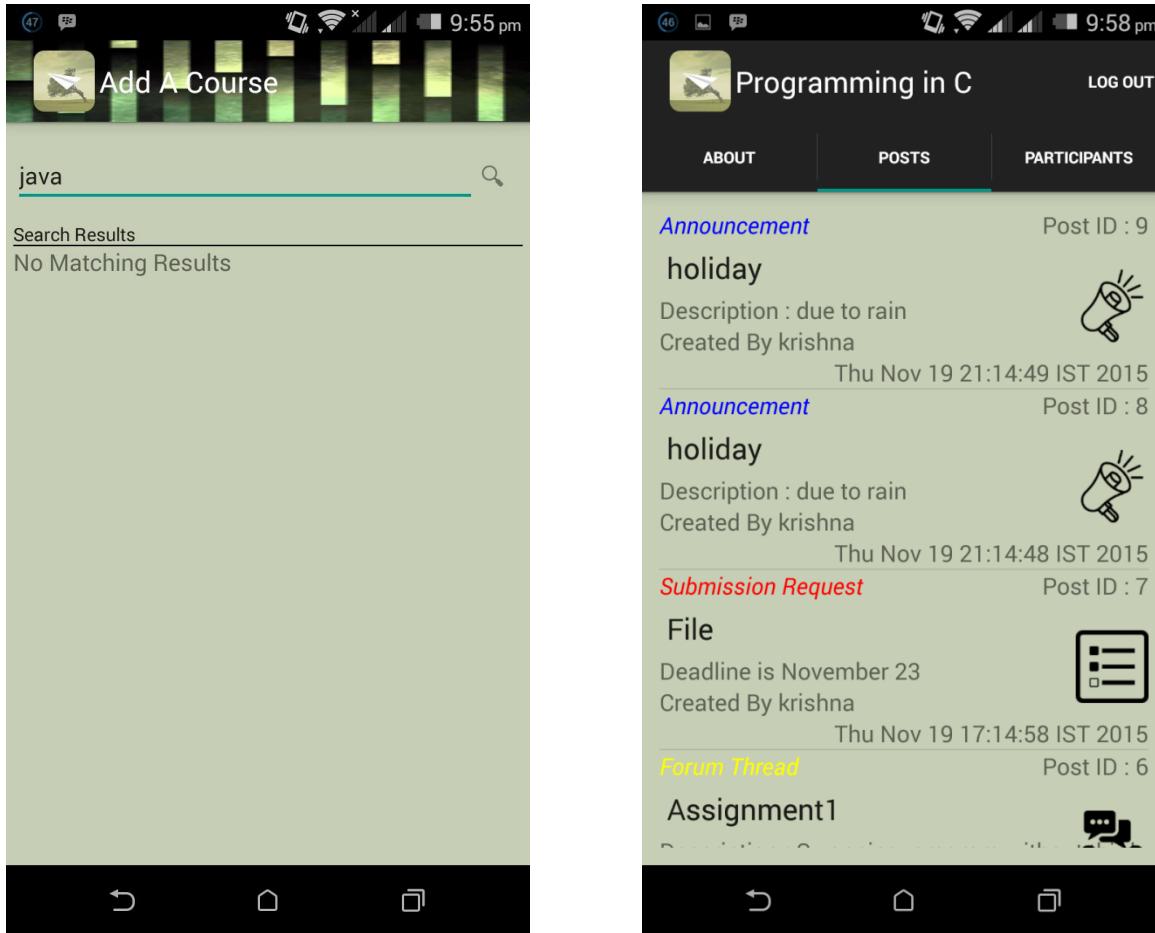
The application performance is concentrated in the certain modules of the unicorn application. The performance is determined in the following screens

Adding a course by the faculty



The faculty screens of adding a course is completely successful. The ease of operation and aptness in the operation determines the successfullness of the operation implemented. This operational functionality is one key performance parameter for faculty login

