UML Documentation

for

Campus Recruitment System

Version 1.0

Prepared by Bobby Patil and Chetan Prajapati

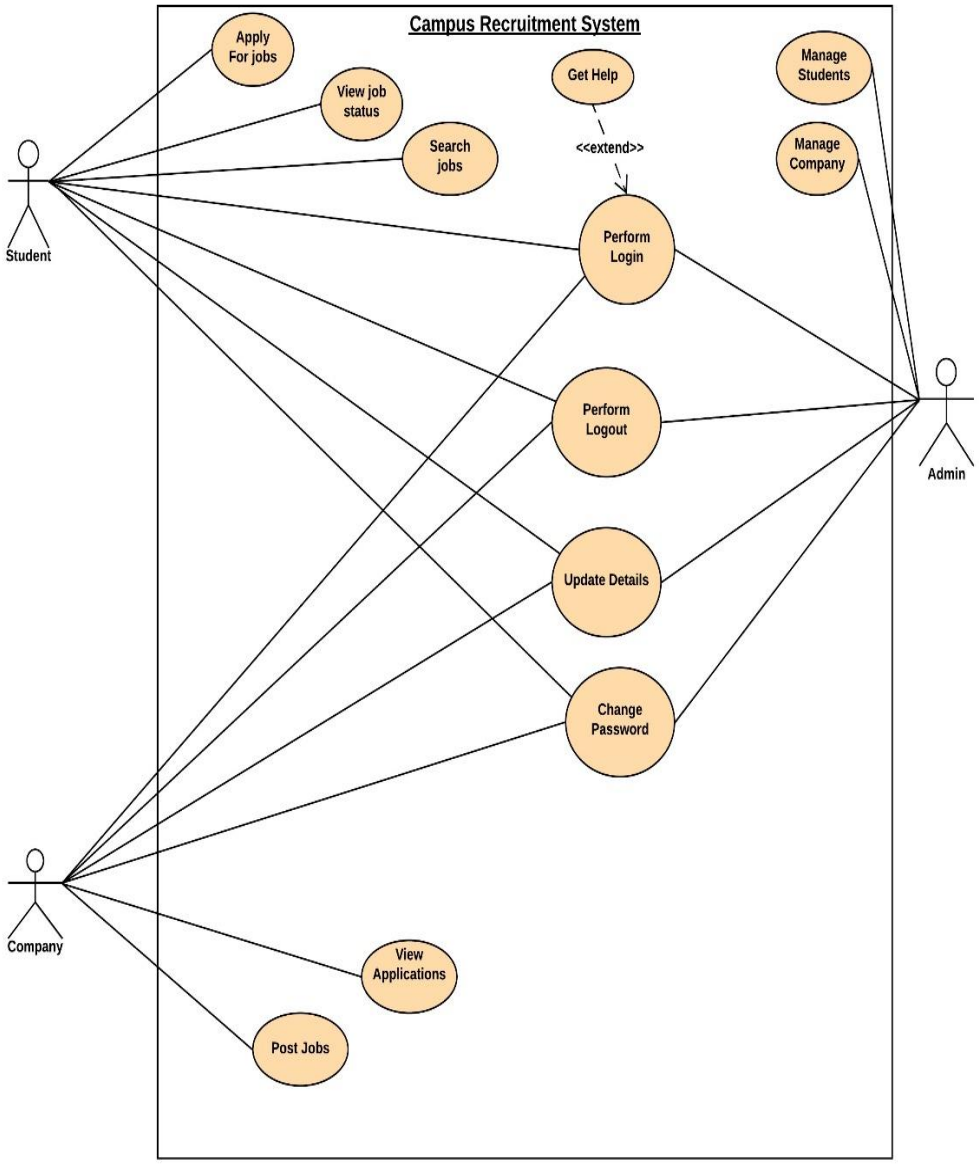
National Institute of Technology Karnataka, Surathkal

21st February 2018

Revision History

Name	Date	Reason For Changes	Version
Use Case Diagram	21/02/2018	Initial Version	1.0
Class Diagram	21/02/2018	Initial Version	1.0
Activity Diagram		Initial Version	1.0
Sequence and Collaboration Diagram	21/02/2018	Initial Version	1.0
State Machine Diagram	21/02/2018	Initial Version	1.0
Component Diagram	21/02/2018	Initial Version	1.0
Deployment Diagram	21/02/2018	Initial Version	1.0

