

Cairo University

Faculty of Computers and Information Department of Computer Science

FCI E-Campus

The Way to Better Education

**Academic Year 2017-2018**

**Final Year Documentation of Graduation Project**



Cairo University

Faculty of Computers and Information Department of Computer Science

**Supervised by**

***Dr. Mohammed Nassef***

***TA. Nora Abdel Hamid***

Implemented by

|  |  |
| --- | --- |
| ***20140169*** | ***Abdul-Aziz Mahmoud Abdul-Aziz*** |
| ***20140163*** | ***Abdulrahman Mahmoud Mohamed*** |
| ***20140197*** | ***Amr Madgy Mohammed Ezzat*** |
| ***20140030*** | ***Ahmed AbdelMohsen Ahmed*** |
| ***20140170*** | ***Abdallah Abdelazim Abdallah*** |

**Academic Year 2017-2018**

**Final Year Documentation of Graduation Project**

Table of Contents

[Introduction 4](#_Toc518225110)

[Abstract 4](#_Toc518225111)

[Introduction to the main area of the project 4](#_Toc518225112)

[Motivation 5](#_Toc518225113)

[Problem definition 5](#_Toc518225114)

[Project Objective 6](#_Toc518225115)

[Gantt chart of project time plan 7](#_Toc518225116)

[Project development methodology 10](#_Toc518225117)

[The used tools in the project 10](#_Toc518225118)

[Report Organization 10](#_Toc518225119)

[Related work 12](#_Toc518225120)

[System Analysis 15](#_Toc518225121)

[Project specification 15](#_Toc518225122)

[Functional requirement 15](#_Toc518225123)

[Non-functional requirement 21](#_Toc518225124)

[Future work 22](#_Toc518225125)

[Use case Diagrams 24](#_Toc518225126)

[Use Case Tables 26](#_Toc518225127)

[System test cases 31](#_Toc518225128)

[System Design 37](#_Toc518225129)

[System Component Diagram 37](#_Toc518225130)

[System Class Diagrams 38](#_Toc518225131)

[Sequence Diagrams 40](#_Toc518225132)

[Project ERD 42](#_Toc518225133)

[System GUI Design 45](#_Toc518225134)

[Implementation and Testing 56](#_Toc518225135)

# **Introduction**

## Abstract

Organizing time for a faculty student is an essential part of his success. This has been always a problem with no definite solution.

The students at Faculty of Computer and Information, Cairo University are heavily distracted between many online places.

Facebook groups, E-com, Acadox, GDrive and a lot more. Cluttered and unorganized, theses websites do bad to students more than they do them good.

Our app is an effective solution for all the problems facing students at FCI. It combines all what they will need in a single place saving them from distraction and ad-flooded websites. Targeted for their needs, it facilitates learning and helps them focus on the more important things, i.e. learning.

## Introduction to the main area of the project

The main area of this project is **software engineering**. Engineering by itself is the application of well-understood scientific methods to the construction, operation, modification and maintenance of useful devices and systems. However, Software Engineering is the sub discipline of Computer Science that attempts to apply engineering principles to the creation, operation, modification and maintenance of the software components of various systems. As with much of Computer Science, the subject of Software Engineering is at a very early stage in its development. It is much more of an art than a science, and at present has little in common with classical engineering.

Someday, Software Engineering may well be concerned with the application of well-understood scientific methods to the construction, operation, modification and maintenance of software. Today, however, Software Engineering is concerned with finding ways in which to produce working software for predictable costs in predictable time. When the problems involved are very simple or when only one person is involved, implementing software to meet their own needs, there isn't much to be said, and we are a long way from having any scientific principles for the production of software. Therefore, the major focus of software engineering today is on well-tested heuristics for the production of software to solve complex problems when many people are involved in the process, as users, as analysts, as programmers, as managers, etc. Therefore, most of the issues in Software Engineering are concerned with interactions among people, rather than with the production of software.

Software engineering can be divided into sub-disciplines. Some of them are:

* Software requirements (or Requirements engineering): The elicitation, analysis, specification, and validation of requirements for software.
* Software design: The process of defining the architecture, components, interfaces, and other characteristics of a system or component. It is also defined as the result of that process.
* Software construction: The detailed creation of working, meaningful software through a combination of programming (aka coding), verification, unit testing, integration testing, and debugging.
* Software testing: An empirical, technical investigation conducted to provide stakeholders with information about the quality of the product or service under test.

Motivation

Organizing time for a faculty student is an essential part of his success. In college, it can be hard to remember due dates, assignments and tasks for each class when you don’t have the same classes every day or professors continually reminding you of important dates. Thus, it’s important the students stay organized, so they can be successful and less stressed.

There’s no worse feeling than walking into class and realizing an assignment was due and you missed the deadline. Generally, when your assignments and key dates are outlined for each class, you’re better able to prepare and allocate your time accordingly, leaving you less stressed. When you’re less stressed, you are able to focus your energy and attention on more important things…like studying!

Also, with so many fun things going on around campus and with your friends, you don’t want an assignment or project you forgot about to hinder your ability to spend some time doing something social. If you have things organized, you don’t have to worry about forgetting a deadline, and you can plan for social events by accomplishing your assignments and studying early!

As a college student, there are inevitably weeks when multiple tests occur or various projects are due, especially around midterms and finals. When this occurs, it’s even more important to have a plan, so you can prioritize your study time and start studying enough in advance to be prepared for all your classes without the need for ineffective late-night cramming

## Problem definition

When it comes to organizing their class schedule, lectures/sections/labs materials and upcoming assignments/exams, students here at Faculty of Computer and Information, Cairo University are in a terrible mess.

Students at FCI are scattered across many places. Some professors and TAs publish their announcements and materials on the faculty’s official website, others use a third-party website such as Acadox, and moreover some professors and TAs send the materials and announcements to a student who is supposed to deliver that to all the other students. Besides that, the faculty’s schedule is provided to students as just a plain document. Students lose track of assignments deliveries and quizzes that they have.

Moreover, the faculty doesn’t provide a forum where the students can ask and discuss about things relating to their courses or things that aren’t clear for them or they just don’t understand. Students have to use social network websites such as Facebook to communicate which is so distracting and it won’t be appropriate for professors and TAs to answer the questions and help the students.

We, as students at FCI, most often find ourselves scattered between the faculty’s website, Acadox, Facebook, GDrive, Dropbox, Mega, etc.… After all of that we and the other student at FCI can’t help being lost and unorganized.

Project Objective

This project aims to create a mobile application that will help students & staff at FCI stay organized. The app will provide students, teacher assistants & professors with their schedule, depending on the courses they have registered. The schedule will tell them what lectures/labs/sections they have and the location of these in the faculty lectures/sections halls and labs. If the student doesn’t know the location of his lecture or lab the app will help him and show him a map to where the lecture/section hall or lab is in the faculty.

Also, users receive new announcements from the faculty notifying them with important events happening at the faculty such as official holidays, the dates of the beginning of the semesters, etc.

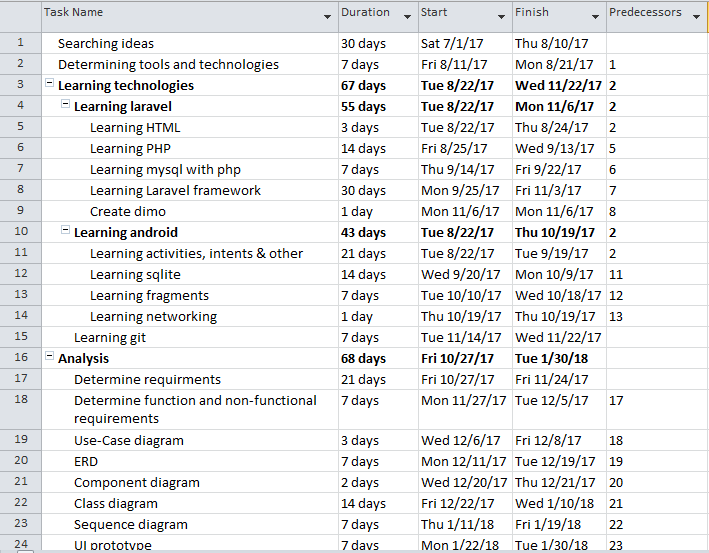
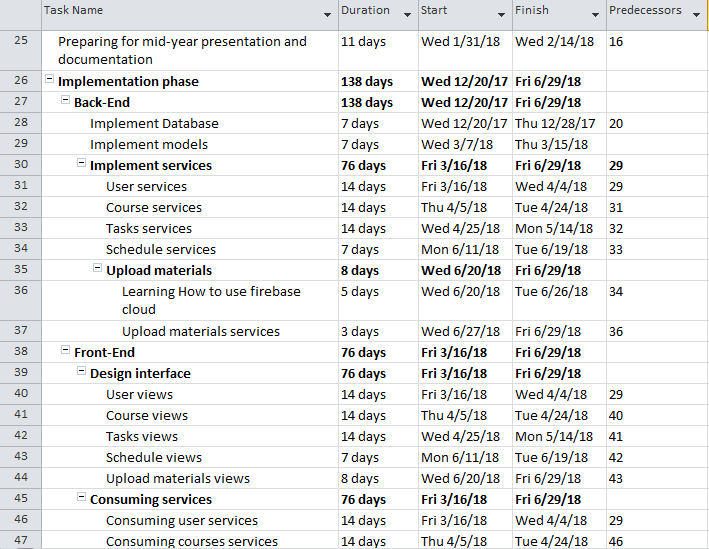
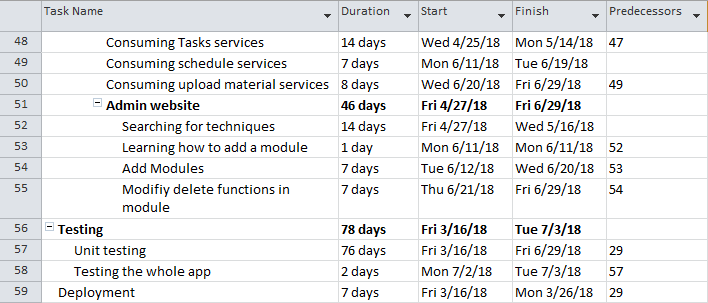
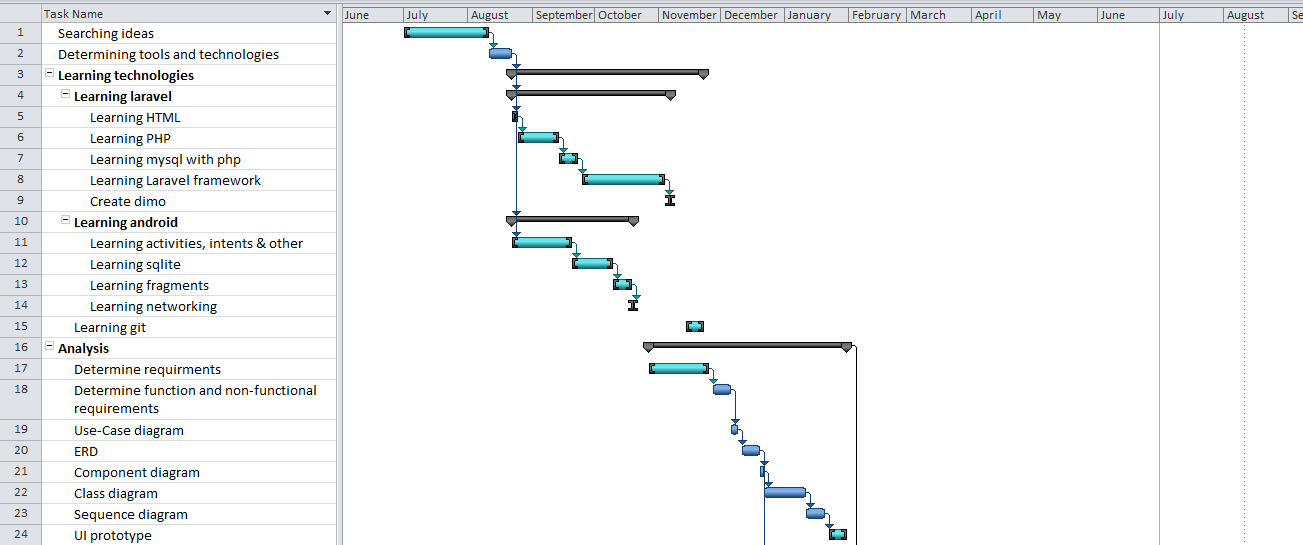
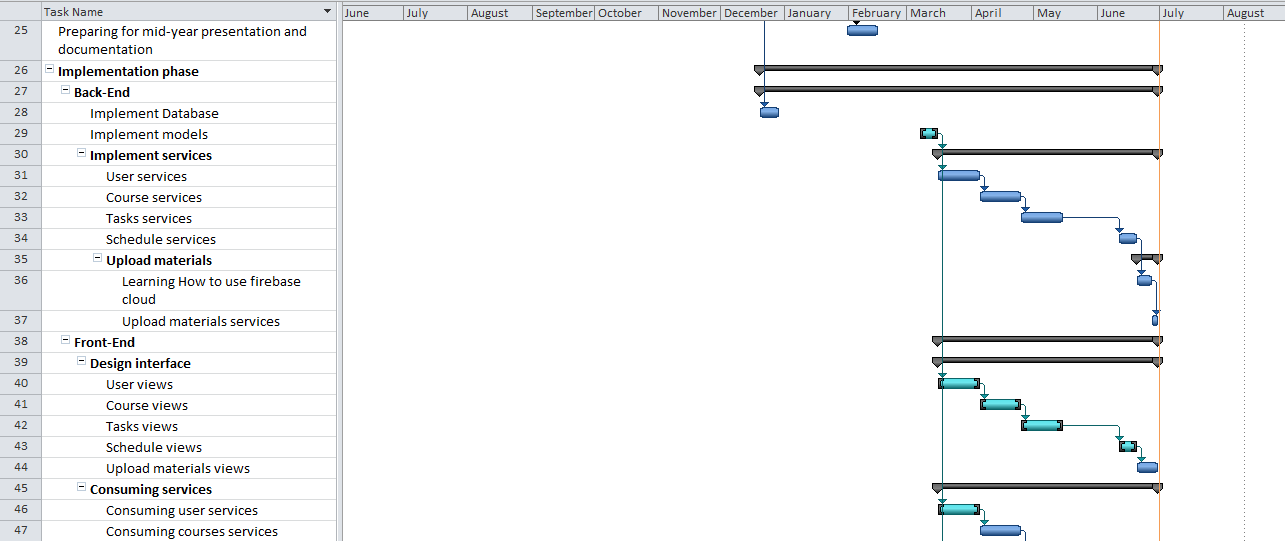
Tasks, quizzes and assignments deliveries due dates were always a problem. To help students keep track with their courses, our app gives an overview with what is due this day and the next. It also gives them a list combining all tasks from all the courses they are registered in. Students will also be notified regularly to remind them of their assignments, quizzes and exams and urge them to do their studies and work early before due date.

