

# 1.Introduction

## 1.1 Scope of Work

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.

## 1.2 Usage Scenarios

The recruitment system allows the job seekers to view the job opportunity through Advertisement and Helps to apply for the job. The organization shortlist the applicants for the interview. The shortlisted applicants undergo through a process of Test and Interview. The HR department selects the Applicant based on the performance in the Test and Interview. Finally the recruited applicants are informed. This system makes the task of the job seeker easier rather than waiting in queue for enrollment. This also reduces the time consumption for both for the job seeker and organization

# 2. Requirement Analysis

## 2.1 Functional Requirements

### 2.1.1 Update information

This functionality will update the information of the student, Company and the Admin.

### 2.1.2 Change Password

This functionality will allow user(student,Company or Admin) to change their account password.

### 2.1.3 Sign Up

This functionality will allow Company,student or Admin to sign up(make a new account).

### 2.1.4 Login

This functionality will allow users(Students,Admin,Company) to login by their respective login credentials.

### 2.1.5 Apply for jobs

This functionality is only used by the students to apply for jobs.

### 2.1.5 Post Jobs/Add Job details

This functionality is only used by the company to post a new job.

## 2.2 Non Functional Requirements

The following are the non-functional requirements:

### 2.2.1 Security Requirements:

The application should not be able to access personal data of the user.

It also ensures that its sensitive information is not revealed to the external world under any condition.

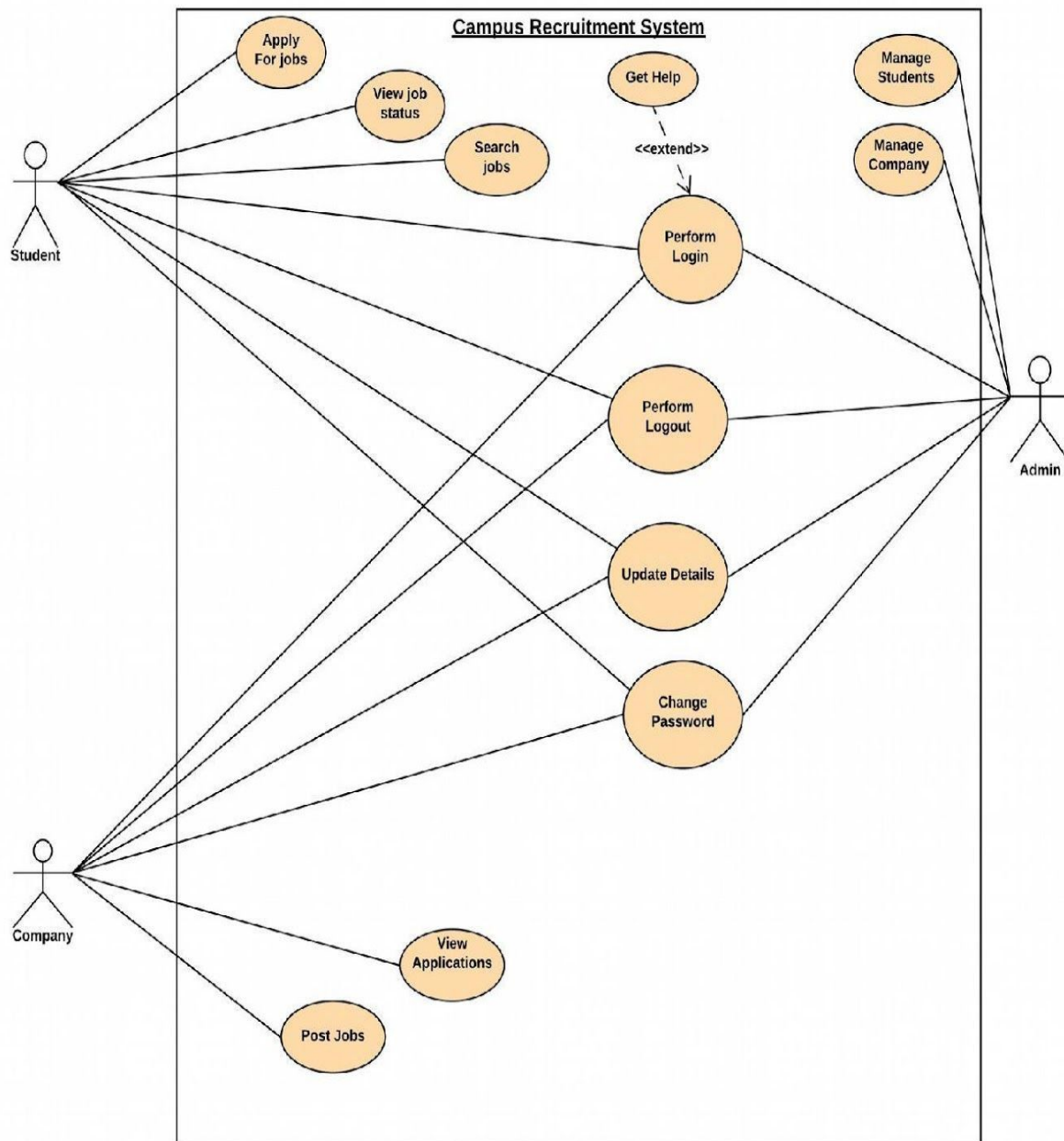
**2.2.2 Performance Requirements:** The response time of the application should be minimum. A user can undo his action any point of time.

**2.2.3 Availability:** All the information that the user requires will be readily available and accessible. The application is also independent of the user's location. The application will be available to the users anytime they wish to use it irrespective of the time and location.

**2.2.4 Usability:** The application has a very simple and user friendly user interface. Any new user can easily use the application. Every screen has self-explanatory buttons and explanations that avoids any confusion for the user. The layout of the user interface is consistent throughout the application. The application supports English language.