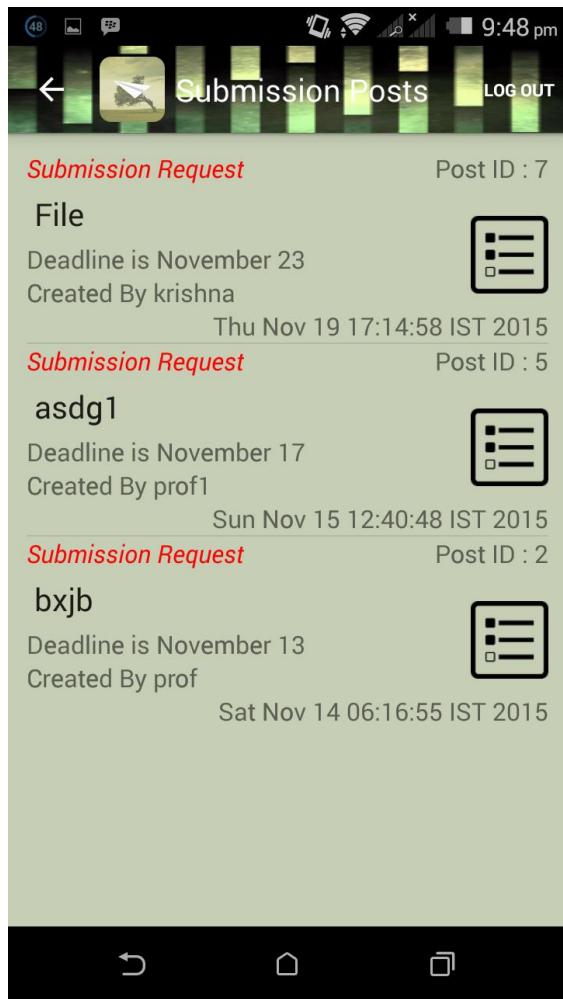
After adding courses by the faculty students will be able to login to their accounts and register themselves for the courses added by the faculty



The student can search for the course and can register himself for the course. After the approval from the faculty the student will be able to view all the posts and submission requests from that particular class.

This functional performance is one of the key areas in the implementation. This is absolutely working. Hence key areas are functioning to the core with the ease of access. This one key functional factor in the application contributing to performance

All the key areas are totally functional to the core. Key areas are extremely tested to ensure best performance at every stage and multitude. The application performance is being affected by several factors like server speed, network connection speed and the processing capability of the mobile device. Inside all these facts the application is completely secure and authenticative in usage.