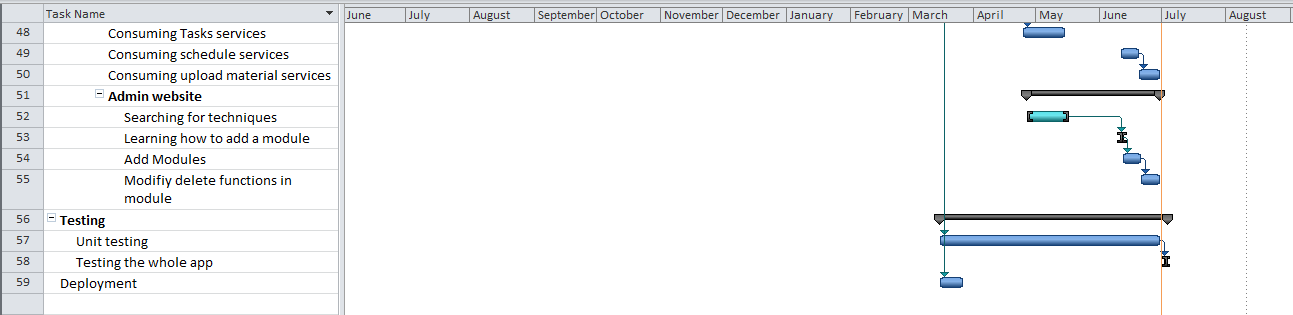
The app will organize downloading and uploading materials and will provide each course’s lectures, sections and labs with its materials that were uploaded.

The app will provide a forum for the students to communicate with each other as well as with TAs and professors. Students can help each other and what will be better is that now it will be appropriate for professors and TAs to enter the forum and answer some of these questions. The forum will be Q/A based. Posts will have a title, body and a marker to indicate whether the problem for this post has been solved and answered or not yet.

Being a mobile application means that all these important features will be available all the time and from anywhere. As nowadays, mobile phones don’t leave one’s hand and pocket. Either the user is at his house or outside catching transportation or doing something, he can open the app and check his courses.

## Gantt chart of project time plan





Project development methodology

In this project we used Scrum Methodology.

Scrum is an agile method for project management. Scrum is characterized by:

* A living backlog of prioritized work to be done.
* Completion of a largely fixed set of backlog items in a series of short iterations or sprints.
* A brief daily meeting (called a scrum), at which progress is explained, upcoming work is described, and obstacles are raised.
* A brief planning session in which the backlog items for the sprint will be defined.
* A brief heartbeat retrospective, at which all team members reflect about the past sprint.

Scrum is facilitated by a scrum master; whose primary job is to remove impediments to the ability of the team to deliver the sprint goal. The scrum master is not the leader of the team but acts as a productivity buffer between the team and any destabilizing influences

The used tools in the project

In this project we build our software in SOA architecture.

The backend:

* Developed using the Laravel PHP framework alongside with MySQL DBMS.
* Developed with Jetbrains PhpStorm IDE and XAMPP stack.
* Deployed on Heroku which is a cloud platform-as-a-service (PAAS).

The frontend (client):

* Android application developed using the native Android framework with Java.
* Developed with Android Studio IDE and Virtual Emulators to test our app.
* Published with the Google PlayStore console.

## Report Organization

The rest of this document gives more in-depth information about the project; its analysis, design, development & testing.

* **Chapter 2: Related work** demonstrates the nearest examples of the project and the main differences between them and this project
* **Chapter 3: System Analysis** gives in complete details the project analysis conducted. This includes the project specification; defining its functional & non-functional requirements. It contains our use case diagram as well as the system test cases.
* **Chapter 4: System Design** explains the design criteria of the system built in the project. This includes the system component diagram, system class diagram, sequence diagram, the database entity relationship diagram (ERD) as well as the frontend GUI design.
* **Chapter 5: Implementation and Testing** demonstrates screenshots of the System running and samples of the applied test cases.

At the end of this document we also include the references used throughout this document.

# **Related work**

1. **Acadox:**

A website that helps you manage coursework, connect and collaborate with learners, and document your achievements.

Link:

<https://www.acadox.com>

Difference:

* Our app manages the schedule of the registered courses, it displays the slots time of the labs, sections & lectures of the course that you have registered in.
* It also manages the schedule of a specific day, it displays the slots time of the labs, sections & lectures of the courses you have in that day.
* Our app displays all the tasks you have in different courses and displays all the tasks due to a specific day.
* Student also cannot join a course unless they pass its prerequisites courses.
* It also shows a static map for the faculty of computers and Information – Cairo University.
* In addition to our app will be officially targeted to our faculty “faculty of computers and Information – Cairo University” with a verified TAs and Professors.

1. **E-com:**

The official Community website for the faculty of computers and information – Cairo University that contains all the student, TAs and doctors, it displays the grades, attendance & GPA, doctors and TAs can upload materials and send announcements, it contains many other functionalities that manages the communication between the faculty stuff and the students.

Link:  
<http://ecom.fci.cu.edu.eg>

Difference:

* Our app manages the schedule of the registered courses, it displays the slots time of the labs, sections & lectures of the course that you have registered in.
* It also manages the schedule of a specific day, it displays the slots time of the labs, sections & lectures of the courses you have in that day.
* Our app displays all the tasks you have in different courses and displays all the tasks due to a specific day.
* It is also more organized in displaying the resources of a specific course.
* It also has a forum for each course that helps students to add post to ask about anything related to that course and also helps TAs & professors to add post to make announcement for the students joined that course.
* It also shows a static map for the faculty of computers and Information – Cairo University.
* In addition to our app is an android app while the E-com is a website.
* E-com has a very poor user interface design. It has low security as it’s written in old PHP.

1. **Facebook:**

The most famous social app, nowadays students of the same level in the faculty creates a group to share materials and information about the courses and the faculty.

Due to the spreading of the Facebook it becomes the main app in communication between students, even the TAs and sometimes doctors too.

Link:

<https://www.facebook.com>

Difference:

* First of all, Facebook is not that app that its purpose is to manage a University life, organize faculty courses in addition to it is not an official app to communicate with faculty stuff.
* Our app manages the schedule of the registered courses, it displays the slots time of the labs, sections & lectures of the course that you have registered in.
* It also manages the schedule of a specific day, it displays the slots time of the labs, sections & lectures of the courses you have in that day.
* It also displays all the tasks you have in different courses and displays all the tasks due to a specific day.
* It also more organized in displaying the resources of a specific course.
* It also has a form for each course that helps students to add post to ask about anything related to that course and also helps TAs & professors to add post to make announcement for the students joined that course, “each form is for a specific course”.
* It also shows a static map for the faculty of computers and Information – Cairo University.
* In addition to our app will be officially to our faculty “faculty of computers and Information – Cairo University” with a verified TAs and Professors.

1. **“School Planner” Android application:**

School Planner is an Android application published on the PlayStore. It is a handy app for students of all ages that was designed to help you organize your career as a student and have everything under control. Whether you are attending elementary school, high school or college, this is the app for you!

Link (Google PlayStore):

<https://play.google.com/store/apps/details?id=daldev.android.gradehelper>

Difference:

* School Planner is the closest application to our project almost offering most of the features that we offer. It offers courses, schedules, tasks, etc…
* School Planner works completely offline. No backend support at all. Users have to add courses themselves, add their tasks themselves & add their schedules themselves. So basically, you have to do a lot of work just to setup the app. And you’ll have to do every semester.
* Our app has a backend. Providing easy registration for the courses. Student just enters the course no more.
* Communication is an important part of the education process. School Planner has no forum. Our app has one for each course allowing students, TAs & Professors to communicate on the forum sharing knowledge and solving problems.
* School Planner is not targeted for FCI. It’s a general app. Our app’s features are specifically designed for FCI.
* School Planner is a paid software. It contains annoying ads that distracts the user and violates the sole purpose of creating such app; AVOIDING DISTRACTIONS.
* FCI E-campus is a free and open source software (licensed with GPLv3.0 license) published on the PlayStore free of any charge.

# **System Analysis**

## Project specification

### **Functional requirement**

#### **For student**

**Sing up**

Student can create his account in our system.

**Login**

He can login to his account and then continue on the system.

**Update his password**

Student can update his password and set new password.

**Show course**   
 get all course details, the name of course, description, TAs and professors in course, schedule of course, materials and tasks.

**Join course**

Student enters the code of course and the pass code for this course and his group number to join this course.

**Show all his courses**

Student can show all the courses he has joined in a list.

**Show all tasks**

Get tasks information such as start date, end date, description.

**Get all tasks for specific course**

Get tasks and know their due dates.

**Show schedule for specific course**

Show the time and place for lectures, labs and sections

**Show his own schedule**

Get schedules for all his courses and only for his group in one place.

**Overview of his day**

Show all lectures, labs and section in his day and the next to never miss a lecture, lab or section.

**show all materials for specific course**

Get all official materials

**Download material**

Student can download official materials for specify course

**Show static map**

This map is a static graphical map which guides the students by showing the buildings and the location of each lecture/section hall , the mosque, the cafeteria, etc.

**Show all Announcements**

All the students signed up to the app can see announcements.

Announcements does not relate to a specific course. The announcements will be information related to the faculty in general (ex. informing students that a day is an official holiday, informing students about the date of the first day of the term, etc…)

**Add post to forum in specific course**

Every course has a Q/A forum to add posts in it, so students can post to ask any thing about this course and the post will appear to all users who has joined in this course

**Mark posts as answered**

If the problem or the inquiry of the post has been answered, student can mark the post as answered.

**Add comments to post**

Student can comment to any post in the course forum he has joined in.