**2.2.5 Documentation:** Software Requirements Specification is provided for the users as well as the developers and System Design Description is provided for the developers and the maintenance engineers.

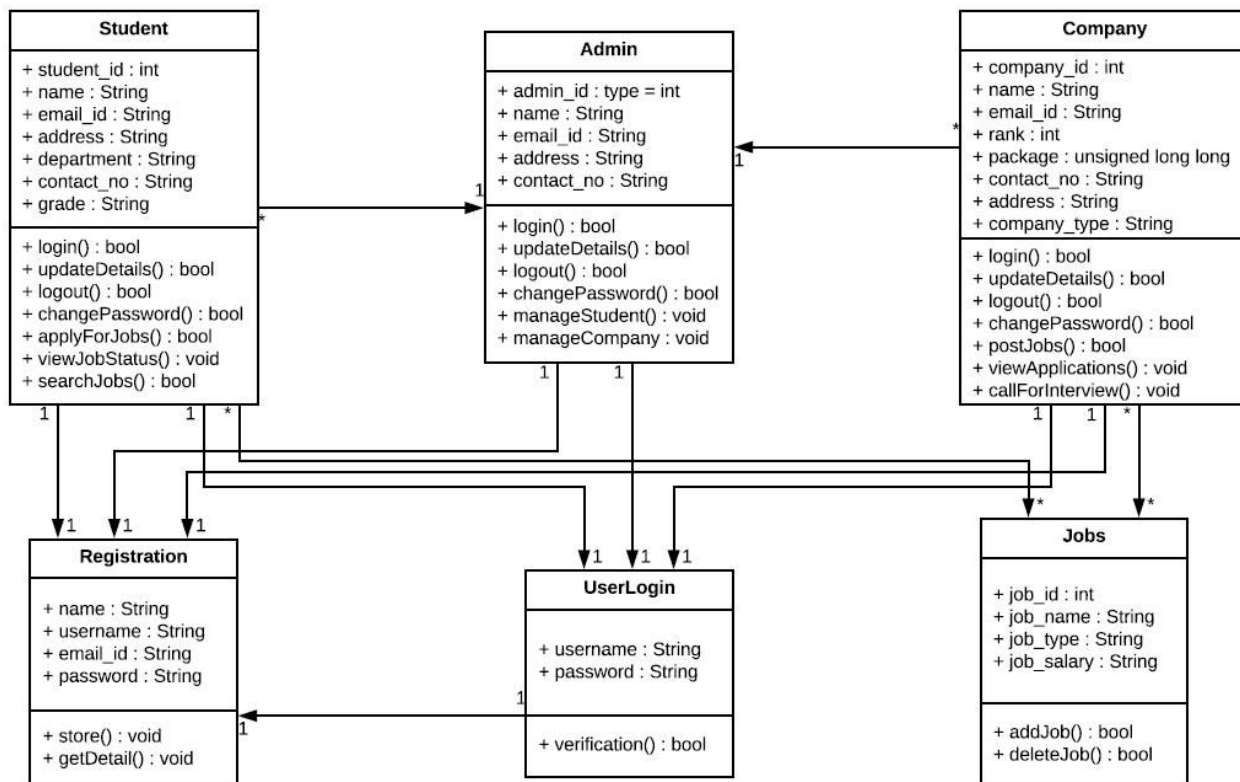
## 2.3 Use Case Scenarios



## 2.4 CLASS DIAGRAM

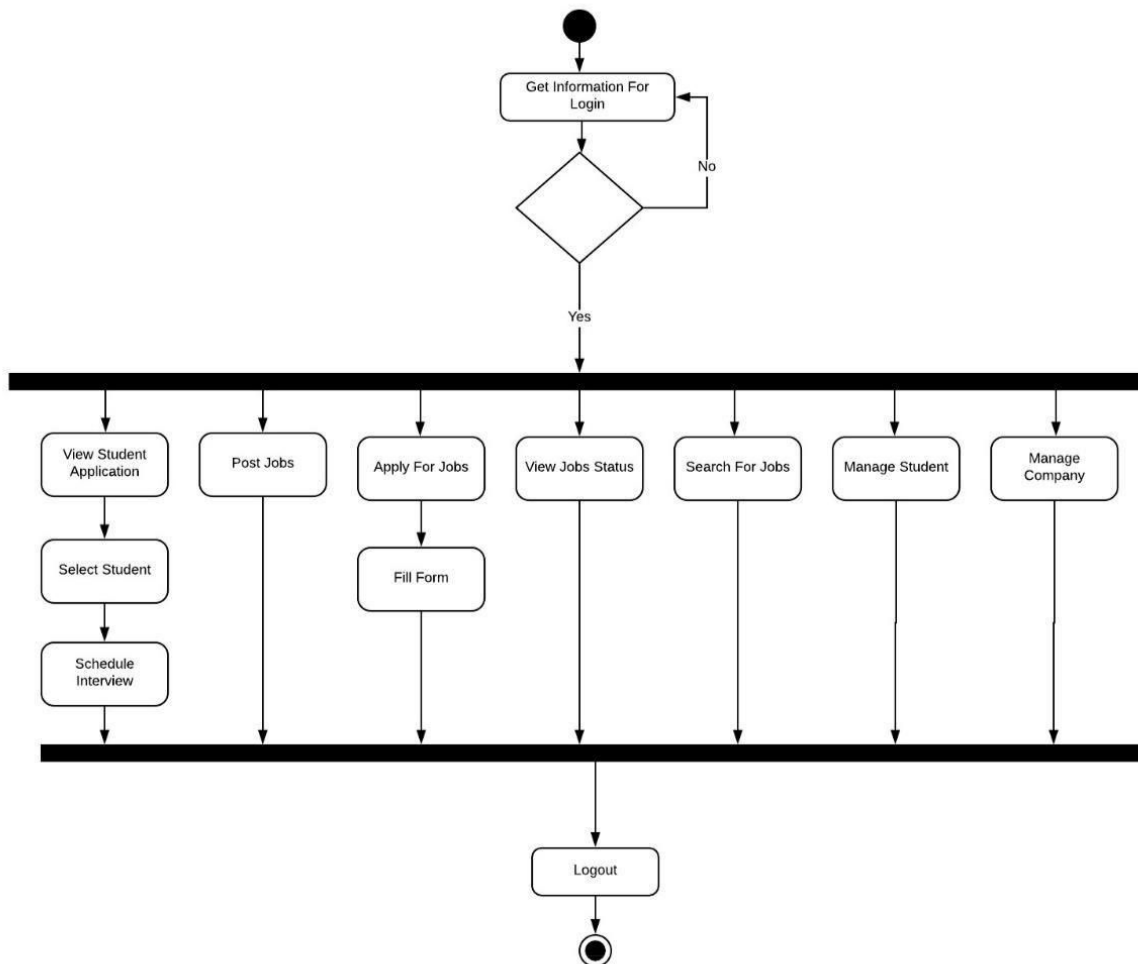
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code.

