

# **ST.XAVIER'S SCHOOL**

**Submitted By:**

**1.Hiral Thakkar**

**2.Jay Thakkar**

**3.Rushin Kadiya**

As a part of submission of Software Development Project Semester-5



**External Guide**

**Mr.Sanjay patel**

**Internal Guide**

**Dr.Bhavik h Pandya**

**Navgujarat College of Computer Applications**

**Ashram Road, Ahmedabad**

**December -2020**



# ST. XAVIER'S SCHOOL

(ENGLISH MEDIUM)

Matikhai Bus Stand, Gatrad Road, Ramol Ring Road, Ramol, Ahmedabad.  
M. 85110 64686

Ref. No. :

Date : 27/08/2020

To The principle,  
St.Xavier's school, Ramol,  
Opp- Matikhai Bus Stop,  
Ramol- Gatrad Road,  
Ramol, Ahmedabad 382449  
E-mail: xaviersramol2020@gmail.com

The Director  
Navgujarat collage of computer  
Applications  
Ashram Road  
Ahmedabad

## SUBJECT: Confirmation Letter for Project Allocation

Dear Sir,

We hereby confirm that the students from Navgujarat College of Computer Applications of Sem-5 are granted the permission to develop the project for the duration of one year. We will provide the necessary information regarding the project. The project details are as under:

**Project Title: St.Xavier's school**

**Technology: Android, php**


**Student Details:** 1) Hiral Thakkar  
2) Jay Thakkar  
3) Rushin Kadiya

REGARDS,

FR.LAWRENCE MARTIS PAUL

9712397131

xaviersramol2020@gmail.com

  
Principal  
ST. Xavier's School  
Opp. Matikhai Bus Stand,  
Gatrad Road, Ramol



# Navgujarat College of Computer Applications

Navgujarat Campus, Ashram Road Ahmedabad

www.navgujaratbca.com 079-27542859

## Certificate of Completion

This is to certify that Software Development Project “ST XAVIER’S SCHOOL”  
Is submitted to Gujarat University by:

- 1 Hiral Thakkar
- 2 Jay Thakkar
- 3 Rushin Kadiya

They have successfully completed the project for the duration of six months (Sem-V) towards the partial fulfilment for the award of “Bachelors in Computer Applications [B.C.A.] in the year 2020.

This is the original work carried out under our guidance and supervision. We further certify that the details are best to our knowledge and believe that the matter presented in this project is bonafied. We appreciate their enthusiasm and dedication towards work submitted.

Best wishes for their future endeavors.

---

(Internal Guide)

---

(Hon. Director)

---

(External Examiner)

# Acknowledgement

We express our heart gratitude to number of people who extended their full support and cooperation in developing this project First , We Would like take this opportunity to thank our College "**NAVGUJARAT COLLEGE OF COMPUTER APPLICATION**" for giving us this opportunity and a platform for discovering and developing our potential , This kind of experience that we have received while making this project report is so immense the narrating that in few world is difficult.

After putting in such a hard work we have realized that takes to work in school and do a project. Our school "**ST XEVIERS SCHOOL**" and colleagues have been great source of help without them we were unable to do this project. Therefore, our project as if small drop water of sea. We have learnt many think from being a part of concept family,

After that our heartiest thank is our internal guide as well as respected faculty "**mr abcd**" for entrusting upon us the responsible and acting as ray of light in darkness. We find our self-short of world to describe our feeling for the role she played of friend, a philosopher and guide, whenever we were in need.

Lastly, we are thankful to our parents without blessing, Love and Support. We are unable to traverse though this most significant stage of life and also, we would Like to take this opportunity to express our regards to all friends and faculties who have helped us directly or indirectly during the executing of the project. We are privileged and thankful to all in bringing our errors and shortcoming.

## Prepared By:

Hiral Thakkar

Jay Thakkar

Rushin Kadiya

# INDEX

1. PROJECT PROFILE.....	7
2. COMPANY PROFILE.....	8
3. ANALYSIS	
3.1 Existing System.....	9
3.2 Problems with existing system.....	9
3.3 proposed System.....	9
3.4 Advantages of proposed system.....	9
3.5.Tools&Technology.....	10
4. SYSTEM FLOW Diagram	
4.1 Symbols of System Flow Diagram .....	11
4.2 Admin System Flow Diagram.....	12
4.3Faculty System Flow Diagram .....	13
4.4 Student System Flow Daigram.....	14
5. UML DIAGRAM	
5.1 USE CASE DIAGRAM	
5.1.1 Symbol of Use Case Diagram .....	15
5.1.2 Admin Use Case Diagram .....	17
5.1.3 Faculty Use Case Diagram .....	18
5.1.4 Student Use Case Diagram.....	19
5.2 ACTIVITY DIAGRAM	
5.2.1 Symbol of Activity Diagram .....	20
5.2.2Admin/Student Login Activity .....	23
5.2.3Faculty Login Activity .....	24

5.2.4Attendance Activity .....	25
5.2.5Complain Activity .....	26
5.2.6 Exam Activity.....	27
5.2.7 Leave Activity.....	28
5.2.8 Result Activity.....	29
5.3 SEQUENCE DIAGRAM	
5.3.1 Symbol of Sequence Diagram .....	30
5.3.2 Registration Sequence .....	33
5.3.3 Login Sequence .....	34
5.3.4Attendance Sequence .....	35
5.3.5Event Sequence .....	36
5.3.6 Result Sequence.....	37
5.3.7 Leave Sequence.....	38
5.4 CLASS DIAGRAM	
5.4.1 Symbols of Class Daigrams.....	39
5.4.2 Class Daigram.....	40
6. ENTITY RELATIONSHIP DIAGRAM	
6.1 Symbols of E R Diagram.....	43
6.2 E R Daigram.....	47
7. DATA DICTIONARY .....	48
8. REFERENCES .....	61

**1.PROJECT PROFILE**

<b>Project Tital</b>	<b>St. Xavier's School</b>
<b>Numbers of Members</b>	3
<b>Name of Members(Roll no)</b>	Hiral Thakkar(111) Jay Thakkar(112) Rushinkadiya(33)
<b>Project Duration</b>	1 year
<b>Internal Guide</b>	Dr.Bhavik H Pandya
<b>External Guide</b>	Mr.Sanjay Patel

**2.COMPANY PROFILE**

<b>Company Name</b>	<b>St. Xavier's School</b>
<b>Company Address</b>	St. Xavier's school,ramol,opp-matikhai bus stop,ramol-gatrad road,ramol,Ahmedabad-382449.
<b>Contact Person</b>	Fr.LawrenceMartis Paul
<b>Contact No.</b>	9712397131



### 3.ANALYSIS

#### 3.1 EXISTING SYSTEM:-

- In the current system we need to keep a number of records related to the student and want to enter the details of the student and the mark manually in this system only the teacher or the school authority views the mark of the student and they want to enter the details of the student this is time consuming and has much cost.

#### 3.2 PROBLEM WITH EXISTING SYSTEM:-

- Current system is totally works manually.
- Present system require much man power.
- It is very hard to operate and maintain.
- The existing system is paper based.
- In the present system report generation become very difficult.
- Daily attendance though manual making require much man power and time consuming.

#### 3.3 PROPOSED SYSTEM (NEW SYSTEM):-

- School management system consists of tasks such as registering students,attendance record keeping controlling absentees producing report cards,producing official transcript, preparing timetable and producing different reports for teachers and parents.

#### 3.4 ADVANTAGES OF NEW SYSTEM:-

- Effective communication between teachers, parents and students.
- Frequent interaction with teachers.
- Automated student attendance.
- Access of attendance,timetable,marks,grades and examination schedule.
- Prior information about school event and holidays.
- Homework assignment to students.

### **3.5 TOOLS AND TECHNOLOGY:-**

#### **1) TECHNOLOGY:-**

##### **1.Front end**

1.Mobile App:-

❖ Android

2 Web Application:-

❖ Html,css,js,bootstrap

##### **2.Backend:-**

❖ My sql,php

#### **2) TOOLS:-**

❖ Android studio

❖ Wampp server

❖ Sublime text

❖ Microsoft word

❖ Microsoft PowerPoint

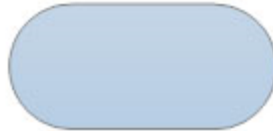
❖ Draw.io

## 4. SYSTEM FLOW DIAGRAM

System flow diagram is a basically a graphical and sequential representation of the major steps involved in a systematic process.

### 4.1 SYMBOLS OF SYSTEM FLOW DIAGRAM

#### ❖ Start/end point



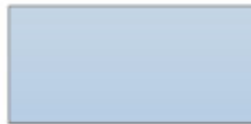
An oval represents a start or end point.

The terminator symbol marks the starting or ending point of the system. It usually contains the word “Start” or “End”.

#### ❖ Arrows

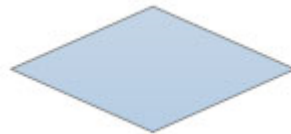
A line is a connector that shows relationships between the representative shapes.

#### ❖ Process



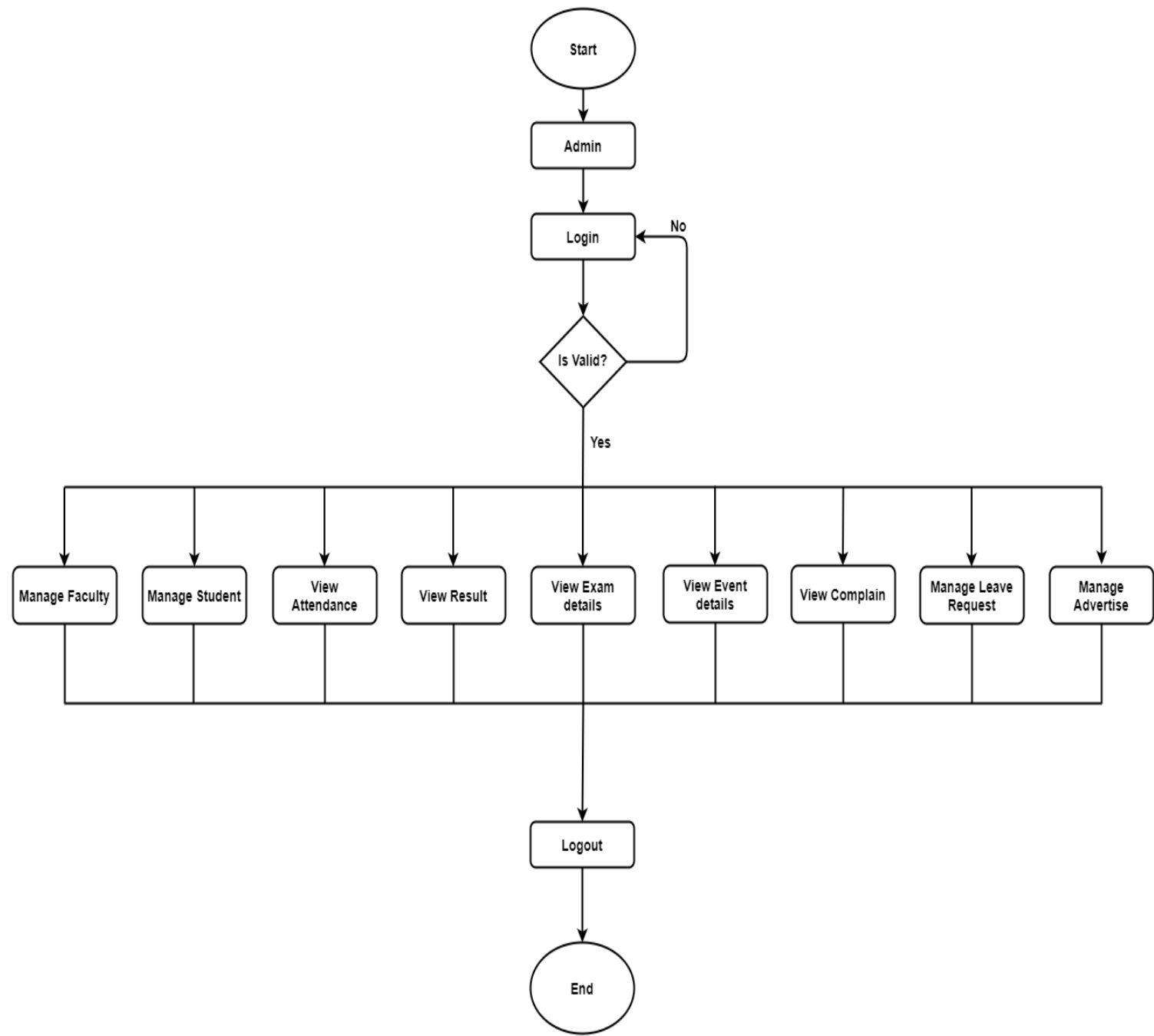
A rectangle represents a process.

