**Complete Report On “Tuition Finder”**

By- Abid Saleh, CSE, CU.

Session: 2014-15

ID: 15701050

INDEX

Chapter 01 Problem Definition Document

* 1. Introduction
  2. Problem Statement
  3. Project Objectives
  4. Preliminary Solution
  5. Project Scopes
  6. Old project Vs Our Project
  7. Conclusion

Chapter 02 System Requirement Specification

2.1 Introduction

2.2 Overviews of System

2.3 User Requirements

2.3.1 Functional Requirements

2.3.2 Non-Functional Requirements

2.4 System Requirement Specification

2.4.1 Functional Requirements

2.4.2 Non-Functional Requirements

2.5 Scenarios

2.6 Use Cases of System

2.7 Use Case Diagram

2.8 Conclusion

Chapter 03 Designing

* 1. Introduction
  2. Activity Diagram
  3. E-R diagram
  4. Class Diagram
  5. Sequence Diagram

3.6 Conclusion

Chapter 04 Implementation and user manual

* 1. Introduction
  2. Main Page
  3. Login Page
  4. Common Interface page
  5. Search Page
  6. Tuition List Page
  7. Update & Delete Window
  8. Post Page
  9. Conclusion

Chapter 01: Problem Definition Document

**1.1. Introduction:**

**Tuition Finder** is an Android based app. By this app one can find tuition based on his locationand subject preference. Also guardians can advertise to find suitable teachers within their budget.

This report illustrate about the project. It includes six sections. They are 'Problem Statement', 'Project objectives', 'Preliminary Solution', 'Project Scopes', 'Old projects Vs our project' and 'Conclusion' respectively. Problem statement describes what problem we are going to solve. Project Objectives illustrates how this project can solve the problem. In Preliminary Solution section, it deals with the solution idea. In Project scope, some features of the project is discussed. briefly. Finally in Old projects vs our project defines comparison between our project and existing projects.

**1.2. Problem Statement:**

Suppose a person 'X' is currently in higher studies or looking for job. Now he has to bear more or less of his expenses. Tuition finder will help him find a tuition where he can use his educational skills and experiences.

Now, let a person ‘Y’ is a concerned guardian or student who is desperately seeking for someone with proficient skills and knowledge to help in his study.

This app will help them finding what they want. This app will also facilitate them with the privilege of finding tutor/tuition associated with subject preference, location, salary etc.

And its best feature is that it’a a cost free medium introduce the guardian with the tutor.

**1.3. Project Objectives:**

This app will help one to find tuition close to his residence and suitable in terms of subject preference and salary.

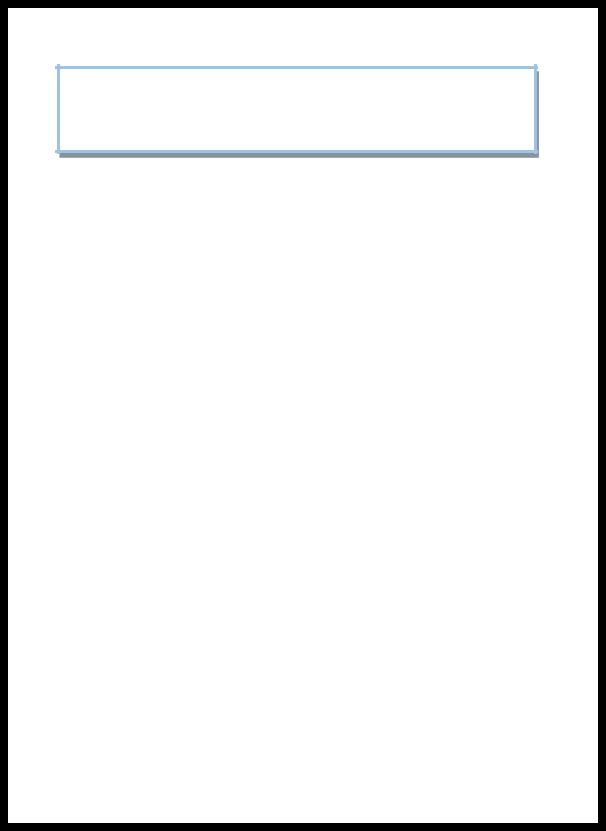
It will also help one to find best tutor within limited budget capability.

**1.4. Preliminary Solution:**

To solve this problem we can develop a system which takes information regarding the educational requirements, exact location, suitable salary, tuition days in a week and so on. It will facilitate the communication between two potential parties in a tuition. The guardian can also state which things are negotiable and which are mandatory.

**1.5. Project Scope:**

This will be an android based system. The system will have the features like – User statistics which will show what kind of educational background is required for particular level of tuition, report misuse, search tuition by location, search tuition by subjects or even by multiple keys, prompt for contact using id, creating user, save the information, mark as booked in which case system will delete the tuition information.



User Creation



User Statistics



Search by locations



Search by multiple keys



Report Misuse

Figure 1.1 : Project Scope

**1.6. Existing system Vs Our system:**

The existing system for communication between guardian and tutor is via tuition media or by passing the tuition news to known people. The first one is not so dependable though they charge a big percentage of tuition fee from both teacher and tutor. And sometimes they are unprofessional-tend to pass high value tuition to less qualified tutors. Also they can’t cover 100% of the potential tuition opportunity. The second one covers much less. So the probability of finding a good tuition is quite low here.

The special feature of our project is that it can cover all the tuition opportunities anywhere anytime, it is completely free of introduction-charge and it presents equal opportunities for each candidate, therefore making the best candidate aware of the opportunity.

Besides, our app presents functionality to prevent misuse.

**1.7. Conclusion:**

We hope that our app will be user friendly so that all classes of people can use this app. It will help one to find and provide tuition opportunities with ease and free of media charge.

Chapter 02: System Requirement Specification

**2.1. Introduction**

**Tuition Finder** is an Android based app. By this app one can find tuition based on his location and subject preference. Also guardians can advertise to find suitable teachers within their budget.

This report illustrates about the project. It includes five sections. They are 'Overview of the system', 'System’s Scenarios', 'System’s Use case', 'Conclusion', and 'Appendix' respectively. Section 2 discuss about the user requirements document that specifies what the user expects the software to be able to do. It also illustrates about the system requirements specification such as a structured document setting out detailed description of the system service. The system architecture, the actors of the system, statements of service that the system should provide, how the system should react to particular inputs and how the system should behave in particular situations (functional requirements) and constraints on the services or functions offered by the system (non-functional requirements) are also discussed in section 2. In section 3, system’s scenarios that illustrates some interaction with a proposed system. Section 4 is on system’s use case that identifies the actors involved in an attraction and the names of the interaction.

**2.2. OVERVIEW OF SYSTEM**

The proposed app on **Tuition finder** will help one to find tuition close to his residence and suitable in terms of subject preference and salary. It will also help one to find best tutor within limited budget capability. In this app there are two actors, Tutor and Provider. User requirement definition, system requirement definition, functional and non-functional requirements of the proposed system are given.

**2.3. User Requirements**

The success of a project lies on the satisfaction of users. To develop a system we first need to find out the requirements of the different users so that full satisfaction of users can be attained which is a measure of successfulness of a project. Requirements are description that the system should do under some constrains. User requirements are statements in natural language along with diagrams of what services the system is expected to provide to system user and the constrains under these services. We have met different types of users of our system. We met different user groups of our system and discussed with them about their requirements. User requirements are as follow.

**2.3.1Functional Requirements:**

1. System shall provide general users to sign up as provider.  
   2. System shall provide general users to sign up as tutor.  
   3. System shall facilitate providers to post tuition advertisement.  
   4. System shall facilitate tutors to search tuitions based their preferences.

5. System shall facilitate tutors to get contact information of chosen provider.  
6. System shall facilitate providers to update tuition.

7. System shall facilitate providers to report abusive tutors.

**2.3.2 Non-Functional Requirements:**

1. System will be fast and user friendly.  
2. System will be responsive in nature.  
3. System will be secure and reliable.

**2.4 System Requirement Specifications**

System requirements contains detailed description of the system’s functions and services along with optional constrains. It defines exactly what is to be implemented. We analyzed the story of the user requirements to find out the requirement of the system.

The requirements of the system can be categorized into two categories, namely:

1. Functional Requirements
2. Non-functional Requirements

***2.4.1 Functional Requirements:*** Functional requirements are description of services. Besides it also includes how system will react for particular input and behave in particular situations. It is supported by non-functional requirements. The functional requirement of our system is given below:

1. The System shall avail user to sign up as provider into the system:

1.a Any person throughout the country can open an account as a provider in our system by providing his/her basic information.

1. The system shall avail user to sign up as a tutor into the system.

2.a Any user throughout the country can open an account in our system as a tutor by providing required information.

1. The system shall facilitate provider to post tuition advertisement.

3.a Provider can post advertisement for a tuition by giving detailed information of the tuition and requirement of his/her desired tutor.

1. The system shall facilitate tutors to search for providers.

3.a Tutors can search for desired tuition by selecting locality and other desired requirements of tuition.

1. The system shall facilitate tutors to get contact information of his/her desired tuition.
2. The system shall facilitate providers to update tuition status.

6.a Provider can update tuition status by marking it as booked.

7. The system shall facilitate the providers to report any tutor for any misbehave.

***2.4.2 Non-functional Requirements:*** Non-functional requirements are the criteria that can be used to judge the operation of a system. These criteria defines some constrains on the services and functions of the system. We have met user groups of our system and tried to find out non-functional requirements. These are enlisted below:

1. The system shall respond quickly, consume less memory.

1.a The system shall start within 1 second.   
1.b The system size shall not exceed 20MB.  
1.c The system UI shall be user friendly.