**CLASS DIAGRAM FOR CAMPUS RECRUITMENT SYSTEM**

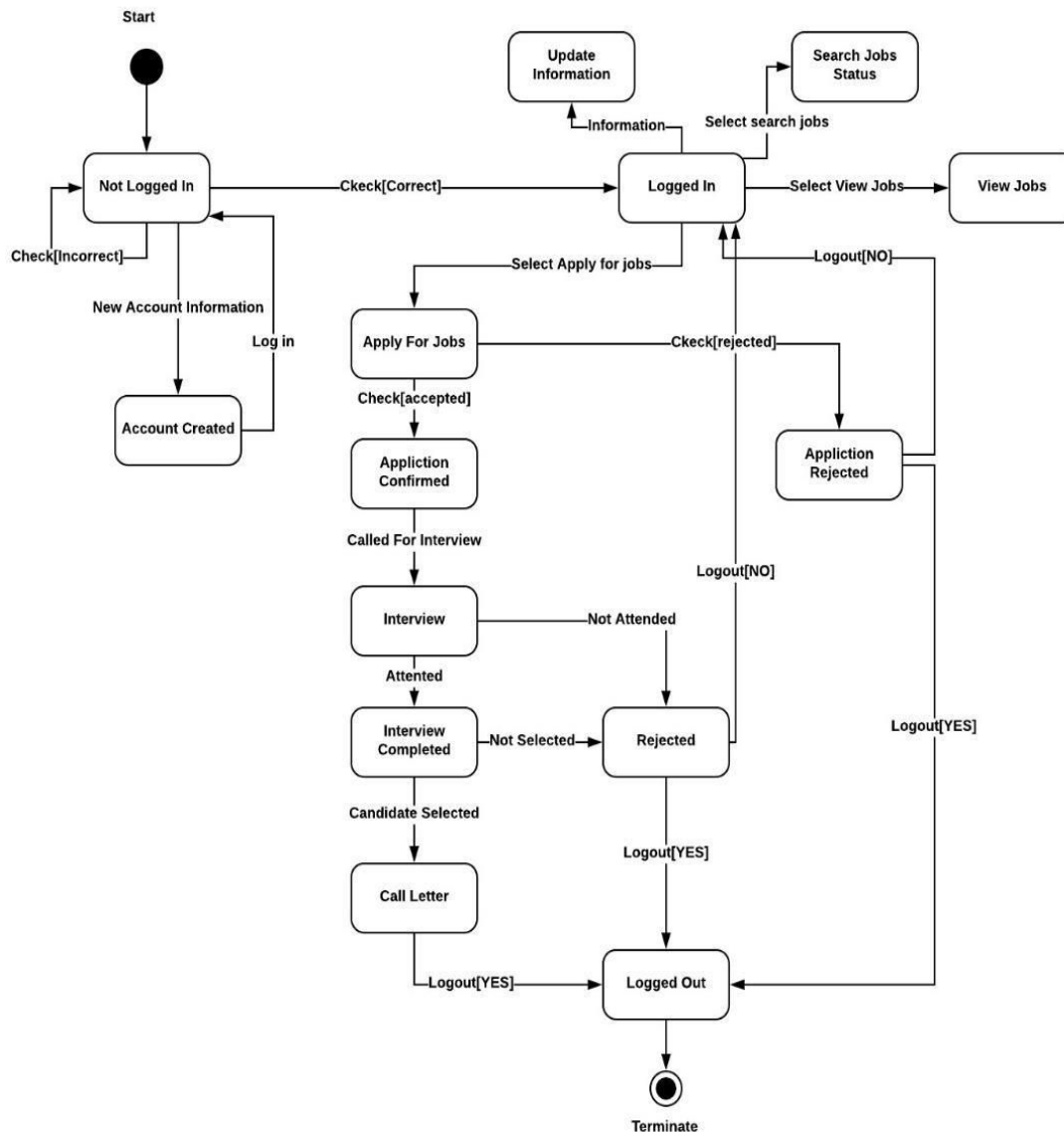


## 2.5 Activity Diagram

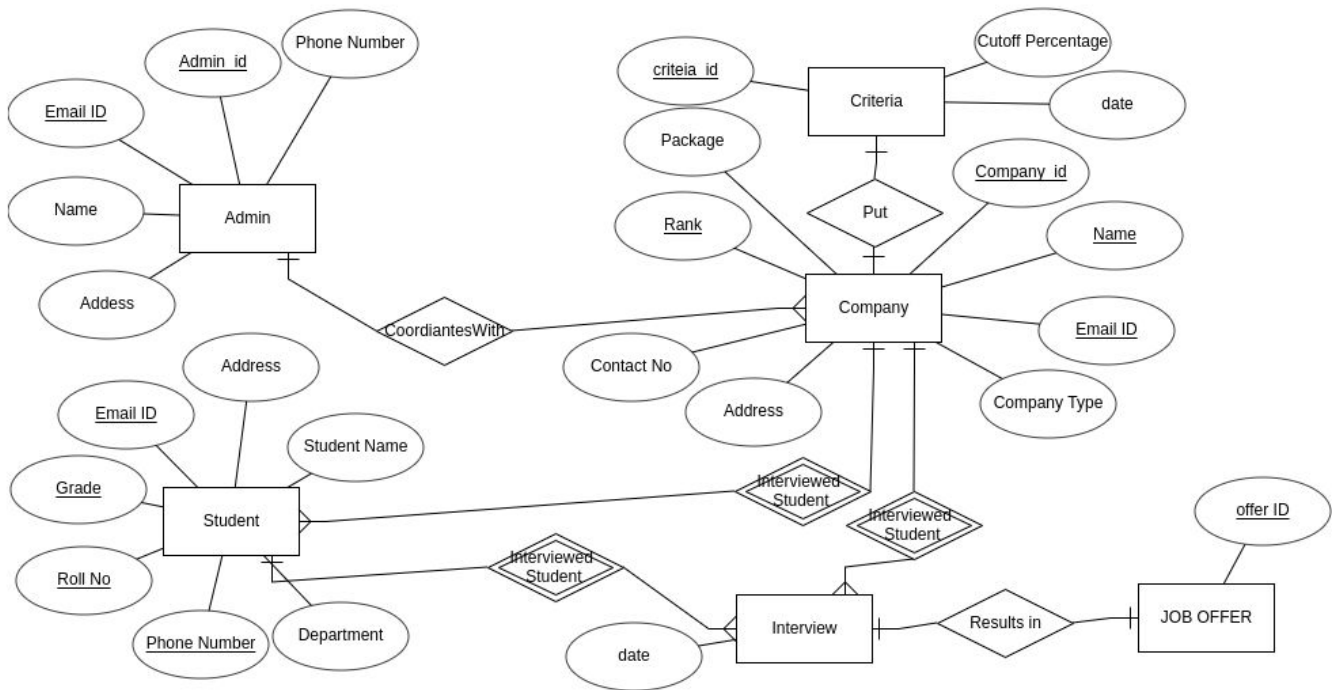
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.



