**CHAPTER 4**

**SYSTEM DESIGN**

Detailed design starts after the system design phase is completed and the system design has been certified through the review. The goal of this phase is to develop the internal logic of each of the modules identified during system design.

In the system design, the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for the modules. In other words in system design attention is on what components are needed, while in detailed design how the components can be implemented in the software is the issue.

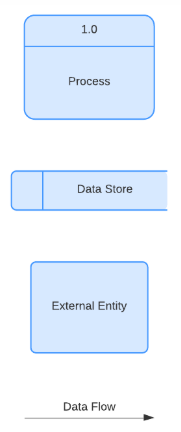
The design activity is often divided into two separate phase system design and detailed design. System design is also called top-level design. At the first level focus is on deciding which modules are needed for the system, the specifications of these modules and how the modules should be interconnected. This is called system design or top level design. In the second level the internal design of the modules or how the specifications of the module can be satisfied is decided. This design level is often called detailed design or logic design.

**4.1 Data Flow Diagram**

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expands it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

* Logical information flow of the system
* Determination of physical system construction requirements
* Simplicity of notation
* Establishment of manual and automated systems requirements

**Basic Notation:**



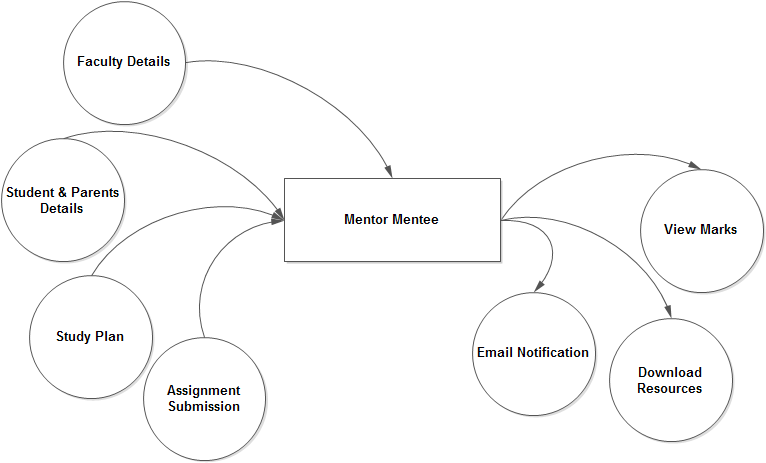
**Process:**any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as “Submit payment.”

**Data store:** files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as “Orders.”

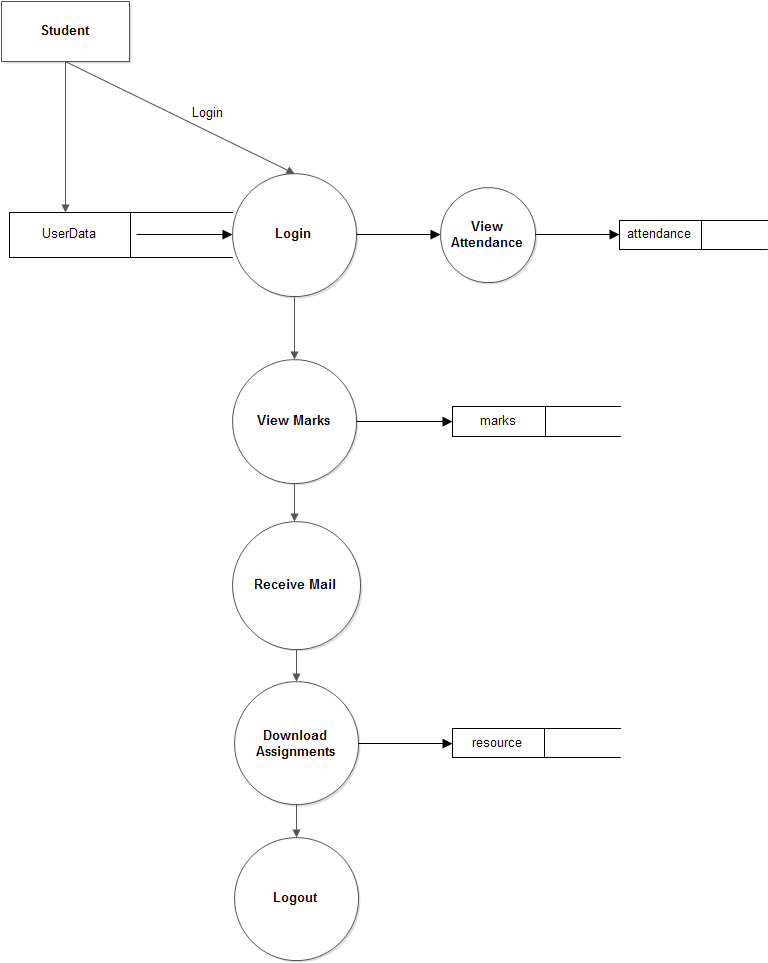
**External entity:** an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram

**Data flow:** the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.

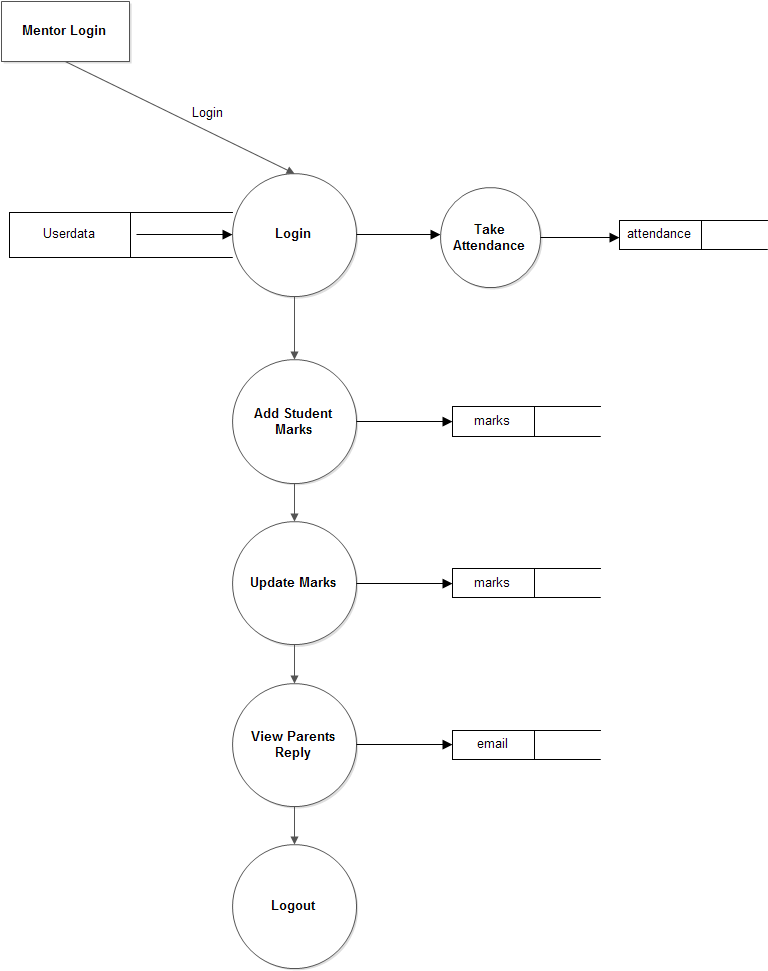
**Data Flow Diagram**



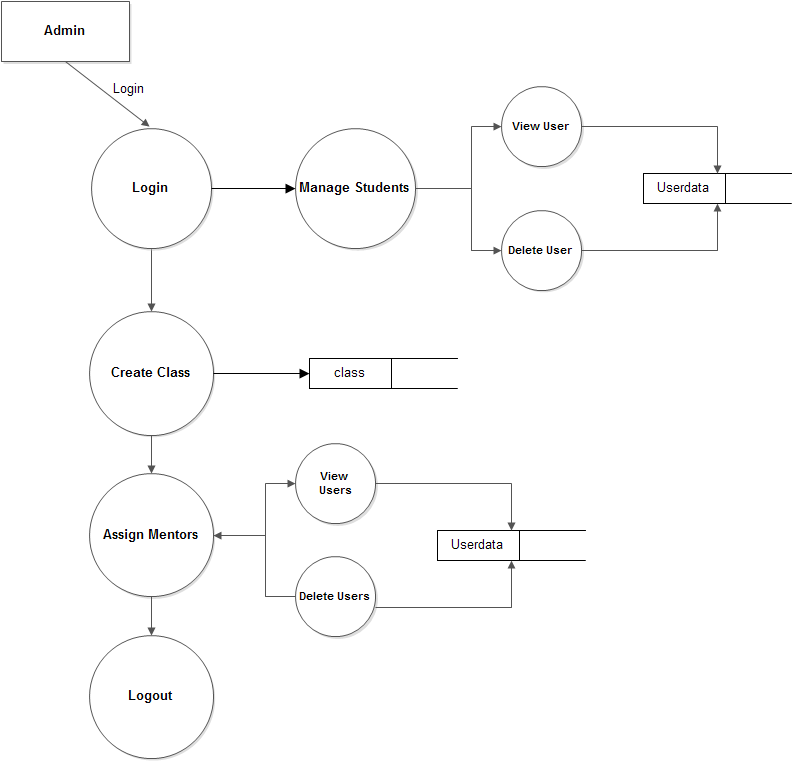
**Level 0 dfd**



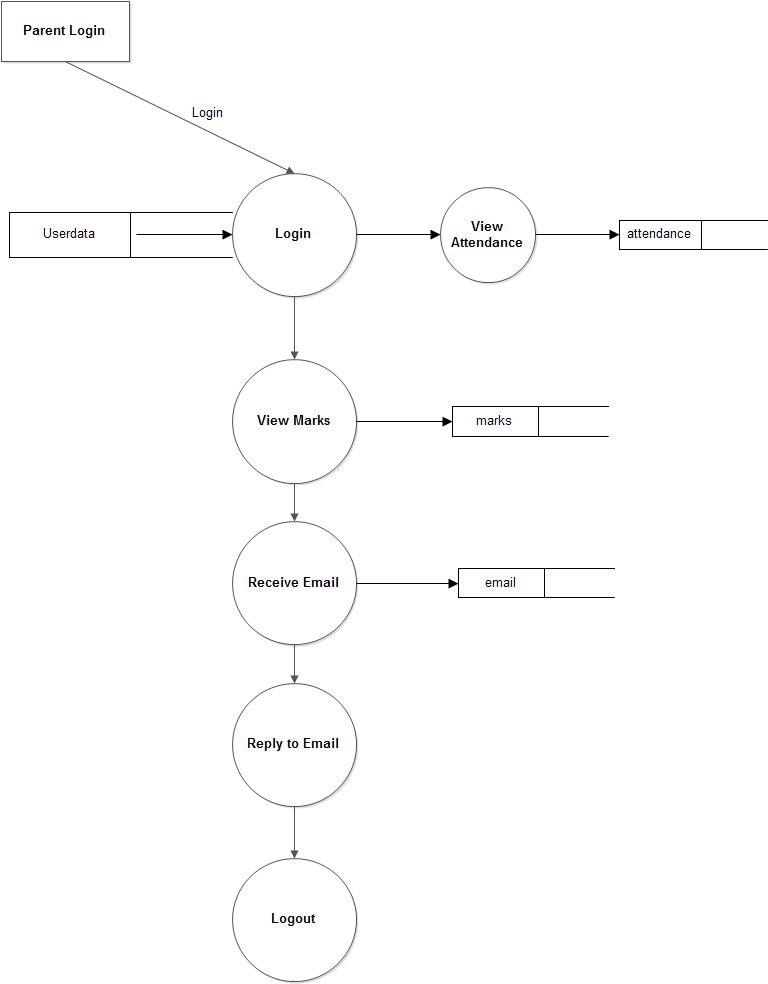
**Student level 1 dfd**



**Faculty level 1 dfd**



**Admin level 1 dfd**



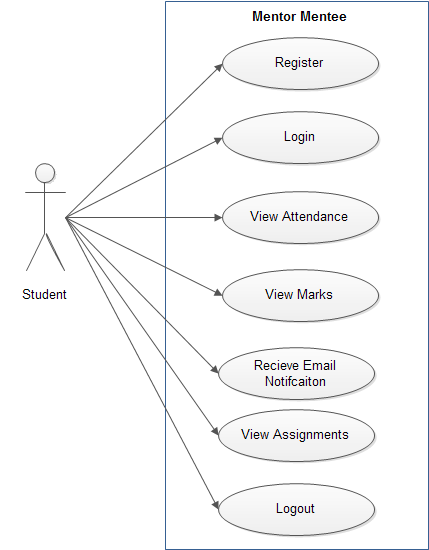
**Parents level 1 dfd**

**4.2 Use case diagram**

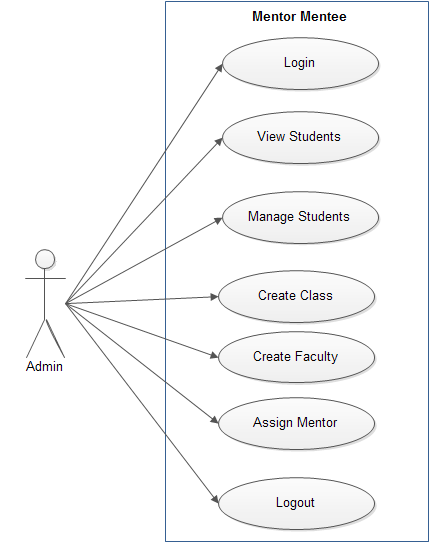
Use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication associations between the actor and the use case. The use case diagram describes how a system interacts with outside actors; each use case represents a piece of functionality that a system provides to its users. A use case is known as an ellipse containing the name of the use case and an actor is shown as a stick figure with the name of the actor below the figure.

The use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into actors and use case. Actors represent roles that are played by user of the system. Those users can be humans, other computers, pieces of hardware, or even other software systems.