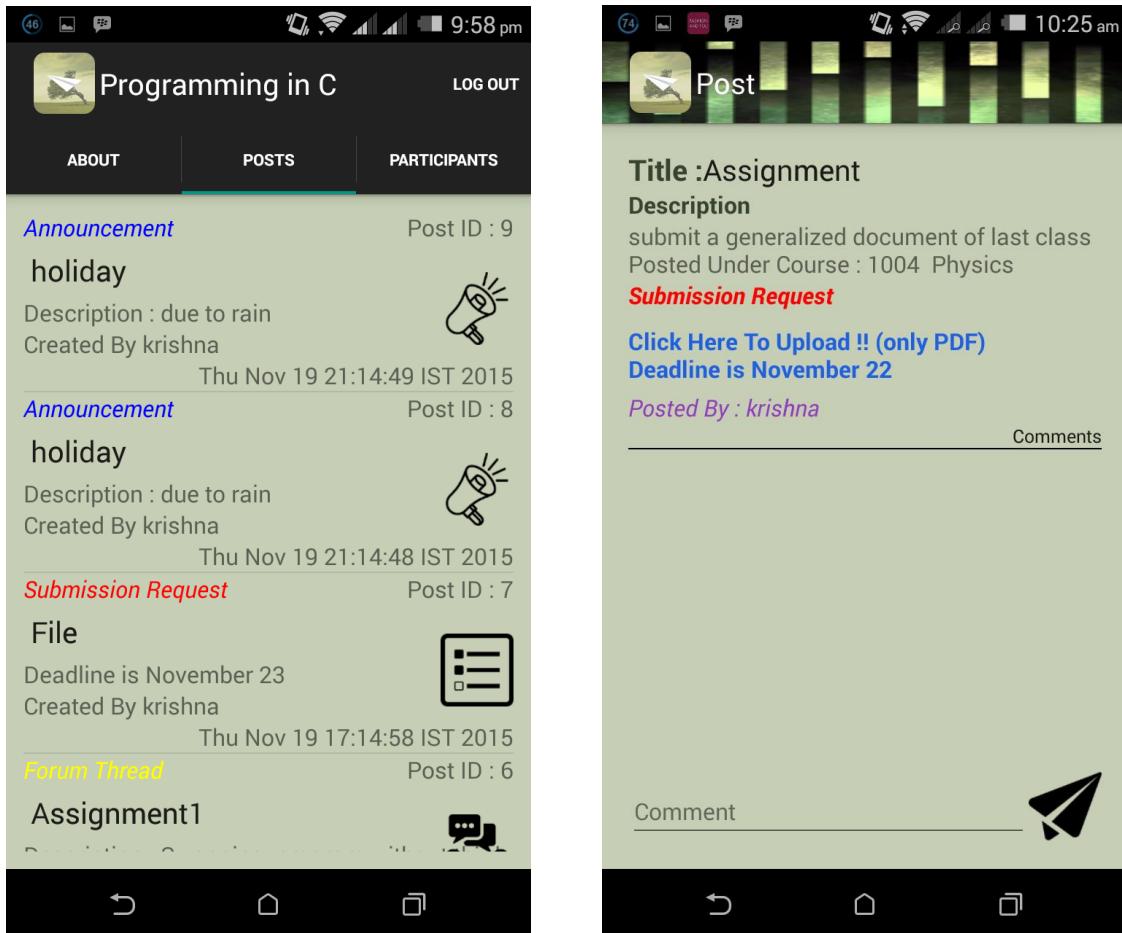
The submission of materials is currently supported to two formats of JPEG and PDF. This is the restriction within the mobile device and this functionality is working properly.

The resulted oriented environment can only be measured in terms of functionality as speed and other factors are determined by the environmental issues.

There are performance oriented authorizations that the application is processing very fastly in terms of performance. Thus the application is totally disruptive in terms of functionality.

Other features of the application are also functional to the core providing access to lot of information from and to the user

Some of the functional screen shots that determine the functionality are



The uploading and downloading of the documents or material submission is fully functional and is functioning as per the design and requirements

These key features are helping in the total functionality of the application

Performance wise the application is functioning to the core.

Chapter 6

Conclusions and Future Enhancements

The application Unisons mere enhanced version of the existing systems. It has the advance features that the current applications are lacking. At the same time UNICON has its own implications and limitations. The Application UNICON can be used for supplementary purposes along with the other Existing Learning Management systems but it cannot be a full length standalone system that can support the knowledge or content management system

The application although performs at its best there are many number of future enhance means that can be done with the system further. Owing to performance wise solutions of the application the performance speed can be farther made to contain the application at its best performance. Certain futures like portability with other Learning management systems like Moodle can be done. The UNICON application can entirely be provided as a service to the users in the form of API's. The application is currently limited to Android environment. This service can further be extended to other environment like windows and IOS.

The application currently addresses many of the issues. But certain enhancements like Push notifications, deployment, Universal customizations etc. can also be made available depending on the user's requirements. The application on the whole bypasses many complex issues that the current systems addresses in order to balance the load on the system. This can be enhanced as well. The Application can be enhanced in numerous ways leaving the development of application to the developer's imagination. Its cohesive integration of modules even with the individual existence of each module supports various modules to be added without any issues.

