1. Introduction :

1.1 Introduction :

- With the popularity of the network and mobile network speed, making people tend to use the way in the mobile client to easily access information.
- Campus is the place to learn every student life, rich and diverse campus information is bound to provide more convenient for our study and life, and the current market, the main function of Application can provide college students with more learning and entertainment, so we develop the establishment of Android-based campus learning life Application---- "fingertips on Institute of Disaster Prevention Science and Technology ".
- Specifically for the campus students to provide all kinds of learning and living real-time information. This paper introduces the development process of this system in detail.

1.2 Scope :

We have put an endless effort on this application & also provided room for the future expansion of this application.

- Interface can be enriched later.
- We can broaden the categories.
- Connect with your friends and other people to make it competitive and exciting.
- Some learning modules and tutorials could be added.
- Help can be provided.
- Security can be increased in the future time.
- More companies can be added and communication between companies and students can be more transparent.

1.3 Project Summary and Purpose :

- The primary purpose to develop this system is to optimize the recruitment process for college. Besides, the qualified applicants could be sort by this system based on their qualifications and company requirements.
- Based on the applicant's skills and areas of interest, the company suitable or the company in which he/she is going to place can be predicted. Another purpose of the software is to facilitate the student (in the college) and the company to register and communicate with placement office.
- Campus Recruitment System enables the user to have the typical recruitment facilities and features at their disposal. It resolves the typical issue of manual staffing processes and activities into a controlled and closely monitored work flow in the architecture of the application.
- The objective of this application is to serve as a common meeting ground for jobseekers and company, locally. This kind of system is specifically designed for organization to help in solving staffing problems and managing human resource department activities at higher degree of optimization.

1.4 Overview Of The Project :

- Final Year for any engineering students as well as their colleges is very crucial, because it is the time when the companies come in. And for every student and company needs each other to work it out and hence this system comes in picture.
- Campus Recruitment System as the name states it is purely used for helping the companies to get the best candidate and directly come in contact with the student, in the same way student get to know about the jobs and the companies directly.
- The College here is the admin and the student and the company both can register with their details and the admin which is a web based system approves or disapproves or takes any action.

- The student when registering has to enter his personal and academic details which can be updated later also. He /She can check for their recent job applications or any new applications filtered according to their categories and can also give a feedback to admin. While the company can post new jobs, see the students or new applications.

1.5 Problem Definition :

Campus Recruitment application is used to create a transparent recruitment system between companies and students. In this application student has to register with his/her username and password. Username can be email ID here. Companies can also register here its details and its requirements. After seeing the details about company any student can register his/her name and apply for the job in the company. Student also have to add his/her academic details, from these details he/she can be eligible for the job. Admin can monitor the system and see how many applications received for particular company.

2. Technology and Literature Review :

2.1 About Tools and Technology :

❖ Android :

- Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.
- Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.
- The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008. On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.
- The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.
- Android has many characteristics. These all characteristics are very much important. The figure of Android characteristics is as below :

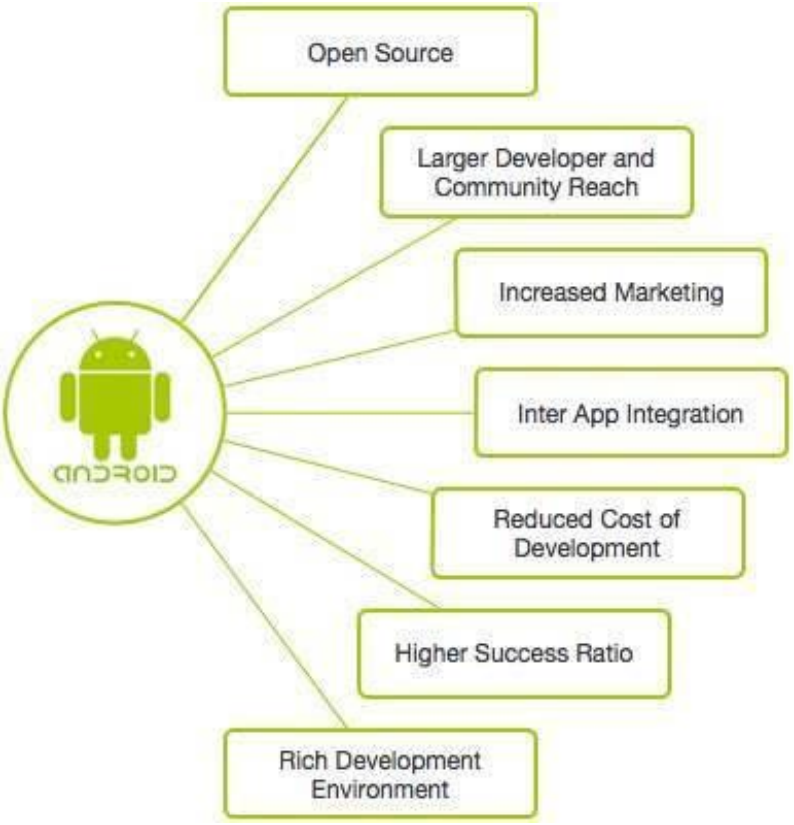


Fig. 2.1(a)- Features Of Android

❖ **Features of Android :**

- Head set layout
- Storage
- Connectivity: GSM/EDGE, IDEN, CDMA, Bluetooth, WI-FI, EDGE, 3G, NFC, LTE, GPS.
- Messaging: SMS, MMS, C2DM (could to device messaging), GCM (Google could messaging), IPC message passing
- API's for location based services such as GPS.
- Multilanguage support
- Multi touch
- Video calling
- Screen capture
- External storage
- Streaming media support
- Optimized graphics

❖ **Architecture of Android OS :**

- Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

✓ **Linux Kernel :**

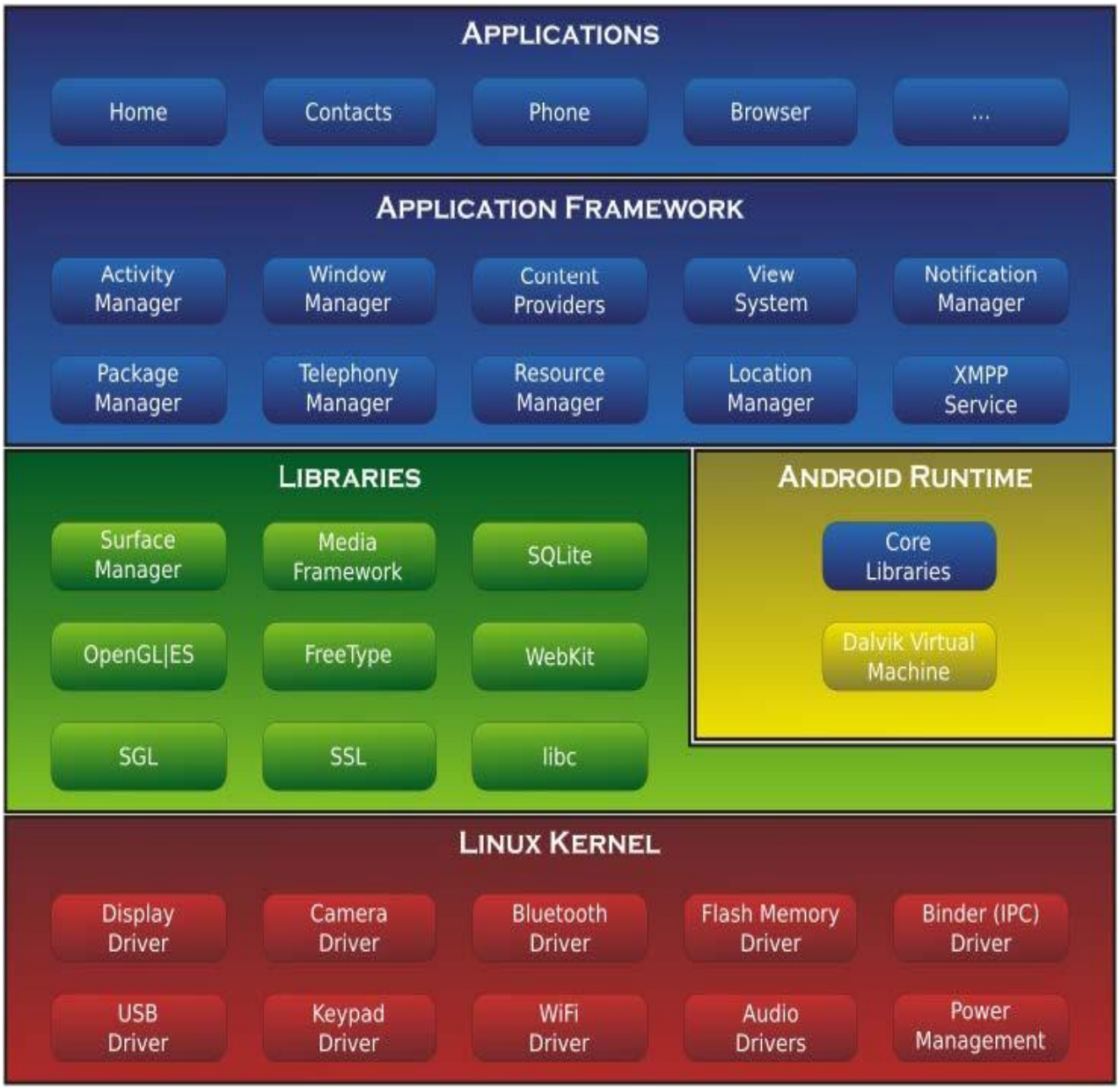


Fig. 2.1(b) – Architecture Of Android

- At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc.
- Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

✓ **Libraries :**

- On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

✓ **Android Libraries :**

- This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows –

android.app − Provides access to the application model and is the cornerstone of all Android applications.

android.content − Facilitates content access, publishing and messaging
between applications and application components.

android.database − Used to access data published by content providers and includes SQLite database management classes.

android.opengl − A Java interface to the OpenGL ES 3D graphics rendering API.

android.os − Provides applications with access to standard operating system services including messages, system services and inter-process communication.

android.text − Used to render and manipulate text on a device display.

android.view − The fundamental building blocks of application user interfaces.

android.widget − A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.

android.webkit − A set of classes intended to allow web-browsing capabilities to be built into applications.