Use-Case Diagram For Campus Recruitment System

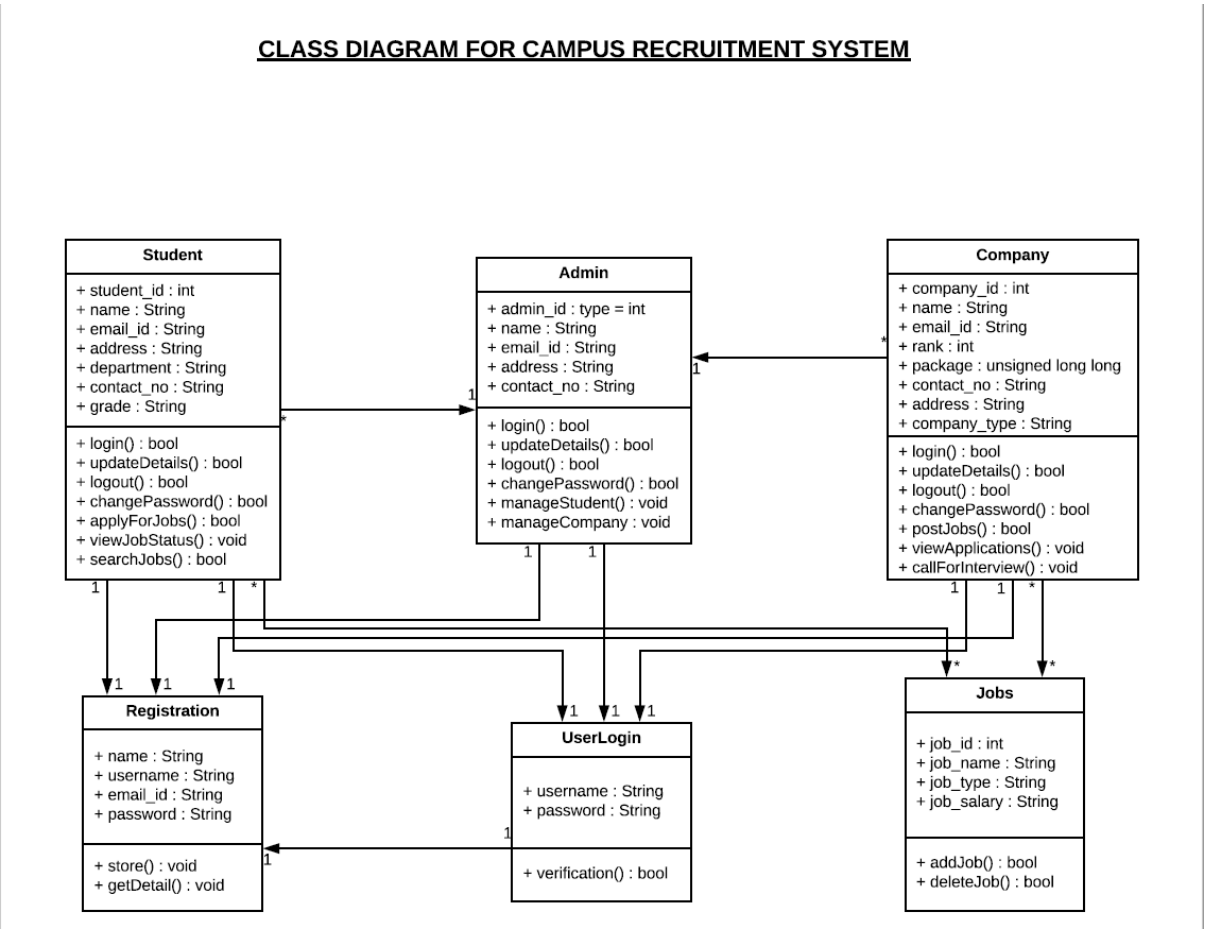


Class Diagram For Campus Recruitment System

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.