The application also leaves the possibility and limitations of the service to be an unanswered question. The application can be made robust to any level irrespective of the environment and handlers to support various funvtioanlities. UNICON provide the basic platform for any application to be developed or service to be developed on its foundation headers.

References:

- [https://docs.moodle.org/26/en/Mobile_app]
- [http://www.academia.edu/11306874/Integrating_Mobile_Learning_Management_System_Using_Activity_theory_In_Higher_Education]
- [<https://net.educause.edu/ir/library/pdf/ers1414.pdf>]
- [https://books.google.co.in/books?id=qWjIQm8iYJgC&pg=PA990&lpg=PA990&dq=search+mobile+lms&source=bl&ots=jX8kpbDiZ8&sig=mL_r2yAVoFR7xcWaBDvO9DGKJXo&hl=en&sa=X&ved=0CGQQ6AEwC2oVChMloPiEpoyPyQIVSgmOCh3GkAF8#v=onepage&q=research%20mobile%20lms&f=false]
- [<http://www.mlearn.org.za/CD/papers/Attewell.pdf>]
- [<https://www.schoology.com/k-12/mobile-app>]
- [<https://www.docebo.com/docebo-mobile-learning-app-ios-iphone-android-smartphone/>]

Scholarly Articles

- [<http://www.irrodl.org/index.php/irrodl/article/viewArticle/348>]
- [<http://ajet.org.au/index.php/AJET/article/view/832>]
- [http://www.researchgate.net/profile/Luigi_Colazzo/publication/228796353_Integrating_a_multi-agent_recommendation_system_into_a_mobile_learning_management_system/links/09e4150e844600ce80000000.pdf]
- [<http://www.irrodl.org/index.php/irrodl/article/viewArticle/349>]

Appendix 1

Signup Activity

```
Public class Signup extends Fragment {
```

```
    private OnFragmentInteractionListener mListener;
```

```
    ImageView iv;Bitmap image = null;
```

```
    public SignUp() {
```

```
        // Required empty public constructor
```

```
    }
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
}
```

```
@Override
```

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
```

```
    Bundle savedInstanceState) {
```

```
        // Inflate the layout for this fragment
```

```
        Return inflater.inflate(R.layout.fragment_sign_up, container, false);
```

```
}
```

```
@Override  
public void onActivityCreated(Bundle savedInstanceState) {  
    super.onActivityCreated(savedInstanceState);  
    getActivity().setTitle(R.string.signup);  
    iv = (ImageView) getView().findViewById(R.id.imageView);  
    image = BitmapFactory.decodeResource(getResources(),  
        R.drawable.no_image);  
    iv.setImageBitmap(image);  
  
    iv.setOnClickListener(new ImageView.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            Intent intent = new Intent(Intent.ACTION_GET_CONTENT);  
            intent.setType("image/*");  
            startActivityForResult(Intent.createChooser(intent, "Select Profile Picture"), 1);  
        }  
    });  
  
    getView().findViewById(R.id.button1).setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            onSignUpButtonPressed();  
        }  
    });  
    getView().findViewById(R.id.button2).setOnClickListener(new View.OnClickListener() {
```

```

@Override
public void onClick(View v) {
    onCancelButtonPressed();
}

});

}

// TODO: Rename method, update argument and hook method into UI event
private void onSignUpButtonPressed() {

    EditText et_name = (EditText) getView().findViewById(R.id.editText1);
    EditText et_email = (EditText) getView().findViewById(R.id.editText2);
    EditText et_userName = (EditText) getView().findViewById(R.id.editText5);
    EditText et_passwd = (EditText) getView().findViewById(R.id.editText3);
    EditText et_confirm_passwd = (EditText) getView().findViewById(R.id.editText4);
    Spinner sp = (Spinner) getView().findViewById(R.id.spinner);

    final ParseUser user = new ParseUser();

    if
        (et_name.getText().toString().length()==0||et_email.getText().toString().length()==0||et_passwd.
        getText().toString().length()==0||et_confirm_passwd.getText().toString().length()==0||et_userNa
        me.getText().toString().length()==0)