#### ❖ Decision

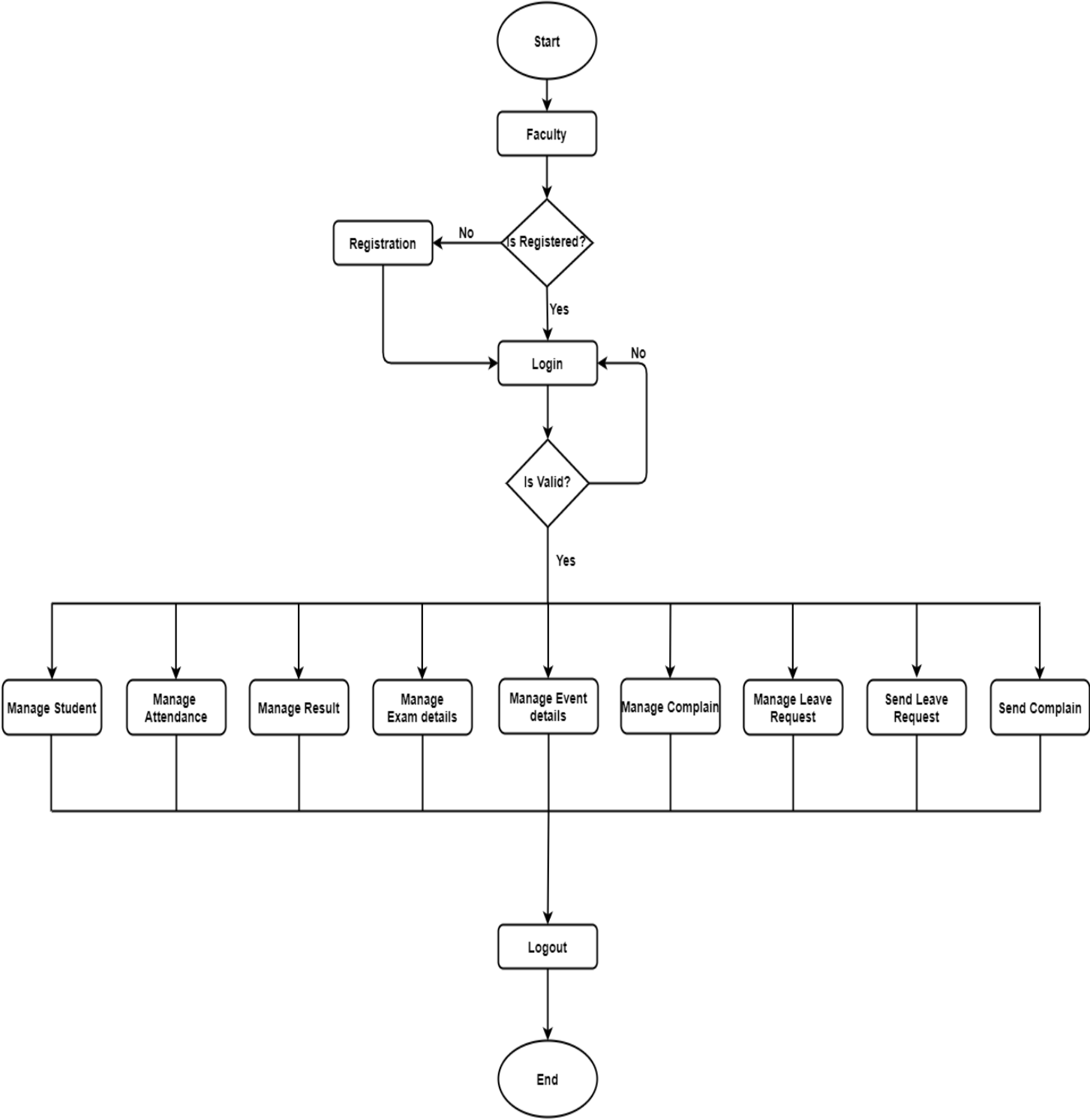


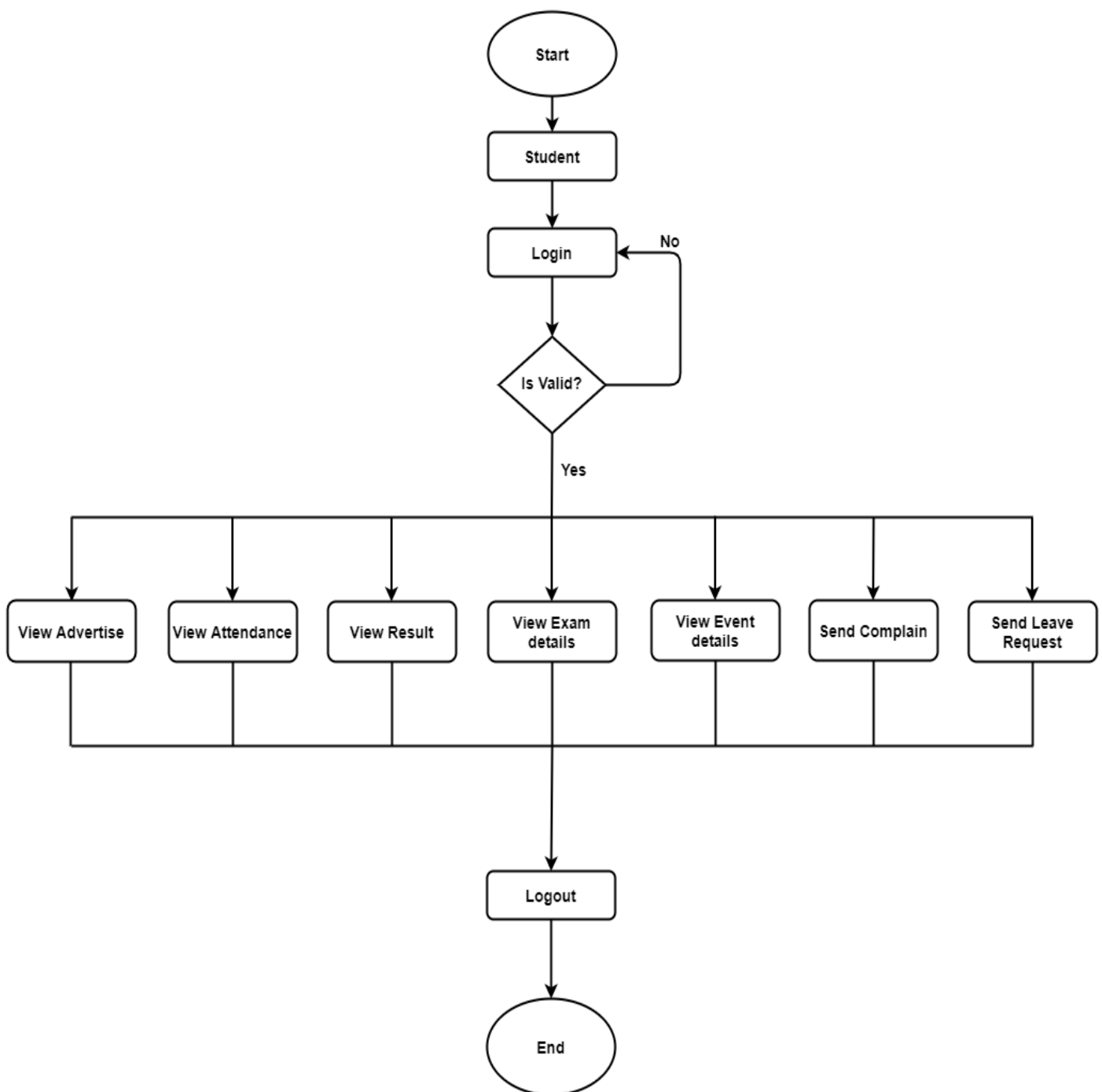
A diamond indicates a decision. It appoints where the outcome of a decision dictates the next step. There can be multiple outcomes, but often there are just two – yes and no or true – false, then branches to different parts of the systemflow diagram.

4.2 ADMIN SYSTEM FLOW DIAGRAM



4.3FACULTY SYSTEM FLOW DIAGRAM



**4.4 STUDENT SYSTEM FLOW DIAGRAM**

## 5. UML DAIGRAMS

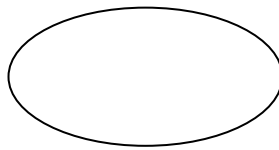
### 5.1 USE CASE DAIGRAM

- Use case is a set of scenarios that describing an interaction between a user and a system.
- Use case diagram displays the relationship among actors and use cases.
- The two main components of a use case diagram are use cases and actors.
- An actor represents user or another system that will interact with the system.
- Use case is an external view of the system that represents some actions the user might perform in order to complete a task.

#### 5.1.1 SYMBOLS OF USE CASE DIAGRAM

##### ❖ Use Case

Use case



Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

##### ❖ Actors



Actor

Actors are the users of a system. When one system is the actor of another system label the actor system with the actor stereotype.

##### ❖ System

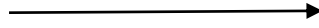


System

Draw your system's boundaries using a rectangle that contains use cases, Place actor outside the system's boundaries.