1.d The search result shall have to be shown in less than 1 second.

1.e. The duration of receiving contact information should not be more than 1 seconds.

1. The system shall fit screen of any type of device such as smartphone, tab etc.

2.a The system shall support down to Gingerbread version of Android.

1. The system shall require internet connection for viewing information.

3.a Acquires and updates information regarding providers list and their availability from software server via internet.

1. The system data should be secure and reliable.

4.a The system shall be amenable to the rules and regulations.

4.b The data shouldn’t be modified without login.

4.c The data should not be contradictory.

* 1. **Scenarios**

A scenario is a narrative of measurable interactions of user roles (in the UML known as ‘actors’) and the technical system, which is usually, includes computer hardware and software. It describes one way that a system is or is look upon to be used in the context of activity in defined time frame. The system in our Mr. Provider scenario is given below.

***2.5.1 Initial Assumptions:***

The user has an active smartphone and out system is installed in the phone.

***2.5.2 Normal:***

The user clicks on the menu icon. Installed apps will be shown. User browses and clicks on tuition finder icon. The system will be launched. Welcome message is shown for a moment and options will be shown for both provider and tutor. For both provider and tutor app has separate profile. Both users can view his/her profile through login. In provider’s profile there is a “See Interested tutors” button to view interested tutors. Tutor’s profile has “SEARCH” button to find tuitions in a particular area and “Get contact” button to communicate with a of a specific provider.

***2.5.3 What Can Go Wrong:***

The system may wrong show info. User should restart the application. Network connection may be off when rating application. User should open network connection to rate the application.

***2.5.4 Other Activities:***

User can only view the information but can’t modify unless login into the system.

***2.5.5 System State on Completion:***

Appropriate page and message is shown for particular task.

**2.6. Use Cases of System**

A use case represents the actors involved in an interaction. This is then supplemented by additional information describing the interaction with the system. The additional information may be textual description or one or more graphical models such as UML sequence or state charts.

We have found four main use cases of our system. These are listed below:

***2.6.1 Use cases of our system:***

1. Signing up as a provider.
2. Signing up as a tutor.
3. Posting advertisement.
4. Search for tuition.
5. Updating tuition status.

These use cases are described in detail in the following.

***UC1:*** Signing up as a provider.

***Actor:*** Provider.

***Precondition:***

1. System displays home page of Tuition finder.

***Main Success Scenario:***

1. System displays “I am Tutor” and “I am Provider” options.
2. User clicks on the “I am Provider” option.
3. System displays provider’s activity page.
4. System shows options “log in” and “register”.
5. User clicks on “register” option.
6. System displays registration page containing input fields: “Full Name”, “Username”, “Password”, “Re-enter the password”, “Phone no”, “Email Address”, “Locality”.
7. User fills up the fields with accurate information.
8. User clicks on “Confirm”.
9. System authenticates that information provided by user are valid.
10. System displays message “successfully registered”.

***Post Condition:*** User is successfully registered into the system as a provider.

***Alternative Course:***

2.a User clicks on “I am tutor” option.

2.a.01. Use case ends.

4.a User clicks on “log in” option.

4.a.01 System displays log in page containing input fields “Username” and “Password”.

4.a.02 User fills up the fields with accurate information.

4.a.03 User clicks on “Confirm” from “Confirm” and “Cancel”.

4.a.04 System authenticates username and password as valid.

8.a User clicks on “Cancel”.

8.a.01 Resume at 4.

9.a System authentication returns as the information are invalid.

9.a.01 System displays message “Invalid Information”.

9.a.02. Resume at 4.

***UC2:***  Signing up as a tutor.

***Actor:*** Tutor.

***Precondition:***

1. System displays home page of Tuition Finder.

***Main Success Scenario:***

1. System displays “I am Tutor” and “I am Provider” options.
2. User clicks on the “I am Tutor” option.
3. System displays tutor’s activity page.
4. System shows options “log in” and “register”.
5. User clicks on “register” option.
6. System displays registration page containing input fields: “Full Name”, “Username”, “Educational Background”, ”“Password”, “Re-enter the password”, “Phone no”, “Email Address”.
7. User fills up the fields with accurate information.
8. User clicks on “Confirm”.
9. System displays message “successfully registered”

***Post Condition:*** User is successfully registered into the system as a tutor.

***Alternative Course:***

2.a User clicks on “I am provider” option.

2.a.01. Use case ends.

5.a User clicks on “log in” option.

5.a.01 System displays log in page containing input fields “Username” and “Password”.

5.a.02 User fills up the fields with accurate information.

5.a.03 User clicks on “Confirm” from “Confirm” and “Cancel”.

5.a.04 System authenticates username and password as valid.

12.a User clicks on “Cancel”.

12.a.01 Resume at 4.

***UC3:*** Posting advertisement.

***Actor:*** Provider.

***Precondition:***

1. User is successfully logged in into the system as a provider.

***Main Success Scenario:***

1. System displays two options “Update status” and “Post a tuition”.
2. User selects “post a tuition”.
3. System displays dropdown menus “group”, “class”, “locality”, “days in a week” and input fields levelled as “salary”, “subjects”, “contact number” and “other description”.
4. User fills up the input fields and dropdown menus with appropriate information and then clicks on “submit” button.
5. System displays message “Successfully posted”.

***Alternative Scenario:***

1.a.01. User click on “Update status”.

1.a.02. Use case ends.

***UC4:***  Search for tuition.

***Actor:*** Tutor.

***Precondition:***

1. User is successfully logged in into the system as a tutor.

***Main Success Scenario:***

1. System displays dropdown menus “Locality”, “Group”, “Class Range”.
2. User selects from those dropdown menus according to his preference.
3. System displays the list of available tuitions of particular group and class range in a particular area.
4. User clicks on the “get contact” of provider of his/her choice.
5. System displays contact number of provider and his/her unique id to the provider.

***Post Condition:*** Contact number of provider is successfully delivered to tutor.

***Alternative Course:***

4.a User clicks on “Exit”.

4.a.01 Use case ends.

***UC5:*** Updating tuition status.

***Actor:*** Provider

***Precondition:***

1. User is successfully logged in into the system as a provider.

***Main Success Scenario:***

1. System displays two options “Update status” and “Post a tuition”.
2. User clicks on update status.
3. System displays options “See Interested tutors” and “Mark a tuition as booked”
4. User clicks on the “Mark a tuition as booked” option.
5. System shows list of tuitions posted by the provider.
6. User selects a particular tuition to mark it as booked.
7. System removes the associated tuition post from User Interface.

***Post Condition:*** System displays message “Post has been removed”.

***Alternative Course:***

1.a User clicks on “Post a tuition”.

1.a.01 Use case ends.

3.a User clicks on “See interested tutors”.

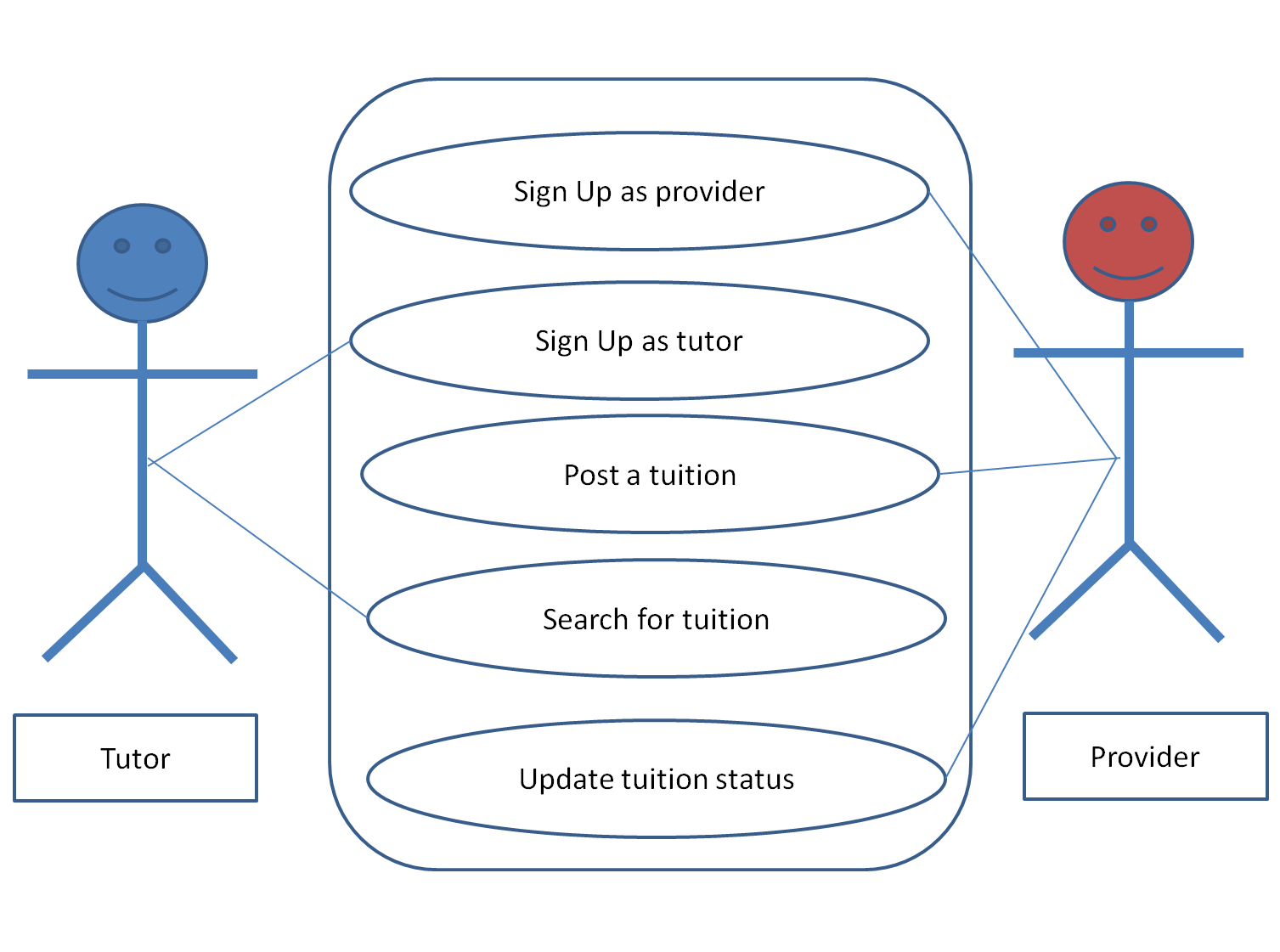
3.a.01. System displays list of tutors, their names, educational background and tutor id.