            Toast.makeText(getActivity(), "Fields Shouldn't Be Empty",
            Toast.LENGTH_SHORT).show();

    else if (et_passwd.getText().toString().equals( et_confirm_passwd.getText().toString())){

        final ProgressDialog pd = new ProgressDialog(getActivity());
        pd.setTitle("Signing Up A New Account...");
        pd.show();
}

```

```

user.setPassword(et_passwd.getText().toString());
user.setUsername(et_userName.getText().toString());
user.setEmail(et_email.getText().toString());
user.put("name", (String) et_name.getText().toString());
user.put("role", (String) sp.getSelectedItem().toString());
ByteArrayOutputStream stream = new ByteArrayOutputStream();
image.compress(Bitmap.CompressFormat.JPEG, 100, stream);
byte[] imageData= stream.toByteArray();
if(imageData != null){
    final ParseFile pf = new
ParseFile(et_userName.getText().toString() + ".jpg", imageData);
    pf.saveInBackground(new SaveCallback() {
        @Override
        public void done(ParseException e) {
            user.put("profile_pic", pf);
            user.signInInBackground(new SignUpCallback() {
                public void done(ParseException e) {
                    if (e == null) {
                        Toast.makeText(getApplicationContext(), "Sign Up successful",
                        Toast.LENGTH_SHORT).show();
                        mListener.onSignUpButtonPressed();
                    } else {
                        Toast.makeText(getApplicationContext(), "Sign Up Failed, Email or User Name
Already Exists", Toast.LENGTH_SHORT).show();
                    }
                }
            pd.dismiss();
        }
    });
}

```

```
    }

}); }}); } }else Toast.makeText(getApplicationContext(), "Password Doesn't Match",
Toast.LENGTH_SHORT).show();}
```

```
private void onCancelButtonPressed() {
```

```
    if (mListener != null) {
        mListener.onCancelButtonPressed();
    }
}
```

```
@Override
```

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {
```

```
    if(requestCode== 1){
        if(resultCode == Activity.RESULT_OK) {
            Uri imageUri = data.getData();
            try {
                image = MediaStore.Images.Media.getBitmap(getApplicationContext().getContentResolver(),
imageUri);
            } catch (IOException e) {
                e.printStackTrace();
            }
            iv.setImageBitmap(image);
        }
    }
}
```

Login Activity :

```
public void onActivityCreated(Bundle savedInstanceState) {  
    super.onActivityCreated(savedInstanceState);  
    getActivity().setTitle("Log In");  
    Button b1 = (Button) getView().findViewById(R.id.button);  
    Button b2 = (Button) getView().findViewById(R.id.button2);  
    b1.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            onLogInButtonPressed();  
        }  
    });  
    b2.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            onNewUserButtonPressed();  
        }  
    });  
    iv = (ImageView) getView().findViewById(R.id.imageView);  
    timer = new Timer();  
    timer.scheduleAtFixedRate(new TimerTask() {  
        @Override  
        public void run() {  
            getActivity().runOnUiThread(new Runnable() {
```

```
    @Override  
  
    public void run() {  
  
        iv.setImageResource(getResourceInt(i));  
  
        if (i == 3) i = 1;  
  
        else i++;  
  
    }  
  
});  
  
}  
  
, 0, 6000);  
  
}  
  
private int getResourceInt(int num) {  
  
    switch (num){  
  
        case 1 : return R.drawable.quote2;  
  
        case 2 : return R.drawable.quote1;  
  
        case 3 : return R.drawable.quote3;  
  
    }  
  
    return 0;  
  
}  
  
// TODO: Rename method, update argument and hook method into UI event  
  
private void onLogInButtonPressed() {  
  
    timer.cancel();
```

```

final EditText et_email = (EditText) getView().findViewById(R.id.editText);
final EditText et_passwd = (EditText) getView().findViewById(R.id.editText2);
final ProgressDialog pd = new ProgressDialog(getActivity());
pd.setTitle("Logging In... ");
pd.show();
ParseUser.logInInBackground(et_email.getText().toString(),
et_passwd.getText().toString(), new LogInCallback() {

@Override
public void done(ParseUser user, ParseException e) {
    if (et_email.getText().toString().length() == 0)
        et_email.setError("Enter ID");
    else if (et_passwd.getText().toString().length() == 0)
        et_passwd.setError("Enter Password");
    else if (user != null) {
        mListener.onLogInButtonPressed();
    } else {
        Toast.makeText(getActivity(), "Log In Failed, Invalid Email or Password",
Toast.LENGTH_SHORT).show();
    }
    pd.dismiss();
}
});

}

private void onNewUserButtonPressed() {