## ❖ Relationships

<<include>>



Relationship

<<extend>>

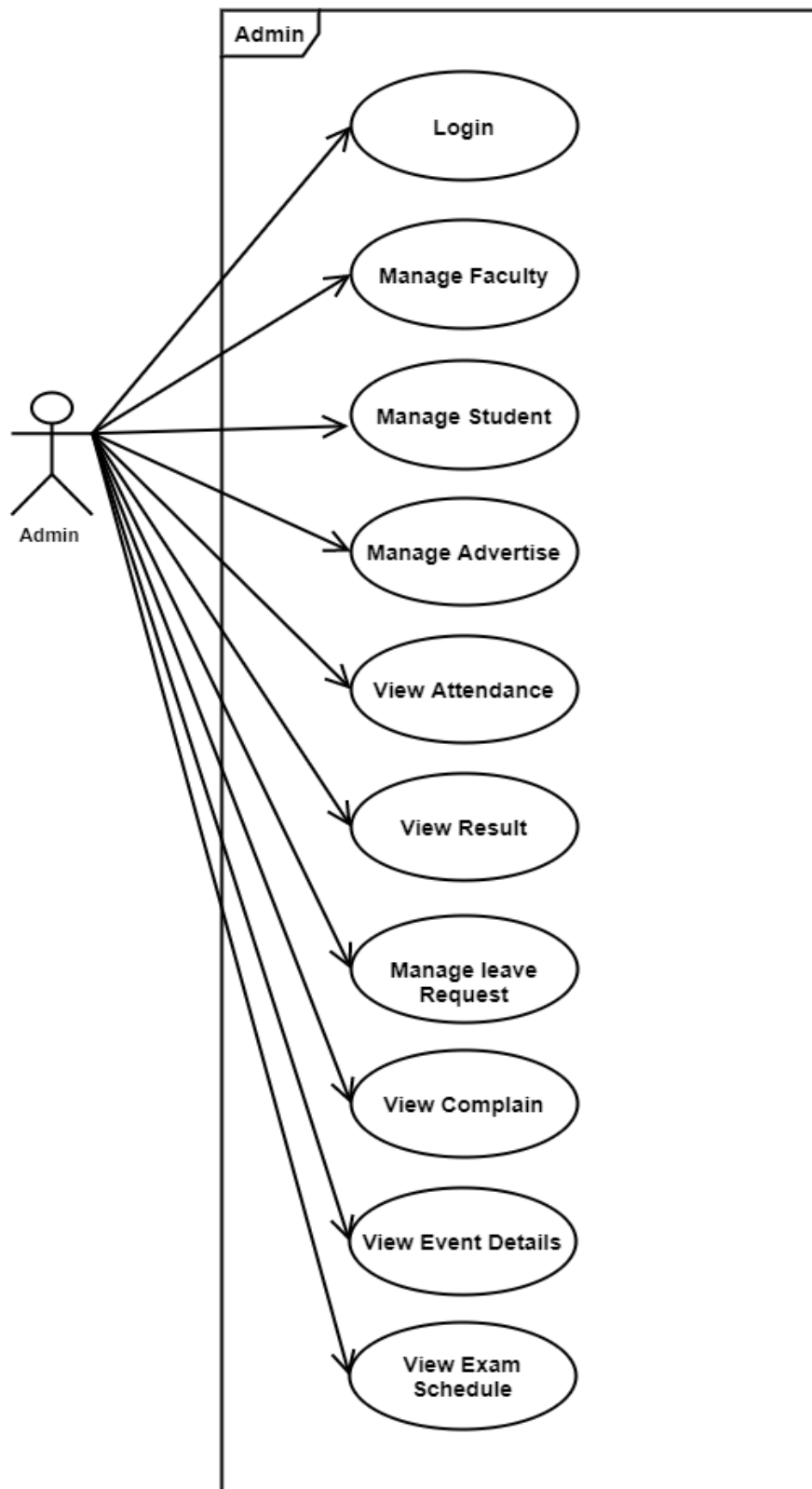


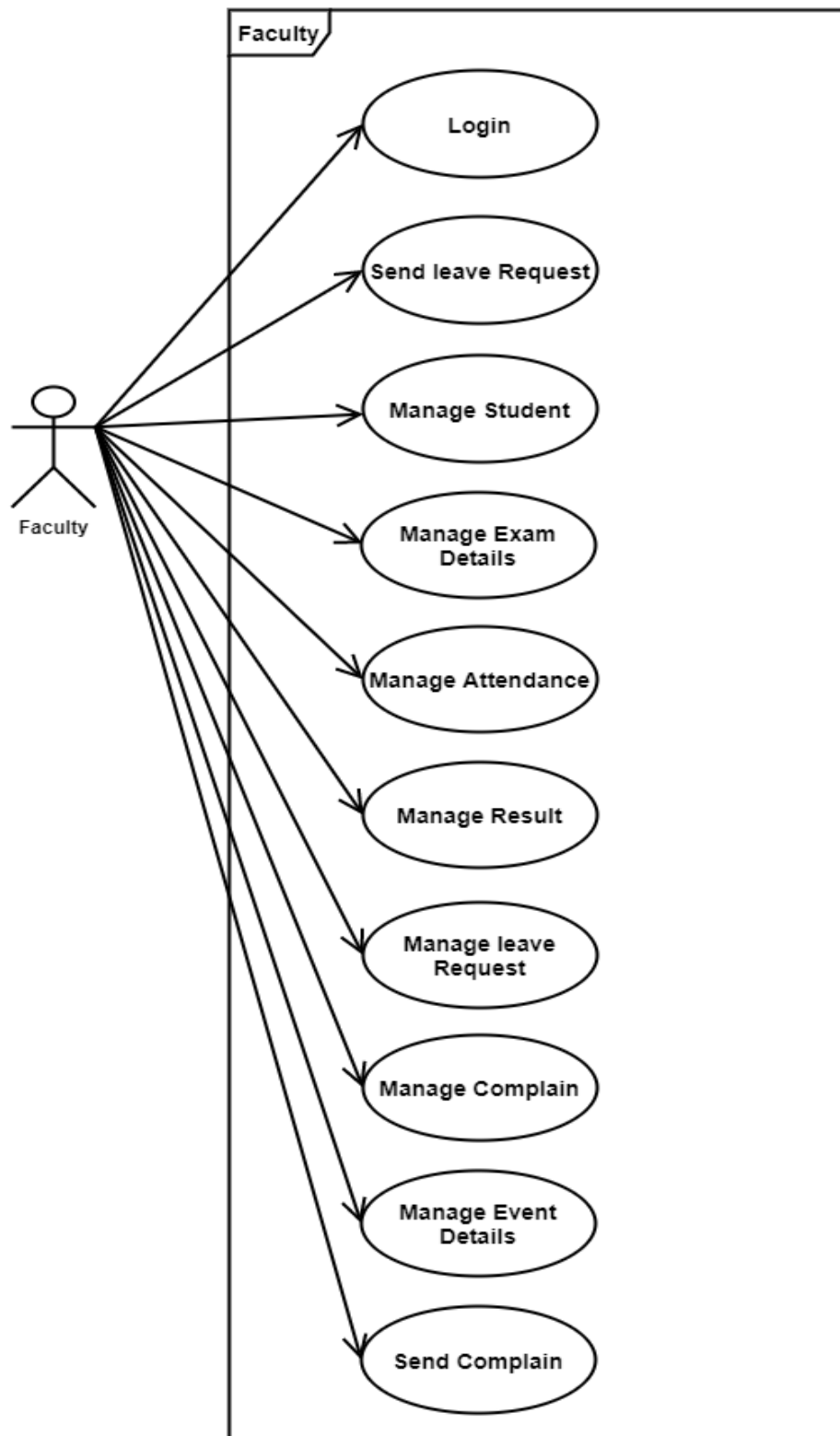
Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either “uses” or “extends.” A “uses” relationship indicates that one use case is needed by another in order to perform a task.

An “extends” relationship indicates alternative options under a certain use case.

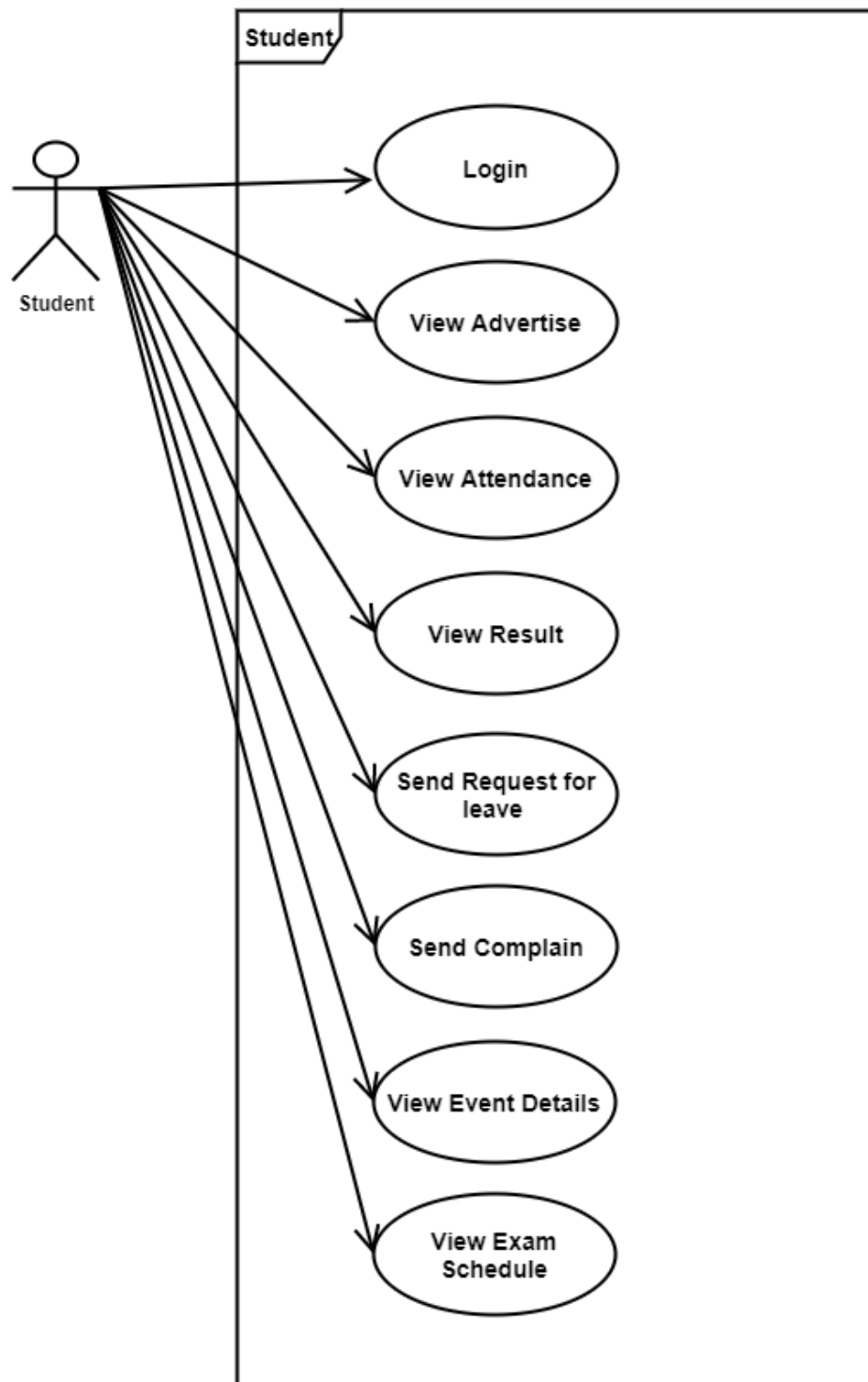


### 5.1.2 ADMIN USE-CASE DIAGRAM



**5.1.3 FACULTY USE-CASE DIAGRAM**

### 5.1.4 STUDENT USE-CASE DIAGRAM



## 5.2ACTIVITY DIAGRAMS

- Activity diagrams are graphical representation of step wise activities and actions with Support for choice, iteration and concurrency.
- Activity diagram can be used to describe the business and operational step – by – step Workflows of components in a system.
- Activity diagrams are typically used for business process modeling, for modeling the logic captured by a single usage scenario, or for modeling the detailed logic.
- Although UML activity diagrams could potentially model the internal logic of a complex operation it would be far better to simply rewrite the operation so that it is simple enough that you don't require an activity diagram.