**Show forum**

Student can see forum in course he has joined in and see posts in this forum and comments for the posts

#### **For TA and professor**

**Sing up**

TA & professor can create his account in our system

**Login**

He can login to his account and then continue on the system

**Activate account**

The account must be activated. The system sends to his academic email the activation code and he inserts it in our app in order to continue.

**Update his password**

Student can update his password and set new password

**Show course**   
 get all course details, the name of course, description, TAs and professors in course, schedule, materials and tasks.

**Show course tasks**

Get task information Start date, end date & description.

**Add task**

He adds tasks to course he has been assigned to.

**Show schedule for specify course**

Show the time and place for lectures, labs and sections

**Show his own schedule for all the courses he assigned to**

Get the schedule for all the courses he has been assigned to. TAs get their schedule by groups he is assigned to only.

**Overview for his day**

get all lectures, labs and section in his day and the next.

**show all materials for specify course**

get all official materials.

**Upload material**

TA and professor can upload lecture and labs to courses they’re assigned to.

**Download material**

TA and Professor can download official materials for specify course .

**Show static map**

This map is a static graphical map which guides the students by showing the buildings and the location of each lecture/section, the mosque, the restaurant, etc.

**Show all Announcements**

All the students signed up to the app can see announcements

announcements that does not relate to a specific course. The announcements will be information related to the faculty in general (ex. informing students that a day is an official holiday, informing students about the date of the first day of the term, etc…).

**Show forum**

TA and Professor can see forum in course he has been assigned to and see posts in this forum and comment in posts.

**Add post to forum in specific course**

Every course has a forum to add posts in, so TA and professor can post to inform students with anything about this course and the post will appear to all users joined in this course.

**Add comment to post**

TA and Professor can comment to any post in his courses forums.

#### **For Admin**

**Add department.**

**Add TA to the system.**

**Add professor to the system.**

**Add group to specific course.**

**Add new course to the system.**

**Add schedule to specific course.**

**Specify prerequisite courses for a specific course.**

**Add Announcement.**

**Assign TAs to Course.**

**Assign professor to Course.**

**Delete department.**

**Delete TA.**

**Delete Professor.**

**Delete group from specific course.**

**Delete course from system.**

**Delete schedule from specific course.**

**Delete Announcement.**

**Delete TAs from Course.**

**Delete Professor from Course.**

### **Non-functional requirement**

1. **Accessibility**
   * Due to the spreading of smart phones that uses android systems our application can be accessed by large community “as it will be an android app”.
2. **Availability**
   * Our app is available 24-hours.
3. **Regulatory compliance**
   * Schedules in our app will be very specific, without any fault and up-to-date with the college schedule.
4. **Cost**
   * Our app will be for free
5. **Deployment**
   * App is available on app store.
   * Back services are deployments
6. **Efficiency**
   * Our app will use only the needed martials “schedule for the student will contains the schedule of the courses that he registered in only without containing the schedule of the other courses.
7. **Effectiveness**
   * Our app will have all the features that the students needed instead of using more than one app.
8. **Extensibility**
   * We can add easily more features in our app in the future “that is what we decide to do in our project there is extra functions that we will added later.
9. **Maintainability**
   * If any user facing a problem in the app he can easily report it and we will repair it.
10. **Modifiability**
    * Some features can be modified easily like the shape of the schedule.
11. **Safety**
    * The data of any user will be completely save as it will not access by any other user.
12. **Open-source**
    * The source code will be available for any one.
13. **Usability**
    * Simple UI that help the user to use its functions easily.

### **Future work**

1. **Moderator students**

There will be extra materials courses which will include the materials uploaded by the moderator students. These resources can be also files or links such as summaries or links to useful websites or videos.

Delete any forum post if they find that it was inappropriate such as unrelated to the course “or asked before (extra)”

1. **Upload task solution**

Tasks can be set to require students to upload their solutions. Students need to upload their solutions before due date. The course’s TAs & professors can download these solutions, correct them and add the grade to each student.

1. **Reminder notifications**

Students will get notifications regularly to remind them of their tasks that has a near due date.

The Notification system in the application will be divided into two subsystems.

**The first subsystem** is responsible for notifying the users for any **new** incoming changes on the application. This will include notifying the users for new announcements, new assignments, new tasks added, new quizzes and new exams.

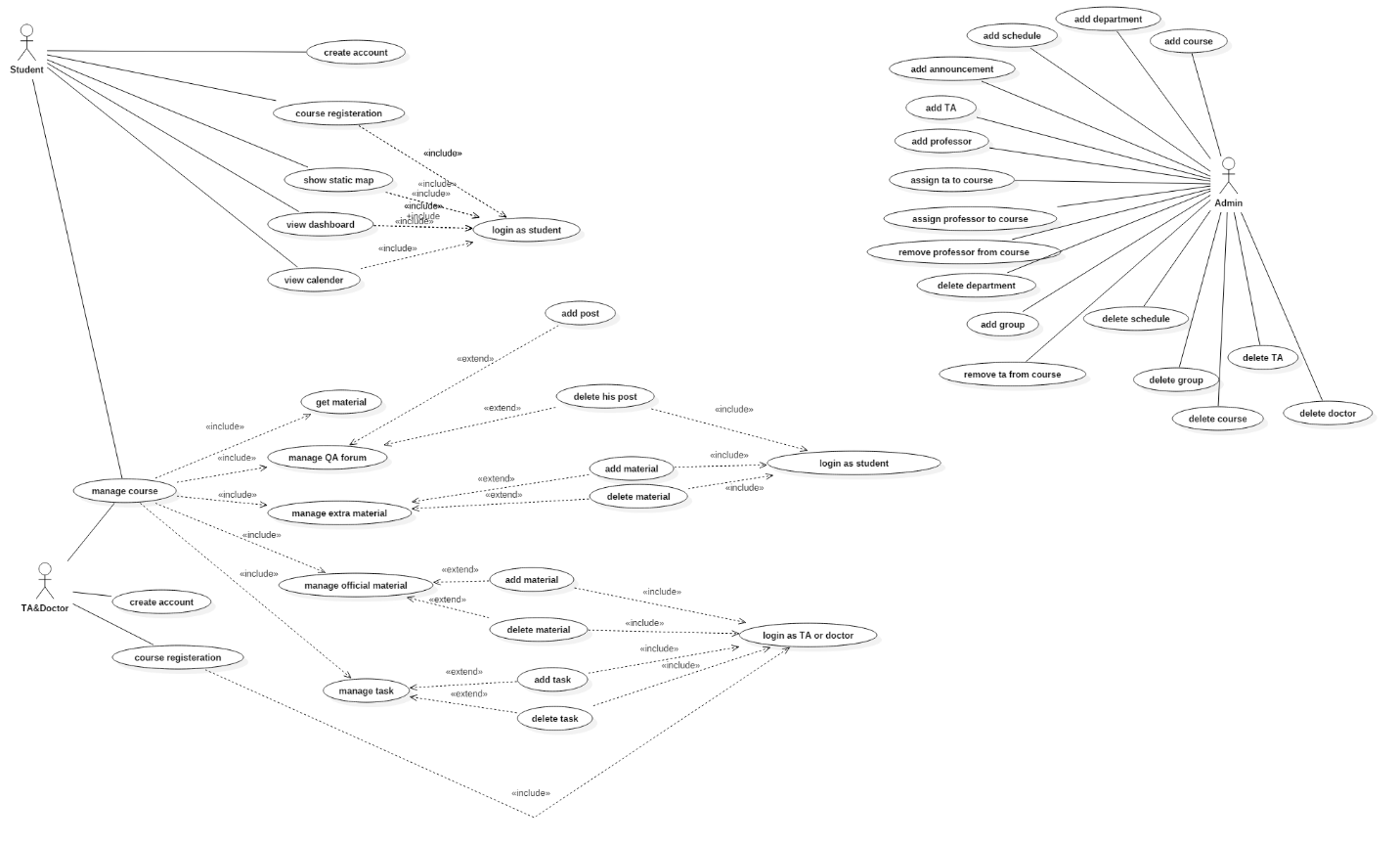
The notification system will not be a background service, but it will collaborate with the synchronization service on the client app. This sync service is responsible for listening to events on the backend. These events will be such as the arrival of a new announcement or the addition of a new assignment on the courses that the user is registered in. The sync service will update the application and calls the notification system to show a notification to the user.

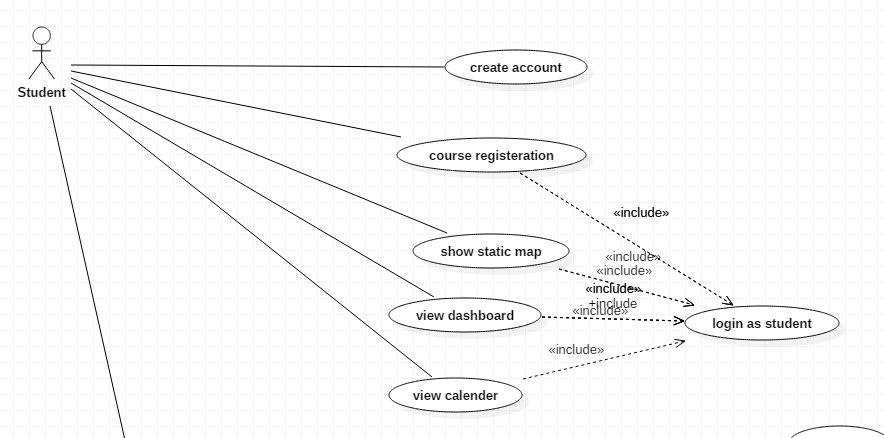
**The second subsystem** is a reminder background service that is responsible for showing reminders notifications. Reminder notifications will be managed locally without the need for any backend sync (very similar to an alarm application). Take this example: If a user has an assignment that will be delivered on Thursday the reminder service will send a notification before a week from the deadline, if the user hasn’t marked this assignment as done, the reminder will continue and sends a notification to the user before 4 days from the deadline, then again if it’s not marked as done it will send a notification to the user before 2 days from the deadline and so on it will continue to send notification before a day, then before 6 hours, … and so on until either the user either marks the assignment as done or the deadline is passed.

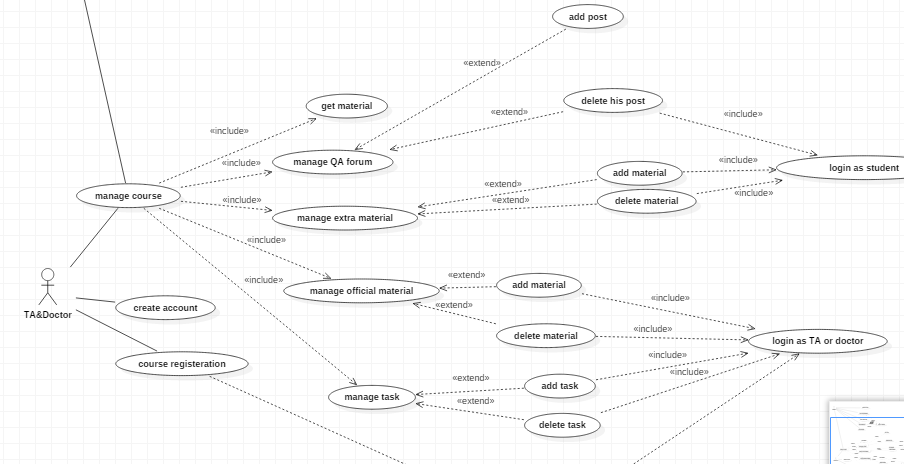
1. **Private messages**

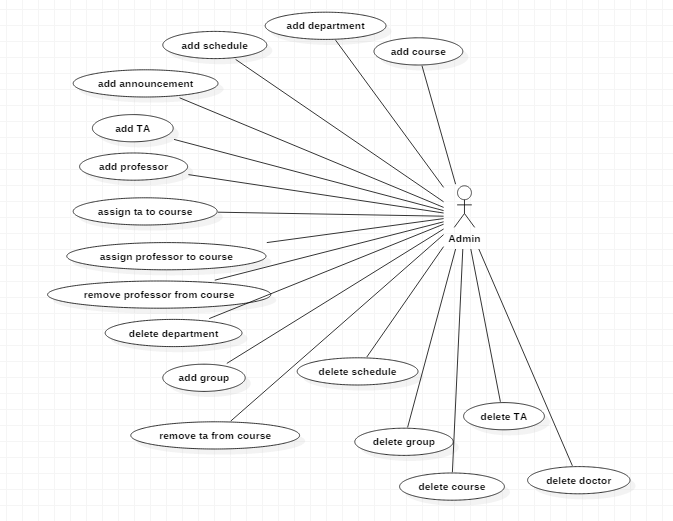
The app will allow users (professors, students or TAs) to send direct private messages to any other user. This is **not a chat service**. The messages will be similar to emails with a title, body & attachments such as images or raw files. The user can search for another user by his name or his ID in our system, then he can send a message to him as described above. Messages will be grouped in an inbox page and categorized and organized by the user it was sent to or received from (similar to chat rooms) so that all messages sent to or received from the same user appear in a single specific place inside the inbox.