3.a.02. User selects a tutor id and clicks on “report this tutor”.

3.a.03. System displays “Successfully Reported”.

**6. Use Case Diagram:**

A use case diagram is the simplest representation of user's interaction with the system that is shown. It shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases. In our system there are two types of user, the provider and the tutor. Tutors have to sign up and looking for a suitable provider in a place and get booked for him/her. The providers have to sign up and check his appointment list and assign his appointment where it is needed. The use case diagram of our system is presented below:



**Figure 1: Use case diagram of Tuition Provider**

**2.8. Conclusion**

System Requirements Specifications document act as a communication medium between the client and the firm. When the system requirements specification is completed and is accepted by all parts, the requirement engineering phase is finished. It is a prerequisite of designing phase.

Chapter 03: Designing

* 1. **Introduction**

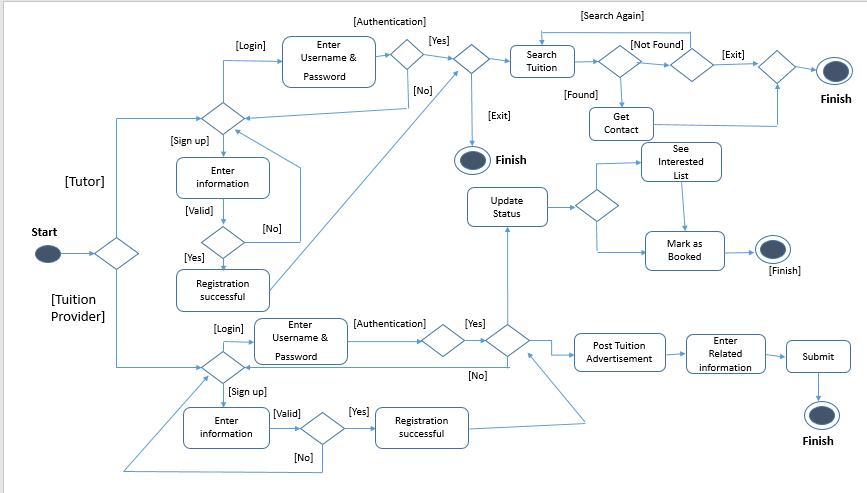
**Tuition Finder** is an Android based app. By this app one can find tuition based on his location and subject preference. Also guardians can advertise to find suitable teachers within their budget.

After completion of finding user requirements and analysing them, a detailed document is needed for starting implementations. A detailed design document (DDD) is a document that contains details of the system to be developed such as system objects, interaction between the objects, architectural pattern etc. It is a guideline for the coders to develop the software. It act as a bridge between the designers and the coders.

Six sections are included in this report. Introduction section contains a brief introduction the project. Activity diagram shows the various states of the system for activities committed by user. In the Class Diagram section, system classes and associations between them are shown. In the Sequence Diagram section, interactions between the system objects are shown. System entities and their relationship is shown in E-R Diagram. The conclusion section contains summery of this report.

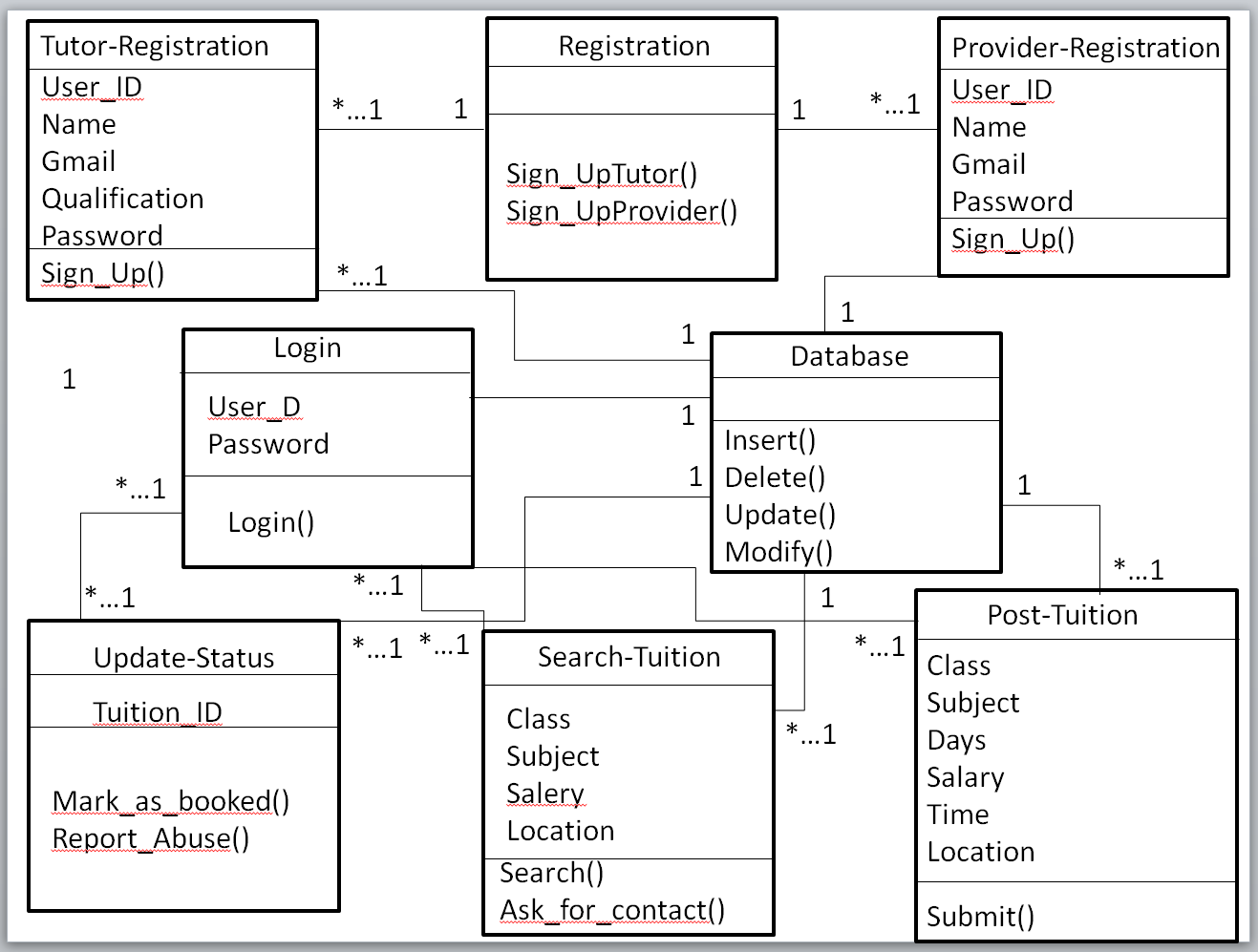
**3.2.1Activity Diagram**

Activity diagram is a diagram that shows the activities that make up the whole system. It shows the business processes and the activities involved in them. It also represents the flow of control. It is not necessarily matched with the code. It is mainly used by the business user. The activity diagram of our system is shown below.

 **Figure 1: Activity diagram of Tuition Finder**

* + 1. **Class Diagram**

In object oriented approach, class is a fundamental concept that represents objects of the system. Class diagram is used to show the necessary classes of the system and association between them. Association is a link between classes that represents that there is a relationship between the classes.