```

```

        timer.cancel();

        if (mListener != null) {
            mListener.onNewUserButtonPressed();
        }
    }

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        try {
            mListener = (OnFragmentInteractionListener) activity;
        } catch (ClassCastException e) {
            throw new ClassCastException(activity.toString()
                    + " must implement OnFragmentInteractionListener");
        }
    }

    @Override
    public void onDetach() {
        super.onDetach();
        timer.cancel();
        mListener = null;
    }

    public interface OnFragmentInteractionListener {

```

```
// TODO: Update argument type and name  
public void onNewUserButtonPressed();  
public void onLogInButtonPressed();  
}  
  
}
```

Full Screen Activity :

```
public class FullscreenActivity extends ActionBarActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_fullscreen);  
        getSupportActionBar().hide();  
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {  
            getWindow().setNavigationBarColor(getResources().getColor(R.color.color_dark));  
            getWindow().setStatusBarColor(getResources().getColor(R.color.color_dark));  
        }  
        new Timer().schedule(new TimerTask() {  
            @Override  
            public void run() {  
                startActivity(new Intent(FullscreenActivity.this, LogIn_SignUp.class));  
            }  
        }, 3500);  
    }  
}
```

Profile activity:

```
package com.help.sd.uni_con;

import android.content.Context;
import android.graphics.Color;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

import com.parse.ParseObject;

import java.util.Calendar;
import java.util.List;
import java.util.Locale;

/***
 * Created by SaideepReddy on 11/14/2015.
 */
public class commentsAdapter extends ArrayAdapter<ParseObject> {

    Context context; int resource; List<ParseObject> objects;

    public commentsAdapter(Context context, int resource, List<ParseObject> objects) {
```

```

super(context, resource, objects);

this.context = context;

this.resource = resource;

this.objects = objects;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

    if(convertView==null){

        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        convertView = inflater.inflate(resource,parent,false);

    }

    TextView tv1 = (TextView) convertView.findViewById(R.id.textView63);

    TextView tv2 = (TextView) convertView.findViewById(R.id.textView64);

    TextView tv3 = (TextView) convertView.findViewById(R.id.textView65);

    ParseObject po = objects.get(position);

    tv1.setText(po.getString("author")+" : ");

    tv2.setText(po.getString("comment"));

    Calendar date = Calendar.getInstance();

    date.setTime(po.getCreatedAt());

    tv3.setText(date.getDisplayName(Calendar.MONTH, Calendar.SHORT,
Locale.ENGLISH)+date.get(Calendar.DAY_OF_MONTH)+" "+

date.get(Calendar.HOUR_OF_DAY)+":"+date.get(Calendar.MINUTE));

    switch(position%4){

        case 0:tv1.setTextColor(Color.parseColor("#FFE0981A"));


```

```

        break;

    case 1:tv1.setTextColor(Color.parseColor("#FF1EE2D5"));

        break;

    case 2:tv1.setTextColor(Color.parseColor("#FF9729E0"));

        break;

    case 3:tv1.setTextColor(Color.parseColor("#FFE024D0"));

        break;

    }

    return convertView;

}

}