## Use case Diagrams









## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 1 | |
| Use Case Name: | Login | |
| Actors: | TA/Doctor – Student – Moderator student | |
| Pre-conditions: | User register in the app and validate his account | |
| Post-conditions: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Flow of events: | **User Action** | **System Action** |
| * User Selects to login |  |
|  | * System returns Login page |
| * User enters his username * User enters his password |  |
|  | * System Verifies his information * System logs him in to his account |
| Exceptions: | **User Action** | **System Action** |
| * User Enter username and Password |  |
|  | * username does not exist or password is wrong * System rejects the Login * System demands the user to enter correct username and password |
| Includes: |  | |
| Notes and Issues: | Login will identify if the account corresponds to TA/Doctor, Student or Moderator student. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 2 | |
| Use Case Name: | TA/Doctor register | |
| Actors: | TA/Doctor | |
| Pre-conditions: | TA/Doctor has a valid academic E-mail | |
| Post-conditions: | TA/Doctor validate his account from the mail sent to his academic mail | |
| Flow of events: | **User Action** | **System Action** |
| * TA/Doctor select to register |  |
|  | * System returns registration page |
| * Enter the required information |  |
|  | * System check that neither of his data violate the specifications * System sent a verification code to the entered academic mail |
| * TA/Doctor enters the verification code |  |
|  | * System match the verification code * creates the account |
| Exceptions: | **User Action** | **System Action** |
| * TA/Doctor entered data that violate the specifications |  |
|  | * System recognize that the entered data violate the specifications * System demand the TA/Doctor to enter his data again in right manner. |
| Notes and Issues: | If TA/Doctor enter invalid E-mail he will not be able to validate his account so he will not be able to complete registration. | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 3 | |
| Use Case Name: | Course registration | |
| Actors: | Student | |
| Pre-conditions: | Login | |
| Post-conditions: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Flow of events: | **User Action** | **System Action** |
| * Student Show courses |  |
|  | * System return List of courses |
| * Student select a specific course |  |
|  | * System return course information and some choices |
| * Student Select register in this course |  |
|  | * System return a box to write course code |
| * Student enter course code |  |
|  | * System check that the course code is right * System add this course to Student courses |
| Exceptions: | **User Action** | **System Action** |
| * Student entered invalid course code |  |
|  | * System demand to enter course code again |
| Includes: |  | |
| Notes and Issues: | You should have the course code to register in the course | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 4 | |
| Use Case Name: | Add Material | |
| Actors: | TA/Doctor – Moderator student | |
| Pre-conditions: | Login – Assign to this course | |
| Post-conditions: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| Flow of events: | **User Action** | **System Action** |
| * User shows the courses he assigned to |  |
|  | * System return List of courses |
| * User select the course he wanted |  |
|  | * System return course information and some choices |
| * User select add material |  |
|  | * System return a box to upload material |
| * User upload material |  |
|  | * System assign the uploaded material to the course * System send notification to the students assigned to this course |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |
| Notes and Issues: | If you are student the add material button will not appear to you | |

System test cases

|  |  |
| --- | --- |
| **Name** | Assign Slots to Courses |
| **Scenario** | Admin login to his account 🡪 select assign Slots to course 🡪 select the course and enter the data 🡪 save |
| **Description** | It allow the admin to add schedule to the faculty courses “assign lab slots, section slots or lecture slots” |
| **Steps** | 1. Open admin website 2. Login with admin account 3. Select Assign Slots to Course 4. Choose course title & enter the data 5. Select save |
| **Data** | Day = Mon Start Time = 22:45:30 Duration = 90 Group ID = 2 Slot Type = 2 Place = Lab 7 Course Title = OS1 |
| **Expected Output** | The data has been added ! |
| **Actual Output** |  |
| **Name** | Assign Professor to Course |
| **Scenario** | Admin login to his account 🡪 select assign professor to course 🡪 select the professor and the course 🡪 save |
| **Description** | It allow the admin to assign professors to the faculty courses |
| **Steps** | 1. Open admin website 2. Login with admin account 3. Select Assign Professor to Course 4. Choose course title & professor username 5. Select save |
| **Data** | Course Title = ALG Professor Username = Abdallah |
| **Expected Output** | The data has been added ! |
| **Actual Output** |  |

|  |  |
| --- | --- |
| **Name** | Student Signup |
| **Scenario** | Student signup with valid data |
| **Description** | Student will insert his/her data to register to be able to use the app |
| **Steps** | 1-Open app “FCI-Campus”  2-Select student section  3-Select “No account yet? Create one”  4-Enter necessary data  5-Click Create Account |
| **Data** | First name = Ahmed  Last name = Ali  Email = [ahmed.ali1@gmail.com](mailto:ahmed.ali1@gmail.com)  Mobile number = 012\*\*\*\*\*\*\*\*  Date of birth = 1/1/1996  Username = ahmed.ali123  Password = 123456  Renter password = 123456  Faculty id = 20140001  Major department = General  Minor department = General |
| **Expected Output** | Account created successfully and redirect to Login Activity |
| **Actual Output** | Student Signup |

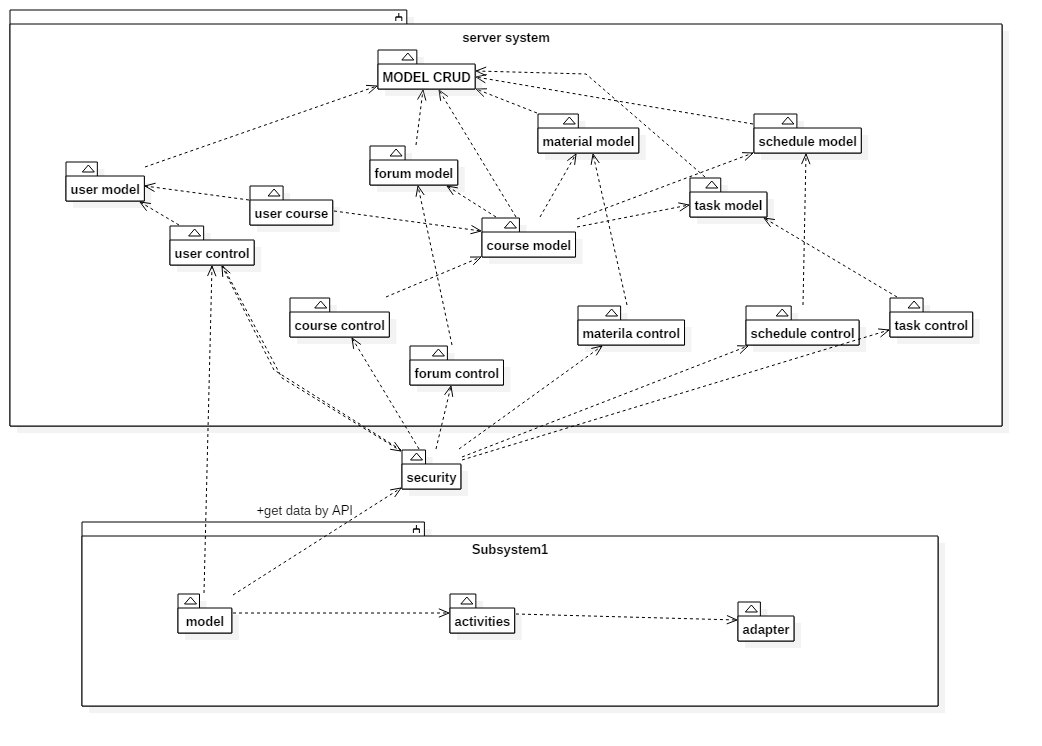
|  |  |
| --- | --- |
| **Name** | Student Join Course |
| **Scenario** | Student join course with valid  Course code and pass code |
| **Description** | Student will insert the course code and  Its pass code to join it |
| **Steps** | 1-Open app “FCI-ECampus”  2-Select My Courses Fragment  3-Click on “+” Button  4-Enter course code  5-Enter pass code  6-Enter group number  7-Click Join |
| **Data** | Course code = CS123  Pass code = ag123  Group number = 2 |
| **Expected Output** | Joined successfully and redirect to  My Courses Fragment  Adding to it the new joined course |
| **Actual Output** | Join Course |

|  |  |
| --- | --- |
| **Name** | Comment on a Post |
| **Scenario** | User comment on a forum  post |
| **Description** | User has ability to comment  on any post in course’s forum |
| **Steps** | 1-Open app “FCI-ECampus”  2-Select My Courses Fragment  3-Select particular course  4-Click Open Forum  5-Select any post  6-Type the comment  7-Submit the comment |
| **Data** | Comment = hello |
| **Expected Output** | Comment added to the post and  be visible to other users |
| **Actual Output** | Comment |

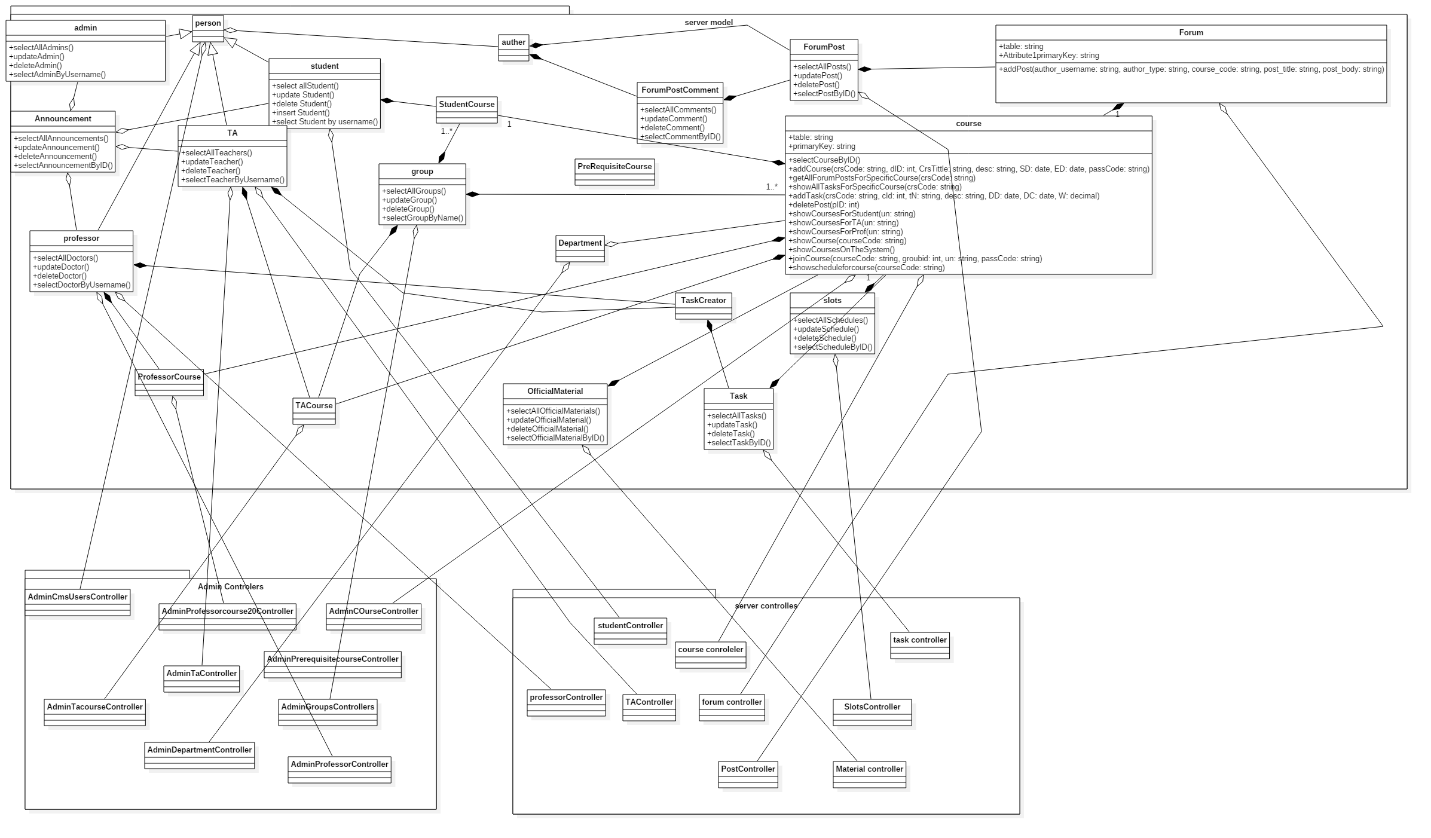
|  |  |
| --- | --- |
| **Name** | Faculty Map |
| **Scenario** | User open faculty map |
| **Description** | Student, TA or Professor can  look at the map of the faulty |
| **Steps** | 1-Open app “FCI-ECampus”  2-Select Faculty Map Fragment |
| **Data** |  |
| **Expected Output** | Simple Image of the faculty  buildings is being showed |
| **Actual Output** | Faculty Map |