**Figure 2: Class diagram of Tuition Finder**

* + 1. **Sequence Diagram**

Sequence diagram is used to model the interactions between the actor and the objects of the system along with interactions between the objects themselves. Sequence diagram shows the sequence of interactions that take place during particular use case. Our system has five use cases namely,

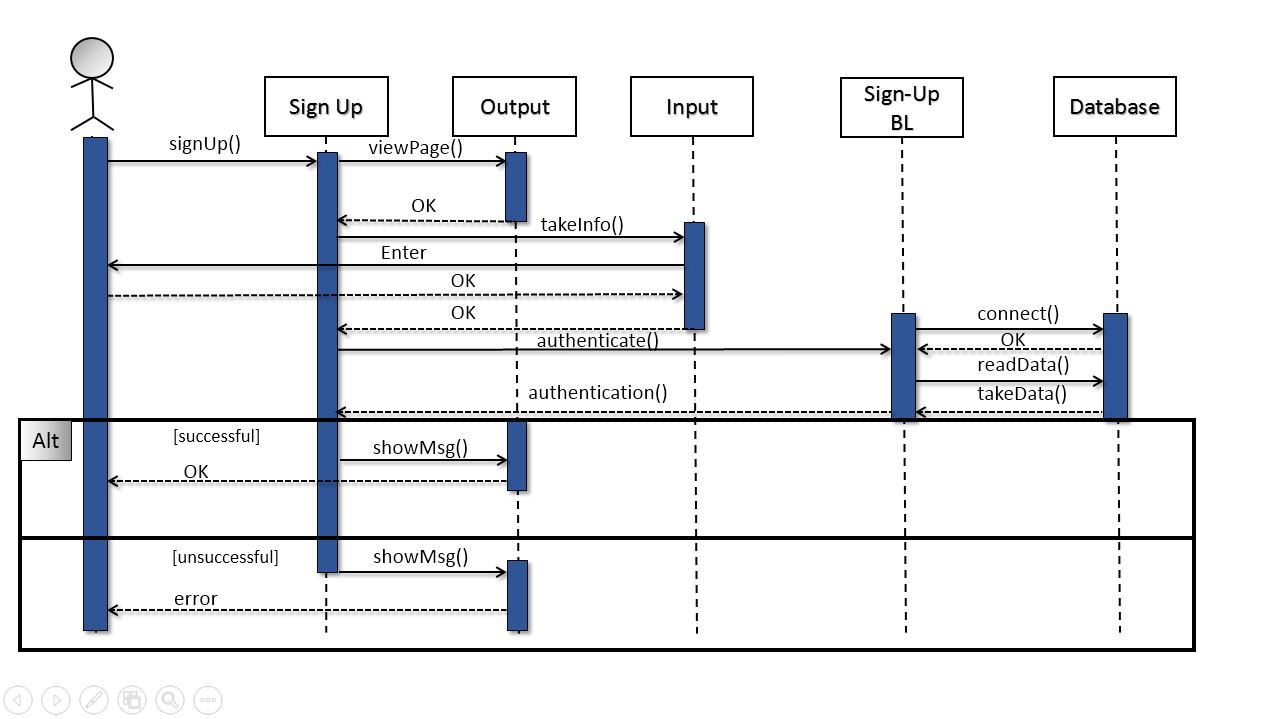
1. Sign up as a provider.
2. Sign up as a tutor.
3. Post advertisement.
4. Search for tuition.
5. Update tuition status.

Sequence diagrams for use cases are shown in below.

**3.3.1 Sign Up as a provider/tutor**

Using this use case user can sign up as a provider.

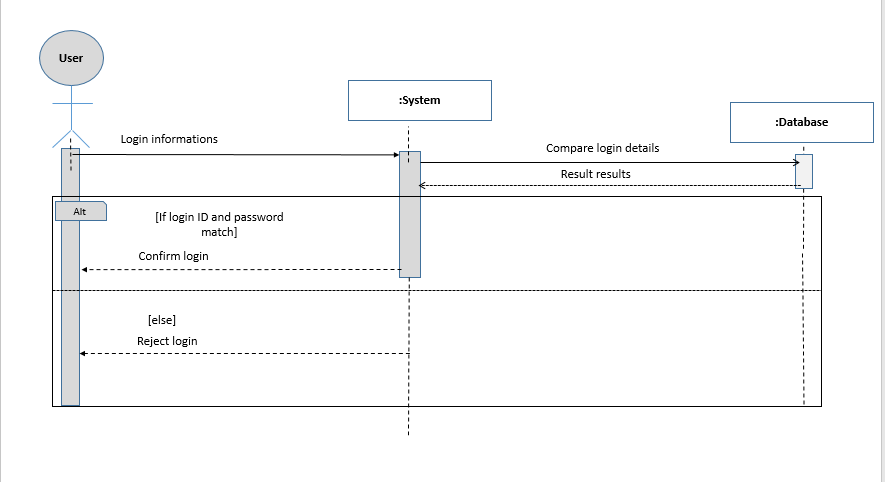
The sequence of interactions for this use case is shown below.



**Figure 3:** **Sequence diagram for sign up as tutor/ provider**

**3.3.2 Log In as provider/tutor**

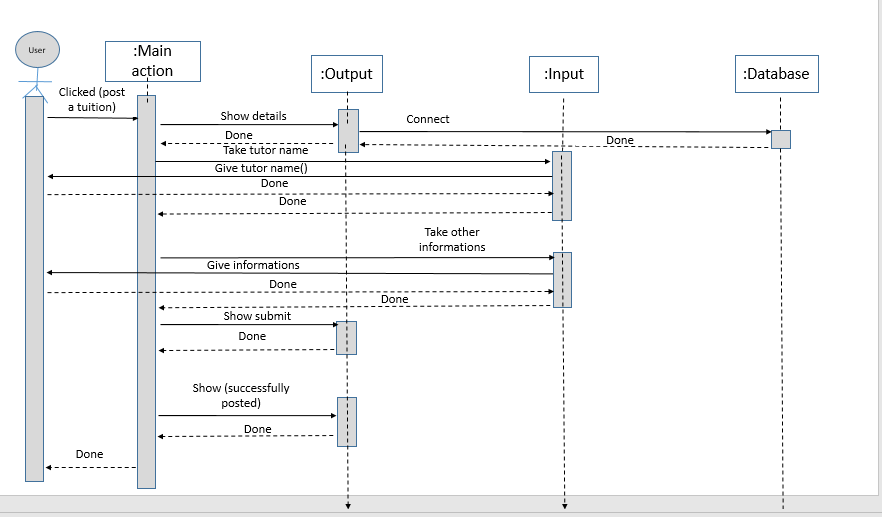
Using this use case user can login as a tutor/provider.



**Figure 4: Sequence diagram for login as tutor/provider**

**3.3.3 Post Advertisement**

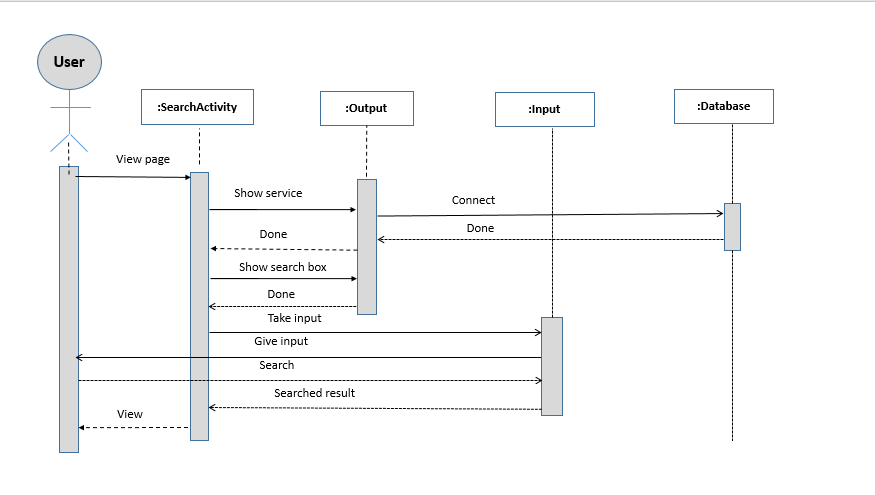
Using this use case provider can post advertisement.

****

**Figure 5: Sequence diagram for post advertisement**

**3.3.4 Search for tuition**

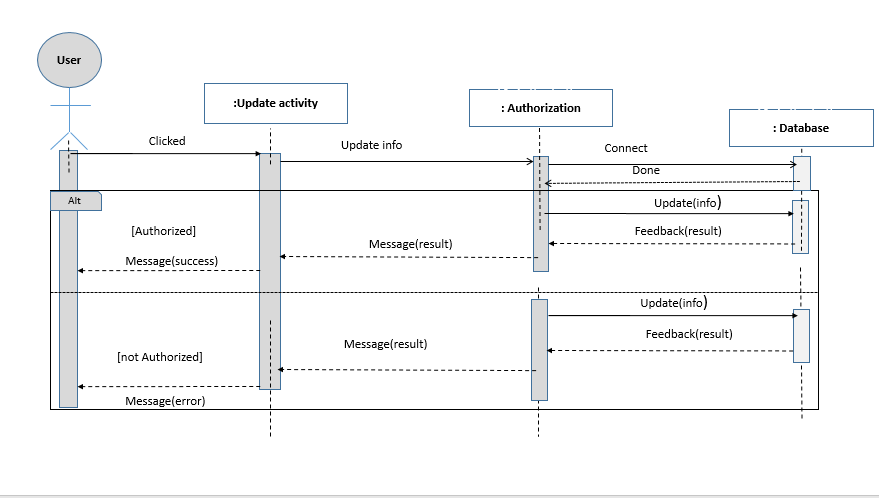
Using this use case tutor can search for tuition.