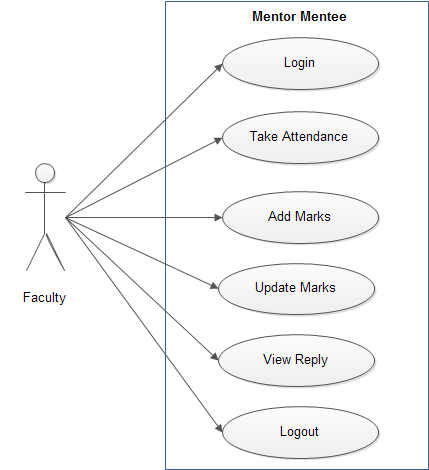
**Student Use case diagram**



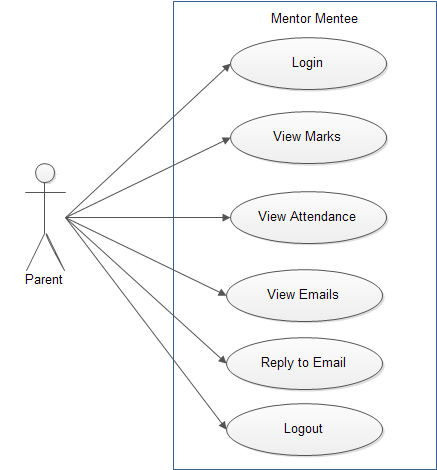
**Admin Use case diagram**



**Faculty Use case diagram**



**Parent Use case diagram**



**4.3 Sequence Diagram**

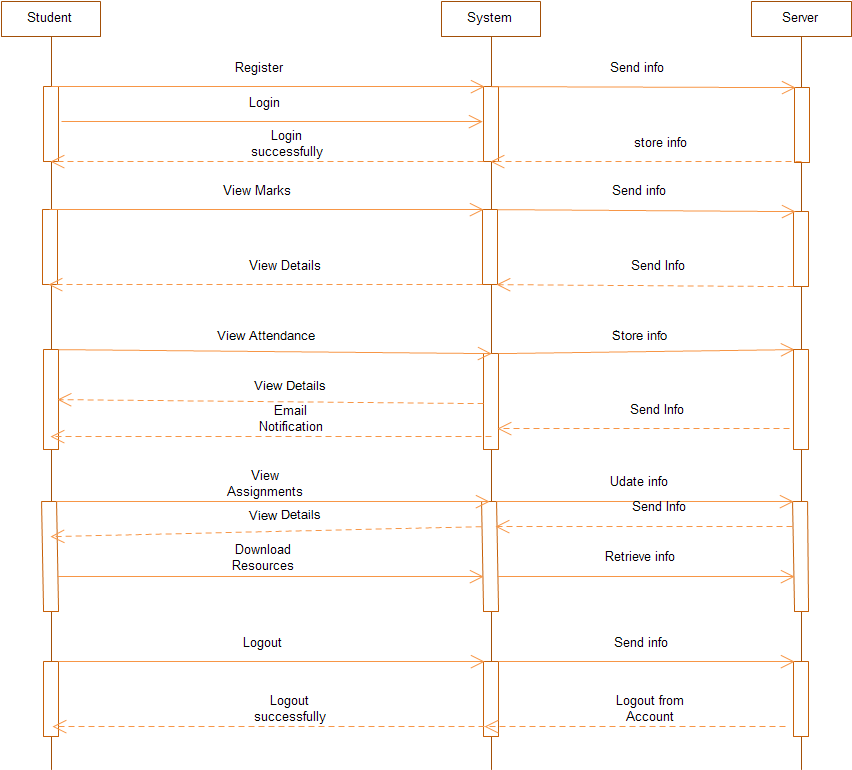
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are sometimes called **event diagrams**, **event scenarios.**

UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram Sequence Diagrams are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior.