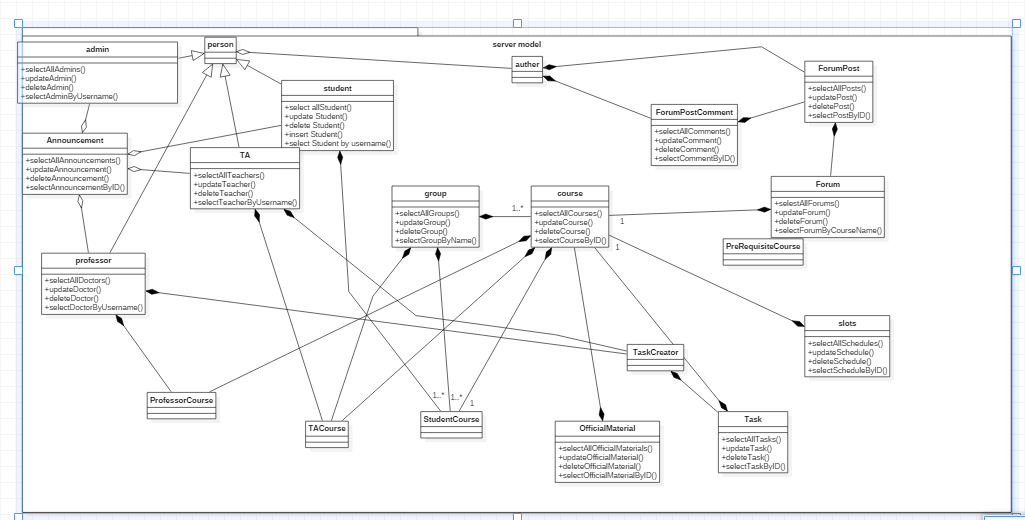
# System Design

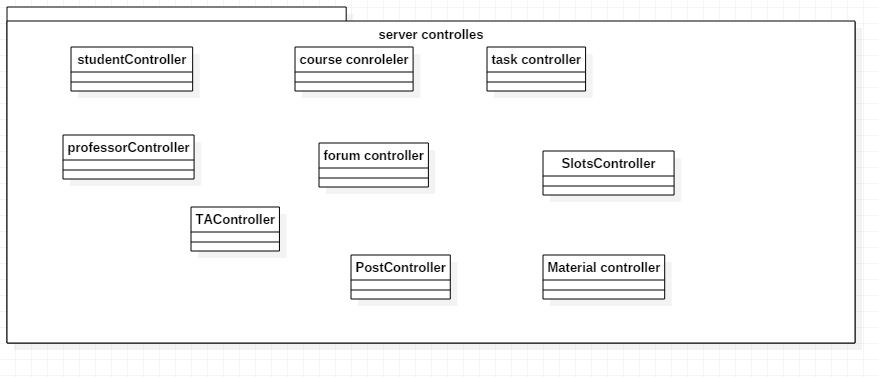
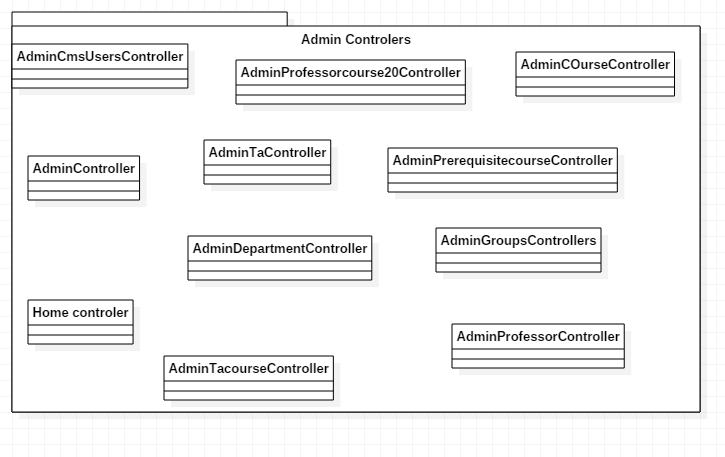
## System Component Diagram