**Figure 6: Sequence diagram for search tuition**

**3.3.5 Update tuition status**

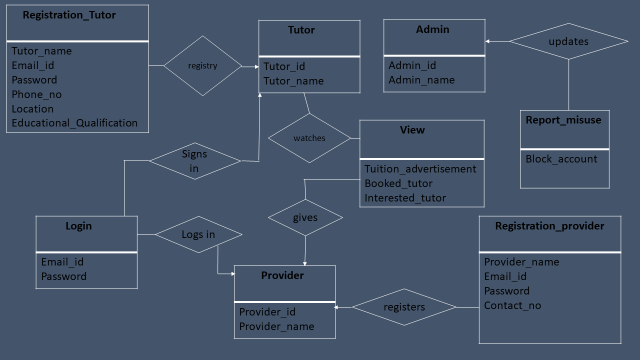
Using this use case tutor can update tuition status



**Figure 7: Sequence diagram for update tuition status**

* 1. **E-R Diagram**

An entity relationship model, also called an entity relationship diagram is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of [data](http://www.webopedia.com/TERM/D/data.html) within [databases](http://www.webopedia.com/TERM/D/database.html) or information systems. Basically, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database.

****

**Figure 8: E-R diagram of Tuition Finder**

**3.4 Conclusion**

A detailed design document gives overall structure of the system. It’s a guideline for the coders. Coders use this diagram to implement the system so that it perfectly fulfils the user requirements and meets the stakeholder’s expectations

Chapter 04: Implementation and user manual

* 1. Introduction

Implementation refers to a critical software development process when engineers create the executable version and a user guide, also commonly known as manual, is a technical communication document intended to give assistance to people using a particular system.

This report covers ten sections. In this sections we have discussed the activity and views of our system.

* 1. **Main Page**

When user click on our system, the system show the home page which contains 2 button and 2 edit text. Button for sign up and login and two edit text field get the sign up data. After filling up sign up field user click on signup button then our system stores the signup data into backend database. For backend database we used firebase database. If the data stored successfully then our system shows the message “ signup successful “. Below, this is our system home page,

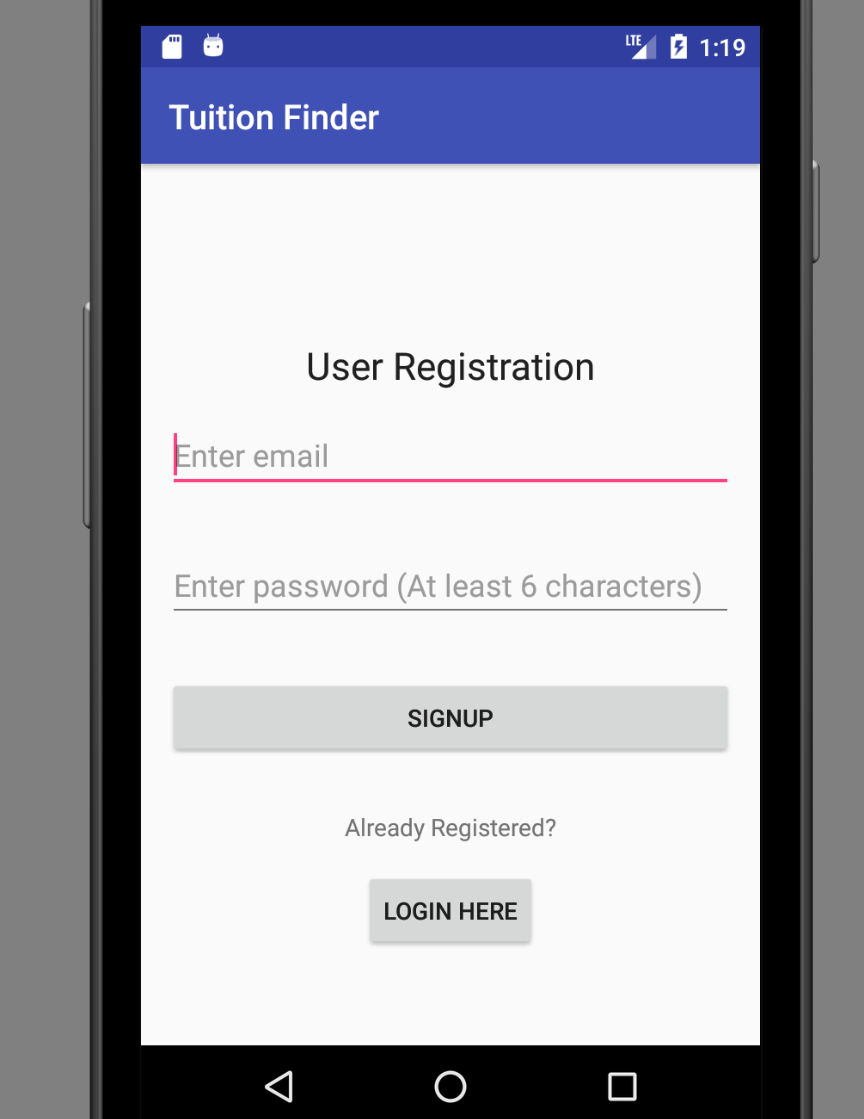


Fig-4.2.1: For Registration Page

4.2.2 Sample code:

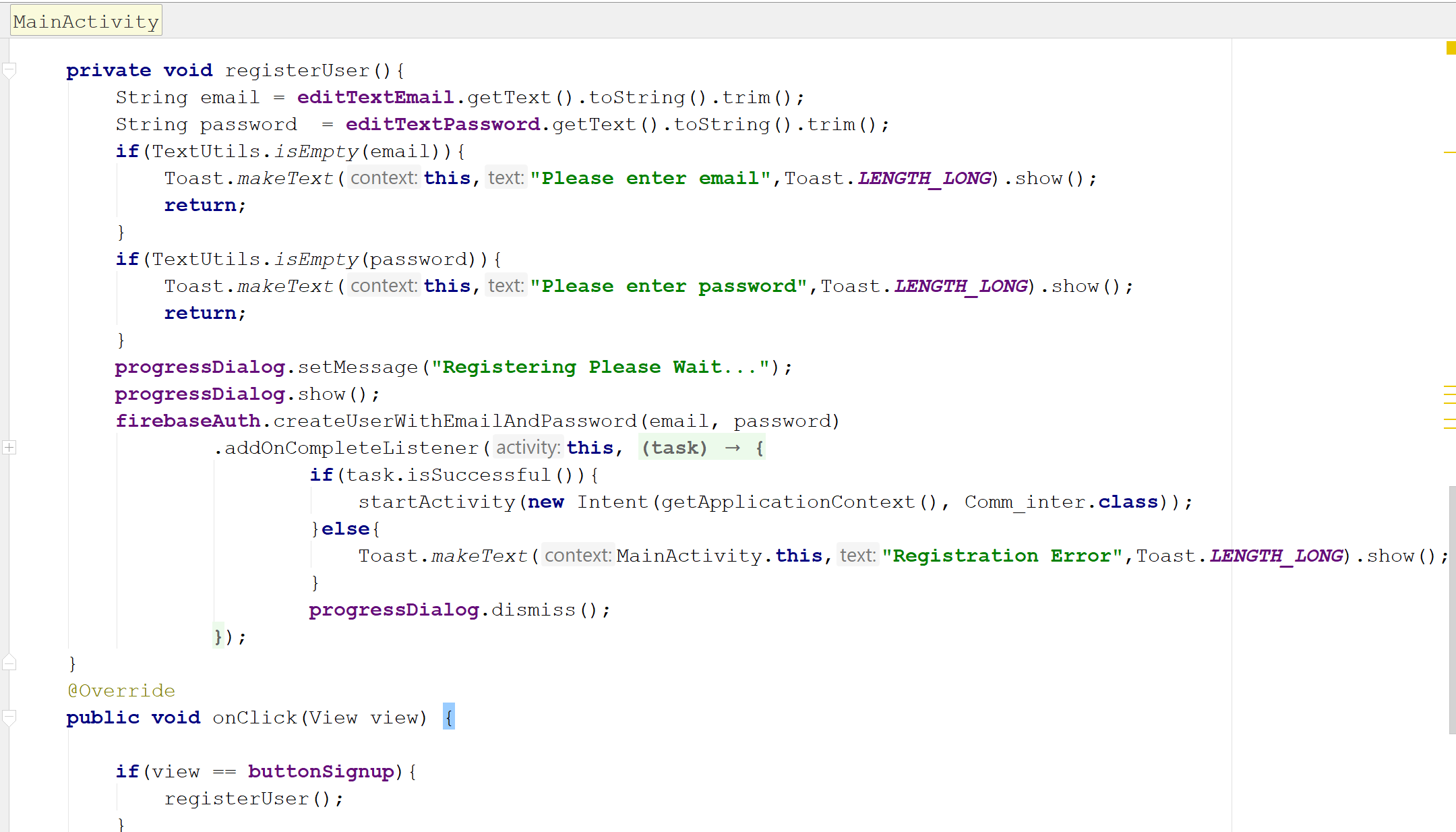


Fig-4.2.2: method for signing up

If user click on login button then our system show login page.