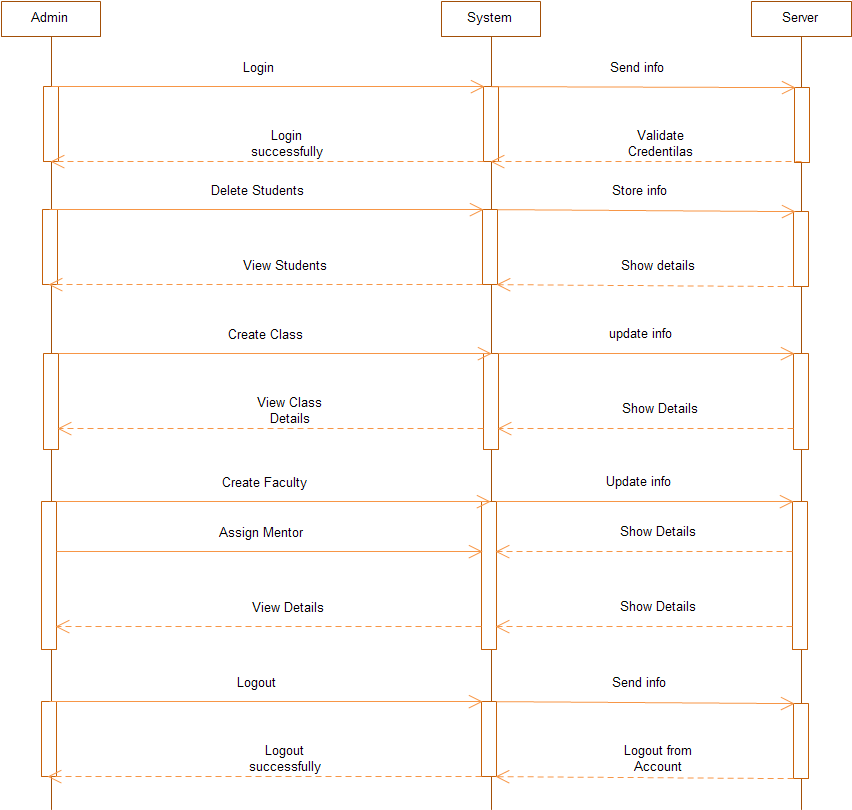
## Purpose

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. One of the primary uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement.