### 5.2.1 SYMBOLS OF ACTIVITY DIAGRAM

#### ❖ Initial State or Start Point

A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swim lanes, make sure the start point is placed in the top left corner of the first column.



#### ❖ Activity or Action State

An action state represents the non-interruptible action of objects. You can draw an action state in Smart Draw using a rectangle with rounded corners.



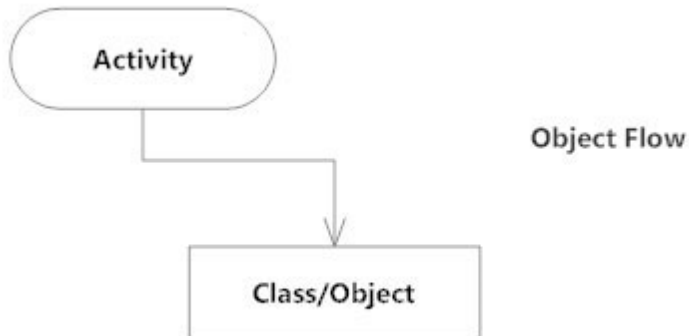
#### ❖ Action Flow

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.



### ❖ Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



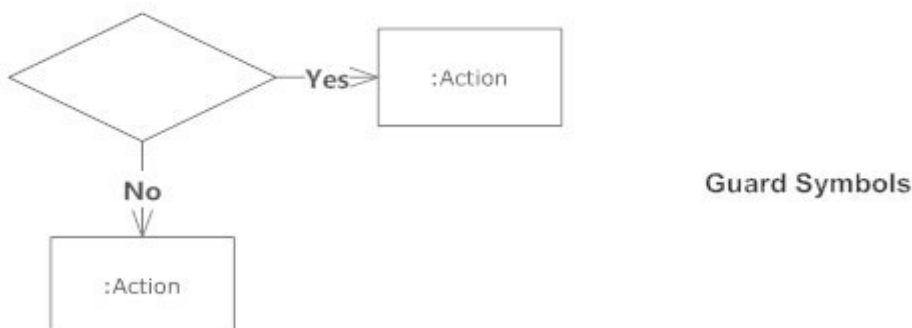
### ❖ Decisions and Branching

A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."



### ❖ Guards

In UML, guards are a statement written next to a decision diamond that must be true before moving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.



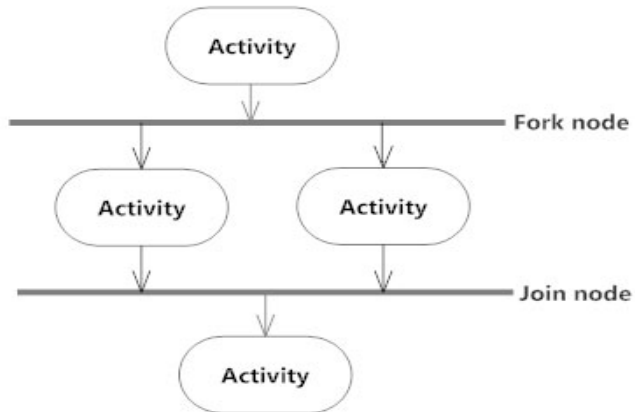
### ❖ Synchronization

A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram.

A join node joins multiple concurrent flows back into a single outgoing flow.

A fork and join mode used together are often referred to as synchronization.

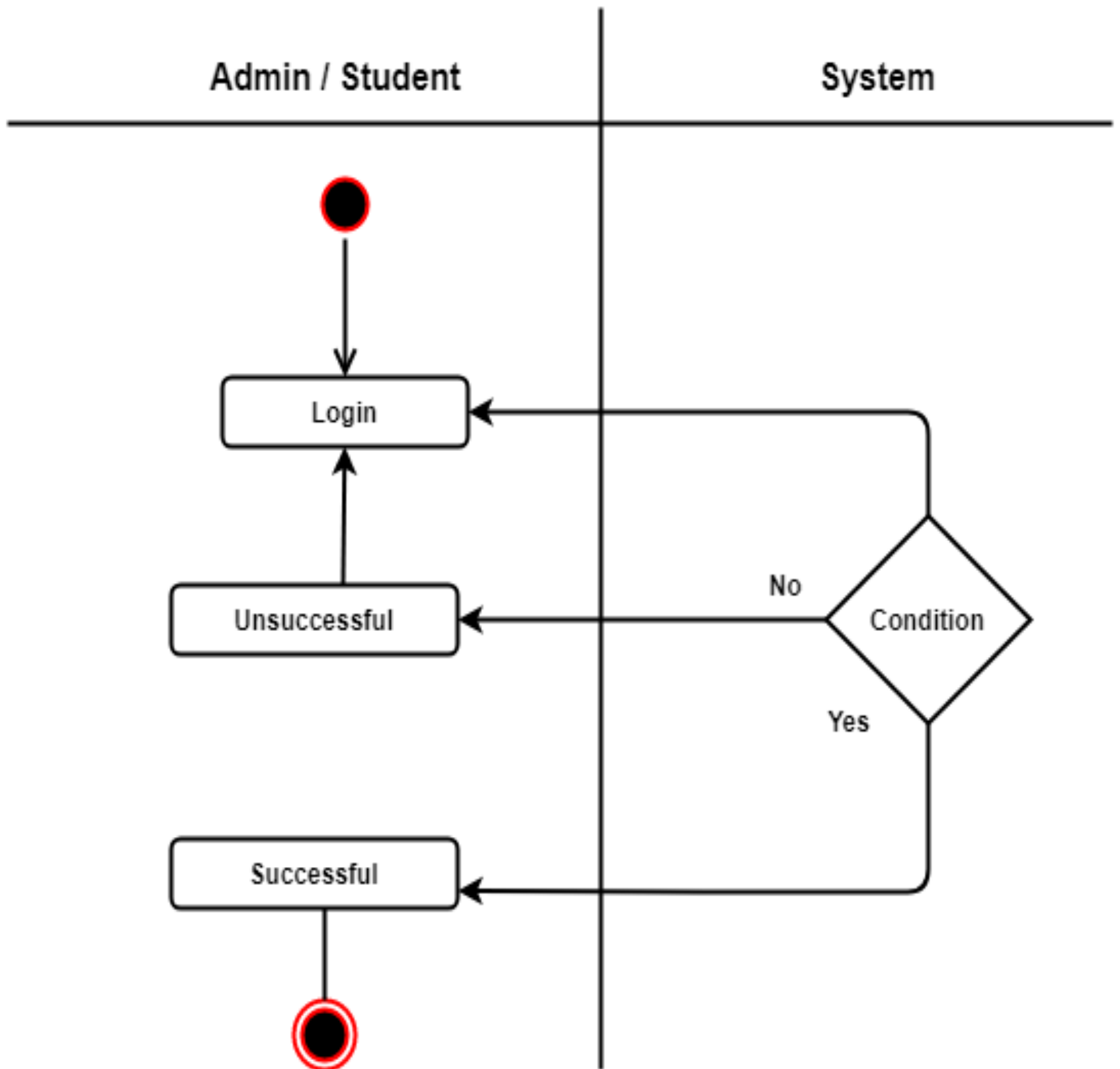
Synchronization

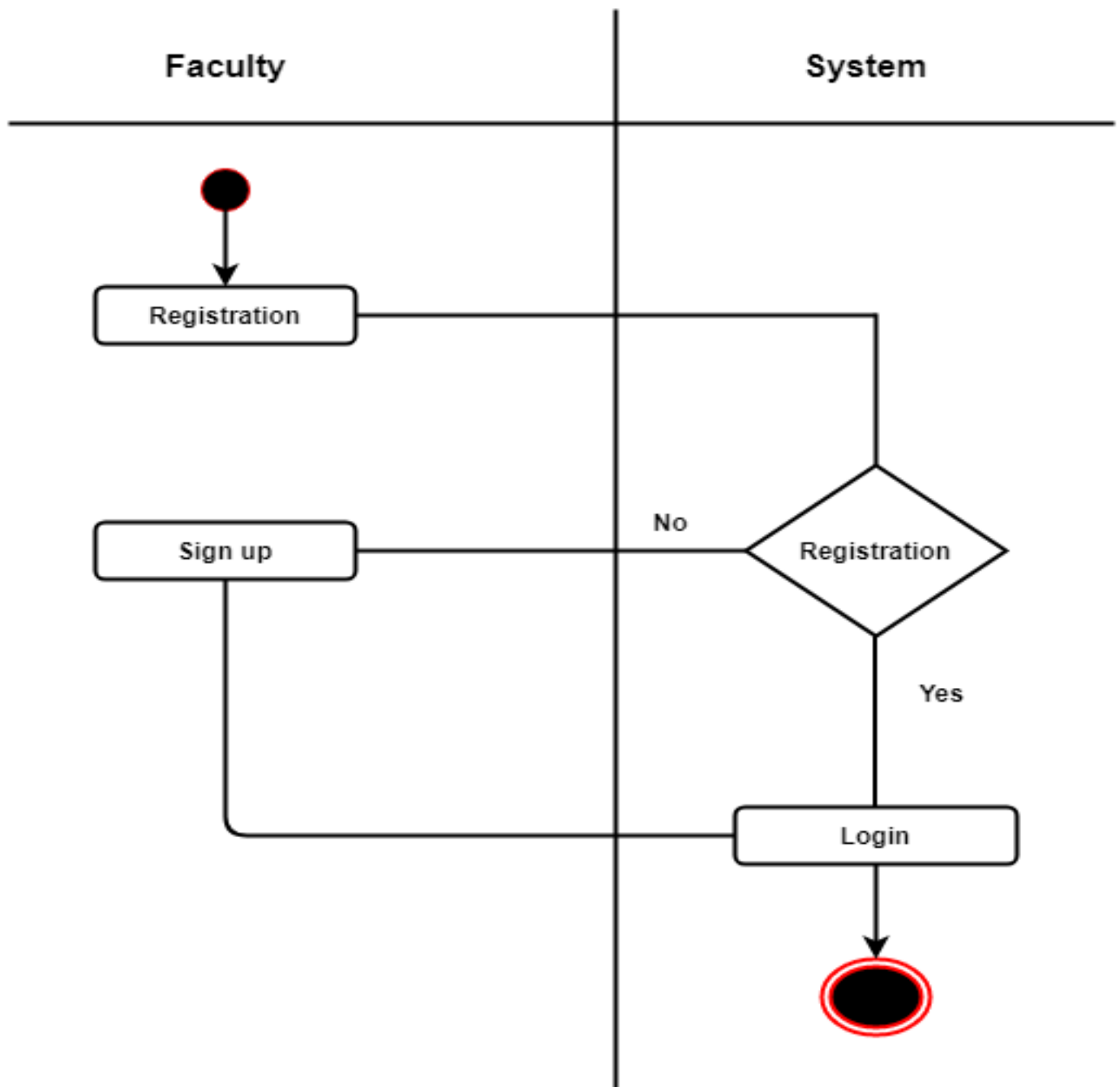


### ❖ Final State or End Point

An arrow pointing to a filled circle nested inside another circle represents the final action state.