* 1. **Login Page**

When user click on login button, the system show the login page which contain 1 button and 2 edit text field. Button for login and two edit text field get the login data. After filling up edit text field user click on login button then our system get the email and password from backend database and check if those data match then system show the message as “ login successful “ this is our system login page,

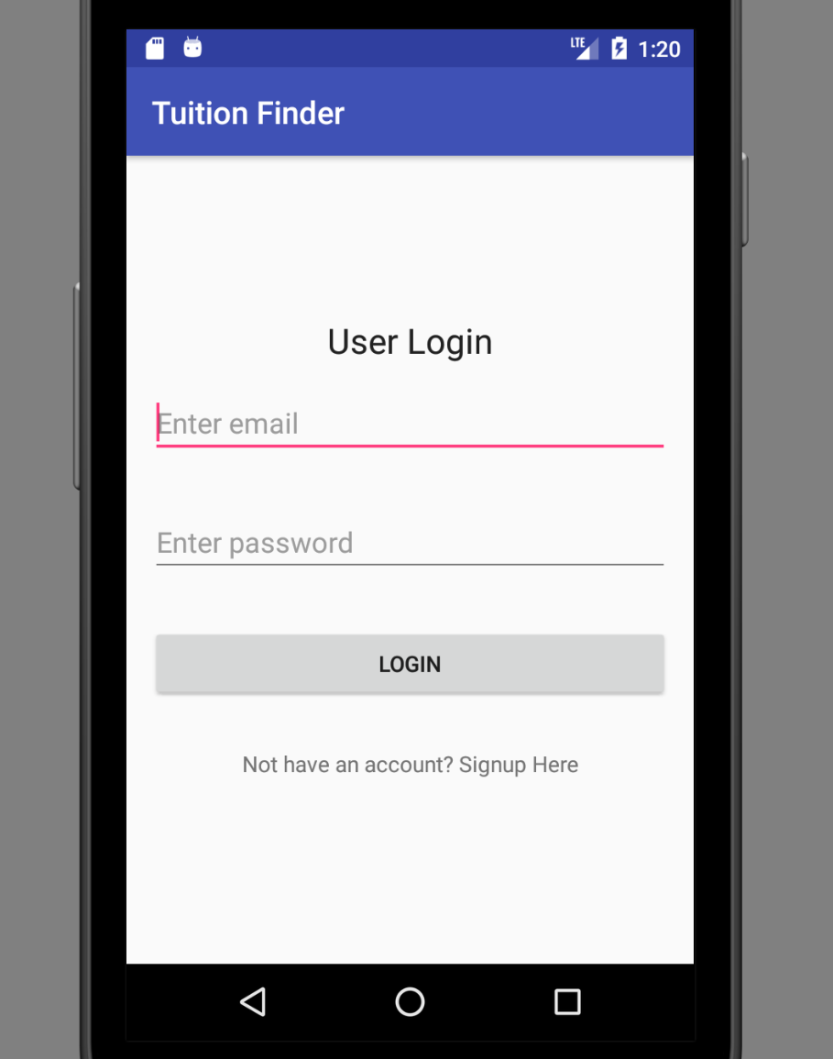


Fig-4.3.1: login page

If logged in successfully then system shows profile activity of the user.

4.3.2 Sample code

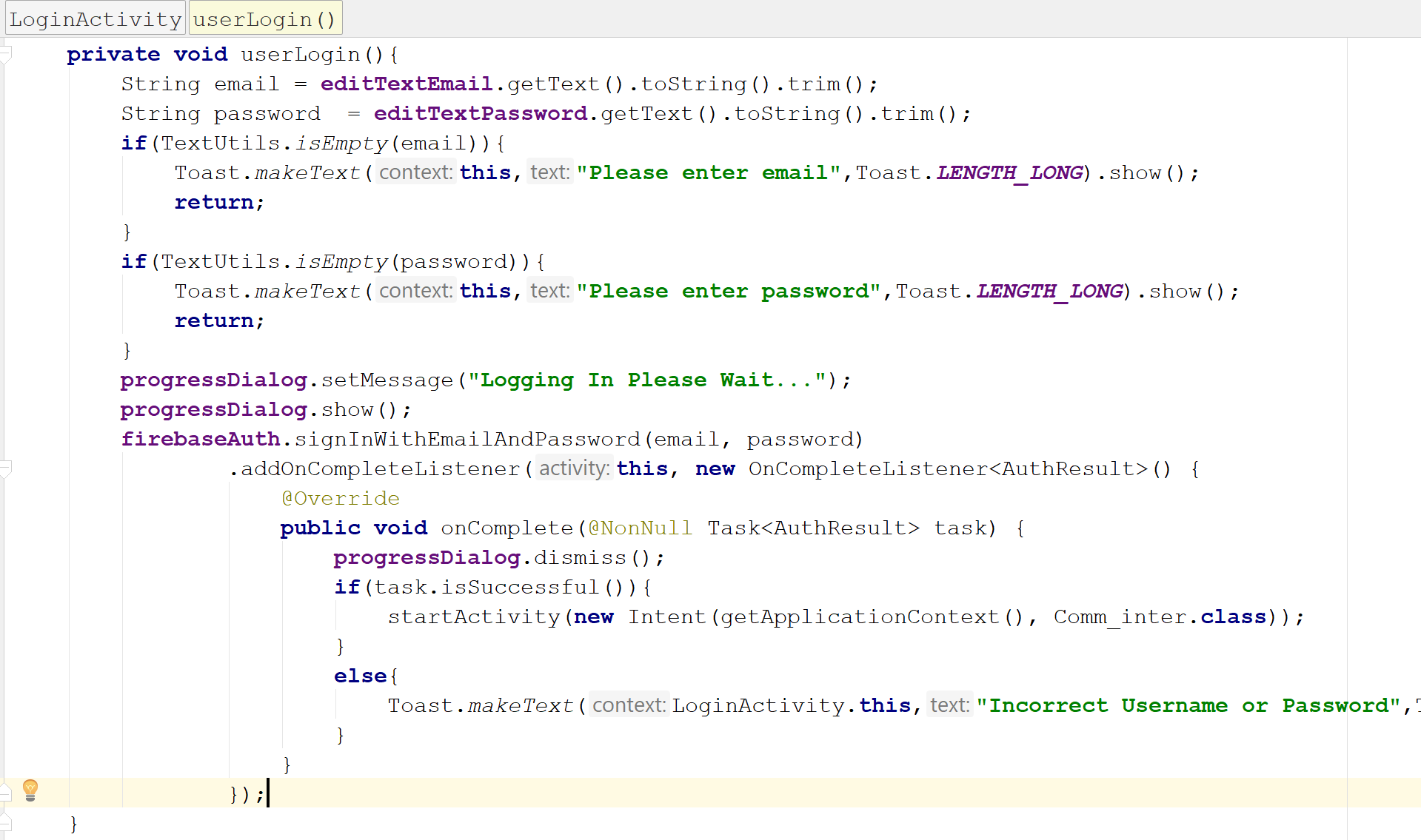


Fig-4.3.2: sample code of login method

* 1. **Common Interface page**

After login successfully system show common interface page with message “Welcome to our app” . Main page contain 3 button. Button 1 for search tuition. Button 2 for post tuition. Button 3 for logout. User can choose one of them at a time.