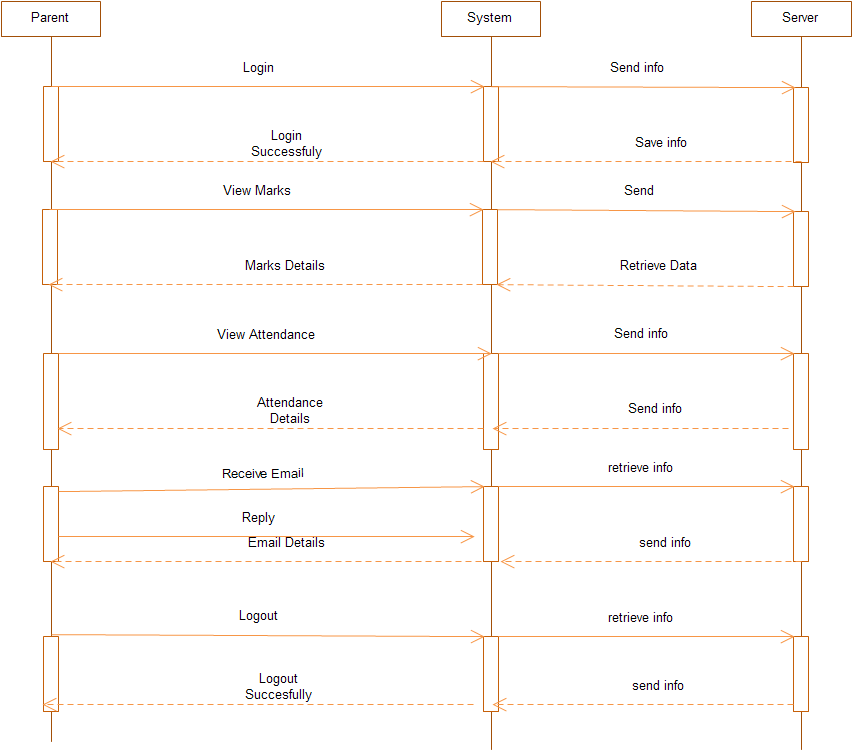
**Student Sequence Diagram**



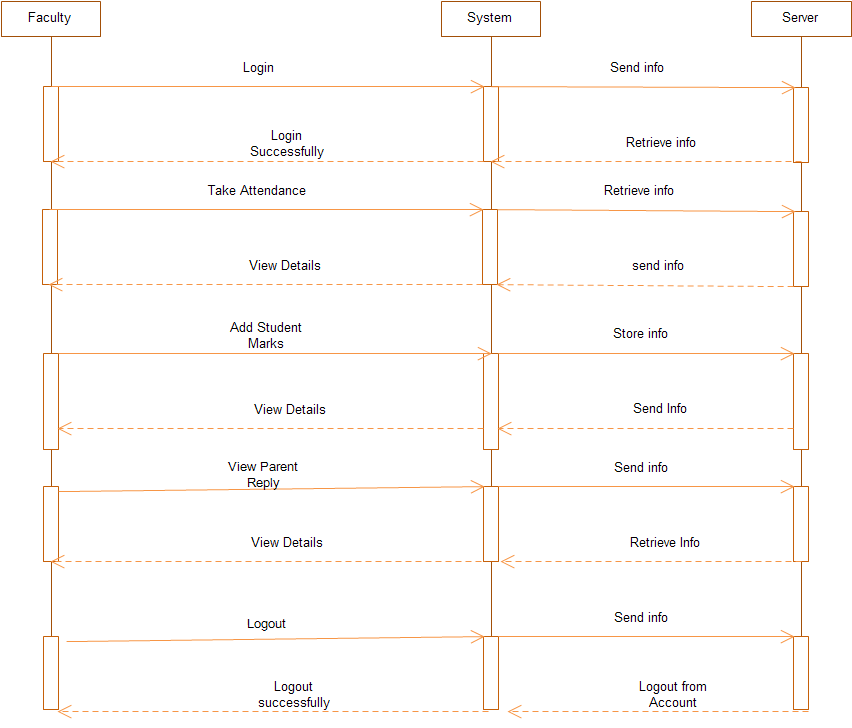
**Admin Sequence Diagram**



**Parent Sequence Diagram**



**Faculty Sequence Diagram**



**5.4 Activity Diagrams**

Activity diagrams represent the business and operational workflows of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state. It is a simple and intuitive illustration of what happens in a workflow, what activities can be done in parallel, and whether there are alternative paths through the workflow.

**Basic Notations**

Initial Activity

This shows the starting point or first activity of the flow. It is denoted by a solid circle.