## System Class Diagrams

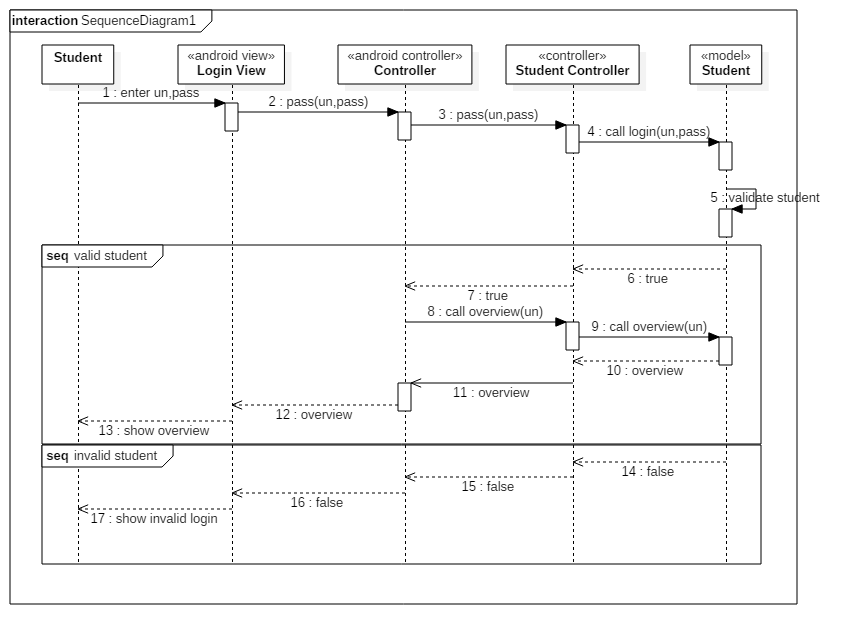




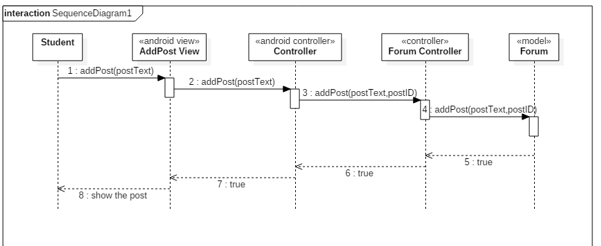


Sequence Diagrams

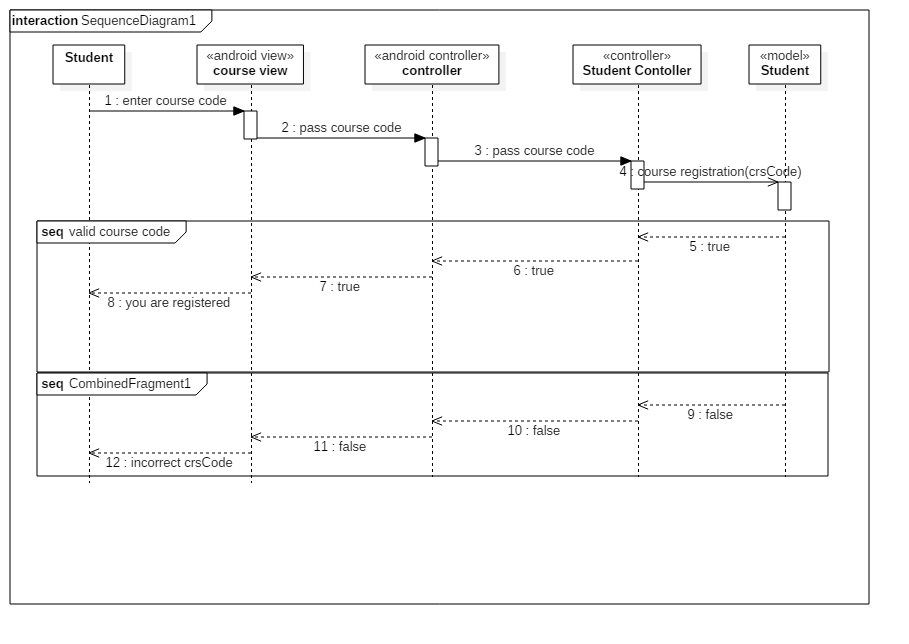
**Student Login**



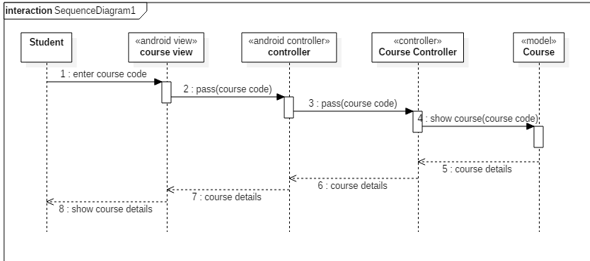
**Add Post (Student)**



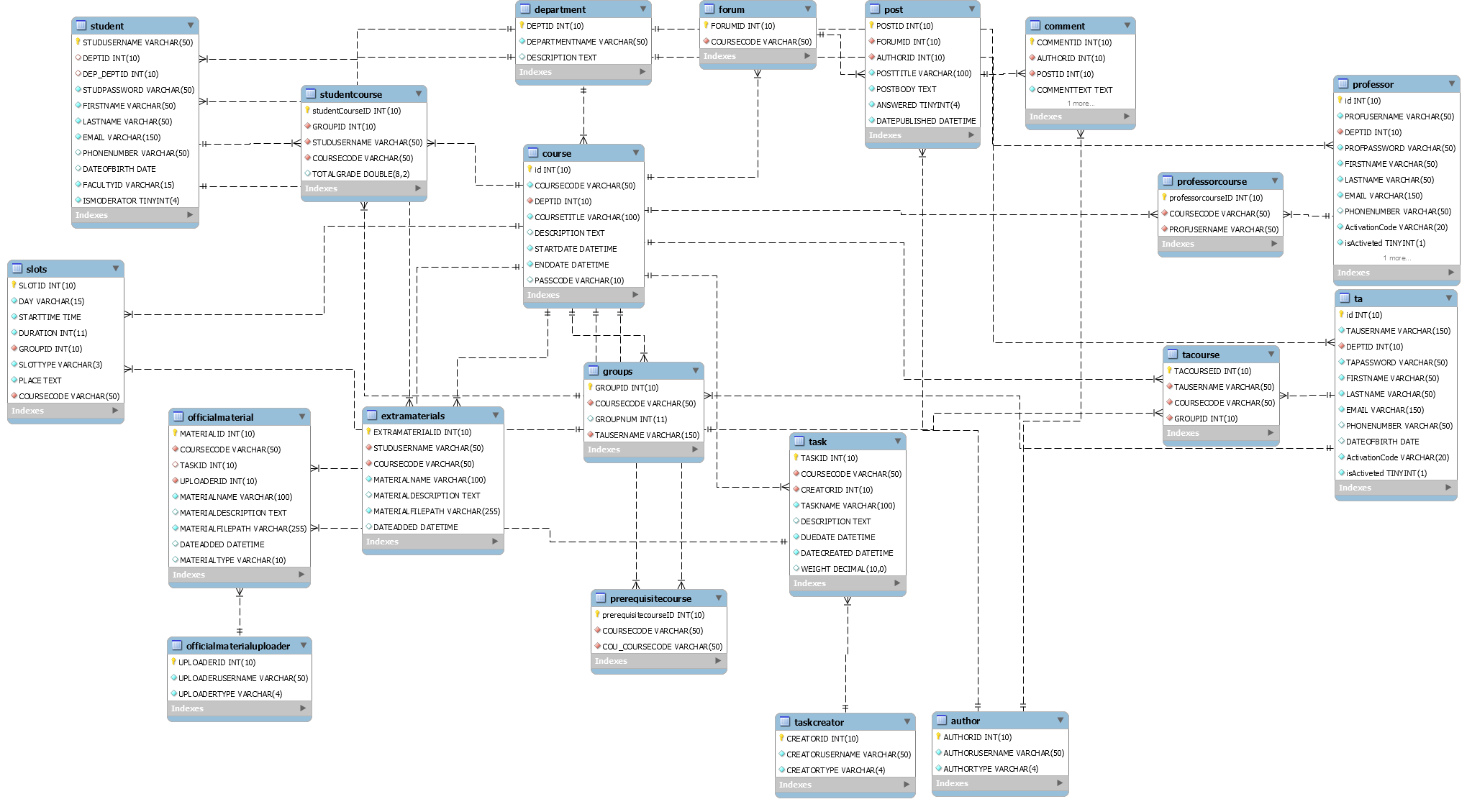
**Course Registration (Student)**

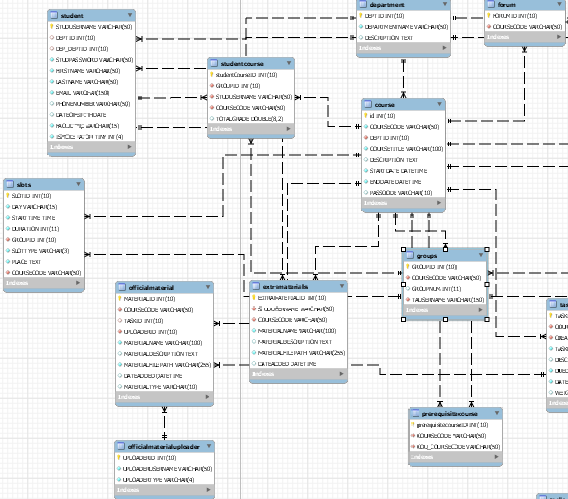


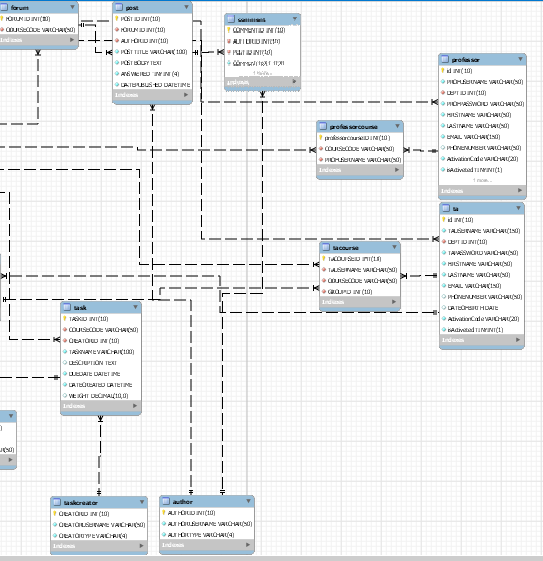
**Show Course Details**



## Project ERD







## System GUI Design

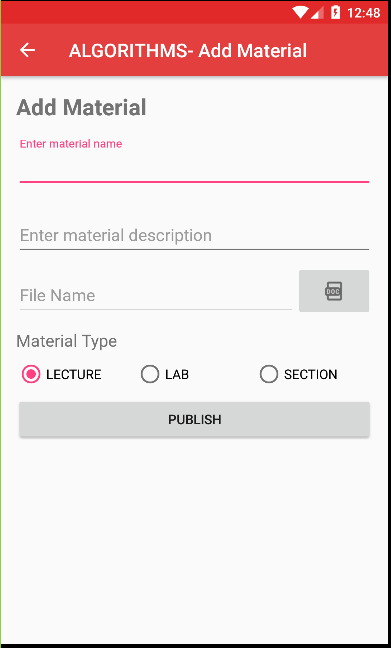
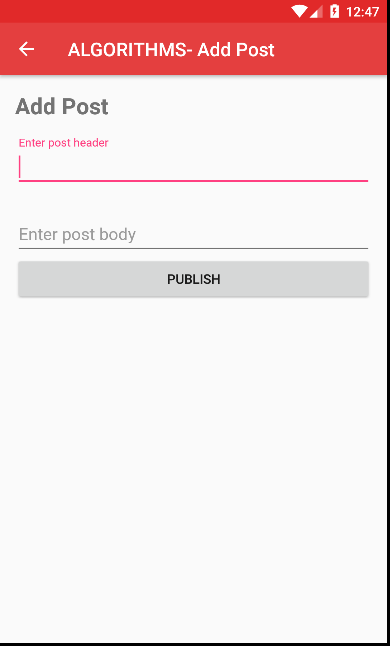


Figure Add post

Figure Add Material

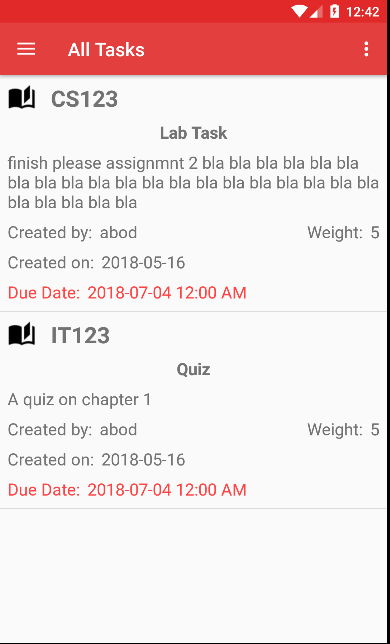
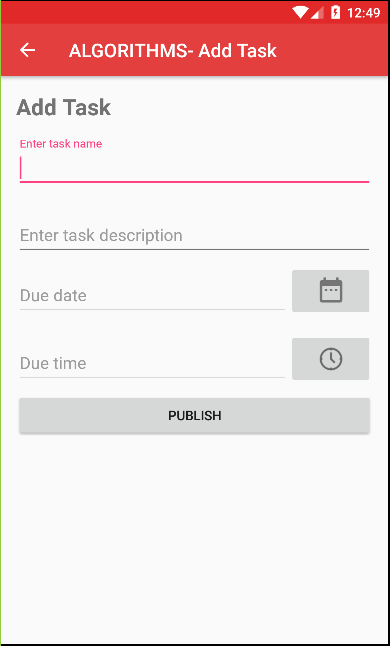


Figure Add task

Figure All tasks

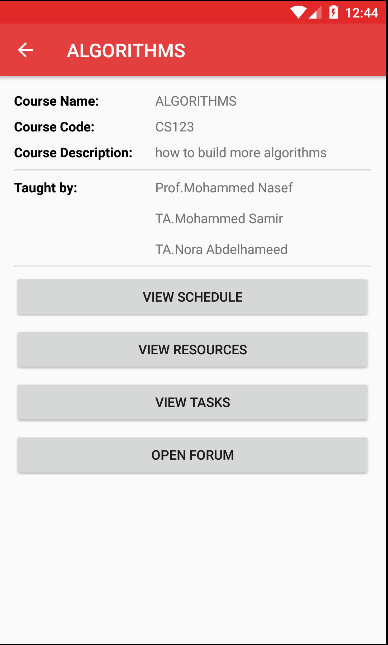
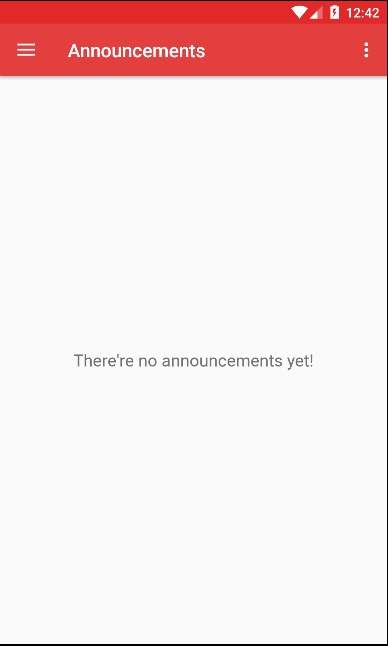


Figure Announcements

Figure Course

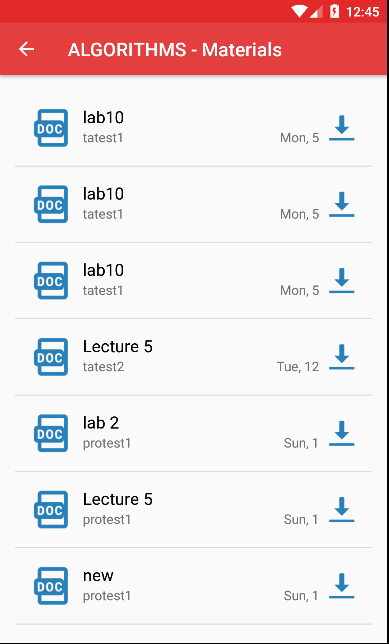
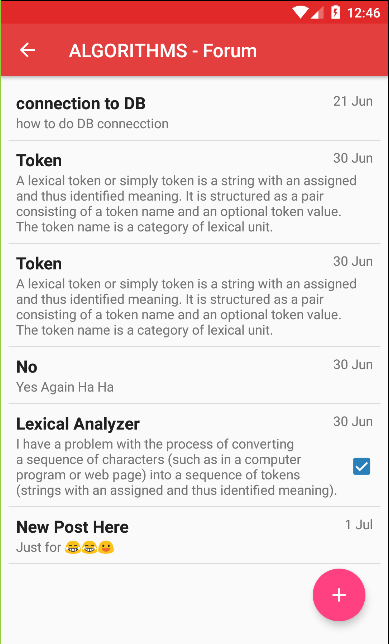


Figure Course Forum

Figure Materials

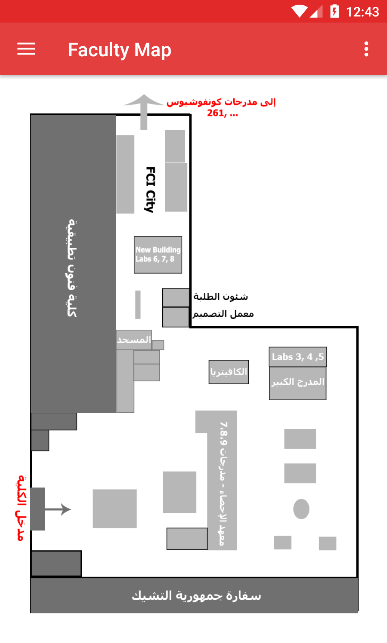
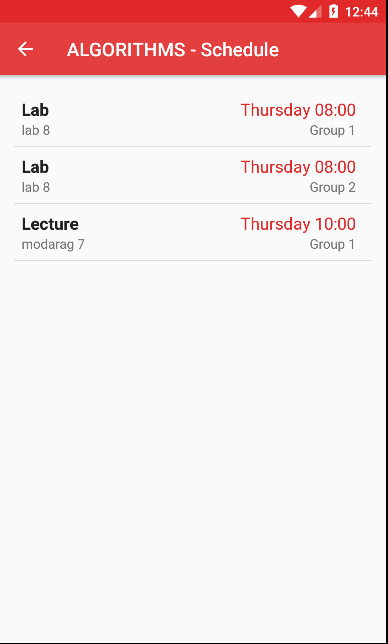