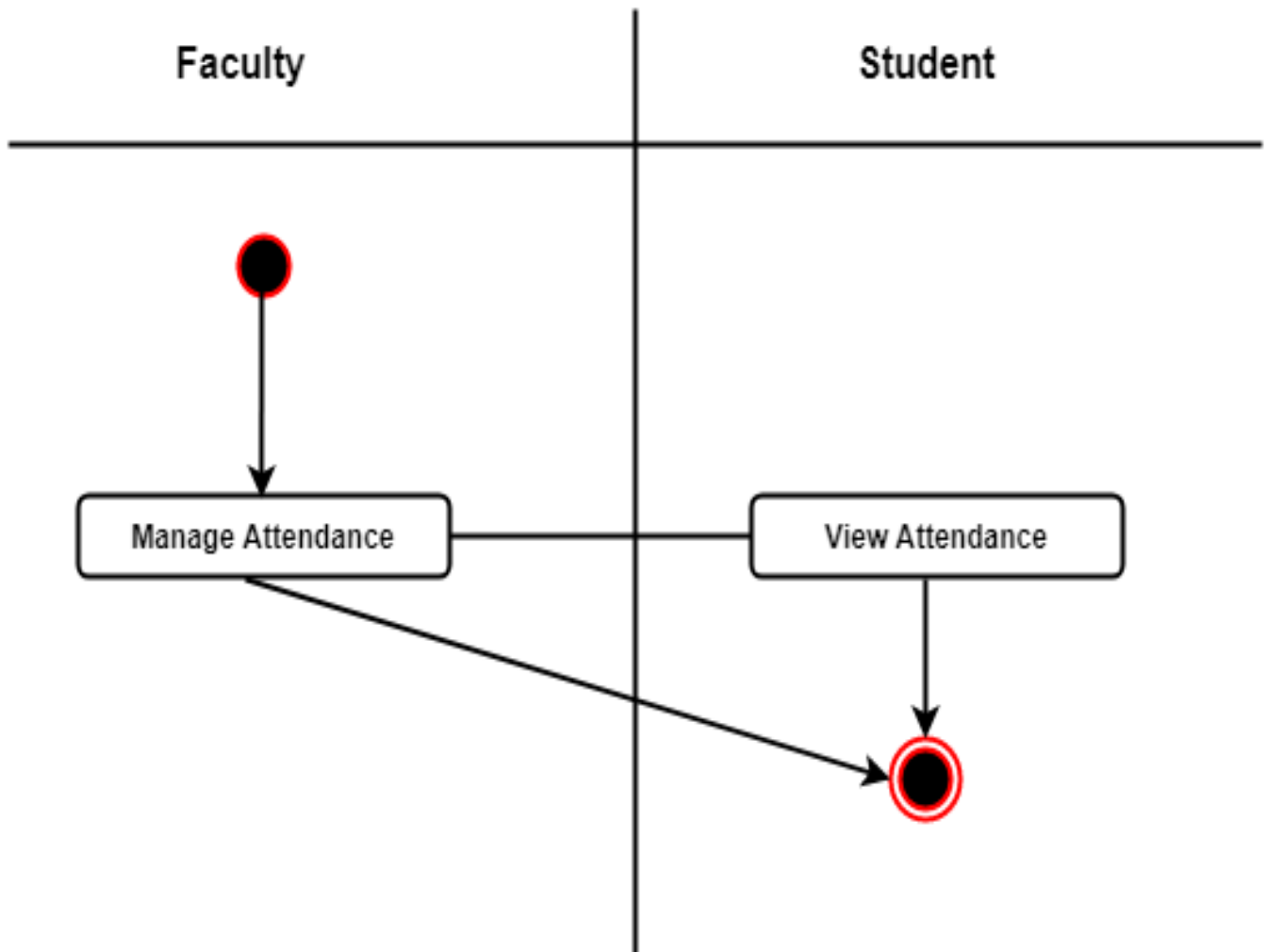


**5.2.2 ADMIN/STUDENT LOGIN ACTIVITY**

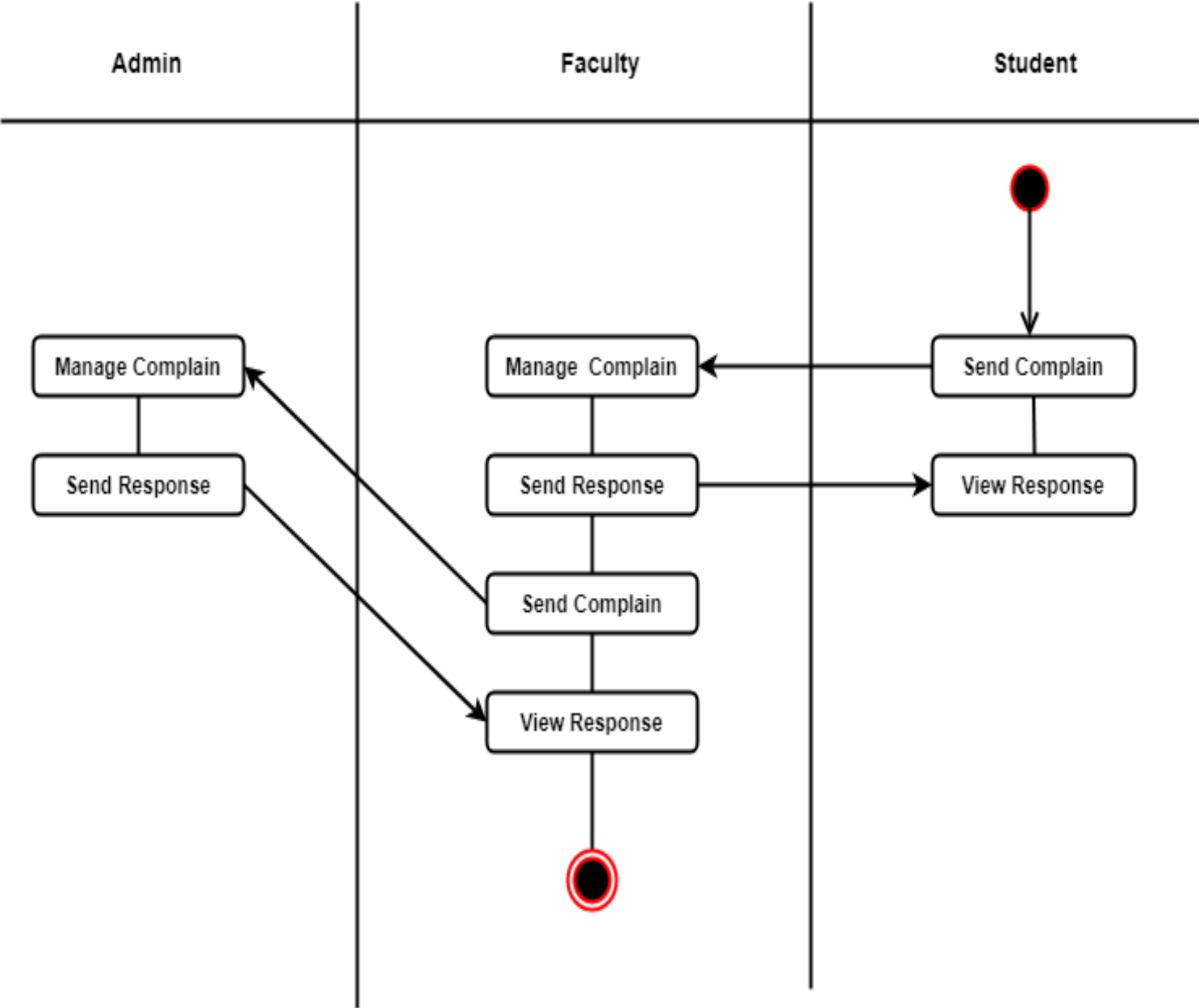
**5.2.3 FACULTY LOGIN ACTIVITY**

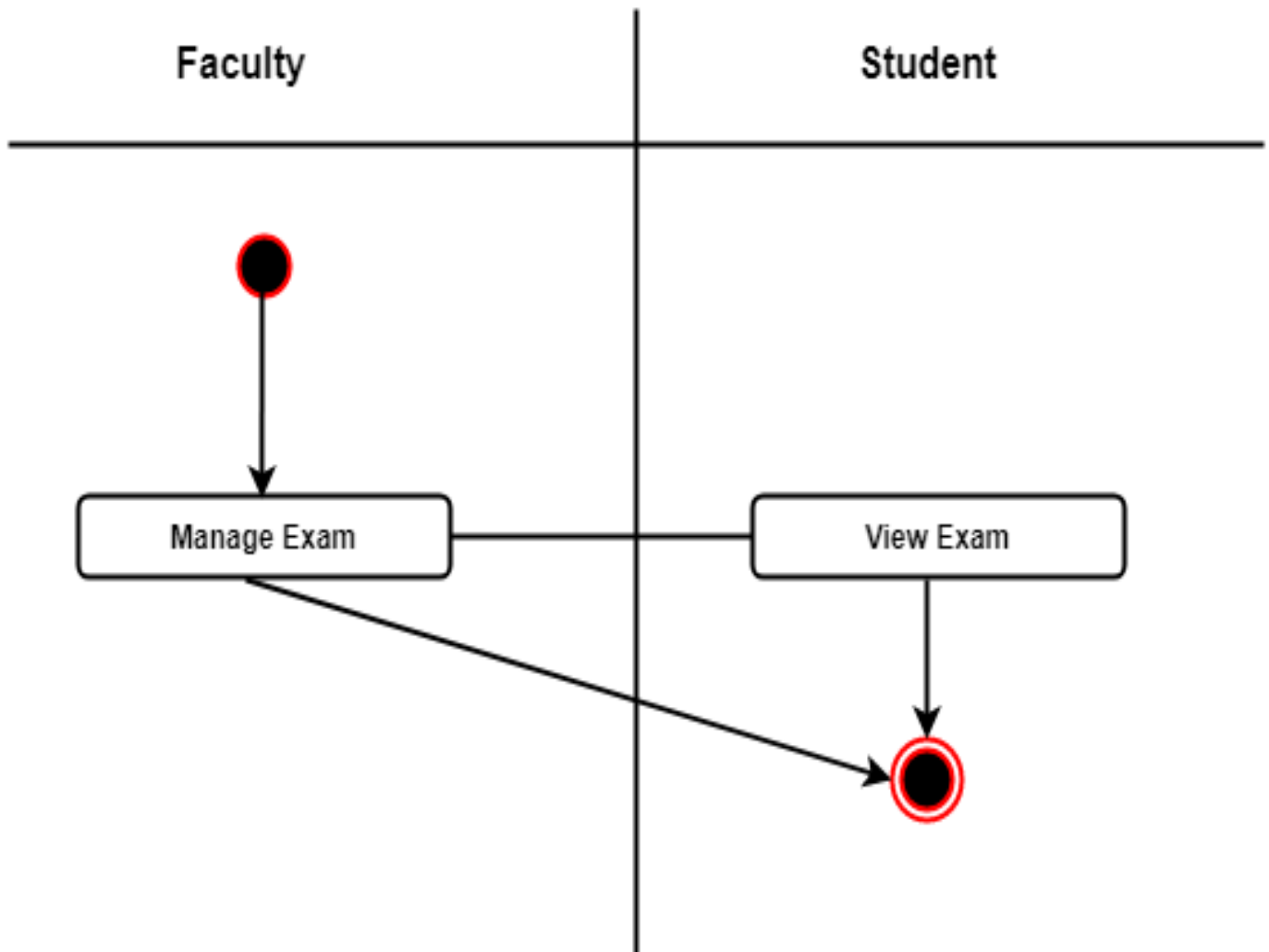


### 5.2.4 ATTENDANCE ACTIVITY

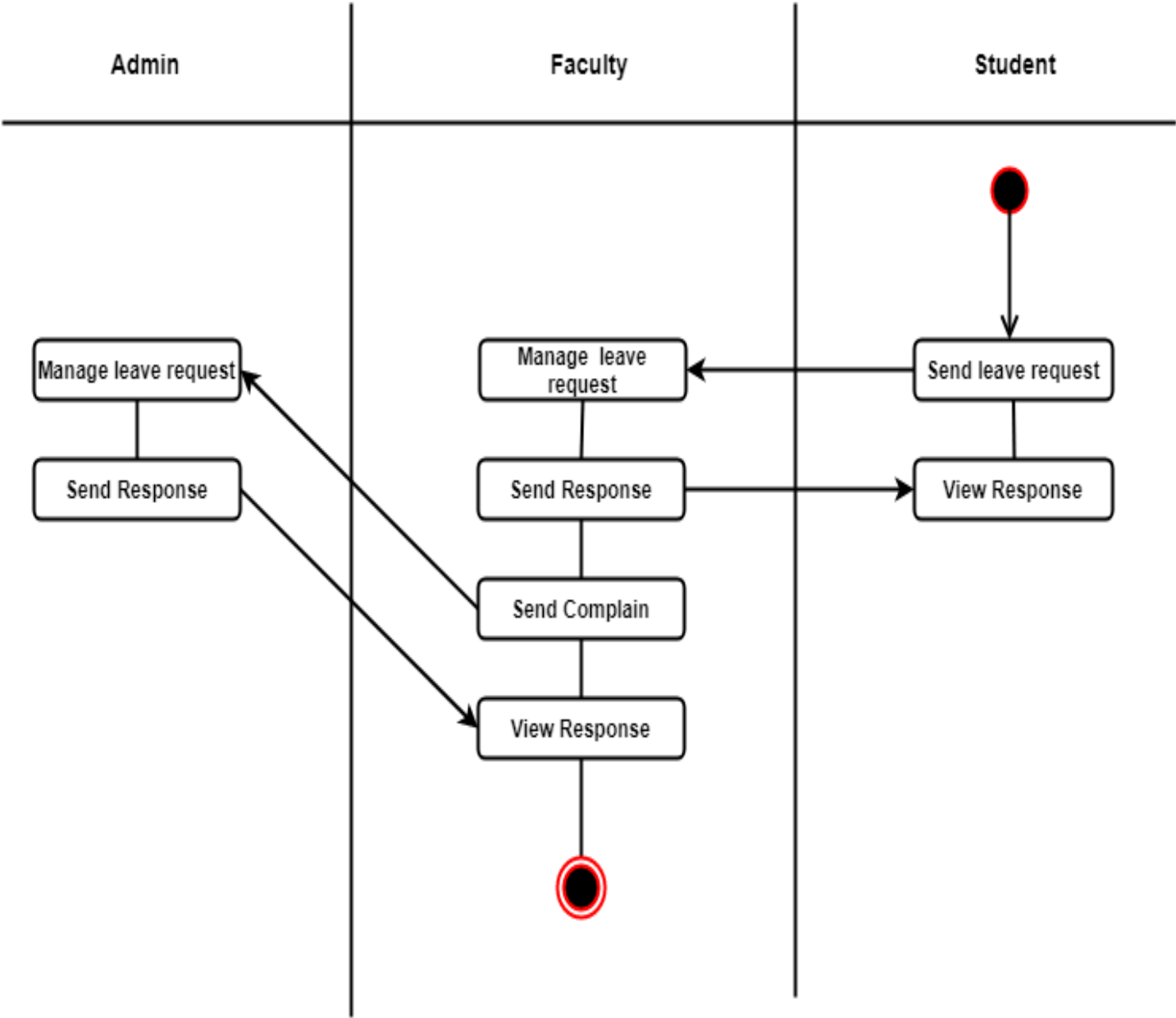


5.2.5 COMPLAIN ACTIVITY

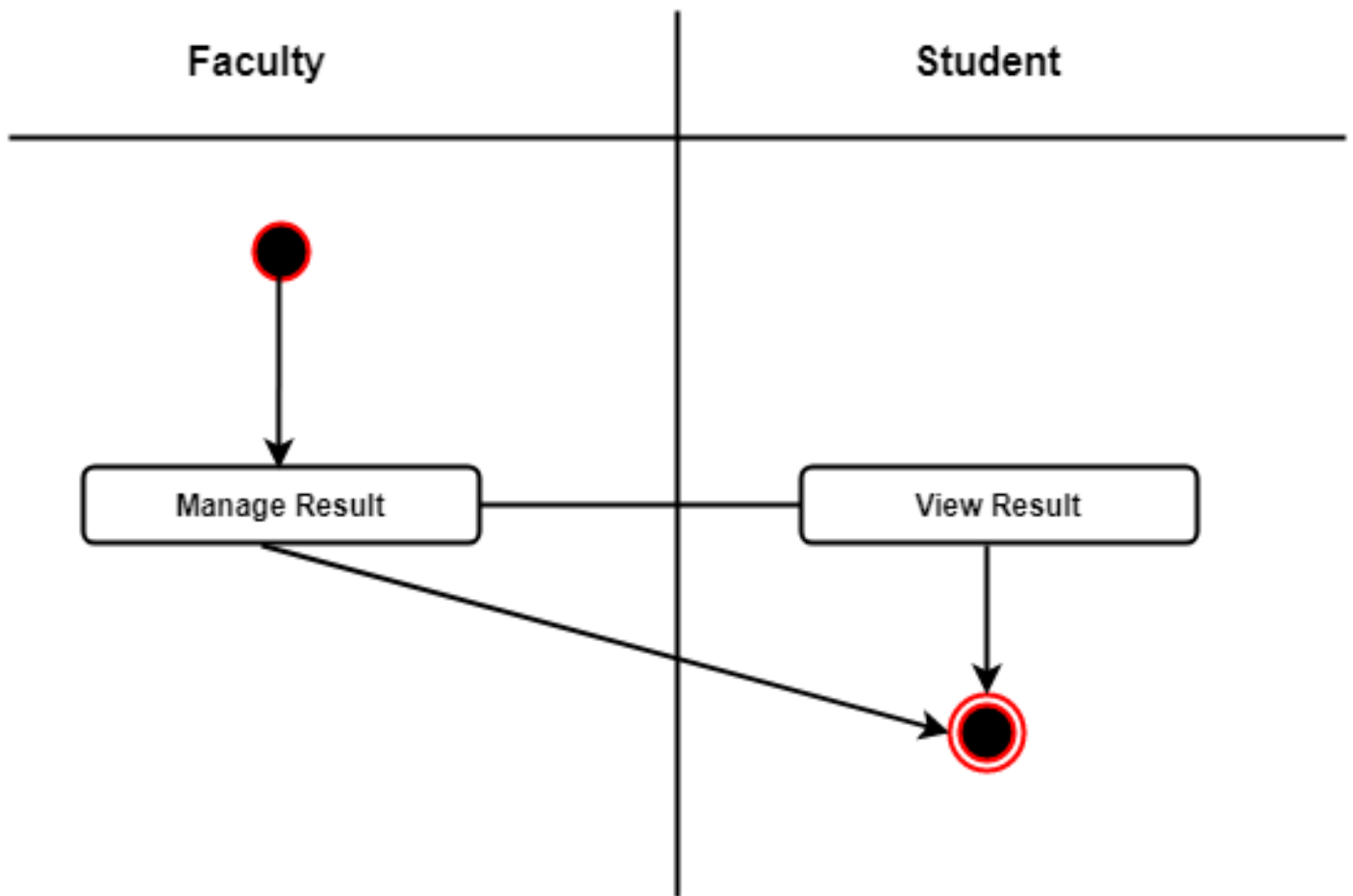


**5.2.6 EXAM ACTIVITY**

5.2.7 LEAVE ACTIVITY



### 5.2.8 RESULT ACTIVITY



## 5.3 SEQUENCE DIAGRAMS

- A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next.
- Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.
- This type of diagram is best used during early analysis phases in design because they are simple and easy to comprehend. Sequence diagrams are normally associated with use cases.

### 5.3.1 SYMBOLS OF SEQUENCE DIAGRAM

#### ❖ Class roles and participants



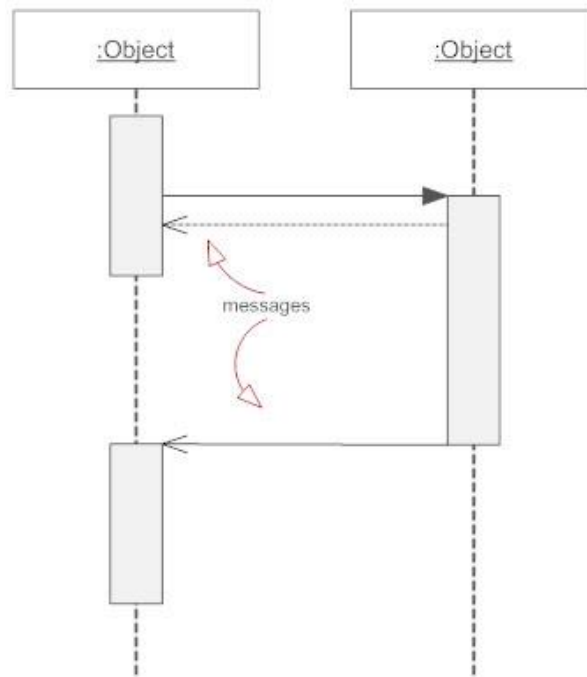
Class roles describe the way an object will behave in context. Use the UML objectsymbol to illustrate class roles, but don't list object attributes.