```

Parse Application

```

package com.help.sd.uni_con;

import android.app.Application;

import com.parse.Parse;
import com.parse.ParseACL;

/**
 * Created by SaideepReddy on 8/15/2015.
 */

public class ParseApplication extends Application {

```

```

@Override
public void onCreate() {
    super.onCreate();

    // Initialize Crash Reporting.
    // ParseCrashReporting.enable(this);

    // Enable Local Datastore.
    //Parse.enableLocalDatastore(this);

    // Add your initialization code here
    Parse.initialize(this, "uvEc8lrTmE1QgsapagJzXR04Lws3iw91T5yslUXp",
    "RLFQVqMYhMoxBq01qQEoNOOrzAY73FYV1fWT6Hs");

    //ParseUser.enableAutomaticUser();
    ParseACL defaultACL = new ParseACL();
    // Optionally enable public read access.
    defaultACL.setPublicReadAccess(true);
    //ParseACL.setDefaultACL(defaultACL, true);
}

}

```

Home Activity:

```
package com.help.sd.uni_con.Home_Activity;

import android.app.AlertDialog;
import android.app.Fragment;
import android.app.FragmentManager;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Build;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.app.ActionBar;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

import com.help.sd.uni_con.LogIn_SignUp_Activity.LogIn_SignUp;
import com.help.sd.uni_con.R;
import com.parse.ParseUser;

public class Home extends AppCompatActivity
    implements NavigationDrawerFragment.NavigationCallbacks {
    boolean studentFlag = false;
    /**
     * Fragment managing the behaviors, interactions and presentation of the navigation drawer.

```

```
*/  
  
private NavigationDrawerFragment mNavigationDrawerFragment;  
  
/**/  
 * Used to store the last screen title. For use in {@link #restoreActionBar()}.  
 */  
  
private CharSequence mTitle;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_home);  
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {  
        getWindow().setNavigationBarColor(getResources().getColor(R.color.color_dark));  
        getWindow().setStatusBarColor(getResources().getColor(R.color.color_dark));  
    }  
  
    findViewById(R.id.container).setBackgroundColor(getResources().getColor(R.color.color_light));  
    if(ParseUser.getCurrentUser().getString("role").equals("Student")) studentFlag=true;  
  
    //  
    // ParseUser.logOut();  
  
    mNavigationDrawerFragment = (NavigationDrawerFragment)  
        getSupportFragmentManager().findFragmentById(R.id.navigation_drawer);
```

```

//mTitle = getTitle();

// Set up the drawer.

mNavigationDrawerFragment.setUp(
    R.id.navigation_drawer,
    (DrawerLayout) findViewById(R.id.drawer_layout));
}

@Override
public void onNavigationDrawerItemSelected(int position) {
    Fragment selection = null ;
    if(position>2&&!studentFlag){
        position++;
    }
    switch(position){
        case 0 : selection = new homeFragment();
        mTitle="Home";
        break;
        case 1 : selection = new profileFragment();
        mTitle="Profile Settings";
        break;
        case 2 : selection = new coursesFragment();
        mTitle="Courses";
        break;
        case 3 : selection = new submissionsFragment();
    }
}

```

```

        mTitle="Submission Posts";
        break;

    case 4 : selection = new forumFragment();
        mTitle="Forum";
        break;

    case 5 : selection = new settingsFragment();
        mTitle="App Settings";
        break;

    case 6 : selection = new supportFragment();
        mTitle="Dev Support";
        break;
    }

FragmentManager fragmentManager = getFragmentManager();
fragmentManager.beginTransaction()
.replace(R.id.container, selection)
.commit();

}

public void restoreActionBar() {
    ActionBar actionBar = getSupportActionBar();
    actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_STANDARD);
    actionBar.setDisplayShowTitleEnabled(true);
    actionBar.setDisplayShowHomeEnabled(true);
    actionBar.setIcon(R.mipmap.ic_launcher);
}