- ✓ **Android Runtime :**

  - This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.The Dalvik VM makes use of Linux core features like memory management and multithreading, which is intrinsic in the Java language.
  - The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

- ✓ **Application Framework :**

    - The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications. The Android framework includes the following key services –

      Activity Manager − Controls all aspects of the application lifecycle and activity stack.
      Content Providers − Allows applications to publish and share data with other applications.
      Resource Manager − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
      Notifications Manager − Allows applications to display alert and notifications to the user.
      View System − An extensible set of views used to create application user interfaces.

- ✓ **Application Layer :**

    - Android ships with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts and others. All applications are built using the Java. Each of the applications aims at performing a specific task that it is actually intended to do.

❖ **Components of Android :**

The basic components of an Android application include Activity, Broadcast Receiver, Service, and Content Provider. Each of the above, which when used for any application, has to be declared in the AndroidManifest.xml. The user

interface of the component is determined by the Views. For the communication among these basic components we use Intents and Intent filters which play crucial role during app development.

- ✓ **Activities :**
  - An activity represents a single screen with a user interface, in-short activity performs actions on the screen. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched.

- ✓ **Service :**
  - A Service is a body of code that runs in the background. It can run in its own process, or in the context of another application's process, depending on its needs. Other components "bind" to a Service and invoke methods on it via remote procedure calls. An example of a Service is a media player; even when the user quits the media selection UI, she probably still intends for her music to keep playing. A Service keeps the music going even when the UI has completed.

- ✓ **Broadcast Receivers :**
  - Broadcast Receivers simply respond to broadcast messages from other applications or from the system. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action.

- ✓ **Content Providers :**
  - A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the Content Resolver class. The data may be stored in the file system, the database or somewhere else entirely.

❖ **Introduction to Java :**

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Over time new enhanced versions of Java have been released. The current version of Java is Java 1.7 which is also known as Java 7. From the Java programming language the Java platform evolved.The Java platform allows software developers to write program code in other languages than the Java programming language and still runs on the Java virtual machine. The Java platform is usually associated with the Java virtual machine and the Java core libraries.

❖ **Java Virtual machine :**

The Java virtual machine (JVM) is a software implementation of a computer that executes programs like a real machine. The Java virtual machine is written specifically for a specific operating system, e.g. for Linux a special implementation is required as well as for Windows.

❖ **Java Runtime Environment vs. Java Development Kit :**

A Java distribution comes typically in two flavours, the Java Runtime Environment (JRE) and the Java Development Kit (JDK).The Java runtime environment (JRE) consists of the JVM and the Java class libraries and contains the necessary functionality to start Java programs. The JDK contains in addition the development tools necessary to create Java programs. The JDK consists therefore of a Java compiler, the Java virtual machine, and the Java class libraries.

❖ **Characteristics of Java :**

- The target of Java is to write a program once and then run this program on multiple operating systems.
- Java has the following properties:

- **Platform independent:** Java programs use the Java virtual machine as abstraction and do not access the operating system directly. This makes Java programs highly portable. A Java program (which is standard complaint and follows certain rules) can run unmodified on all supported platforms, e.g. Windows or Linux.

- **Object-orientated programming language:** Except the primitive data types, all elements in Java are objects.

- **Interpreted and compiled language:** Java source code is transferred into the byte code format which does not depend on the target platform. These byte code instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so called Hotspot-Compiler which translates performance critical byte code instructions into native code instructions.

- **Automatic memory management:** Java manages the memory allocation and de-allocation for creating new objects. The program does not have direct access to the memory. The so-called garbage collector deletes automatically objects to which no active pointer exists.

❖ **JSON :**

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent.
JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

2.2 Brief History Of Work Done :

- Until now we implement mobile campus recruitment application named as “Placemento”. By opening of the app first one can find out the Sign In form and enter into the application by entering username and password if you already signed up. Otherwise you can Sign Up by providing username, password. After signing in one can add academic details.
- Admin can monitor the activities of the application and companies can add their details into the application. Placemento application provides the best facilities for the students who are interested in the placement activities.

3. System Requirement Study :

3.1 Hardware and Software Requirements :

❖ Hardware Requirements :

- Processor – i3
- Hard Disk – 5 GB
- Memory – 2 GB RAM (4 GB Recommanded)
- Mouse
- Keyboard
- Monitor

❖ Software Requirements :

- Windows 7 and above versions
- Android SDK tools
- Java Development Kit (JDK)
- Firebase Pluggin or Web

3.2 Constraints :

3.2.1 Hardware Limitation :

We use minimum API level 21 so other APIs below 21 can't be supported and other user must have android phone because it is not worked with any iOS phone.

3.2.2 Parallel Operations :

Software should be able to support use of multiple user at a time. The database should be able to accommodate thousands of record of the users.

3.2.3 Safety Consideration :

The database may get at certain times due to virus or failure of operating system. Therefore it is required to take back up of database at particular interval.

3.2.4 Security Consideration :

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction or disclosure are describe below.

- Keep Specific log or history data set
- Assign certain functions to different modules
- Restrict communication between some areas of program
- Check data integrity for critical variables

Communication must be restricted when the application is validating user's credentials.

3.3 Assumptions and Dependencies :

- Assume that for creating mobile application user is not interested in knowing that how it will created but instead of it they better know what to do within it. Implementation is not easy for placement. Also all the students of the college are not interested in the placement activities so we have to make the application which is user friendly and easy to use for every student and company.

4. System Analysis :

4.1 Study Of Current System :

- Final Year for any engineering students as well as their colleges is very crucial, because it is the time when the companies come in. And for every student and company needs each other to work it out and hence this system comes in picture.
- Campus Recruitment System as the name states it is purely used for helping the companies to get the best candidate and directly come in contact with the student, in the same way student get to know about the jobs and the companies directly.
- The College here is the admin and the student and the company both has 2 different applications where they have to register with their details and the admin which is a web based system approves or disapproves or takes any action.
- The student when registering has to enter his personal and academic details which can be updated later also. He/ She can check for their recent job applications or any new applications filtered according to their categories and can also give a feedback to admin. While the company can post new jobs, see the students or new applications.

4.2 Problems and Weakness of Current System :

- Student can manipulate the data, so fake data can also be written.
- Student can't edit their application once sent. It would require admin to change the data.
- Requires active internet connection.

4.3 Proposed System :

- In most of the colleges' campus recruitment activities there are emails those are sent by placement officer of the college and in each email there are details about particular company.

- But here in our application there is app that have everything regarding forms and company details. Interested students have to register for the jobs.
- Placement officer sends email for applying to a company and student has to fill the details (academic details) everytime. In this application students do not have to fill the details everytime, as their details are saved in the application.
- There are six features provided by the application.

    i. **Registration:** The student has to register into the system with his personal and academic details.
    ii. **Login:** The student has to sign in and will be signed in till he signs out.
    iii. **Profile:** The user can update his personal as well as academic details.
    iv. **Applications:** List of applications the user has applied while he can also withdraw the applications.
    v. **New Jobs:** List of jobs and its details according to the criteria and eligibility of the student.
    vi. **Feedback:** Student can send a feedback to the admin.

4.4 Requirements of New System :

4.4.1 User requirements :

- Android phone (version 5.0 and above),
- Internet connection

4.4.2 System Requirements :

- Android Software Development Kit (SDK)
- JAVA environment
- Firebase Database
- 2 GB RAM
- i3 Processor

- 10 GB Memory

4.5 Feasibility Study :

- **Product :**

  The project is a Quiz android application. It will help its users to improve their general knowledge.

- **Technical Feasibility :**

  The android application will be developed using Android Studio. The team is competent in that.

- **Social Feasibility :**

  Some training for the users/admin are required but all users are IT literate.

- **Market Research :**

  Market research says that this application would be useful for the users as it could seamlessly help them for their lifestyle.

- **Economic Feasibility :**

  The application can be developed within budget.

- **Alternate Solution :**

  Could be a desktop system but that would not be as portable.

4.6 Features of New System :

- The system can be used by anyone as it is easy to understand.
- Admin can keep Question in Server updated with all things going on in the World.
- It makes Users happy as their knowledge is challenged in very nice way.
- Admin can get instant result.
- The system is flexible and secured to be used.

4.7 Use Case Diagram :

- A use case diagram at its simplest is a representation of a user's interaction with the system that 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 and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

Components of Use Case Diagram :

- **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
- **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
- **Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

The use case diagram for " Campus Recruitment System " is as below :