This is our main page,

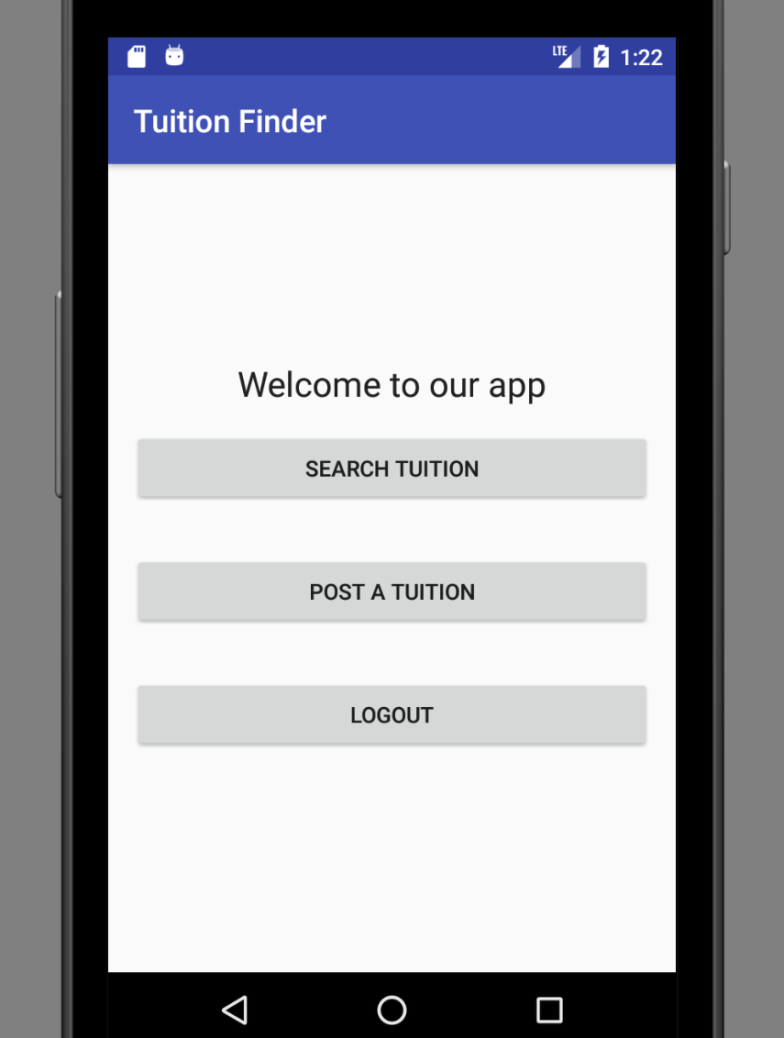


Fig-4.4.1: common interface page

4.4.2 Sample Code

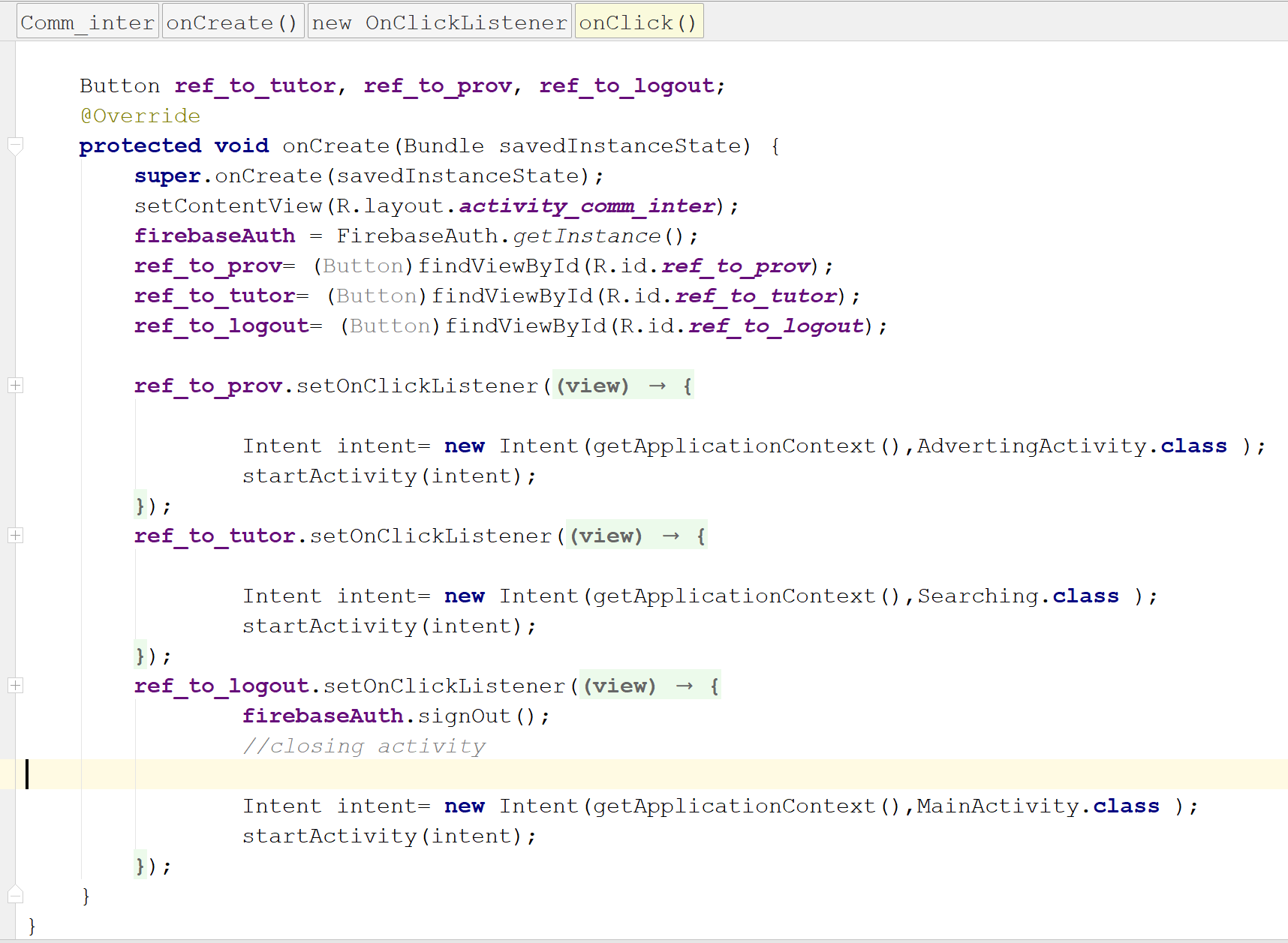


Fig-4.4.2: code for common interface

* 1. **Search Page**

If user clicked on search button in main page then system shows search page, where user can search for tuition according to location

This is our search page

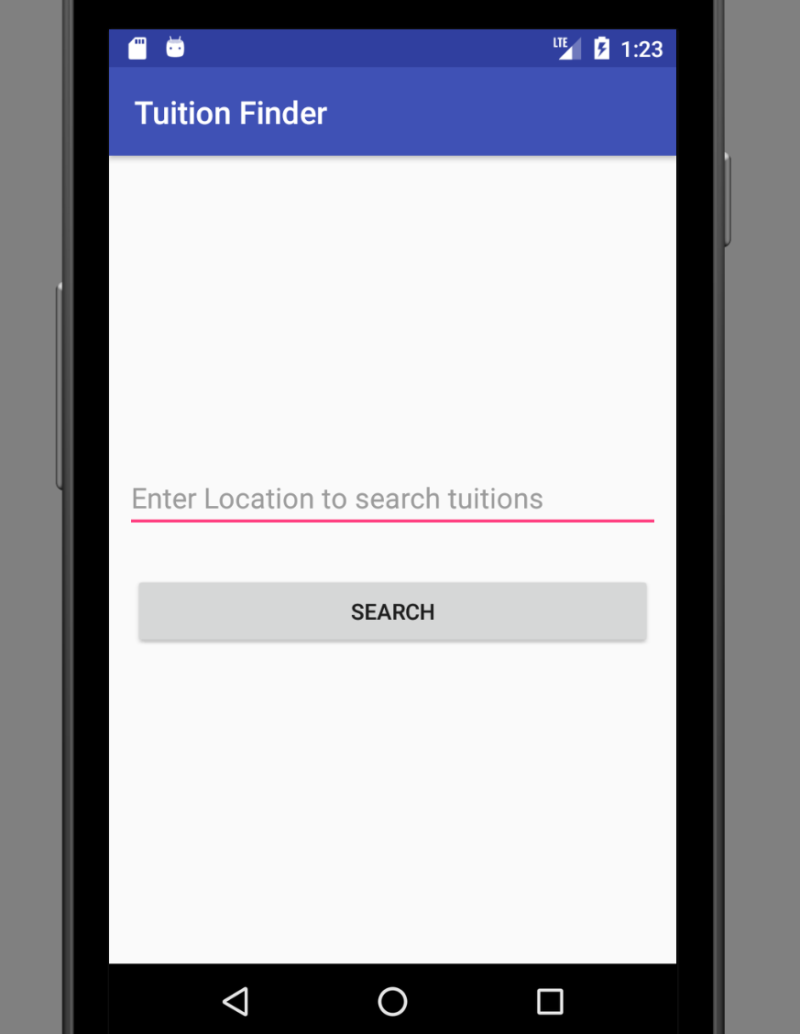


Fig-4.5.1: for search page

If user fill up edit text field with location then system shows the tuition list according to location.

4.5.2 Sample code



Fig-4.5.2 sample code of searching page

* 1. **Tuition List Page**

When User clicked on search button in search page then system shows tuition list page.

This is tuition list page

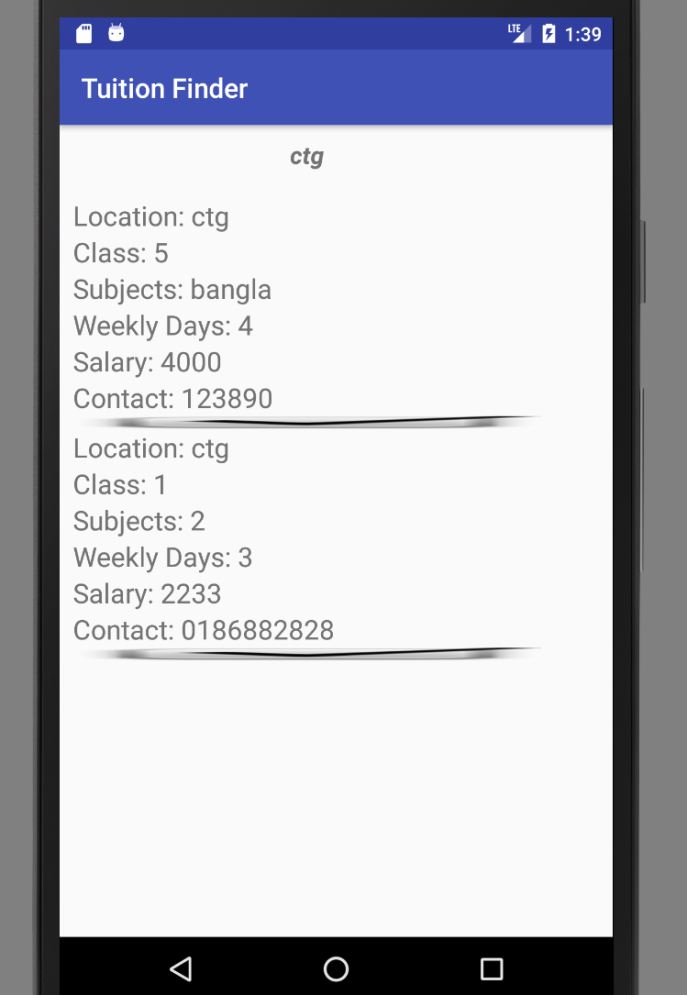


Fig-4.6.1: for tuition list

If user press on specific tuition for a few second, system shows update and delete window.

4.6.2 Sample Code



Fig-4.6.2: sample code for tuition list

* 1. **Update & Delete Window**

When user pushed for some moment on specific tuition then system show this page where user can update and delete tuition information. This window contain 2 button. Button 1 for update and button 2 for delete.