#### ❖ Activation or Execution Occurrence



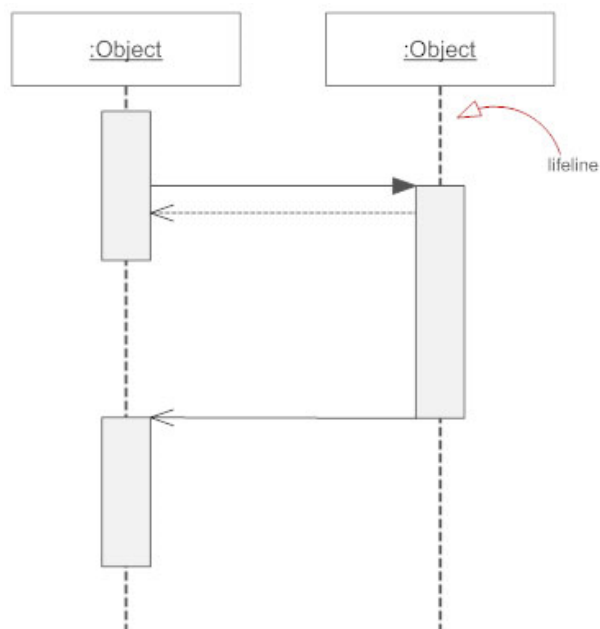
Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.

#### ❖ Messages



Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks. For message types, see below.

### ❖ Lifelines



Lifelines are vertical dashed lines that indicate the object's presence over time.

## Types of messages in sequence diagram

### ❖ Synchronous Message



Synchronous

A synchronous message requires a response before the interaction can continue. It's usually drawn using a line with a solid arrowhead pointing from one object to another.

### ❖ Asynchronous Message



Asynchronous

Asynchronous message doesn't need a reply for interaction to continue. Like synchronous messages, they are drawn with an arrow connecting two lifelines; however, the arrowhead is usually open and there's no return message depicted.

### ❖ Reply or Return Message



Reply or return message

A reply message an object sends to itself, usually shown as a U-shaped arrow pointing back to itself.

### ❖ Create Message

&lt;&lt;create&gt;&gt;



Create message

This is a message that creates a new object. Like a return message, it's depicted with a dashed line and an open arrowhead that points to the rectangle representing the object created.

### ❖ Delete Message

&lt;&lt;destroy&gt;&gt;

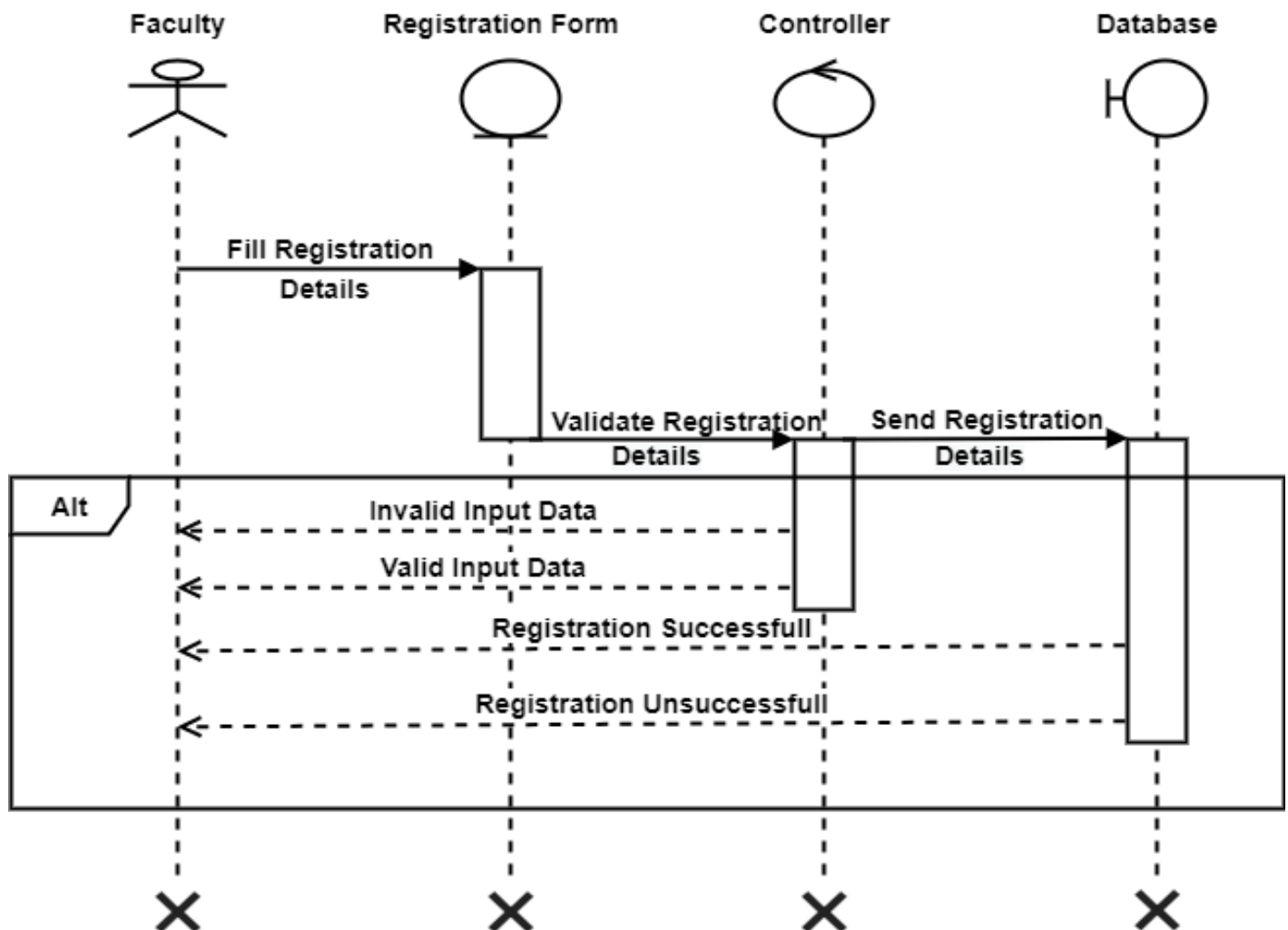


Delete message

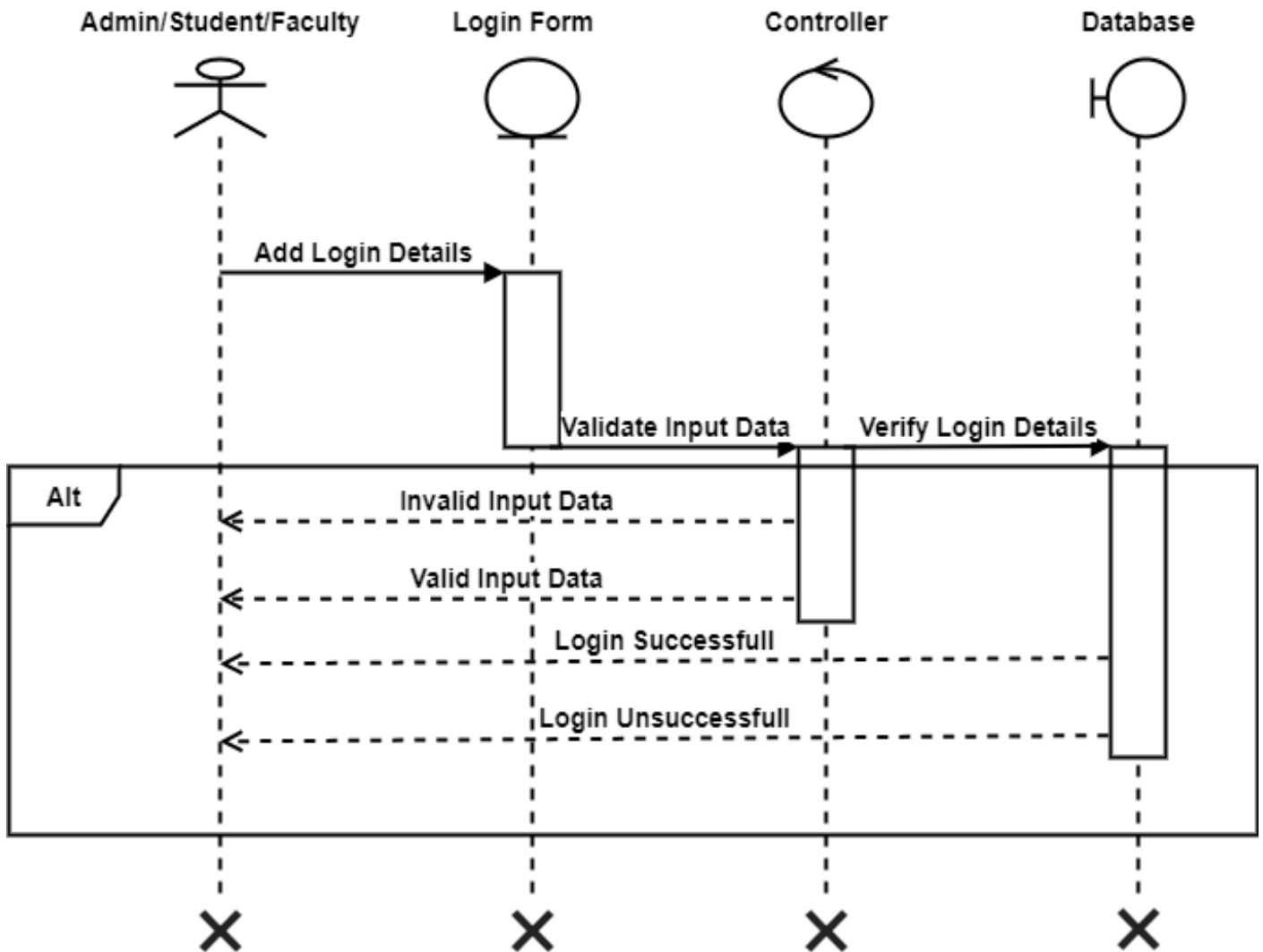
This is a message that destroys an object. It can be shown by an arrow with an x at the end.