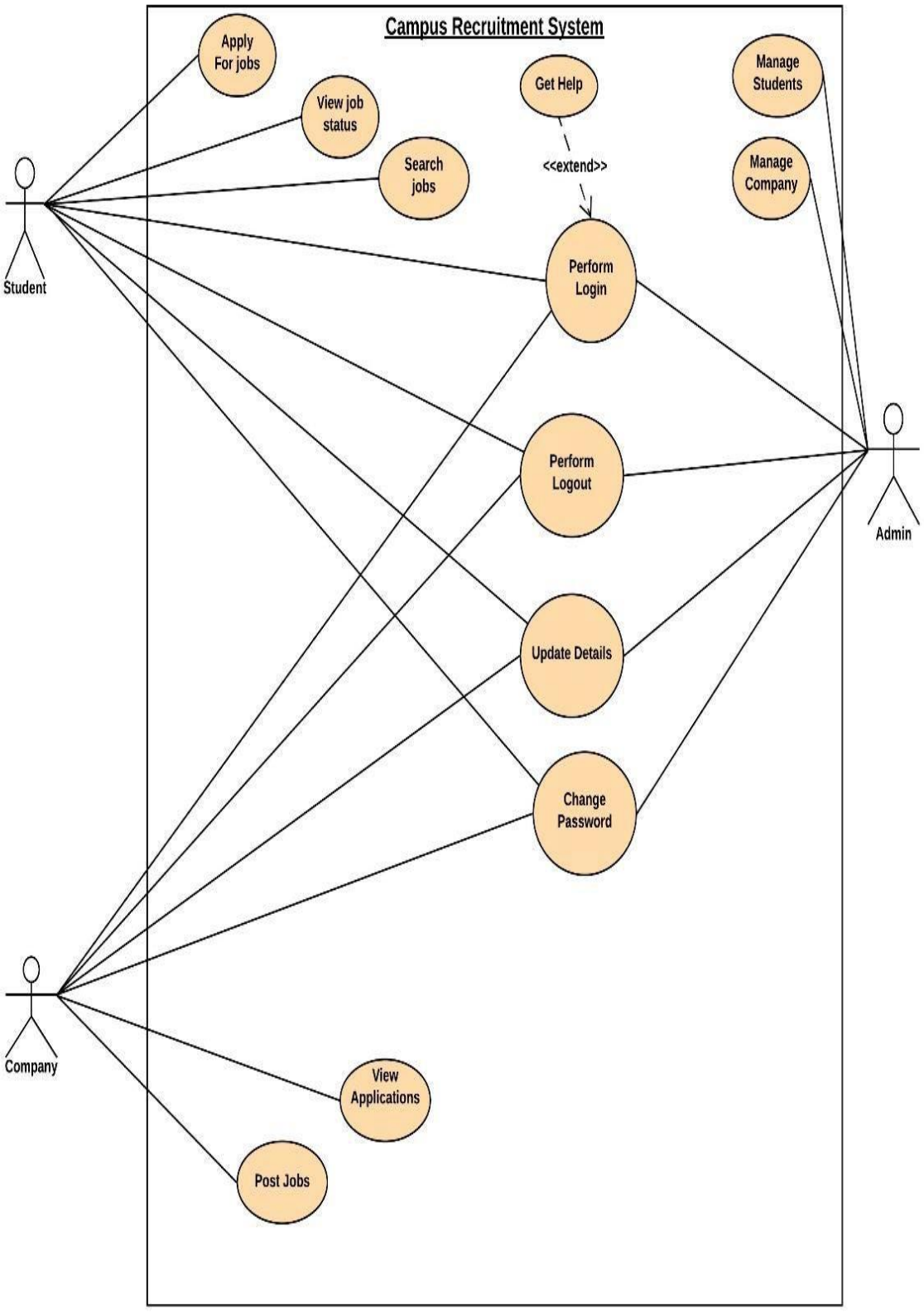


Fig. (4.7). - Use case Diagram

4.8 Activity Diagram :

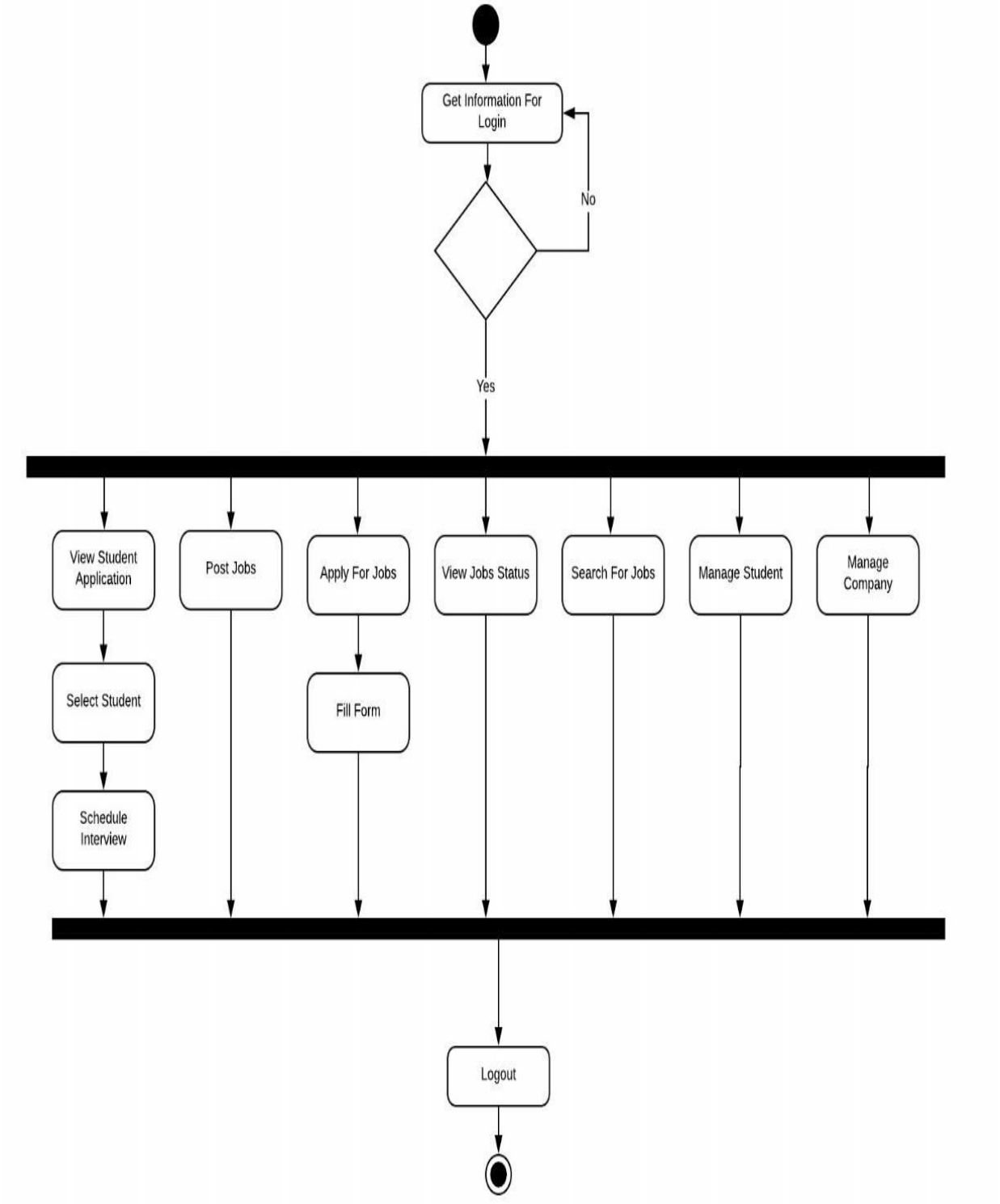


Fig. (4.8). - Activity Diagram

4.9 Sequence Diagram :

- A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.
- A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

a) Apply For Job :

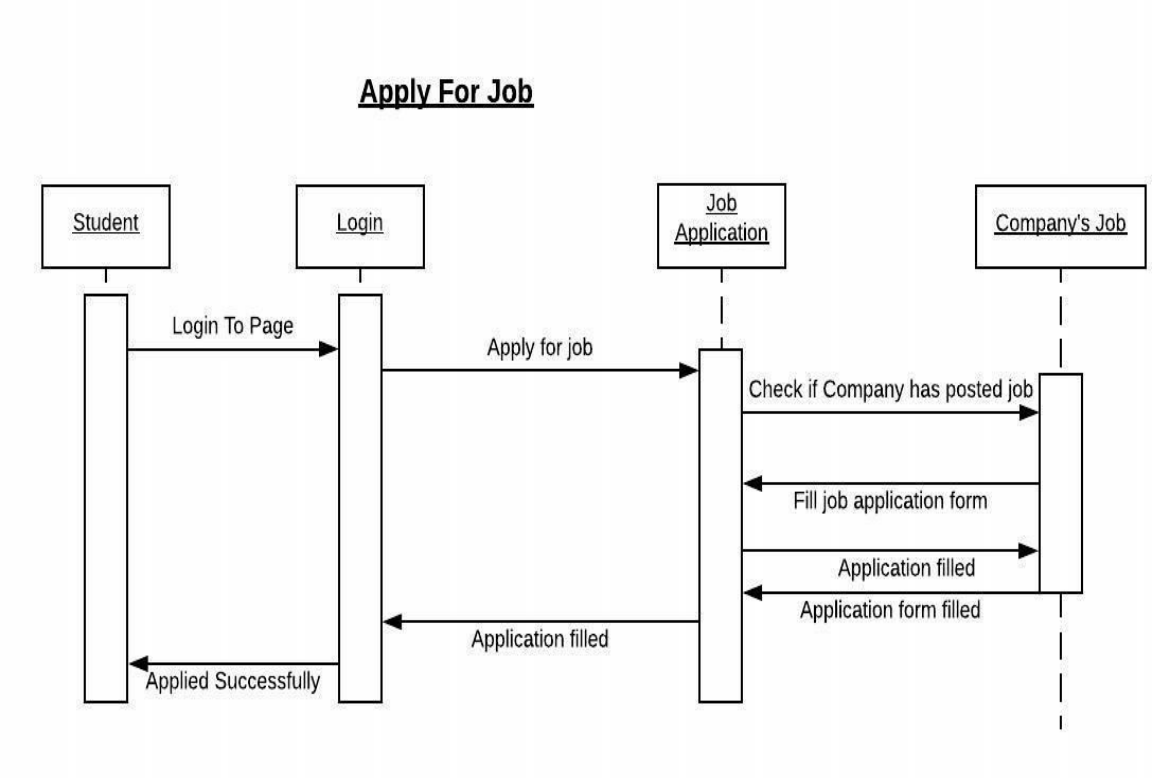


Fig. (4.9).(a) – Apply For Job

4.10 Functional Requirements :

Following table describes the functional requirements:

Feature	Remark
Update Information	This functionality will update the information of the student.
Sign Up	This Functionality is used to sign up for student, admin and company.
Log In	This functionality is used to Login for admin, student and company.
Apply For Jobs	This functionality is used for students to apply for jobs.
Post Jobs / Add Job Details	This functionality is used by the company to post jobs.
View Job Status	This functionality is used to view job status by the student.
Manage Student	This functionality is used by the admin to manage student.
Manage Company	This functionality is used by the admin to manage company.
View Student Application	This functionality is used to view student student application.