A state machine diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.



## 2.7 ER Diagram



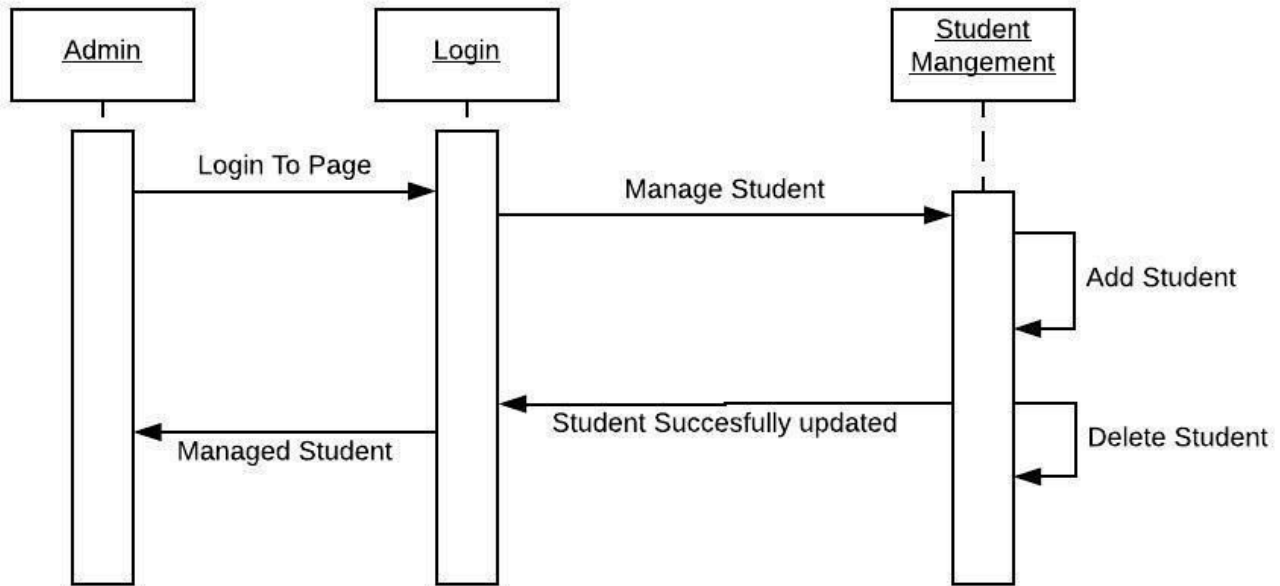
## **2.8 Sequence Diagram**

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.