```

```
    actionBar.setBackgroundDrawable(getResources().getDrawable(R.drawable.action_bar_image));  
    actionBar.setTitle(mTitle);  
}
```

```
@Override  
  
public boolean onCreateOptionsMenu(Menu menu) {  
  
    if (!mNavigationDrawerFragment.isDrawerOpen()) {  
  
        // Only show items in the action bar relevant to this screen  
  
        // if the drawer is not showing. Otherwise, let the drawer  
  
        // decide what to show in the action bar.  
  
        getMenuInflater().inflate(R.menu.home, menu);  
  
        restoreActionBar();  
  
        return true;  
    }  
  
    return super.onCreateOptionsMenu(menu);  
}
```

```
@Override  
  
public boolean onOptionsItemSelected(MenuItem item) {  
  
    // Handle action bar item clicks here. The action bar will  
  
    // automatically handle clicks on the Home/Up button, so long  
  
    // as you specify a parent activity in AndroidManifest.xml.  
  
    int id = item.getItemId();
```

```

//noinspection SimplifiableIfStatement

if (id == R.id.action_logout) {

    ParseUser.logOut();

    finish();

    Intent intent = new Intent(Home.this,LogIn_SignUp.class);

    startActivity(intent);

    return true;

}

return super.onOptionsItemSelected(item);

}

@Override

public void onBackPressed() {

    AlertDialog.Builder builder = new AlertDialog.Builder(Home.this);

    builder.setTitle("Do you wanna quit the application ?")

        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {

            @Override

            public void onClick(DialogInterface dialog, int which) {

                Intent intent = new Intent(Intent.ACTION_MAIN);

                intent.addCategory(Intent.CATEGORY_HOME);

                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

                startActivity(intent);

            }

        }).setNegativeButton("No", new DialogInterface.OnClickListener() {

```

```
@Override  
public void onClick(DialogInterface dialog, int which) {  
  
}); builder.create().show() }
```

Home activity:

```
package com.help.sd.uni_con.Home_Activity;
```

```
import android.app.Activity;  
import android.graphics.Color;  
import android.net.Uri;  
import android.os.Bundle;  
import android.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
import com.help.sd.uni_con.R;  
import com.parse.FindCallback;  
import com.parse.ParseException;  
import com.parse.ParseObject;  
import com.parse.ParseQuery;  
  
import java.util.ArrayList;
```

```
import java.util.List;

public class homeFragment extends Fragment {

    public homeFragment() {
        // Required empty public constructor
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //getActivity().setTitleColor(Color.parseColor("#d3d3d3"));

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_home, container, false);
    }

    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
```

```

super.onActivityResult(savedInstanceState);

final ListView lv1 = (ListView) getActivity().findViewById(R.id.listView2);

final ListView lv2 = (ListView) getActivity().findViewById(R.id.listView3);

final ArrayList<String> ann = new ArrayList<>(),events = new ArrayList<>();

ParseQuery<ParseObject> query = ParseQuery.getQuery("Events");

query.findInBackground(new FindCallback<ParseObject>() {

    @Override

    public void done(List<ParseObject> list, ParseException e) {

        if(list.size()>0){

            for(ParseObject po:list){

                if(po.getString("type").equals("Ann"))

                    ann.add(po.getString("info"));

                else events.add(po.getString("info"));

            }

            ArrayAdapter<String> adapter1 = new ArrayAdapter<>(getActivity(),
            android.R.layout.simple_list_item_1,ann);

            lv1.setAdapter(adapter1);

            ArrayAdapter<String> adapter2 = new ArrayAdapter<>(getActivity(),
            android.R.layout.simple_list_item_1,events);

            lv2.setAdapter(adapter2);

        }

    }

});
```