Figure Course Schedule

Figure Static Map

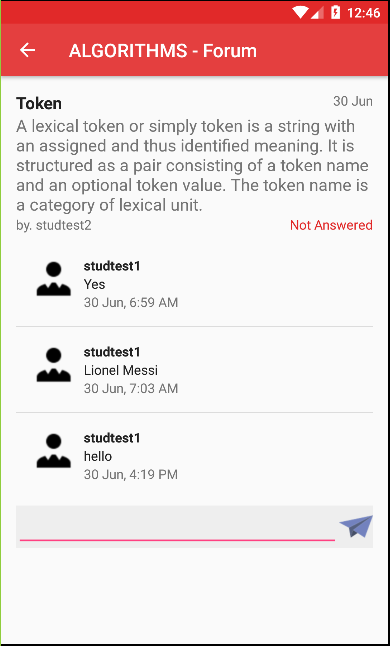
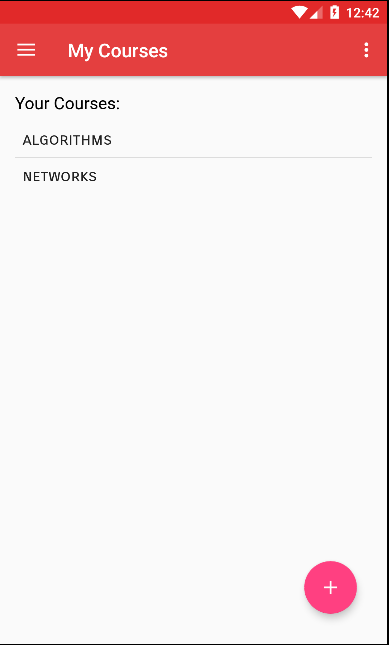


Figure My Courses

Figure Forum Post

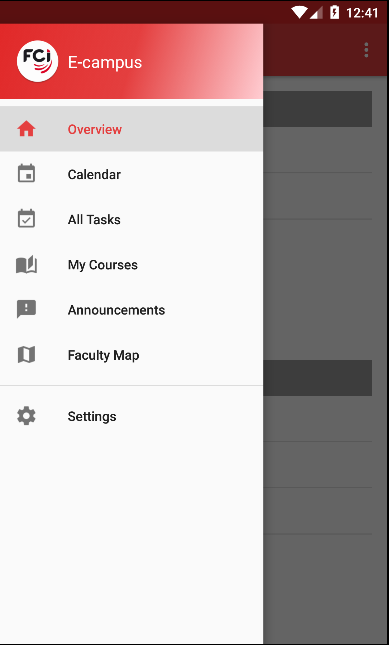
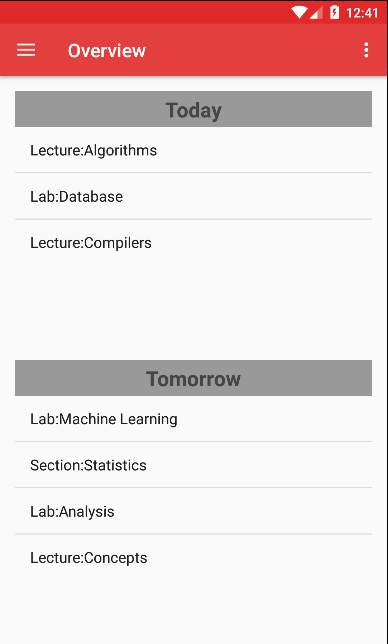


Figure Overview

Figure Navigation Drawer

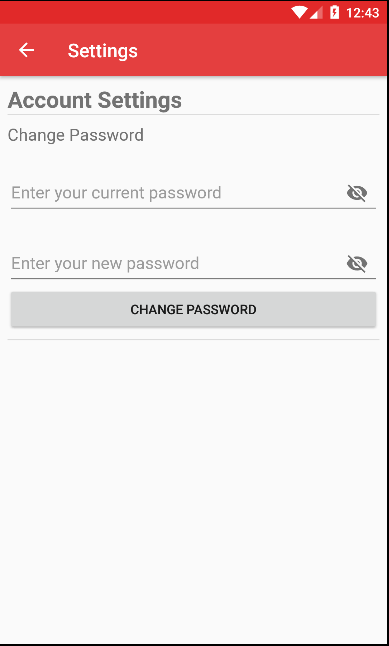
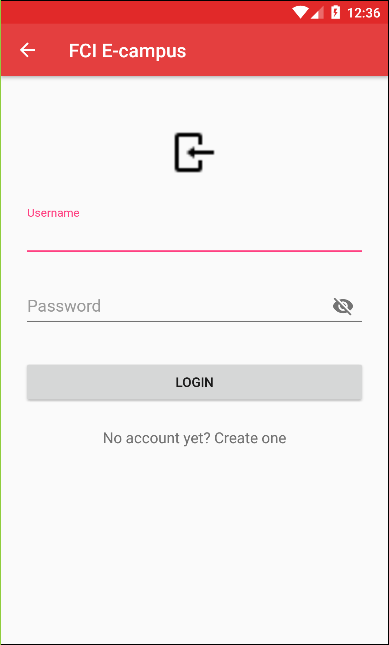


Figure Login screen

Figure Settings

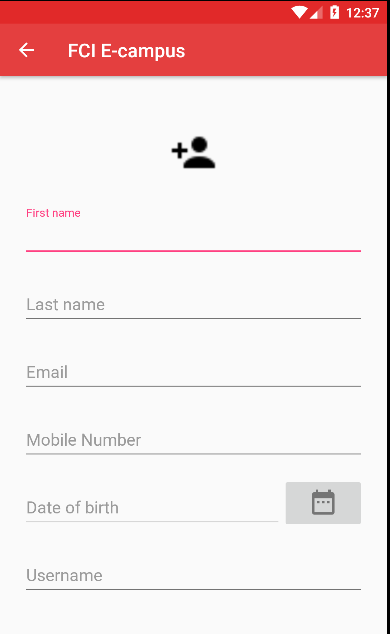
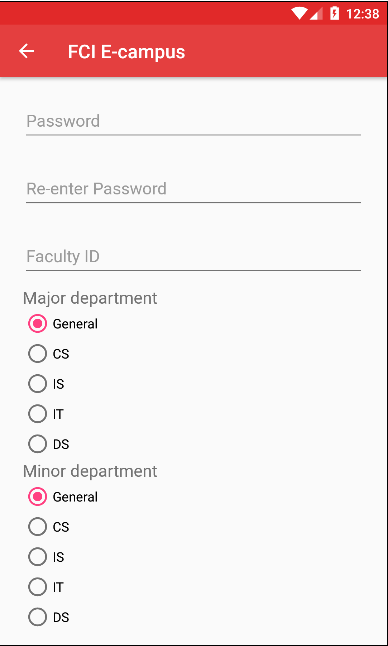


Figure Student signup 2

Figure Student signup 1

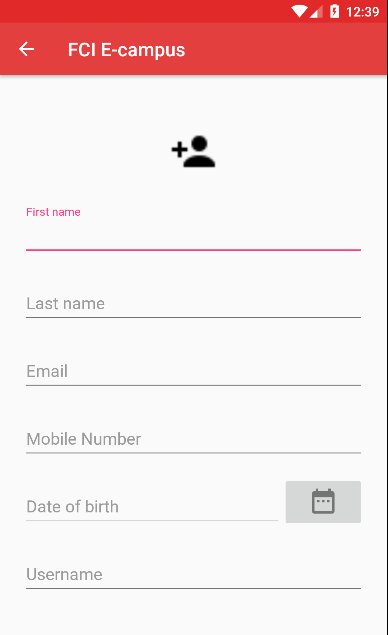
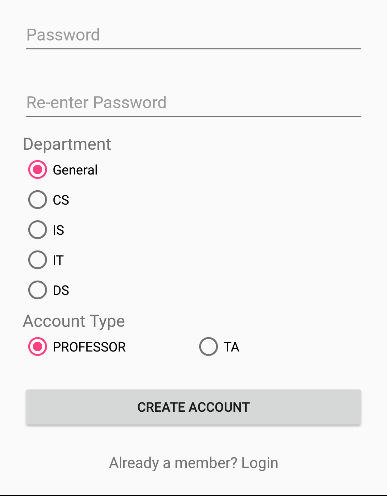


Figure Teacher signup 2

Figure Teacher signup 1

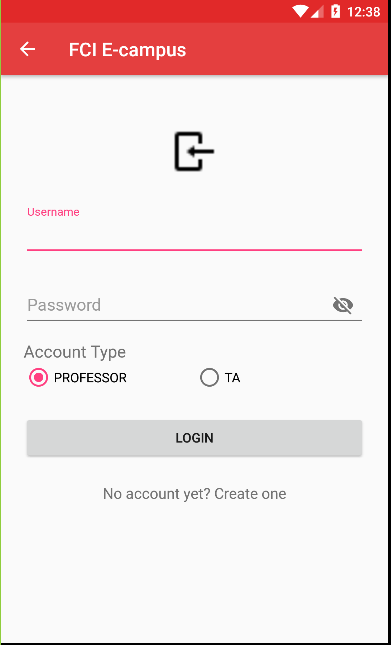
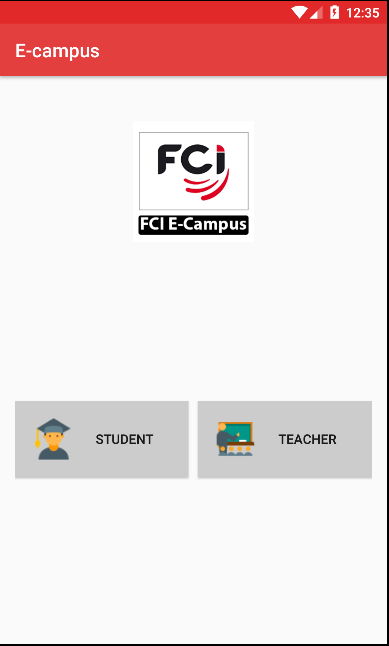