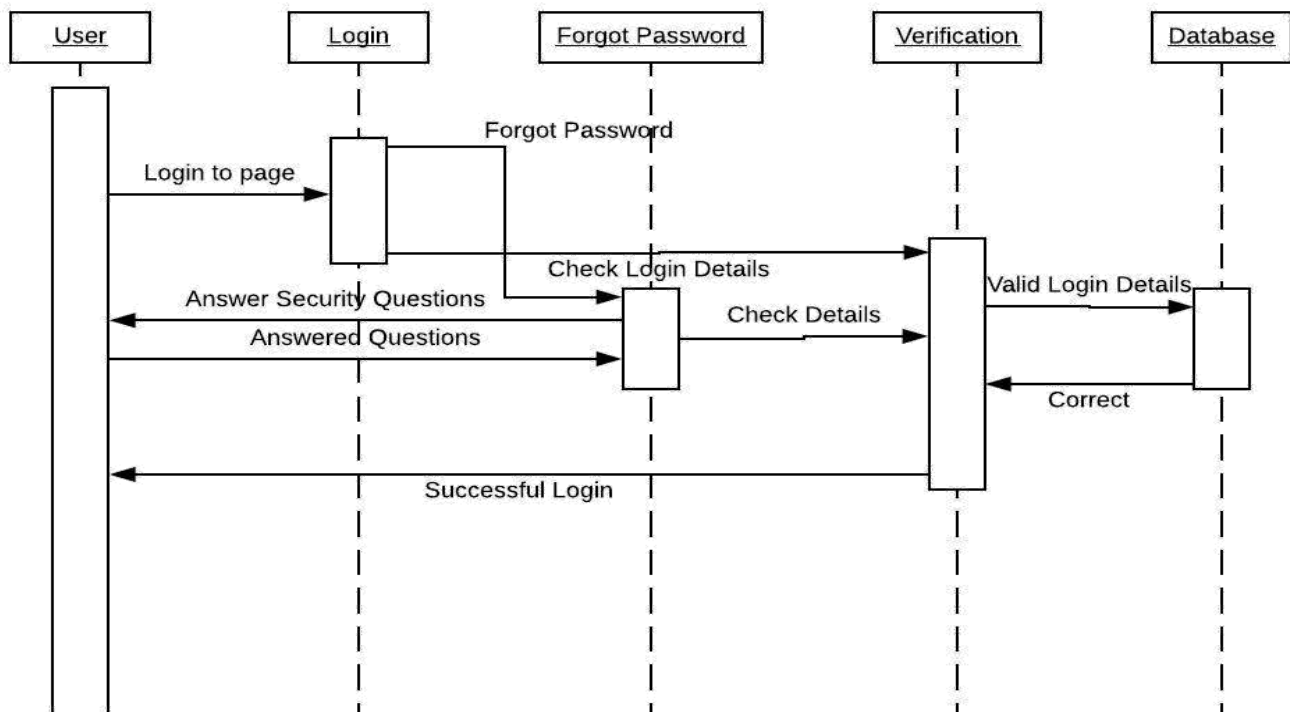
### **2.8.1 Sequence diagram for manage student is shown below:**



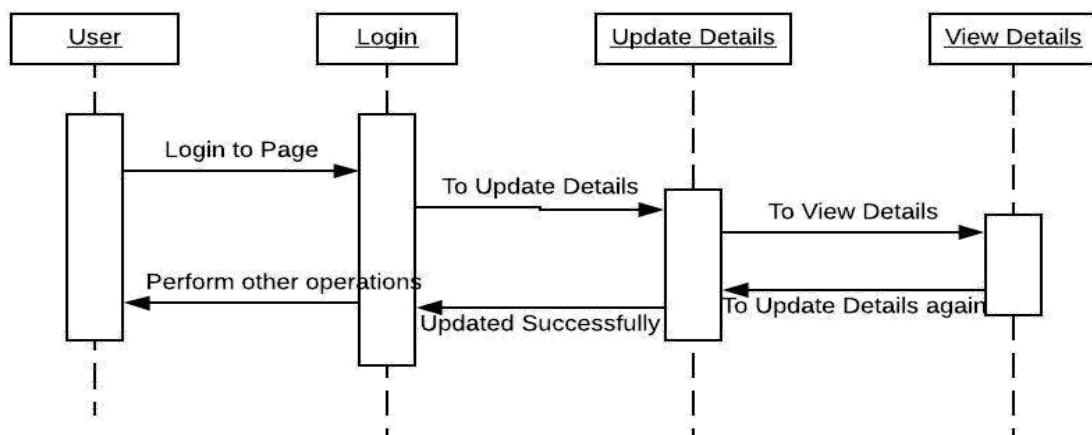
### Manage Student



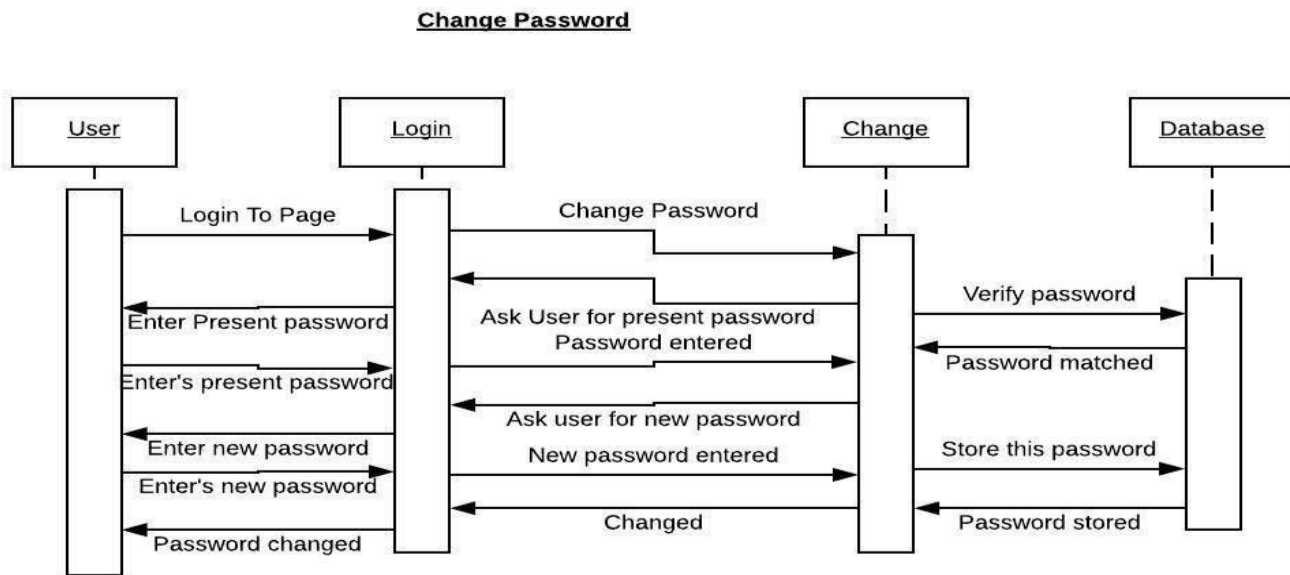
**2.8.2 Sequence diagram for login & update details are shown below:**