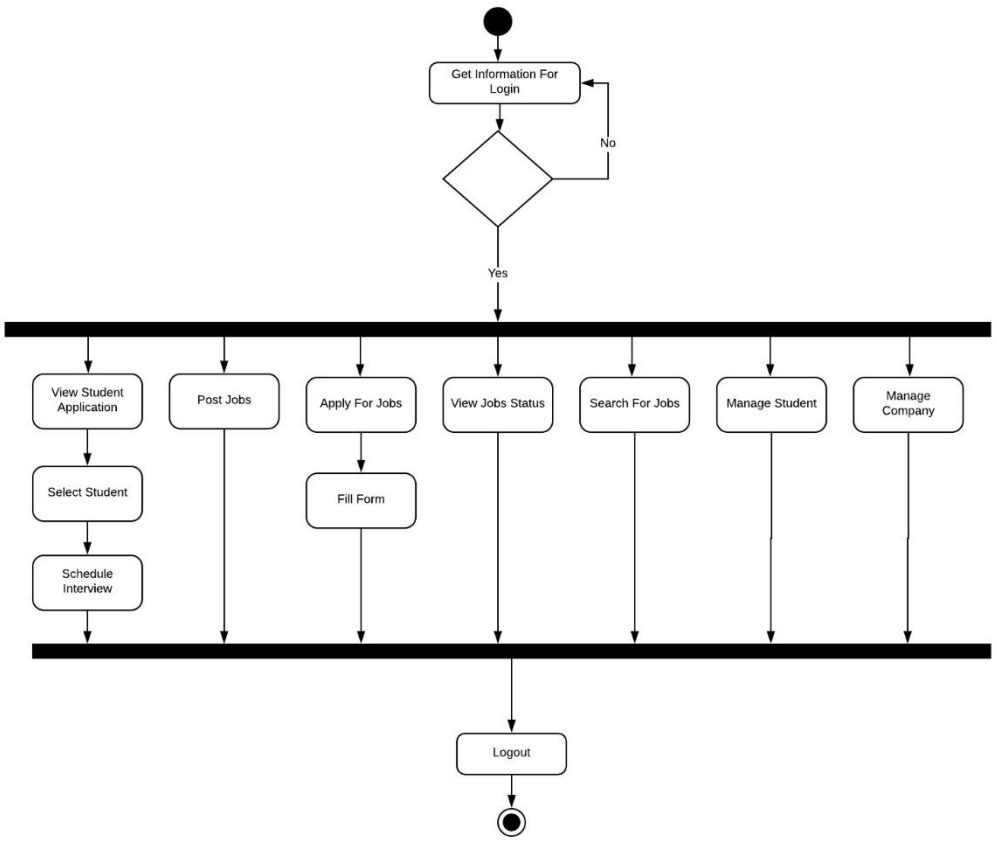
Class diagram of campus recruitment system is shown below:



Activity Diagram For Campus Recruitment System

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.

Class diagram of campus recruitment system is shown below:

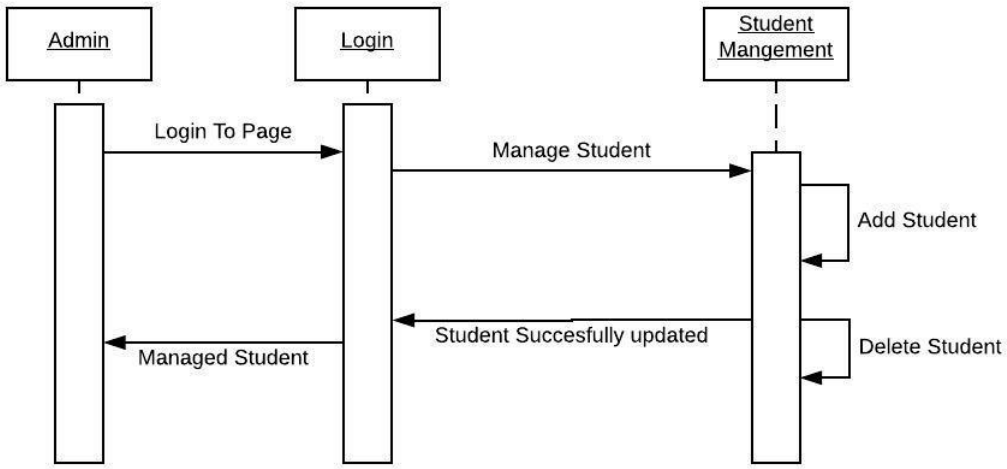


Sequence Diagram For Campus Recruitment System

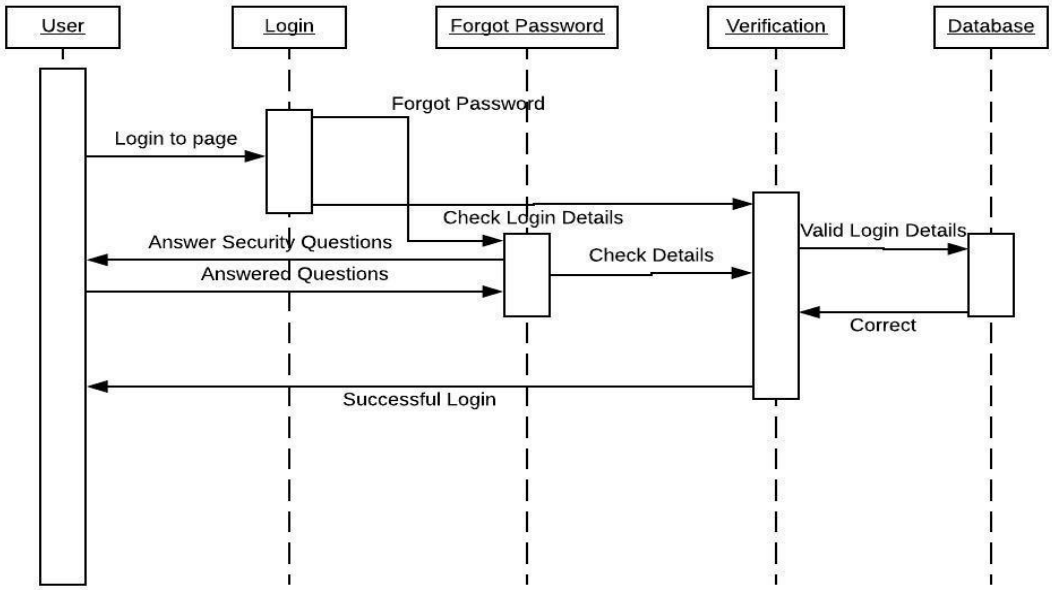
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case 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.

1.Sequence diagram for manage student is shown below:

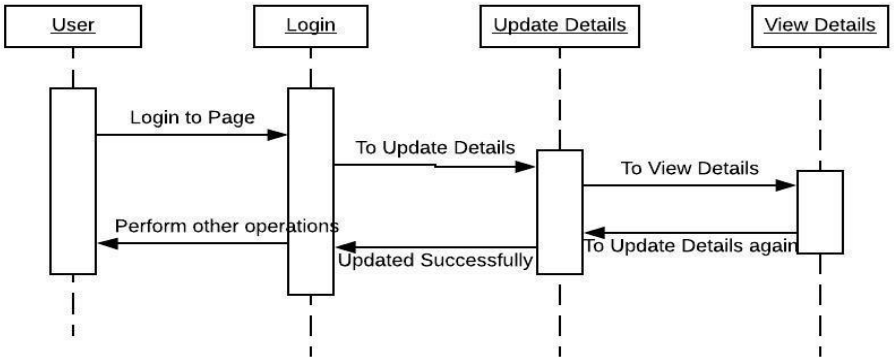
Manage Student



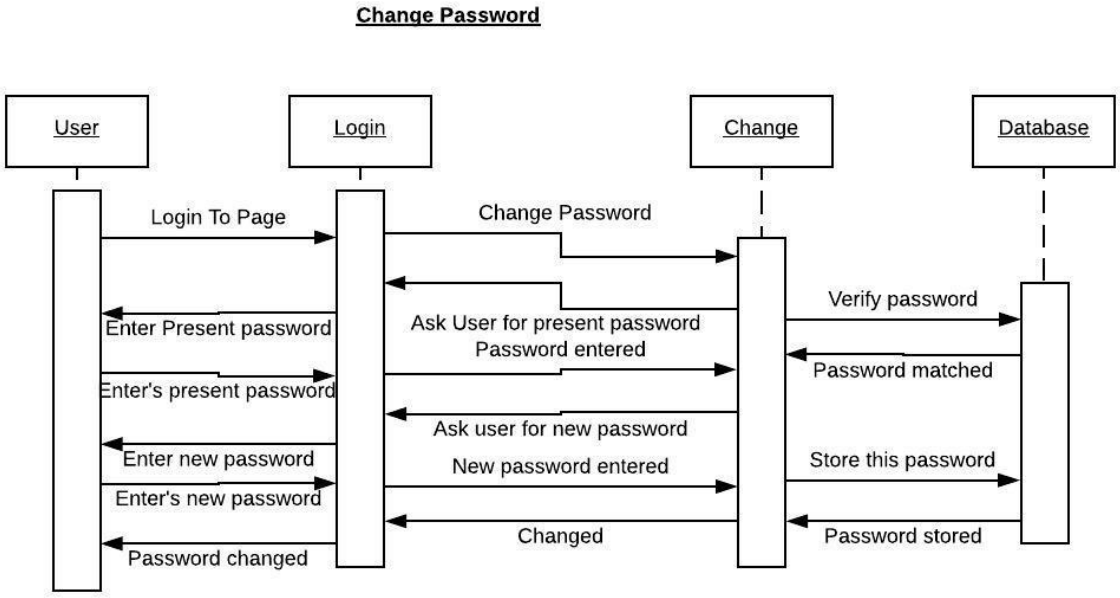
2.Sequence diagram for login & update details are shown below:



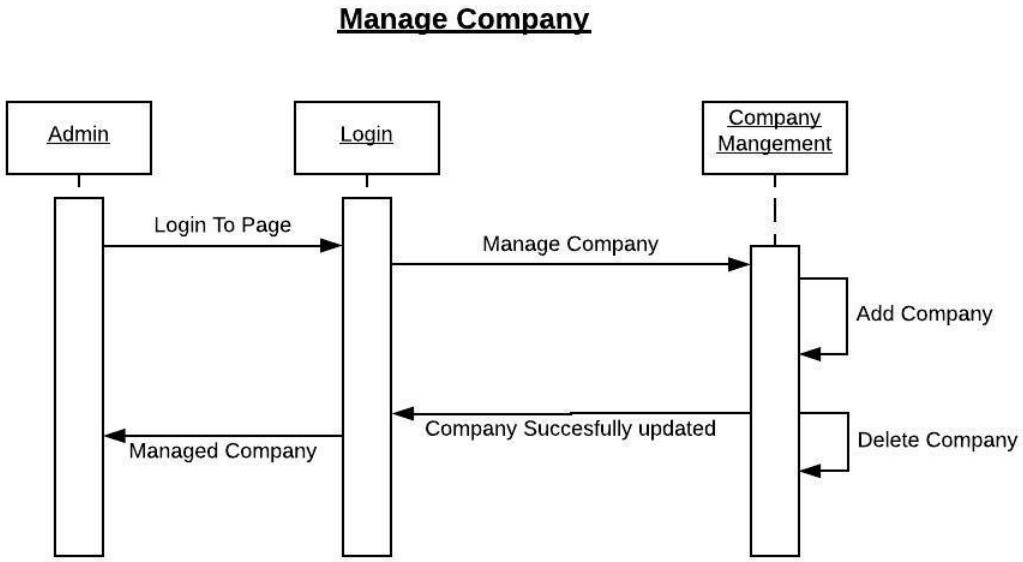
Update Details



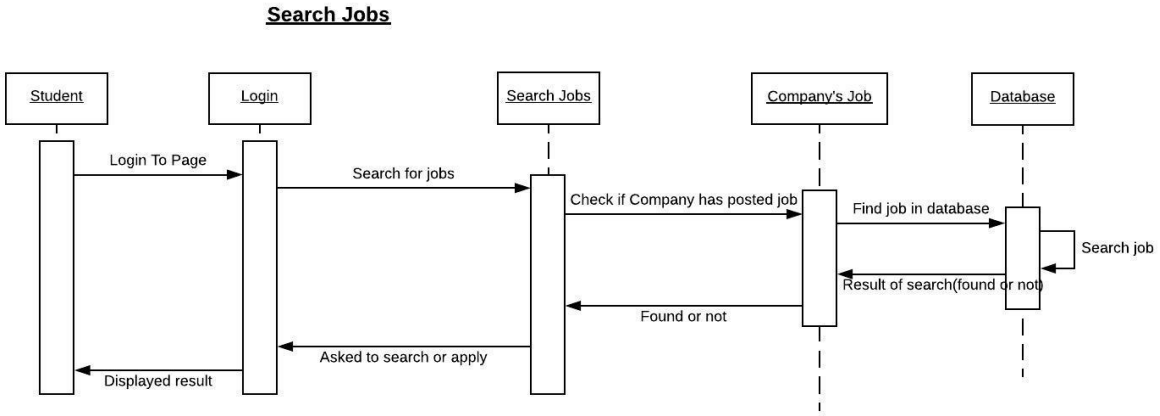
3.Sequence diagram for change password is shown below:



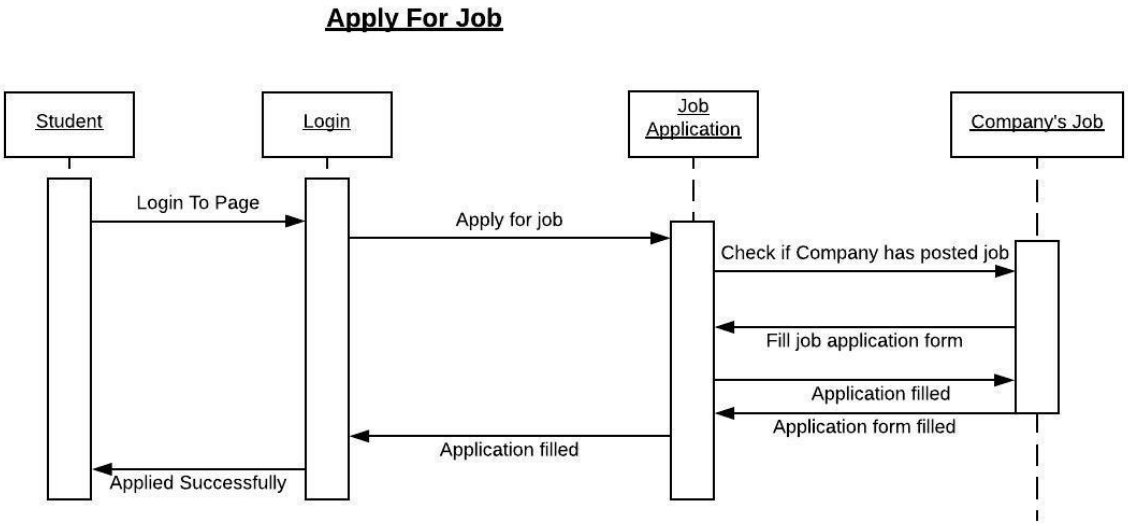
4.Sequence diagram for manage company is shown below:



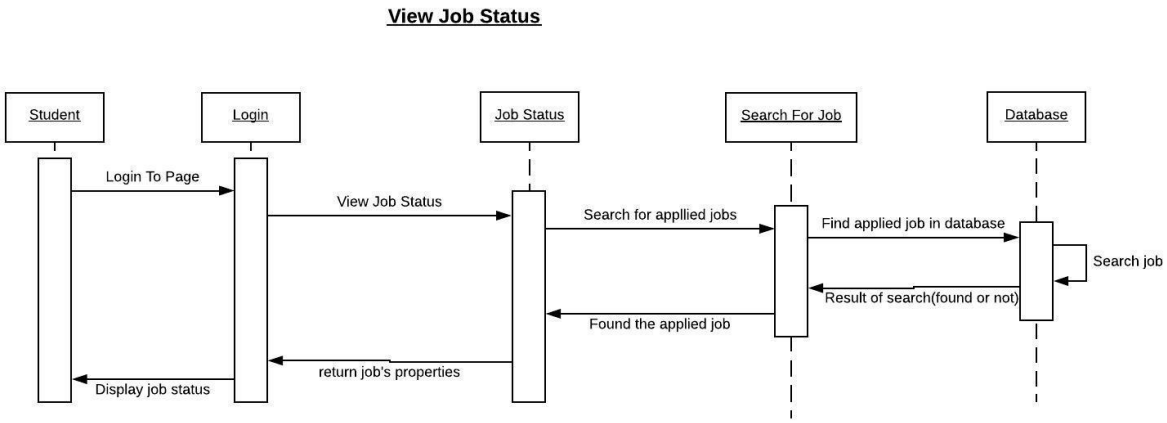
5.Sequence diagram for search jobs is shown below:



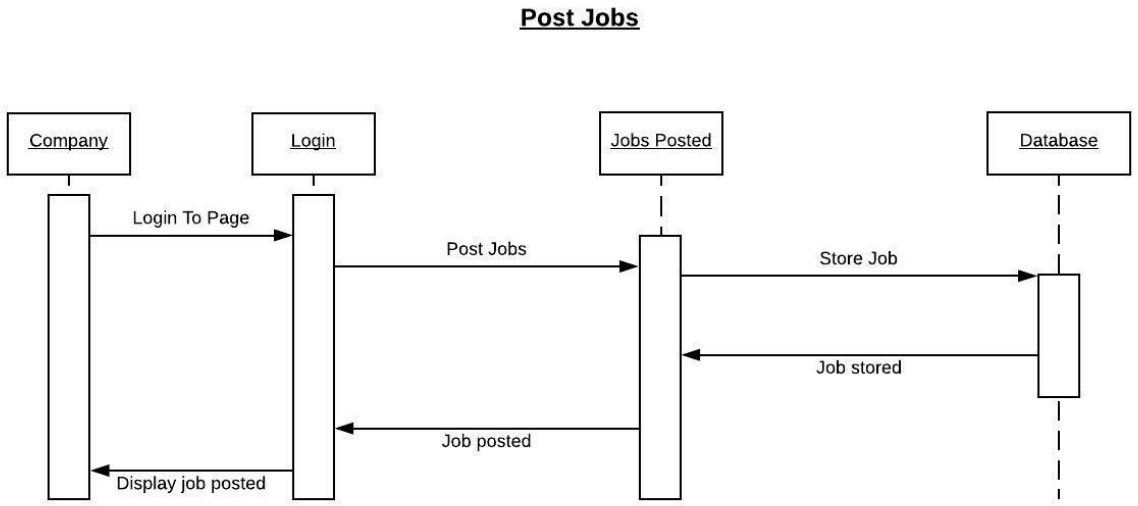
6.Sequence diagram of apply for job is shown below:



7.Sequence diagram for view job status is shown below:

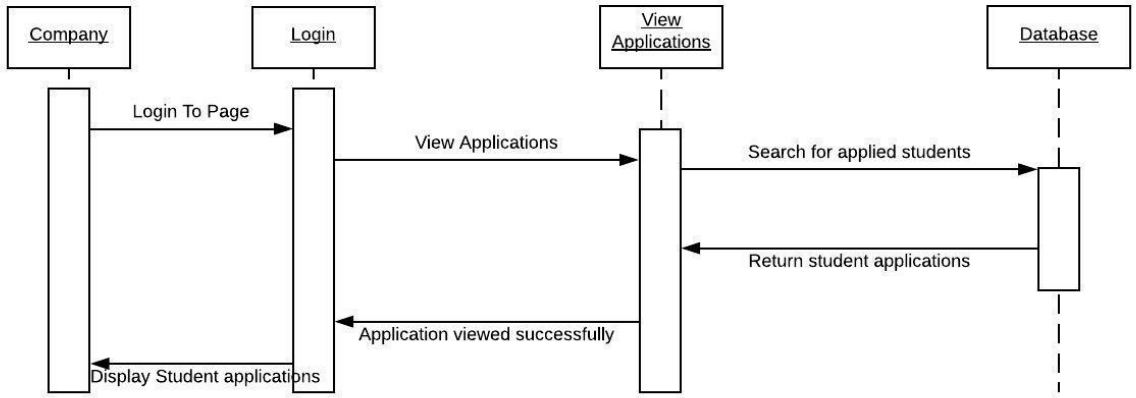


8.Sequence diagram for post jobs is shown below:



9.Sequence diagram for view applications is shown below:

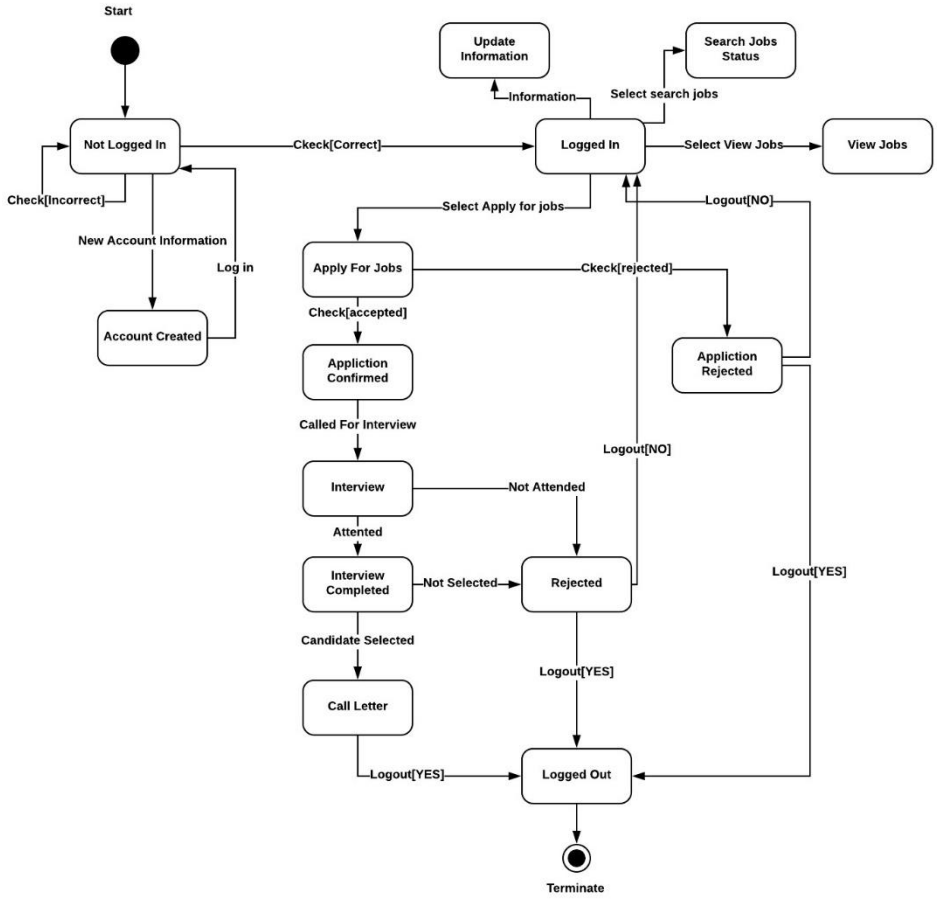
View Applications



State Machine Diagram For Campus Recruitment System

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.

State Machine diagram of campus recruitment system is shown below:

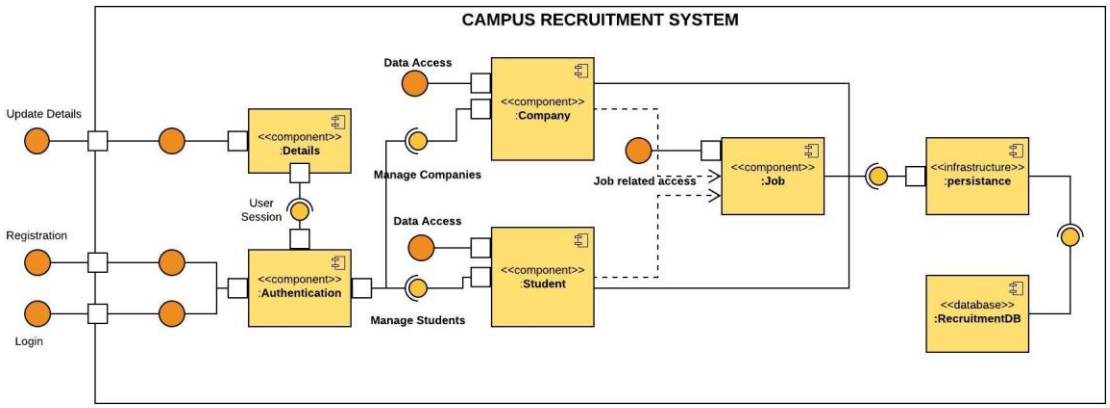


Component Diagram For Campus Recruitment System

In the Unified Modeling Language, a **component diagram** depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

A component is something required to execute a stereotype function. Examples of stereotypes in components include executables, documents, database tables, files, and library files.

Component diagram of campus recruitment system is shown below:



Deployment Diagram For Campus Recruitment System

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have subnodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

Deployment diagram of campus recruitment system is shown below:

Deployment Diagram