4.11 Non-Functional Requirements:

Following are the non-functional requirements of the Campus Recruitment System.

4.11.1 Performance Requirements:

- The completely separate business logic at admin side from the student interface ensures good performance.
- The system exhibits high performance because it is well optimized. The business logic is clearly separate from the UI.

- System is available 24 by 7.

4.11.2 Safety Requirements:

- Errors will be minimized and an appropriate error message that guides the user from an error will be provided.
- Validation of users input is highly essential.
- The time taken to recover from the error is less than 10 second.

4.11.3 Security Requirements:

- The system is provided a high level of security and integrity of the data held by the system.
- Only authorized personnel such as admin can gain access to the to the private data and only the user with valid username and password is allowed to view its user page.

4.12 Class Diagram:

- Campus Recruitment System class diagram describes the structure of Campus Recruitment System classes, their attributes, operations and the relationships among objects. The main classes of the Campus Recruitment System are Student, admin, Company, etc.
- Classes of the Campus Recruitment System are as follows:

i. **Student Class** – Manage all operations of student.

ii. **Admin Class -** Manage all operations of admin.

iii. **Company Class** - Manage all operations of company.

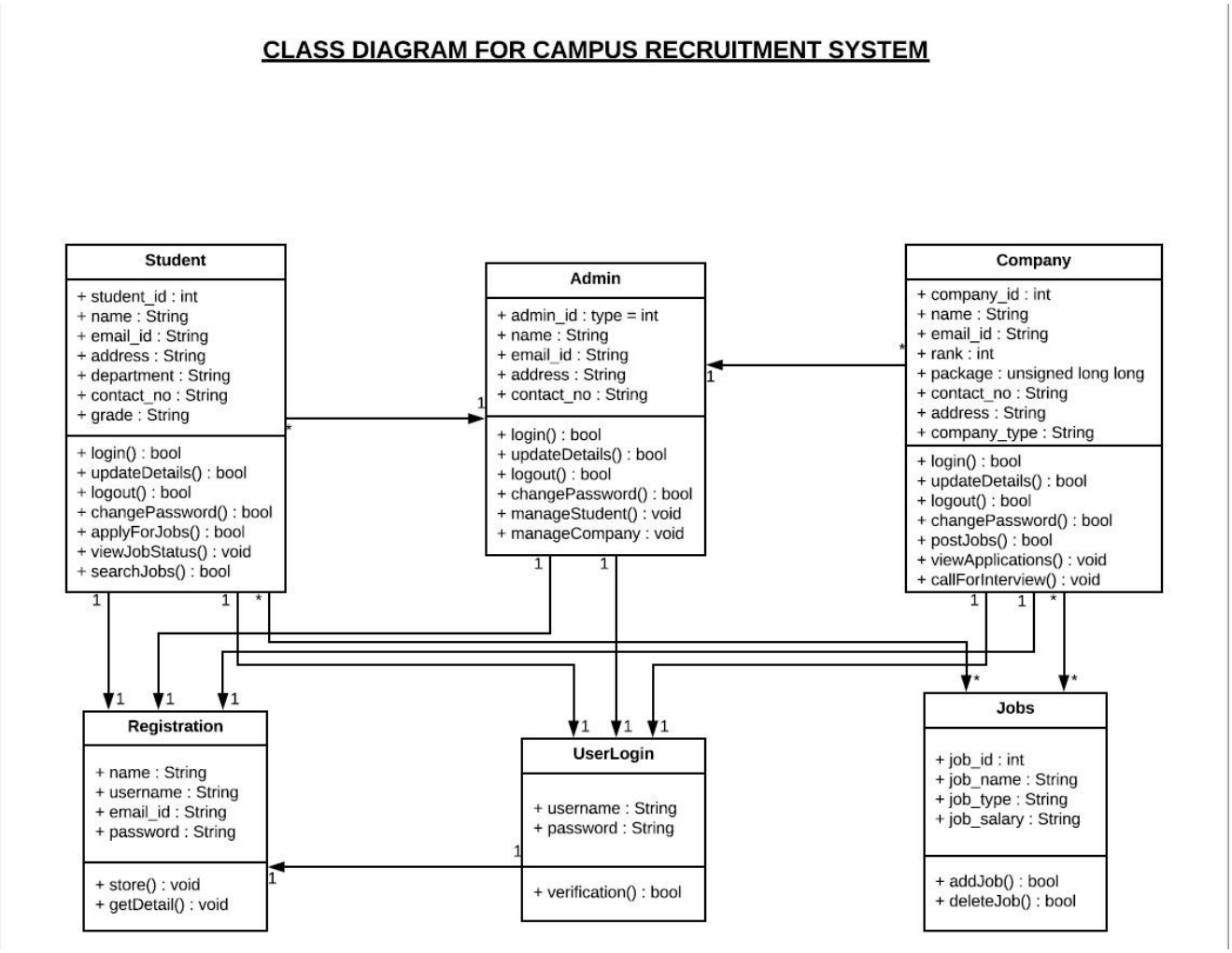


Fig. (4.12). - Class Diagram

5. System Design :

5.1 Database Design/Data Structure Design :

5.1.1 Data Dictionary :

Data Item	Type	Size	Key
AdminID	int	10	Primary Key
UserName	varchar	15	
PassWord	varchar	15	
Enabled	varchar	20	
Role	varchar	20	

Fig (5.1.1) (a). Admin Database Login Table

Data Item	Type	Size	Key
UserID	int	10	Primary Key
UserName	varchar	15	
PassWord	varchar	15	
Email	varchar	20	

Fig (5.1.1) (b). User Table

- The following section describes the Use Cases with Pre and Post Conditions:

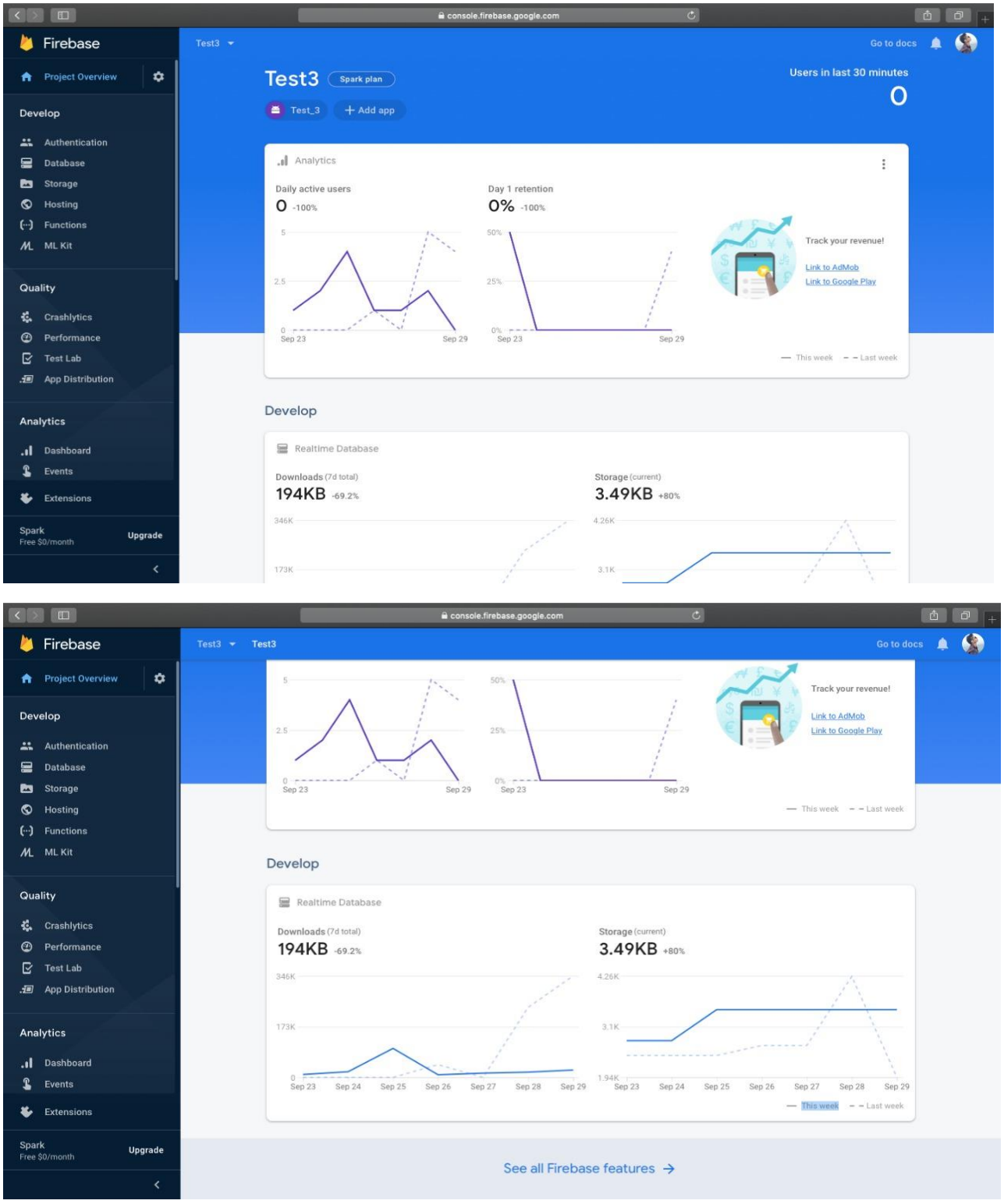
S.No	Use case Name	Description	Pre-condition	Post condition
01	Perform Login	User(admin,student or company) can perform login.	User having account	Login successful
02	Perform Logout	User(admin, student or company) can	User logged in	Successfully logged out
03	Update Details	User(admin, student or company) can	User logged in	View details
04	Change Password	User(admin, student or company) can	User having password.	Password is reset.
05	Manage Company	Admin can manage companies	Existing admin & atleast one company	Validated by admin & company can continue with it's account
06	Manage Students	Admin can manage students	Existing admin & at least one student	Validated by admin & student can continue with it's account
07	Apply for jobs	Student can apply for job	Job should be there posted by company & student must be eligible for job	Applied for job successfully & wait for response from company

08	View Job Status	Student can view job status	Student must have successfully applied for job.	Viewed job status & can accept job if selected else can
09	Search jobs	Student can search for Job	Student must have account & logged in.	Can find a job or not.
10	Post Jobs	Company can post jobs	Company must have account & logged in.	Student can now apply for jobs.
11	View Application	Company can view applications of	Atleast one student must have applied.	Company can react to applications.
12	Get Help	User can get help for login	User should have tried for it or just get help if does not know how to	User will now login using this help.