### Update Details

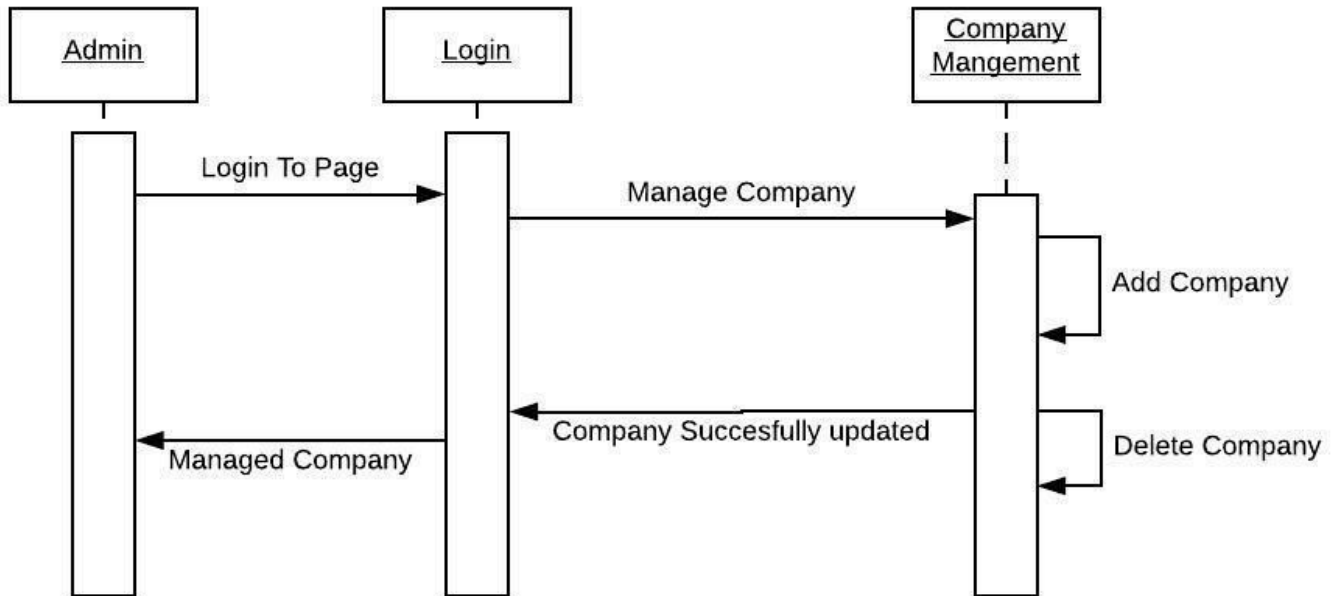


**2.8.3 Sequence diagram for change password is shown below:**



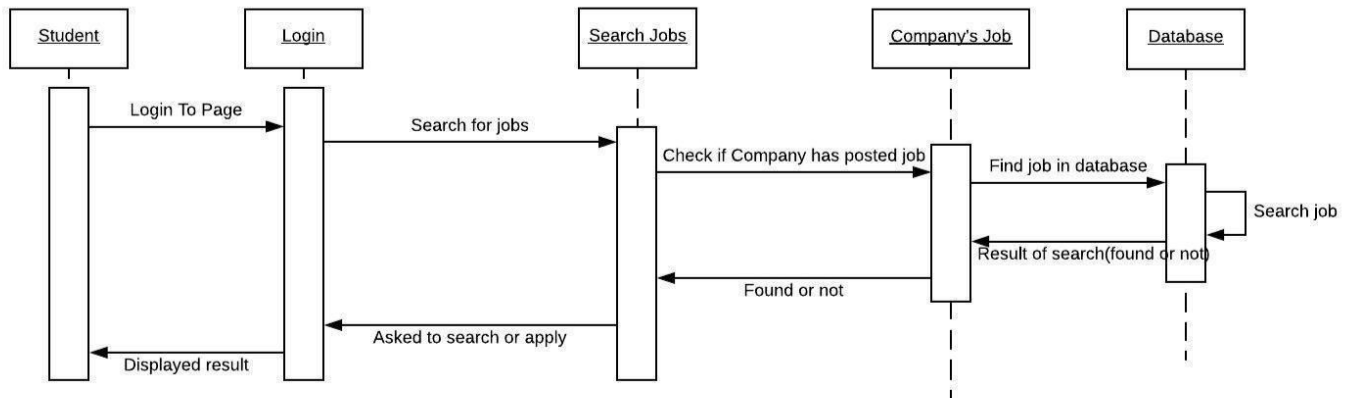
**2.8.4 Sequence diagram for manage company is shown below:**