Final Activity

The end of the Activity diagram is shown by a bull's eye symbol, also called as a final activity.

Activity

Represented by a rectangle with rounded (almost oval) edges

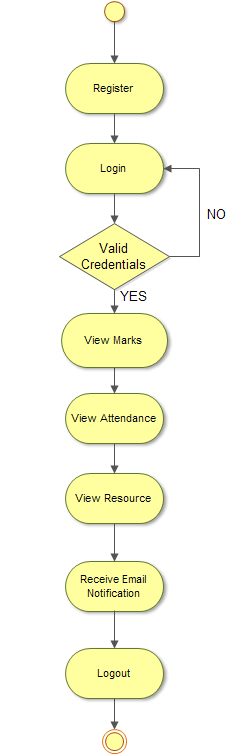
Decisions

A logic where a decision is to be made is depicted by a diamond.

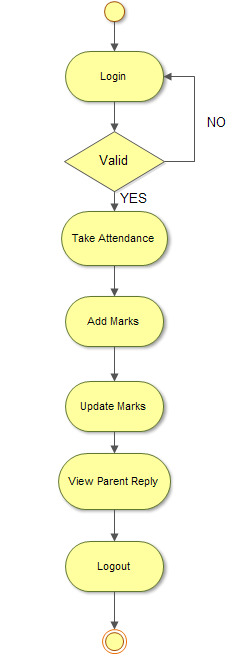
**Workflow**

Workflow is depicted with an arrow. It shows the direction of the workflow in the activity diagram.

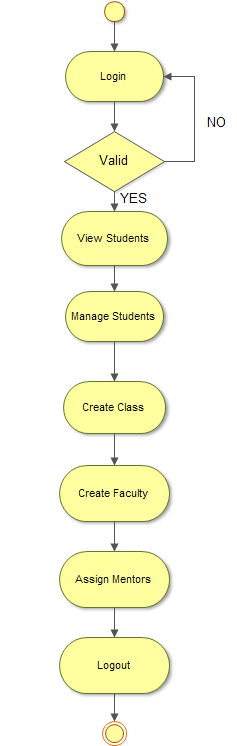
**Student Activity Diagram**



**Faculty Activity Diagram**



**Admin Activity Diagram**



**Parent Activity Diagram**