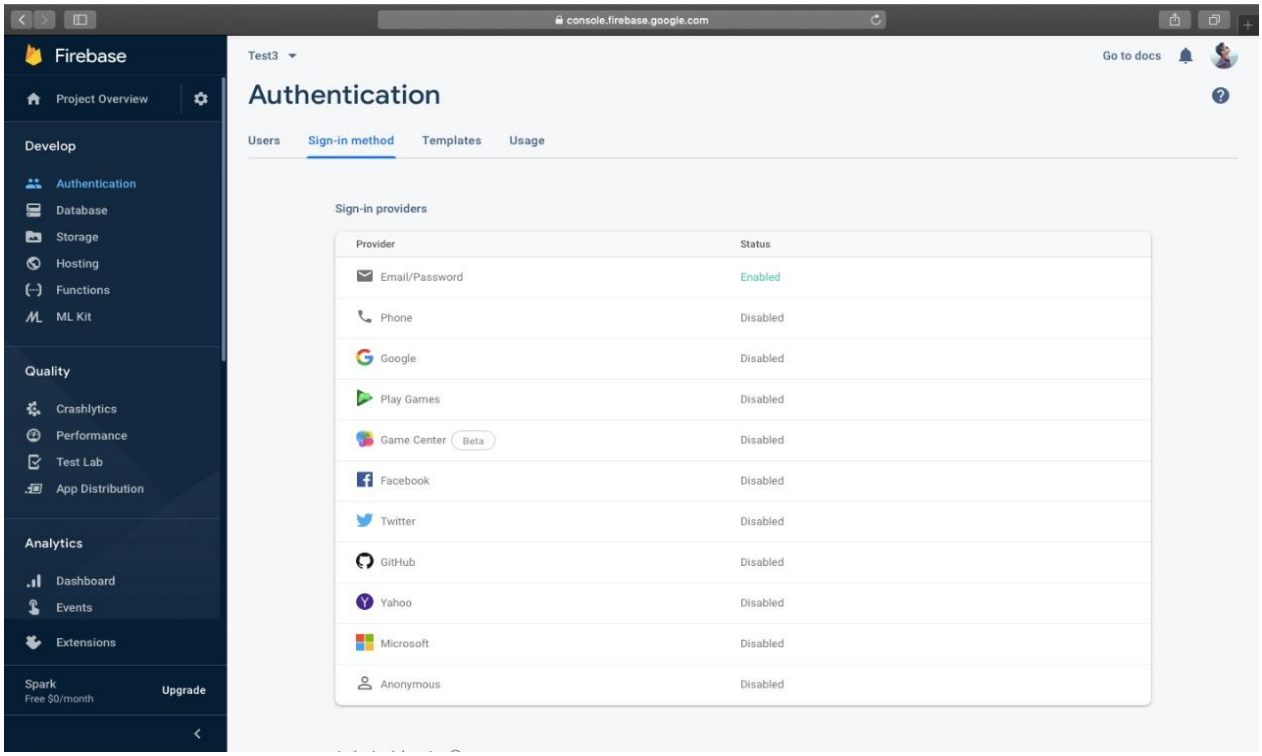
Fig (5.1.1) (c). Use Case Pre & Post Condition Table

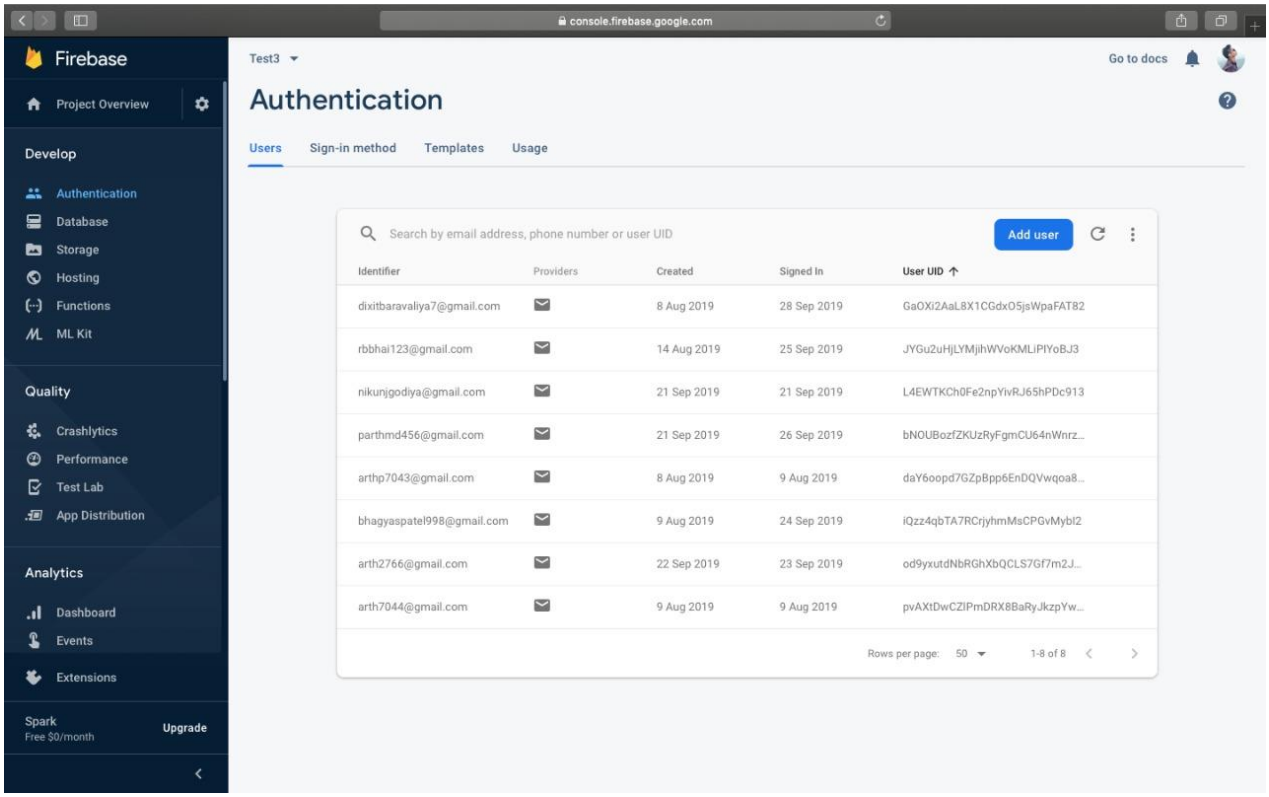
5.2 Input-Output and Interface Design:

5.2.1 Sample of Forms, Reports and Interface:

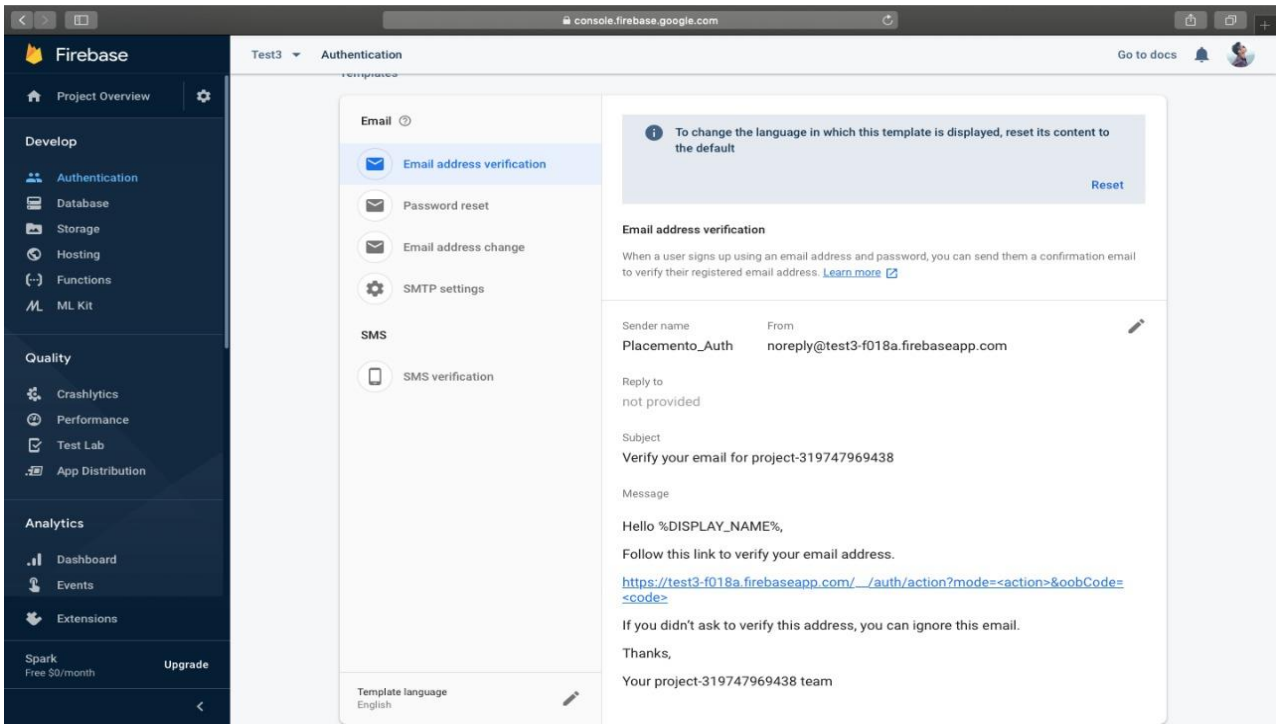


5.2.1(a). Firebase Project Overview

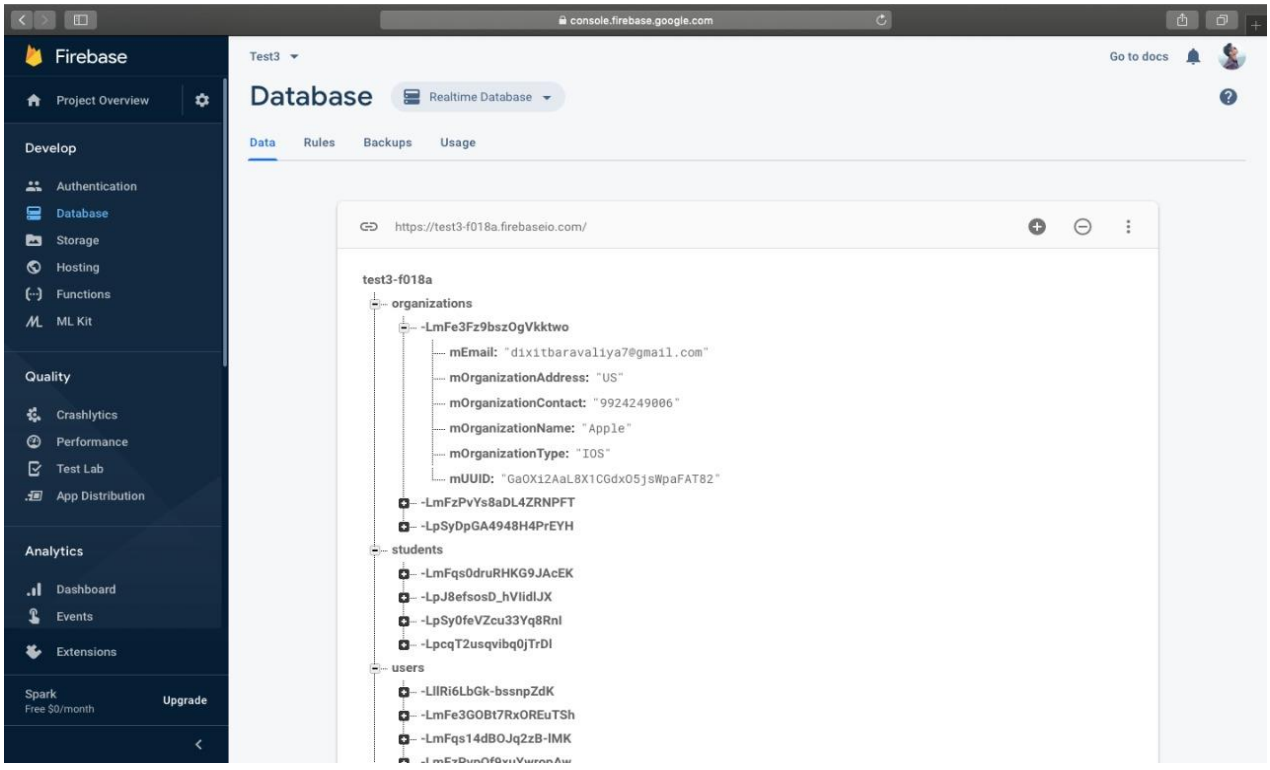




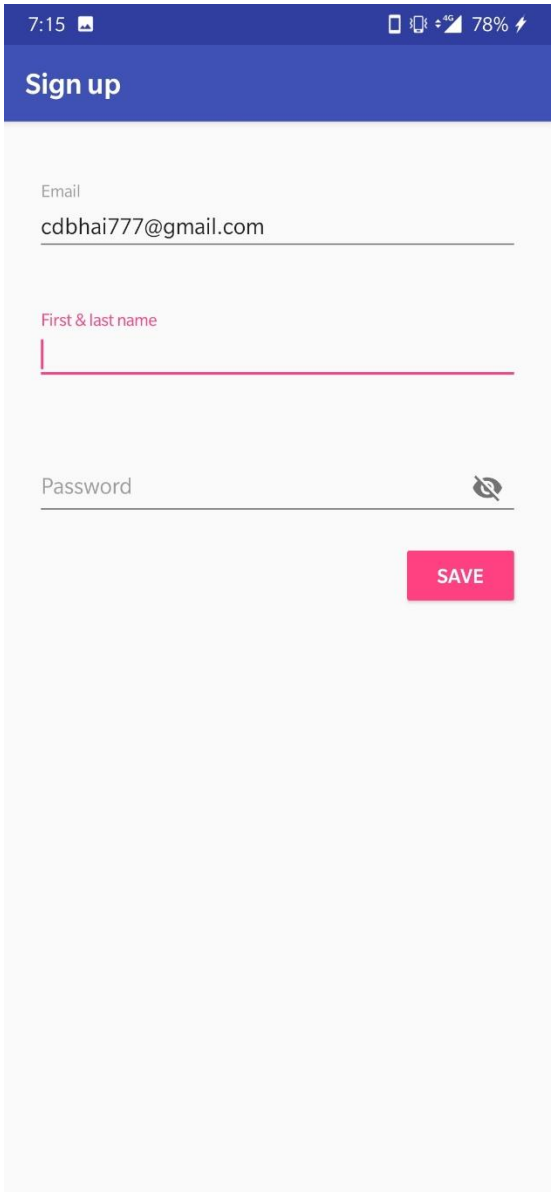
5.2.1(b). Firebase User Authentication



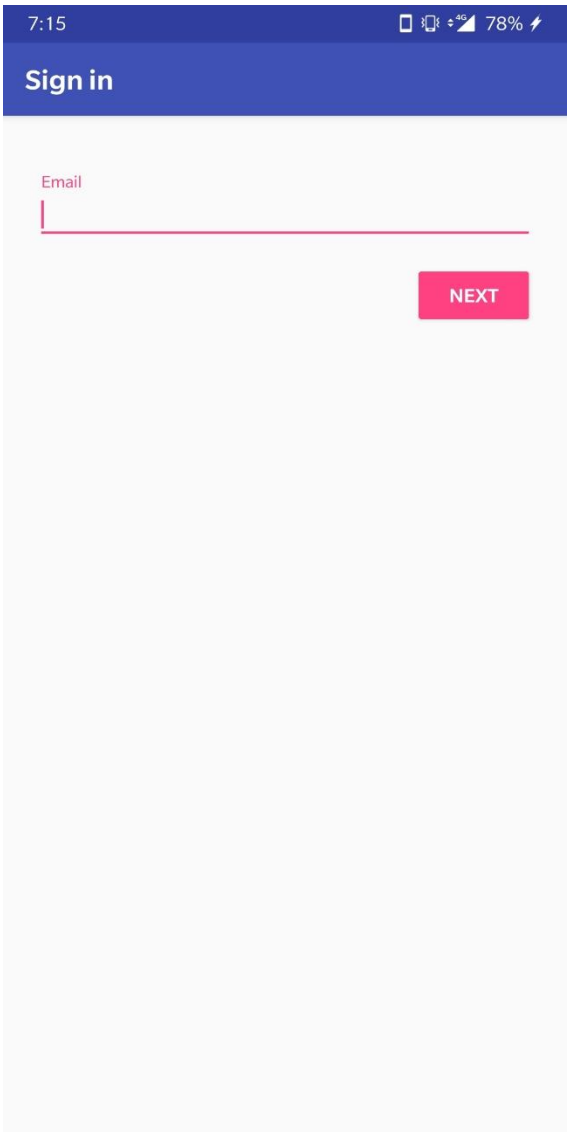
5.2.1(c). User E-Mail Verification



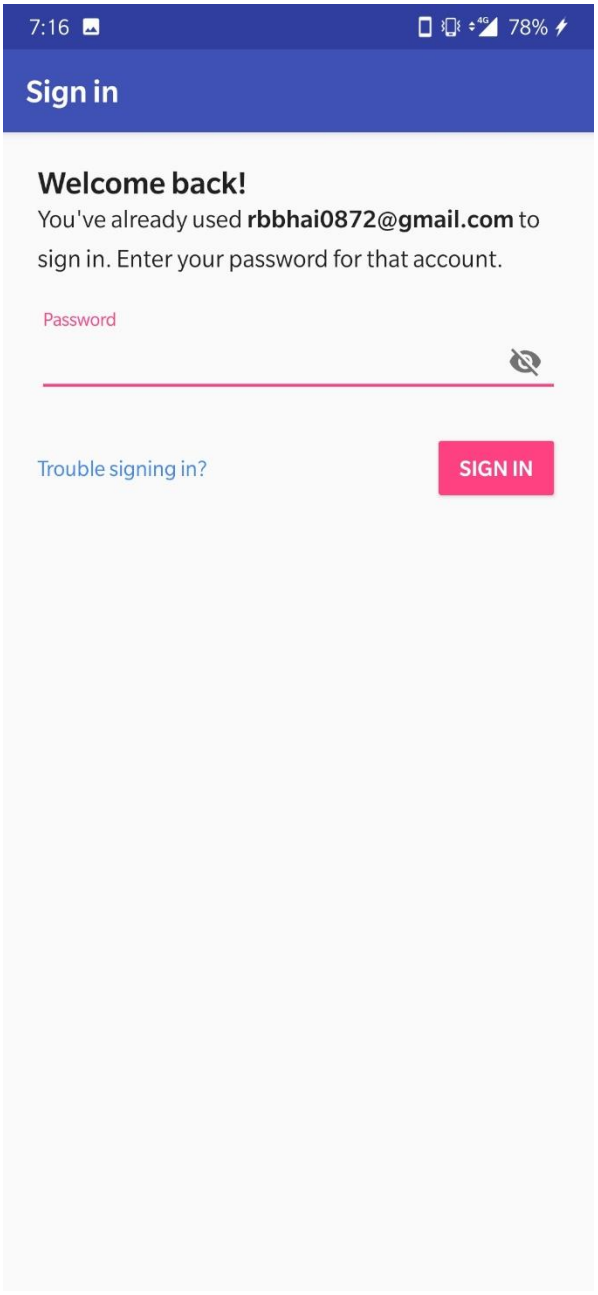
5.2.1(d). Database Record



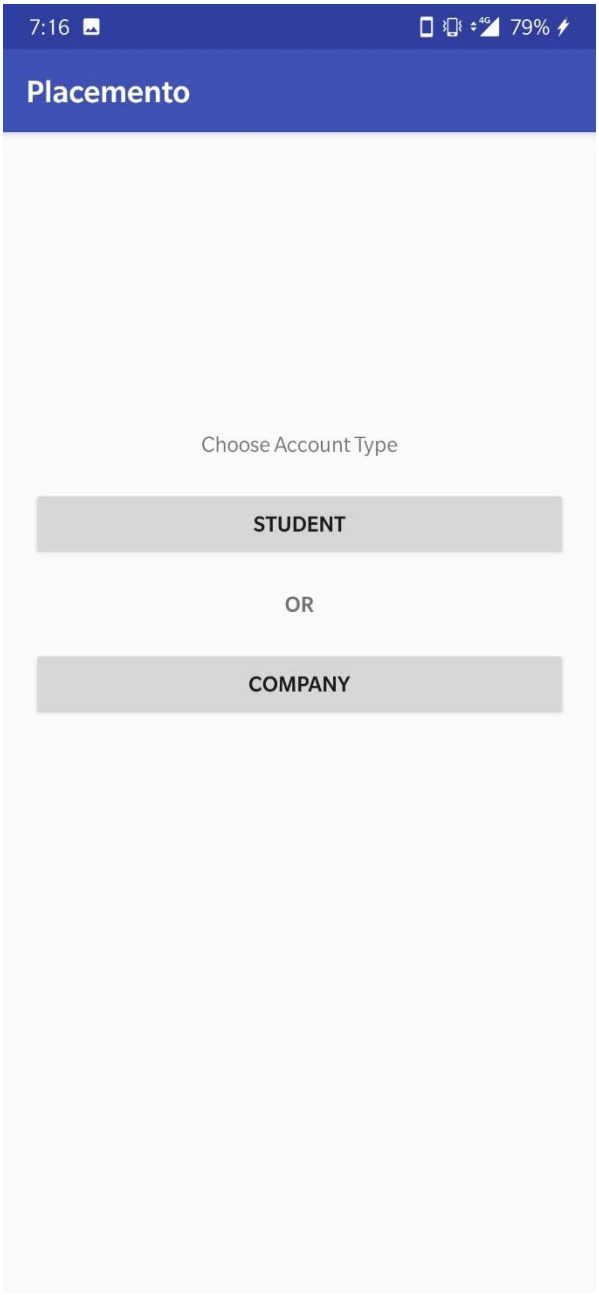
5.2.1(e). Sign Up



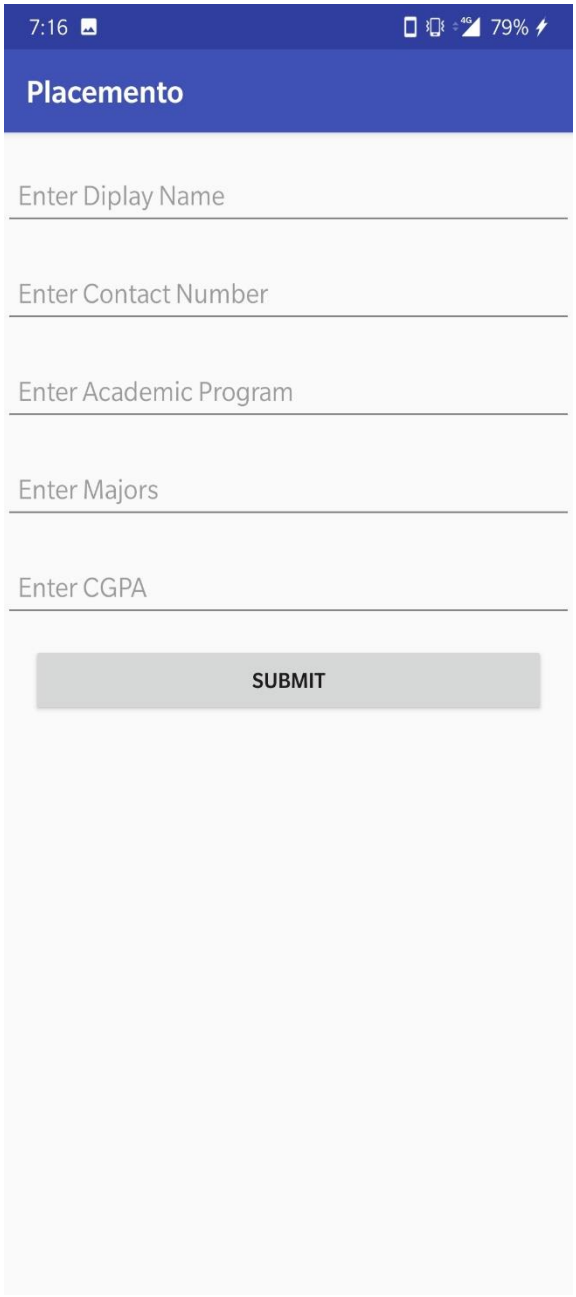
5.2.1(f). Sign In Page1



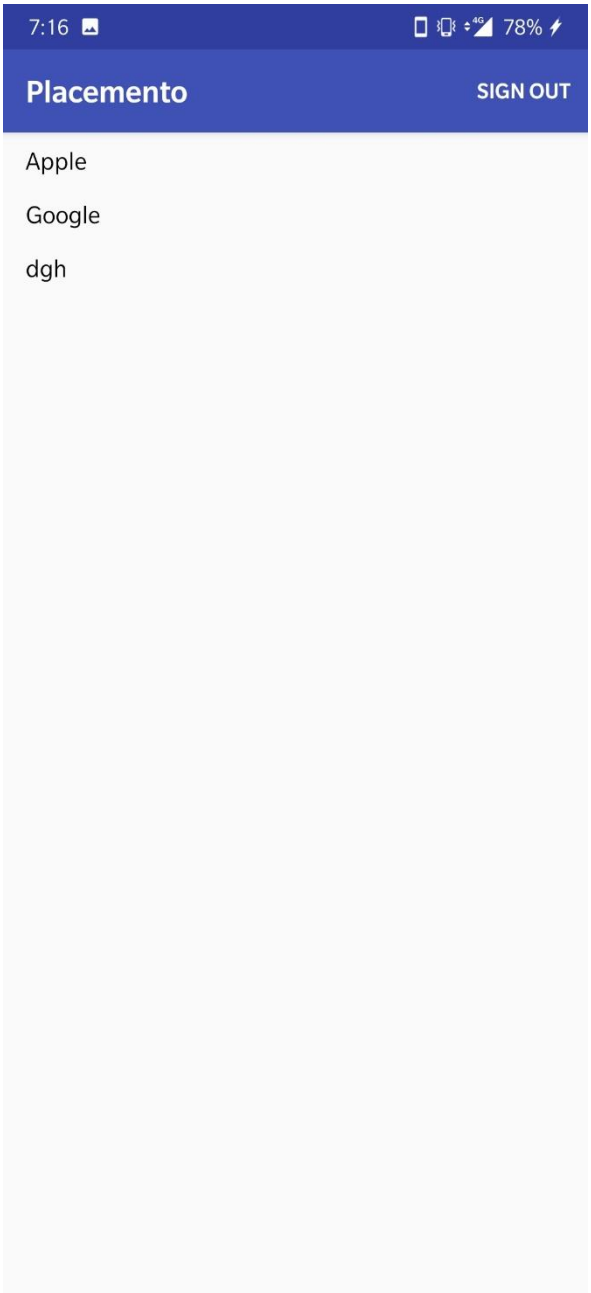
5.2.1(g). Sign In Page2



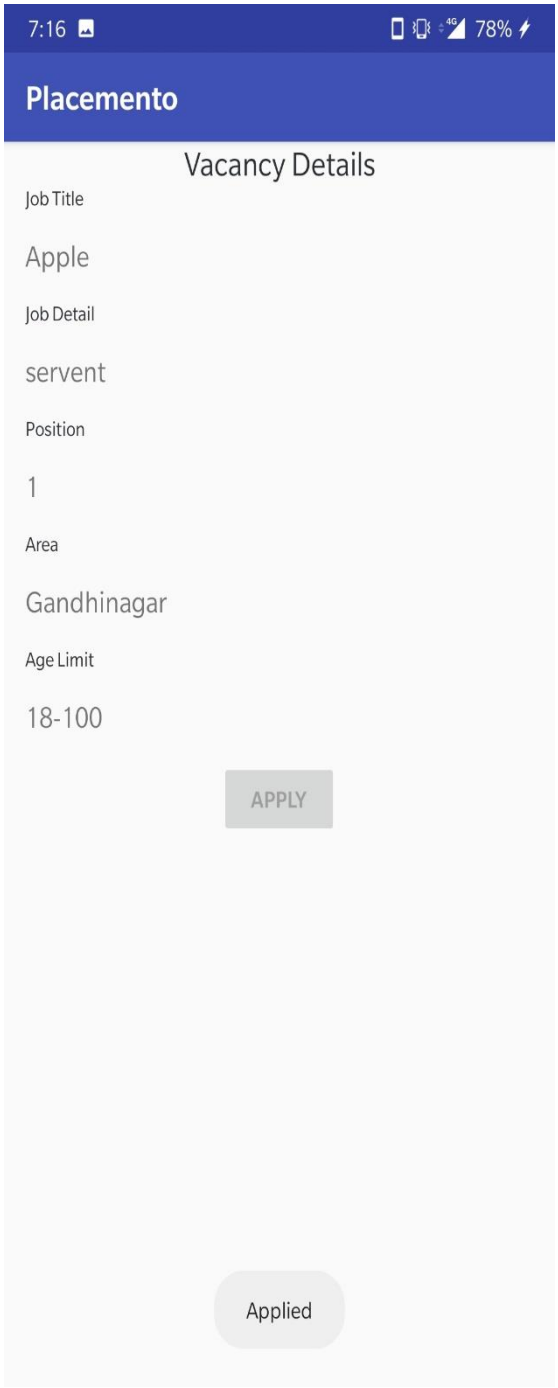
5.2.1(h). Sign In Page3



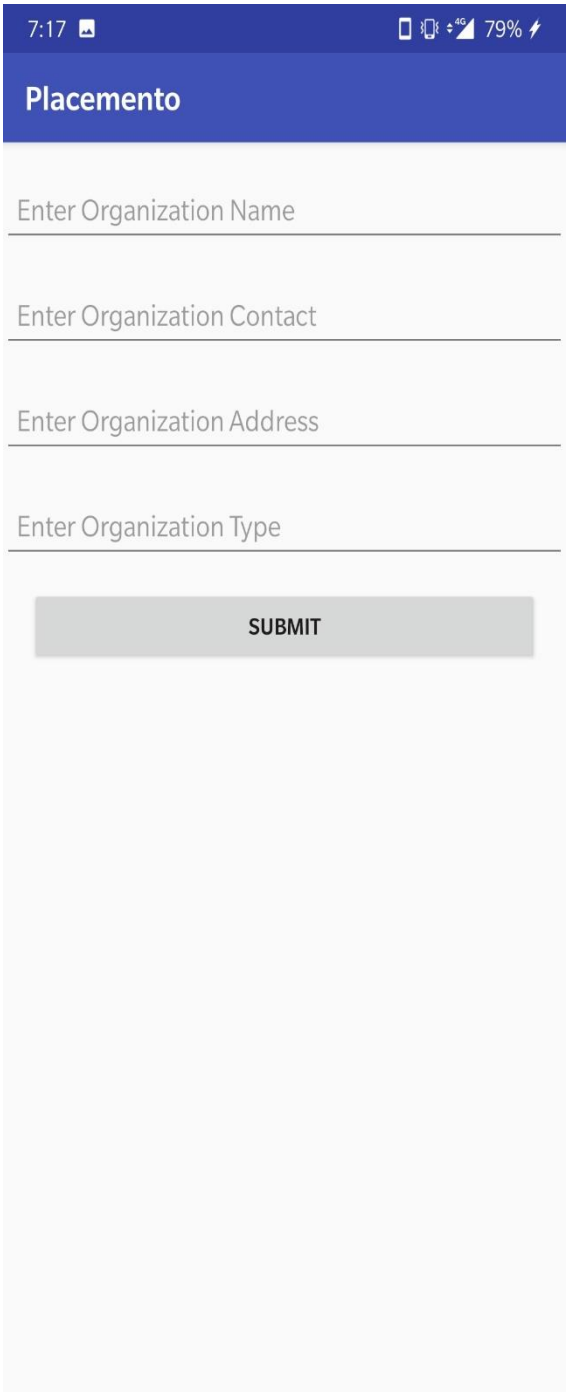
5.2.1(i). Student Details