### Manage Company

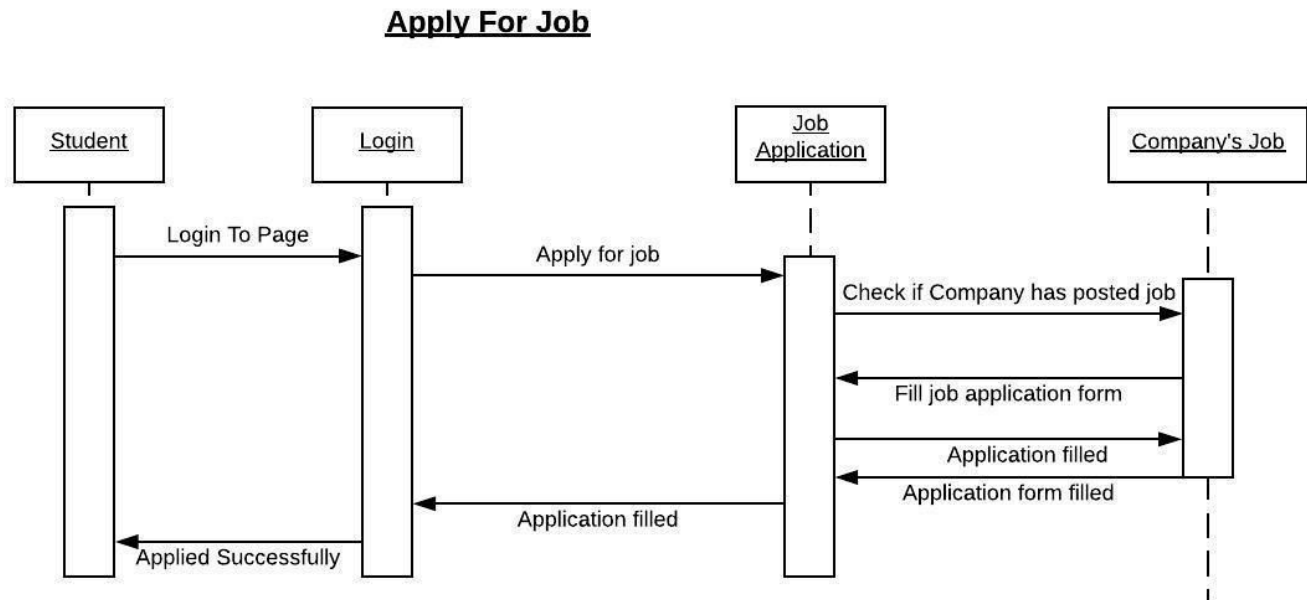


2.8.5 Sequence diagram for search jobs is shown below:

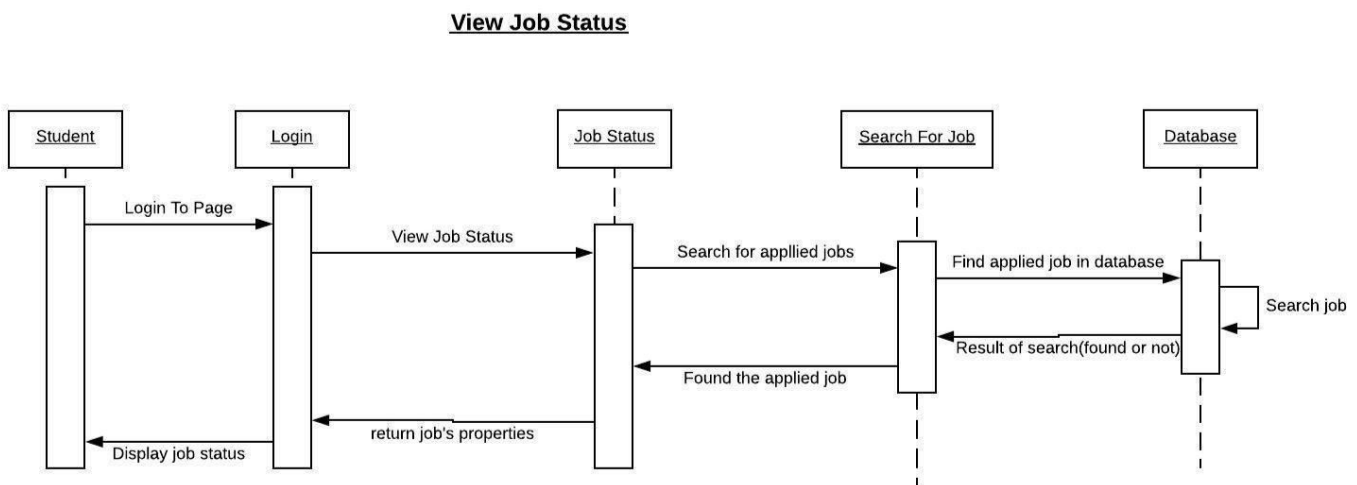
### Search Jobs



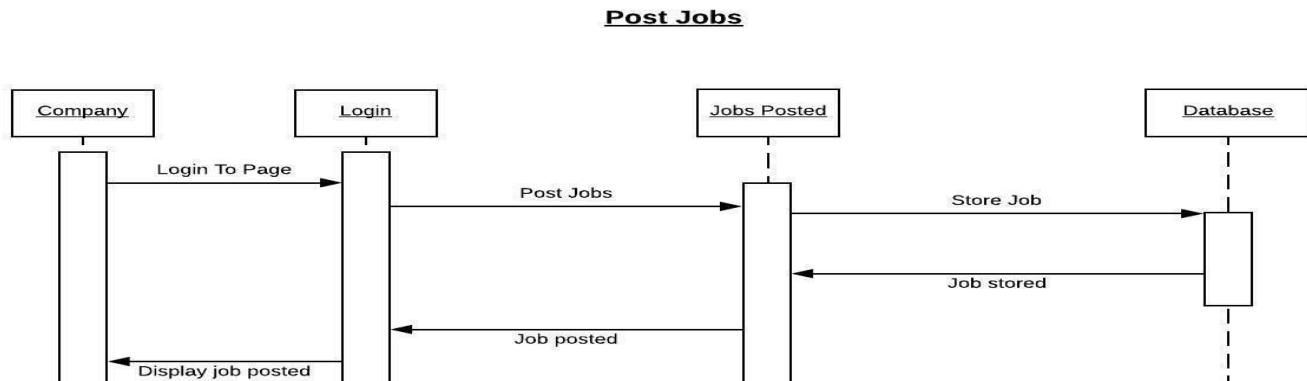
## 2.8.6 Sequence diagram of apply for job is shown below:



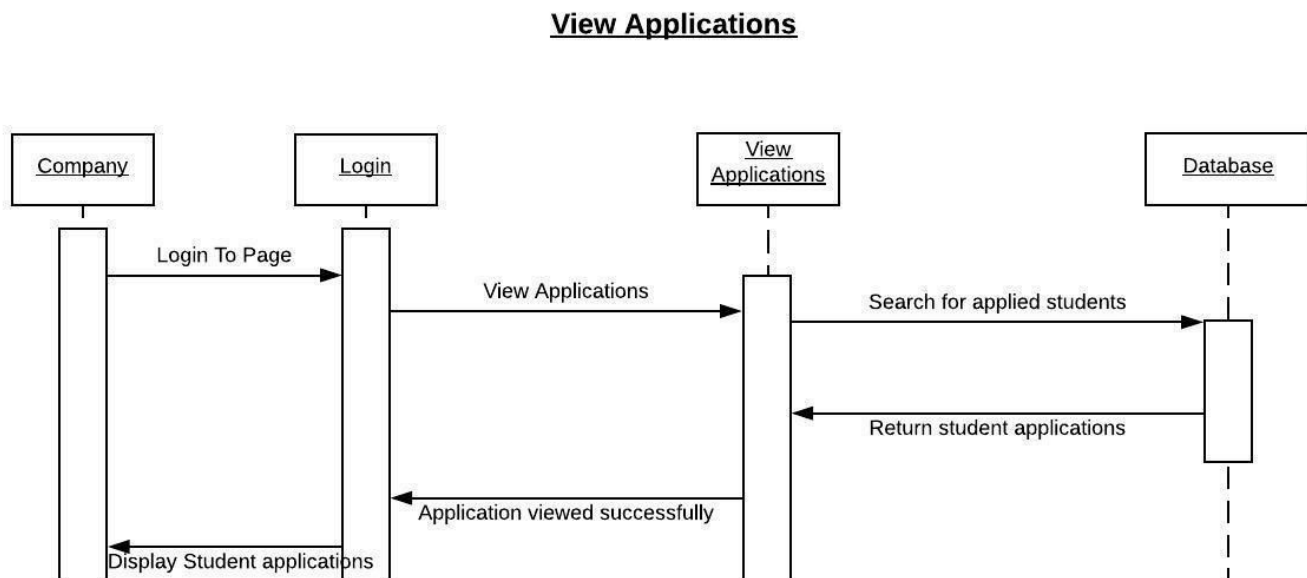
## 2.8.7 Sequence diagram for view job status is shown below:



### 2.8.8 Sequence diagram for post jobs is shown below:



### 2.8.9 Sequence diagram for view applications is shown below:



## **3. SYSTEM DESIGN**

### ***3.1 Design Goals***

The application comprises of many features and hence the system is divided into various components. The main objective of this section is to elaborate the system design and to give an overview of the various components of the application including their interfaces. It also provides information about the relationship between the various components and the different data elements used by each of the components. It also explains the overall system design. The application has a client-server architecture with the application running on the client side and the files residing on the server side with which the user interacts through the application. The following sections contains class diagram, sequence diagram and activity diagrams representing the various components and their interactions and also the detailed description of each of the components.

### ***3.2 Detailed Design Methodologies***

#### **3.2.1 Detailed Description of the User Interface**

The user interface section defines the way various stakeholders interact with the system.

All the screens will be developed to work on android mobile. Error messages will appear as a popup on the screen. The maximum size of error message will be 40 characters. Buttons will there to make the navigation simpler.

A first time user of the mobile should see the login screen when he/she will open the android application. If the user has not registered to the, then he/she should be able to redirect to the sign up page from login screen. Every user should have the profile where he/she can apply for the job. After the creation of account the user can login to the application and will be able to apply for the jobs.

### 3.2.1.1 Login & sign Up Screen Interface

The image displays three mobile application screens for the 'Campus Recruitment System'.

**Screen 1 (Left): Login Options**  
This screen features a blue header with the text 'Campus Recruitment System'. Below the header, the title 'CAMPUS RECRUITMENT SYSTEM' is centered. Three blue buttons are stacked vertically: 'ADMIN LOGIN', 'STUDENT LOGIN', and 'COMPANY LOGIN'. Below these buttons is a horizontal line with the word 'OR' in the center. At the bottom, there is a green button labeled 'CREATE NEW ACCOUNT'.

**Screen 2 (Top Right): Login Form**  
This screen has a blue header with the text 'Campus Recruitment System'. The title 'CAMPUS RECRUITMENT SYSTEM' is centered. Below the title, there are two input fields: 'Username' and 'Password'. Below the 'Password' field is a blue button labeled 'LOGIN'. At the bottom, there is a link that says 'FORGOT PASSWORD?'.

**Screen 3 (Bottom): Sign Up Form**  
This screen has a blue header with the text 'Campus Recruitment System'. The title 'Create New Account' is centered. Below the title, there are seven input fields: 'Name', 'Username', 'UserId', 'EmailId', 'ContactNo', 'Password', and 'Confirm Password'. Below the 'Confirm Password' field is a dropdown menu labeled 'Admin' with a downward arrow. At the bottom, there is a blue button labeled 'SIGN UP'.

### Objects & Actions

#### Login :

Student ,Admin or a company can login by their specific credentials.

#### Sign Up:

User Can create a new account



## **4. Work Done**

### **4.1 Development Environment**

The application being developed would be an Android App and would be called Campus Recruitment System.

- 1.Create a Sqlite database to store the user informations.
- 2.Create Interface for the application
- 3.Connect front end to backend of the application

#### **4.1.1 Create a Sqlite database**

Create Sqlite database for different types of user there will be different tables in the database.

#### **4.1.2 User Interface**

Design user Interface in XML android

### **4.2 Testing**