This is update and delete window,

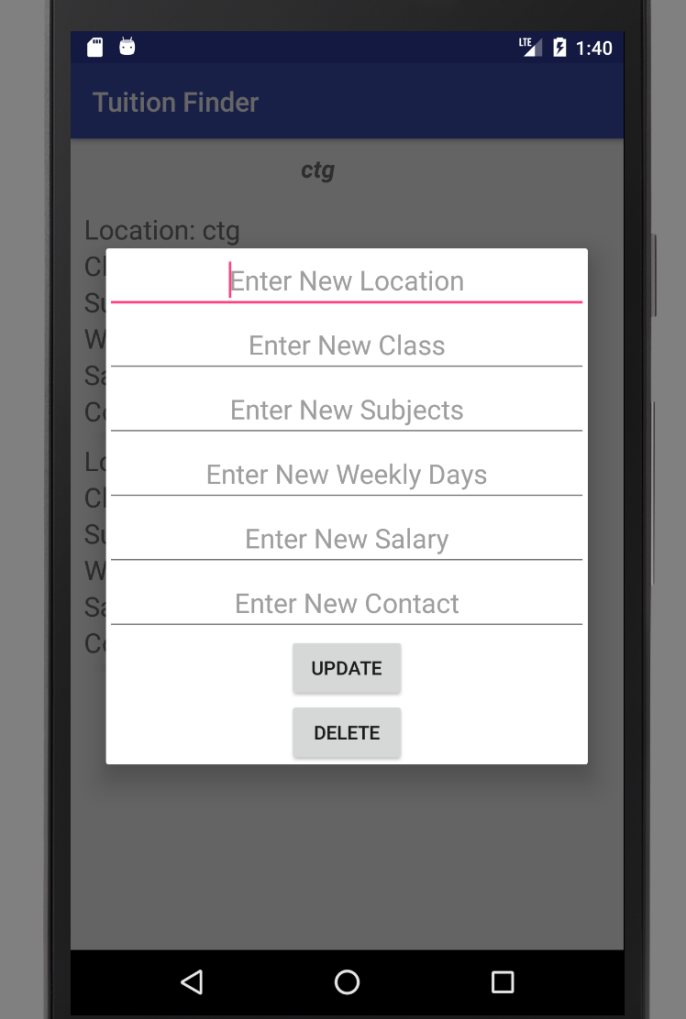


Fig-4.7.1: update and delete window.

* 1. **Post Page**

If user clicked on post button in main page then system shows post page, where user can post for tuition. There are several text field in post page and two button. Text field contain description of the tuition such as location, class, salary, days, subject and contact. Button one for submit and another for logout. If user click on submit button then system store data into backend database from text field.

This is post page,

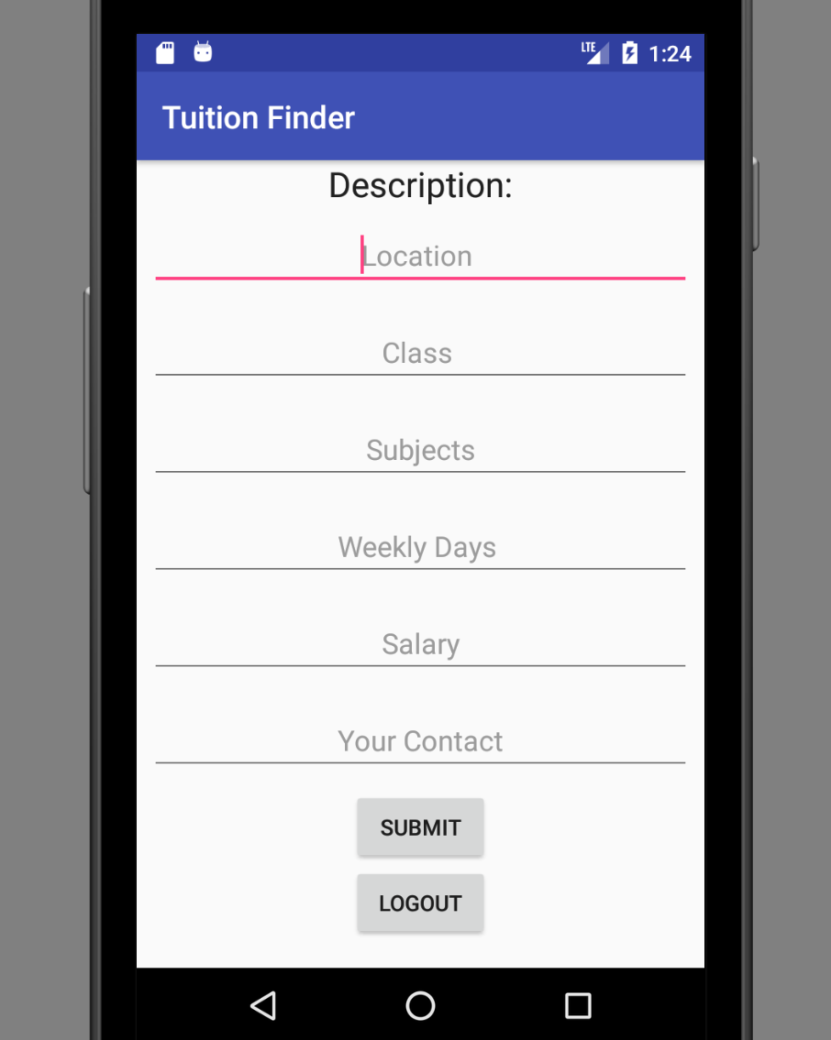


Fig-4.8.1: post page

4.8.2 Sample Code

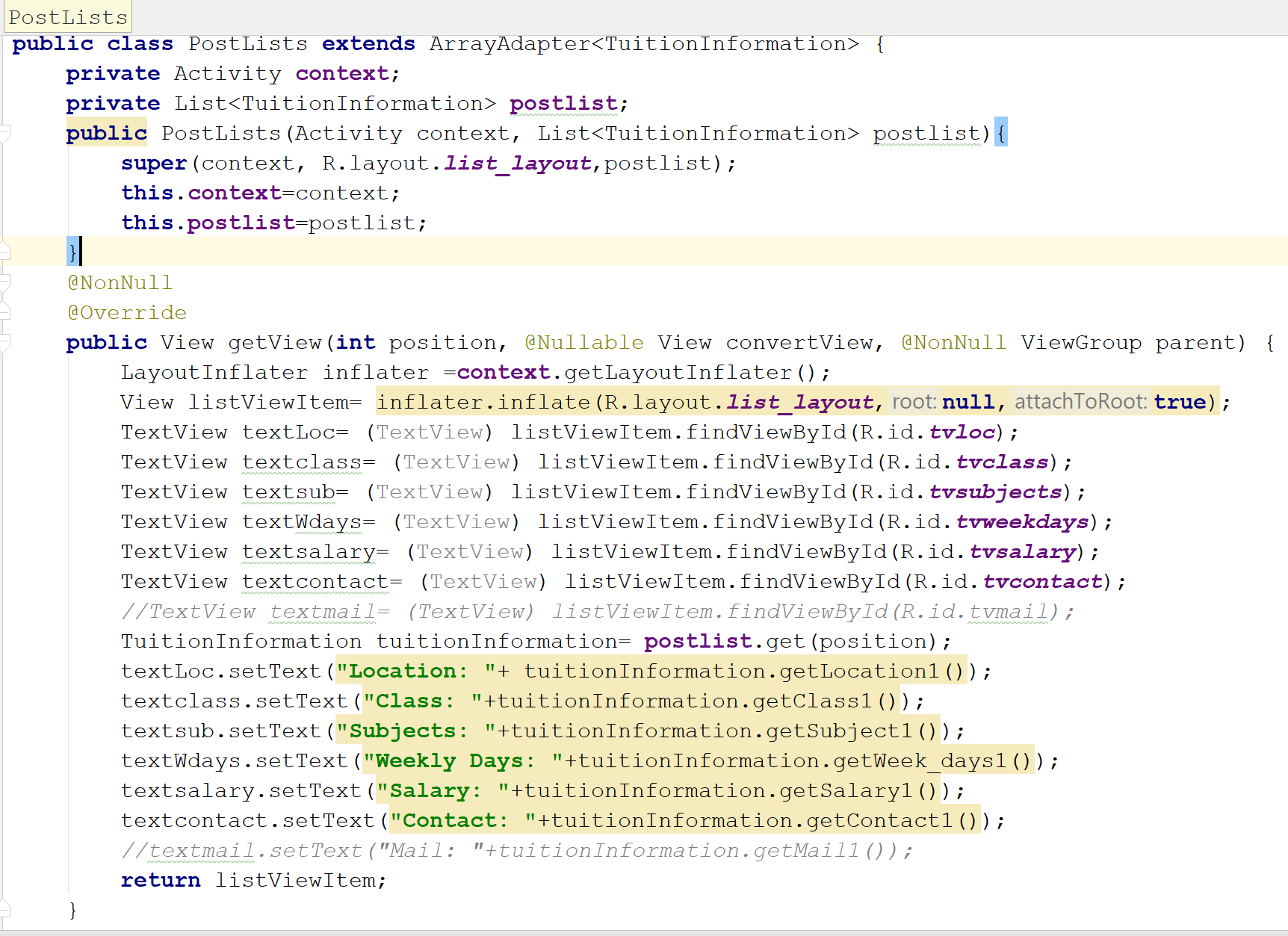


Fig-4.8.2: sample code for post page

* 1. Conclusion

This report provide all the information about the whole project management. The information got from the previous report are combined here. All these report helped us to build our app more efficiently.