Figure Welcome screen

Figure Teacher login

# Implementation and Testing

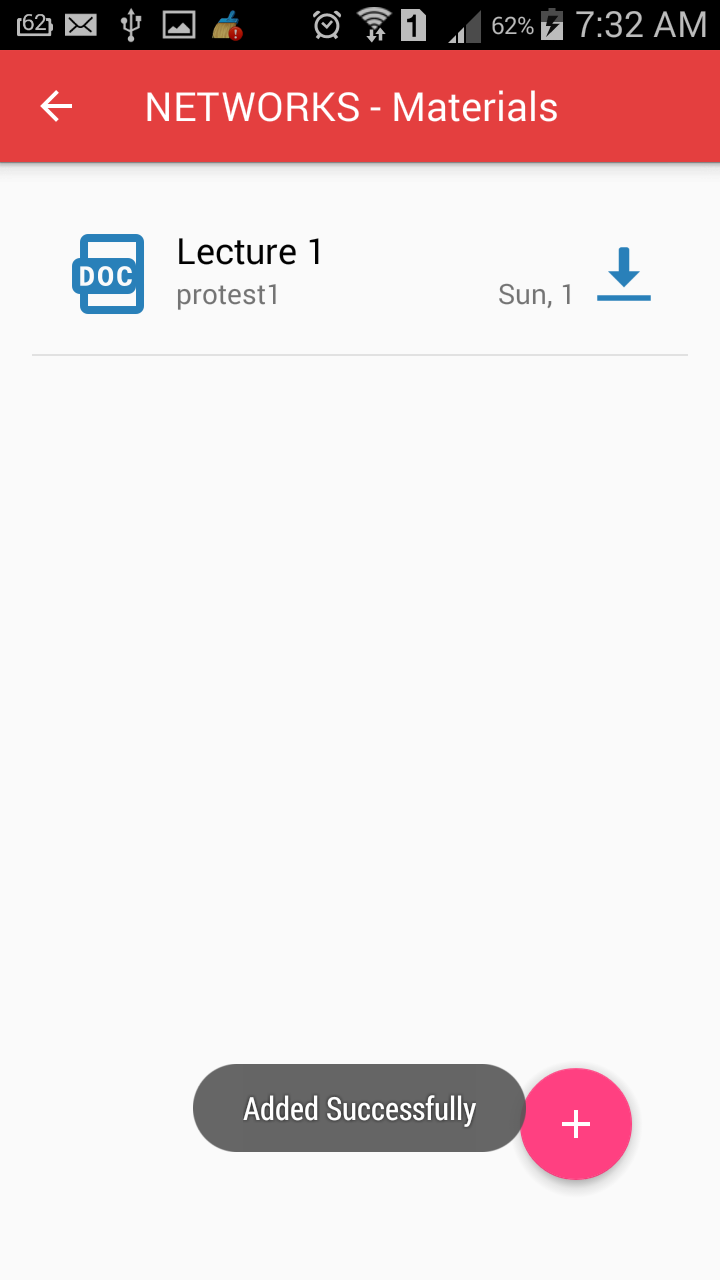


Figure Add Material - OUT

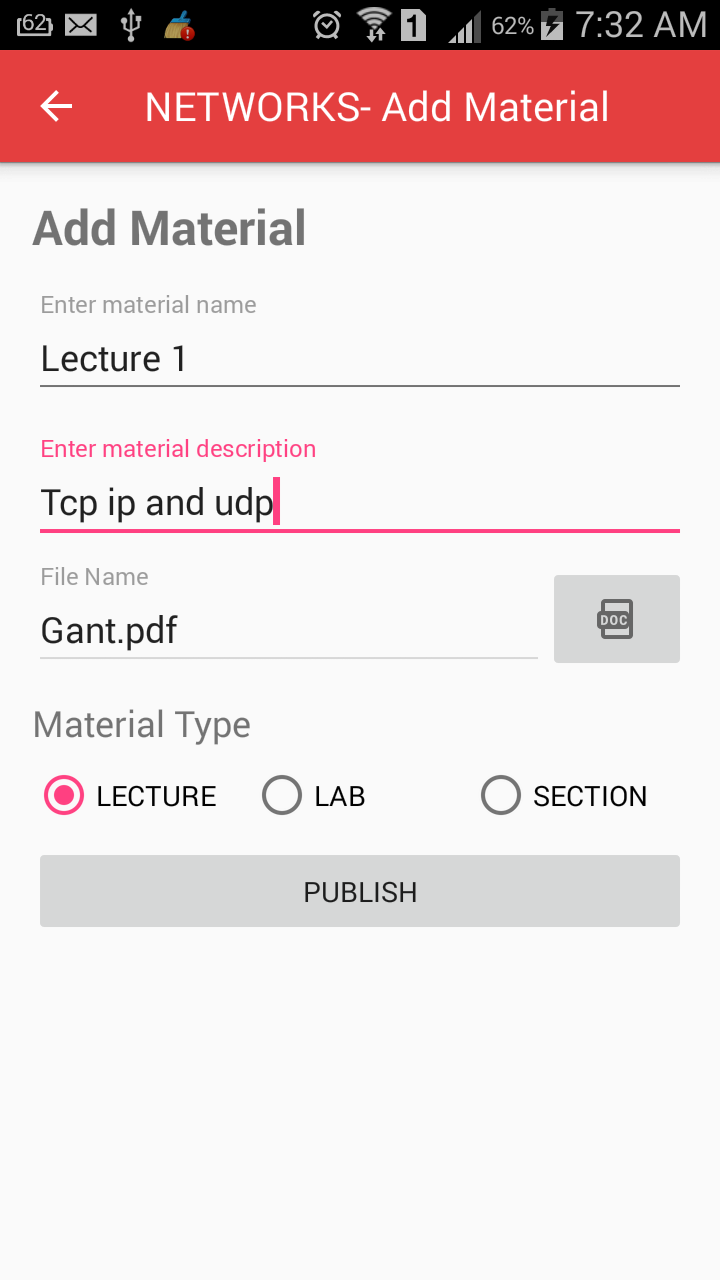


Figure Add material - IN

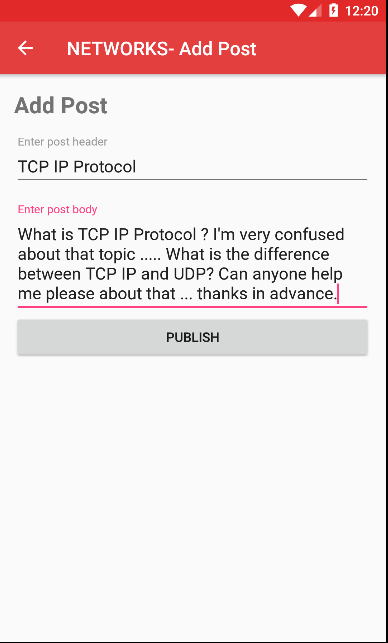
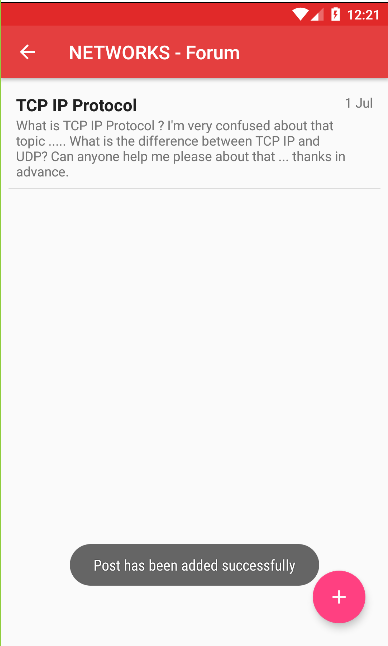


Figure Add post - Out

Figure Add post - IN

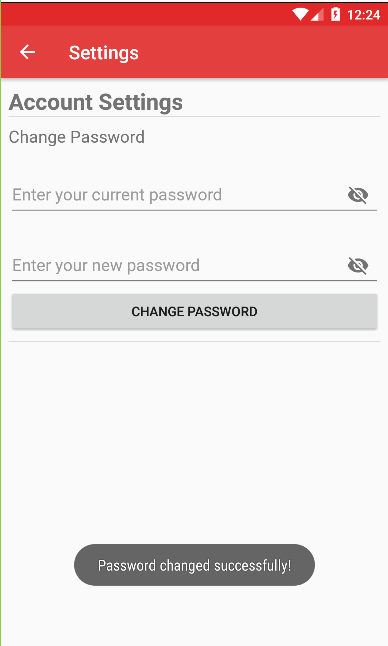
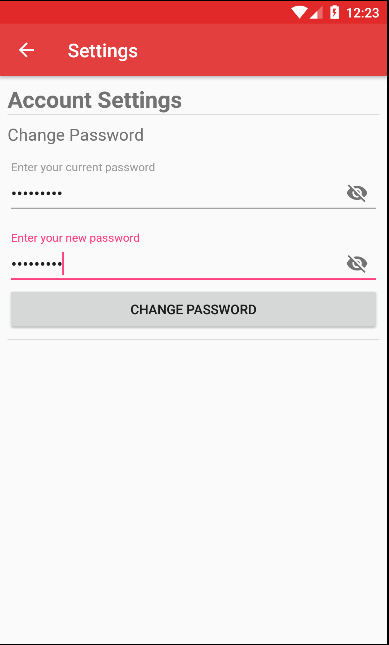


Figure Change password - OUT

Figure Change password - IN

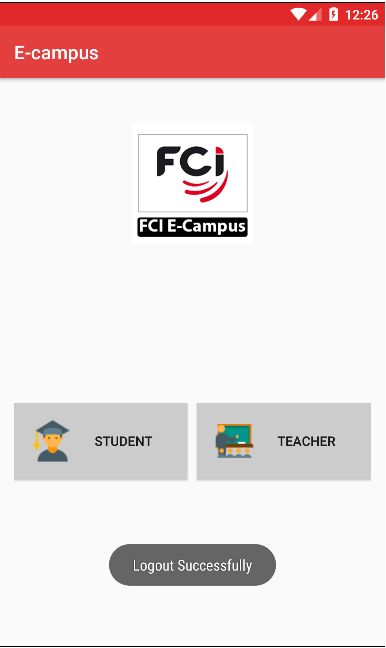
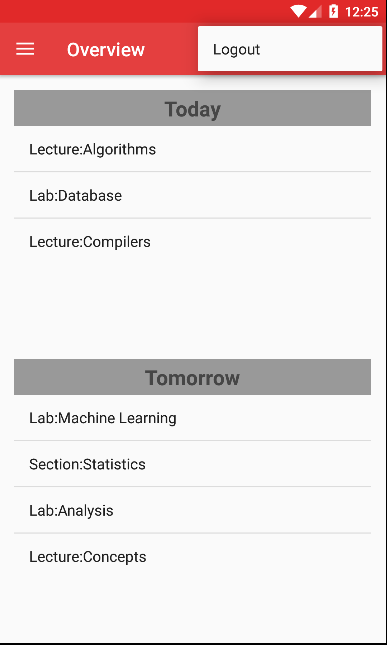


Figure Logout - IN

Figure Logout - OUT

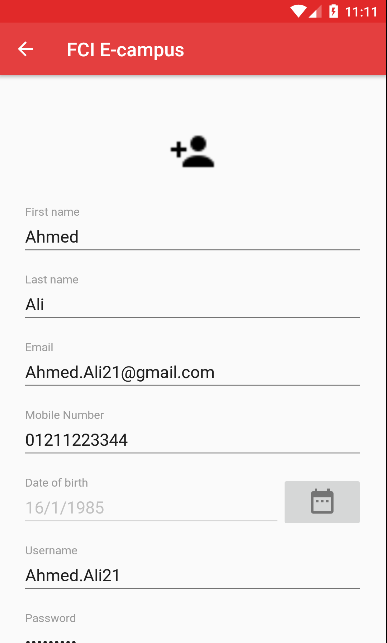
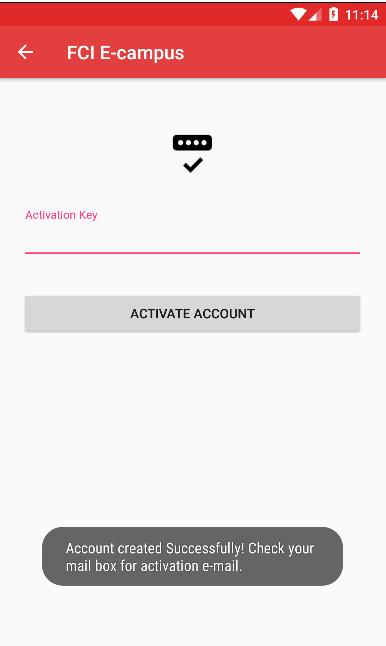


Figure Professor signup - OUT

Figure Professor signup 1 - IN

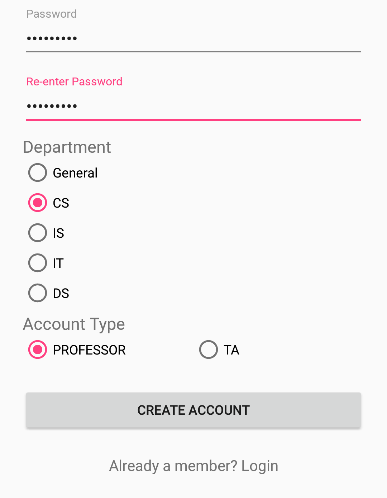


Figure Professor signup 2 - IN

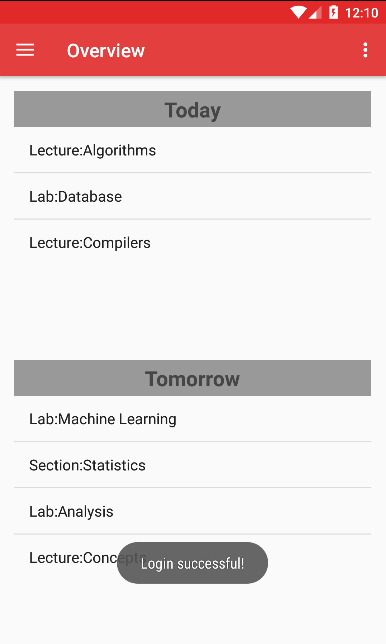
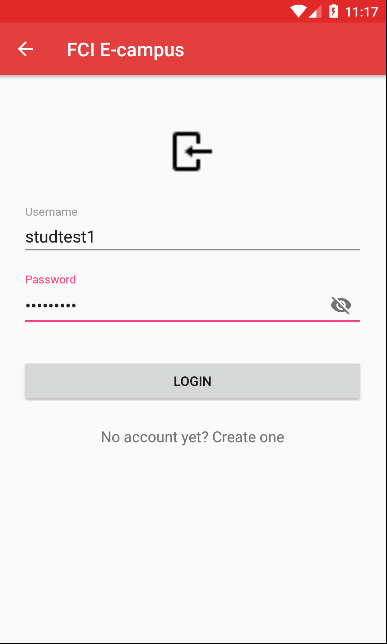


Figure Student login - IN

Figure Student Login - OUT