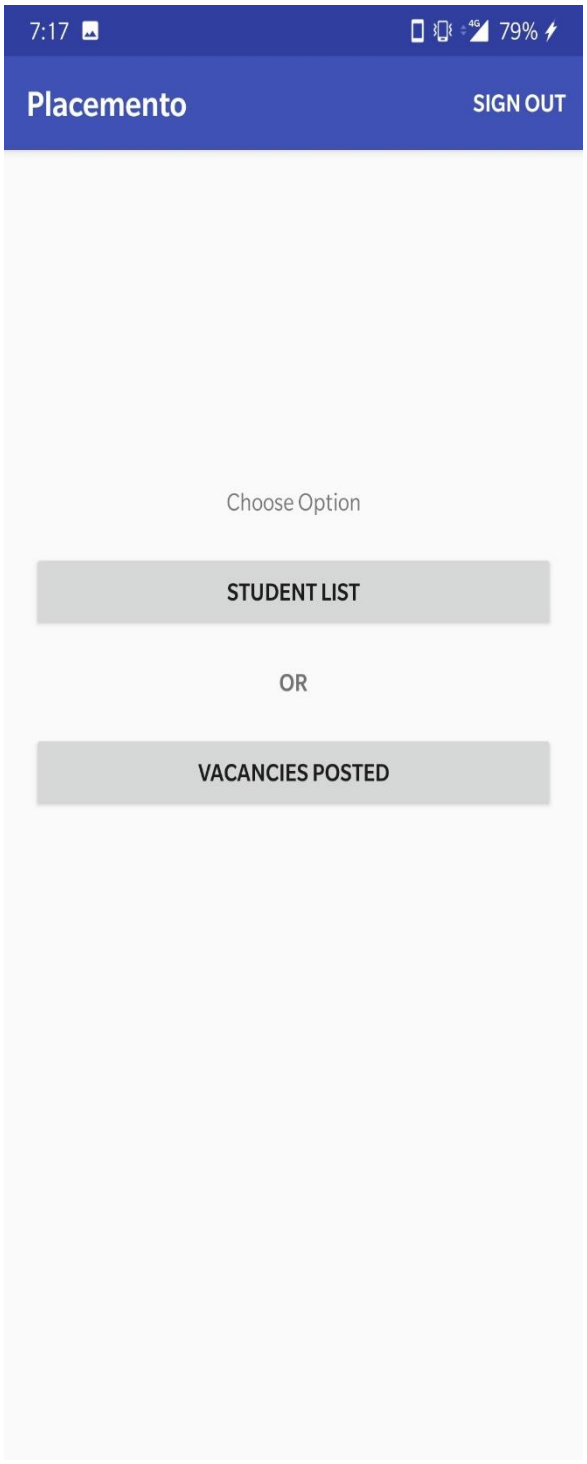
5.2.1(j). Companies List



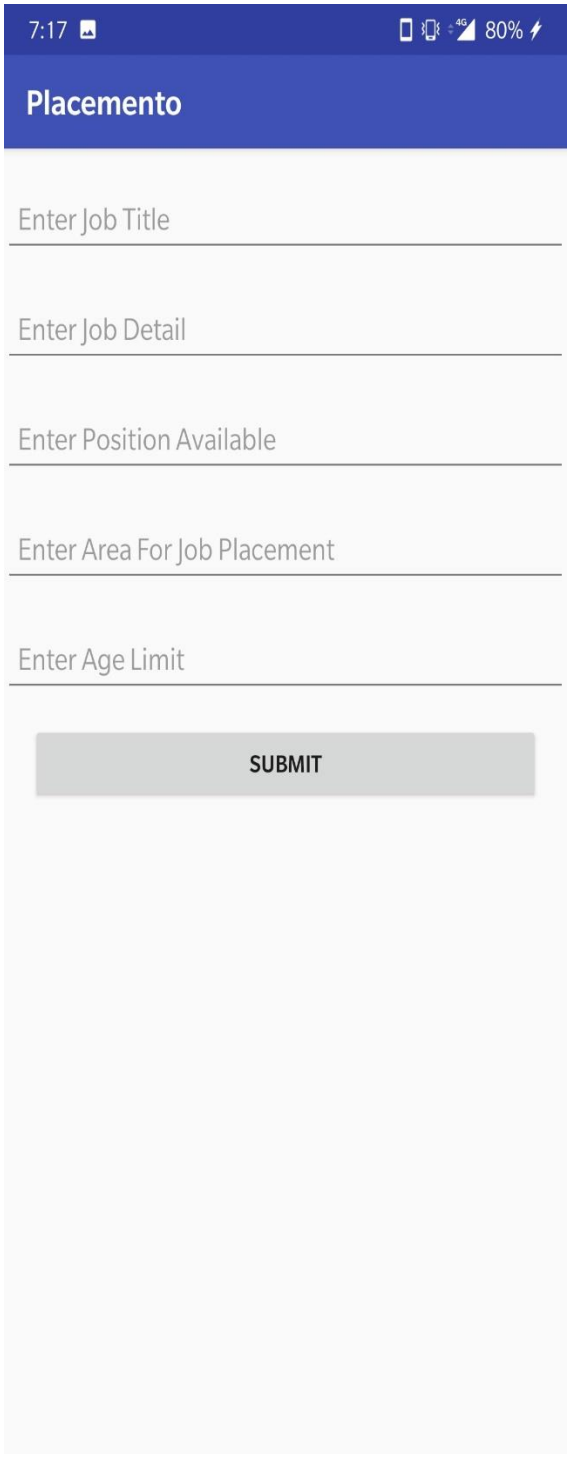
5.2.1(k). Apply For Job



5.2.1(l). Company Registration



5.2.1(m). Company Login



5.2.1(n). Add Job

6. System Testing:

6.1 Test Cases:

Tested By		Arth	
Test Case Type		Unit Testing	
Test Case no		1	
Test Case Name		LogIn/SignUp	
Test Case Description		User should give any user name and password	
Items to be tested			
		Verification Details	
		Input	Result
1	SignUp Verification	Enter User Name , Password and Email	Verify Successfully
2	Email Verification	Send E-Mail for verification	E-Mail Send Successfully
3	SignIn Verification	Enter User Name , Password	Logged In Successfully

6.1(a). SignIn/SignUp Test

7. Conclusion:

- We have successfully created android application which can be used by students. It has plenty of categories makes it attractive and interesting. Various categories help the student to find out his/her desired jobs.

- This system can be vastly used in college campus placement. Many students can find it very useful and they can get jobs according to their eligibility.

8. Bibliography:

- https://firebase.google.com/docs/cloud-messaging/android/client
- https://developer.android.com/studio/profile/inspect-gpu-rendering
- https://firebase.google.com/docs/android/setup
- https://stackoverflow.com
- Android Wireless Application Development By Lauren Darcey and Shane Conder, Pearson Education, 2nd Edition.