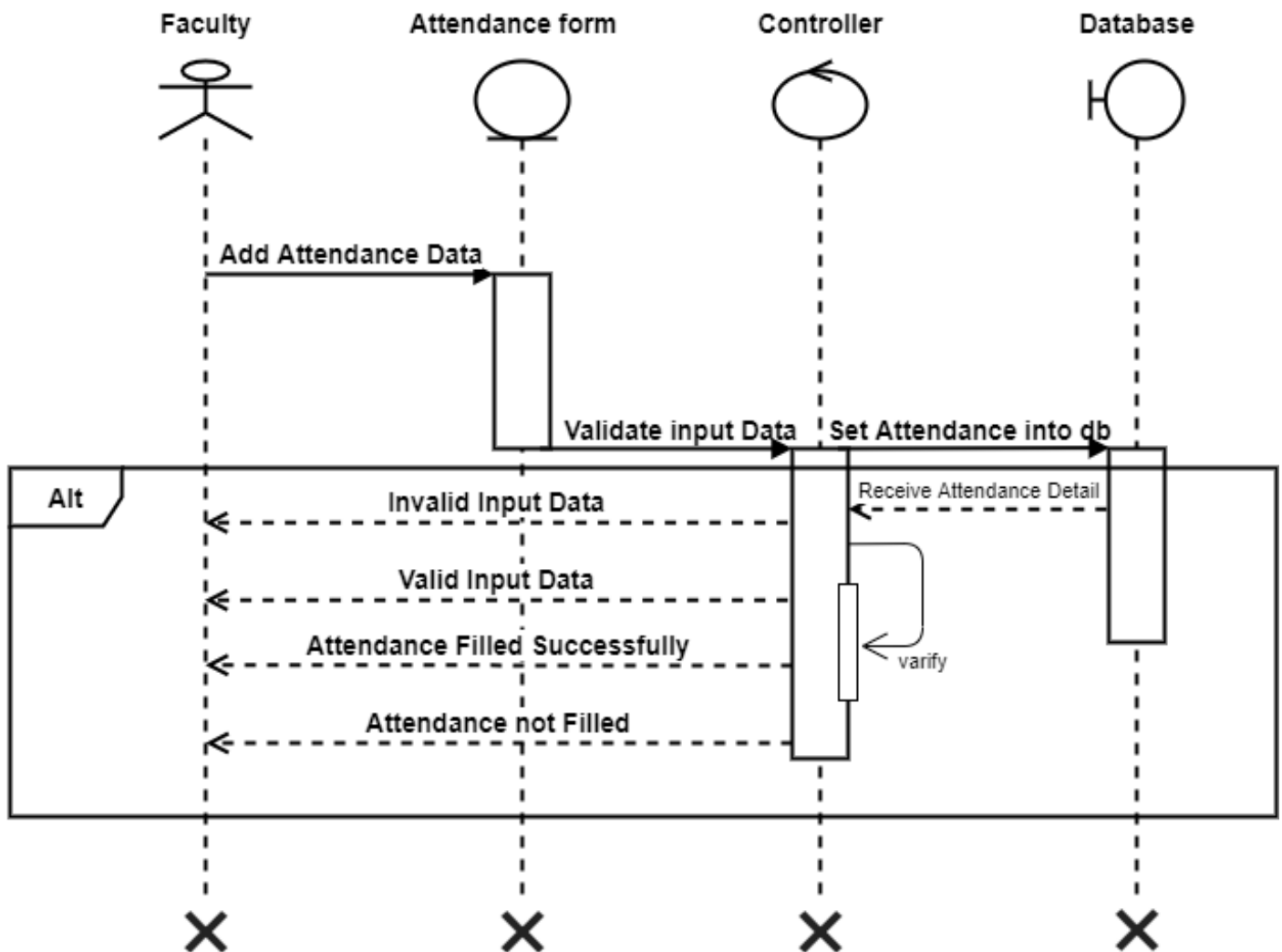
### 5.3.2 REGISTRATION SEQUENCE



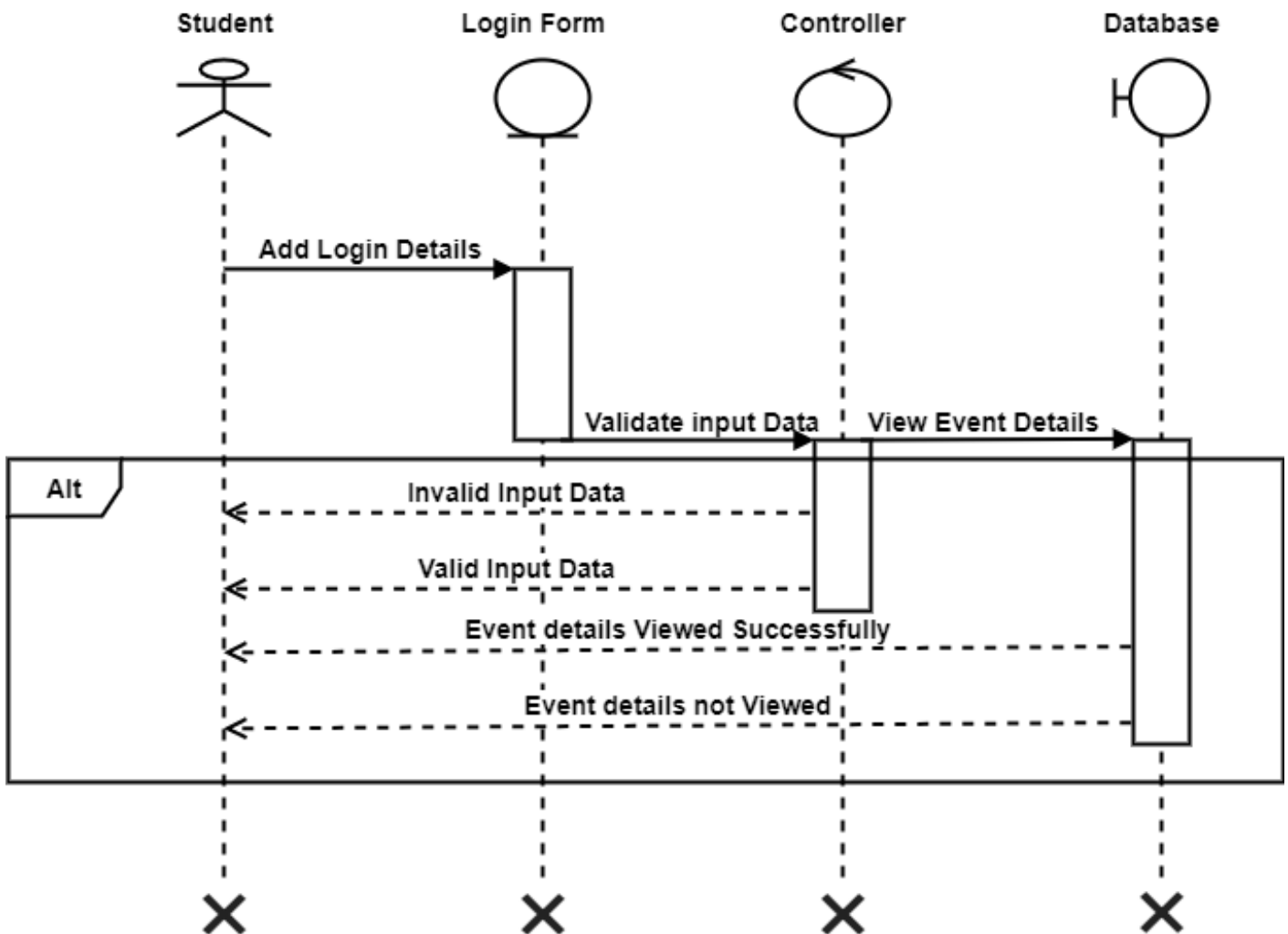
### 5.3.3 LOGIN SEQUENCE



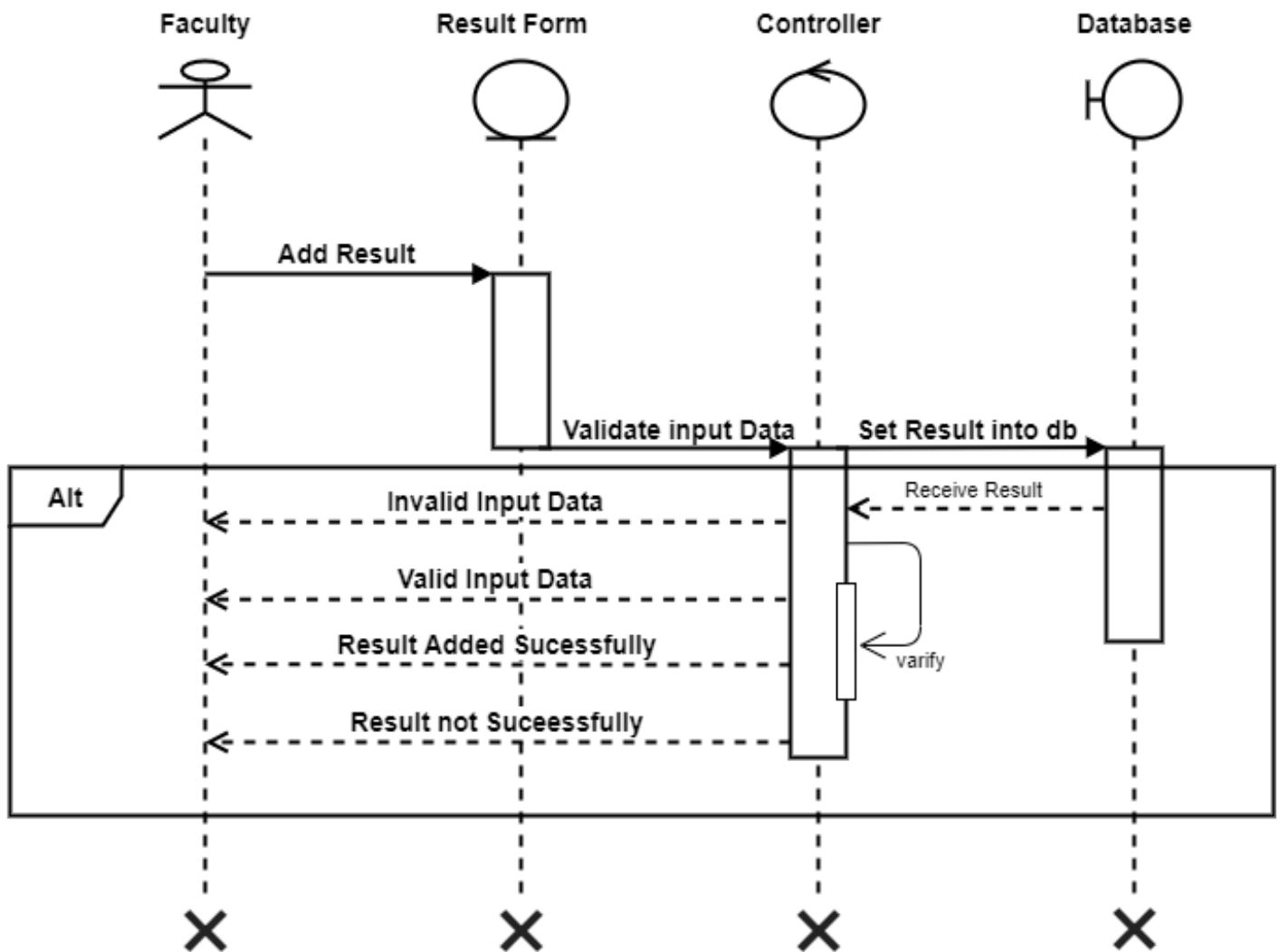
### 5.3.4 ATTENDANCE SEQUENCE



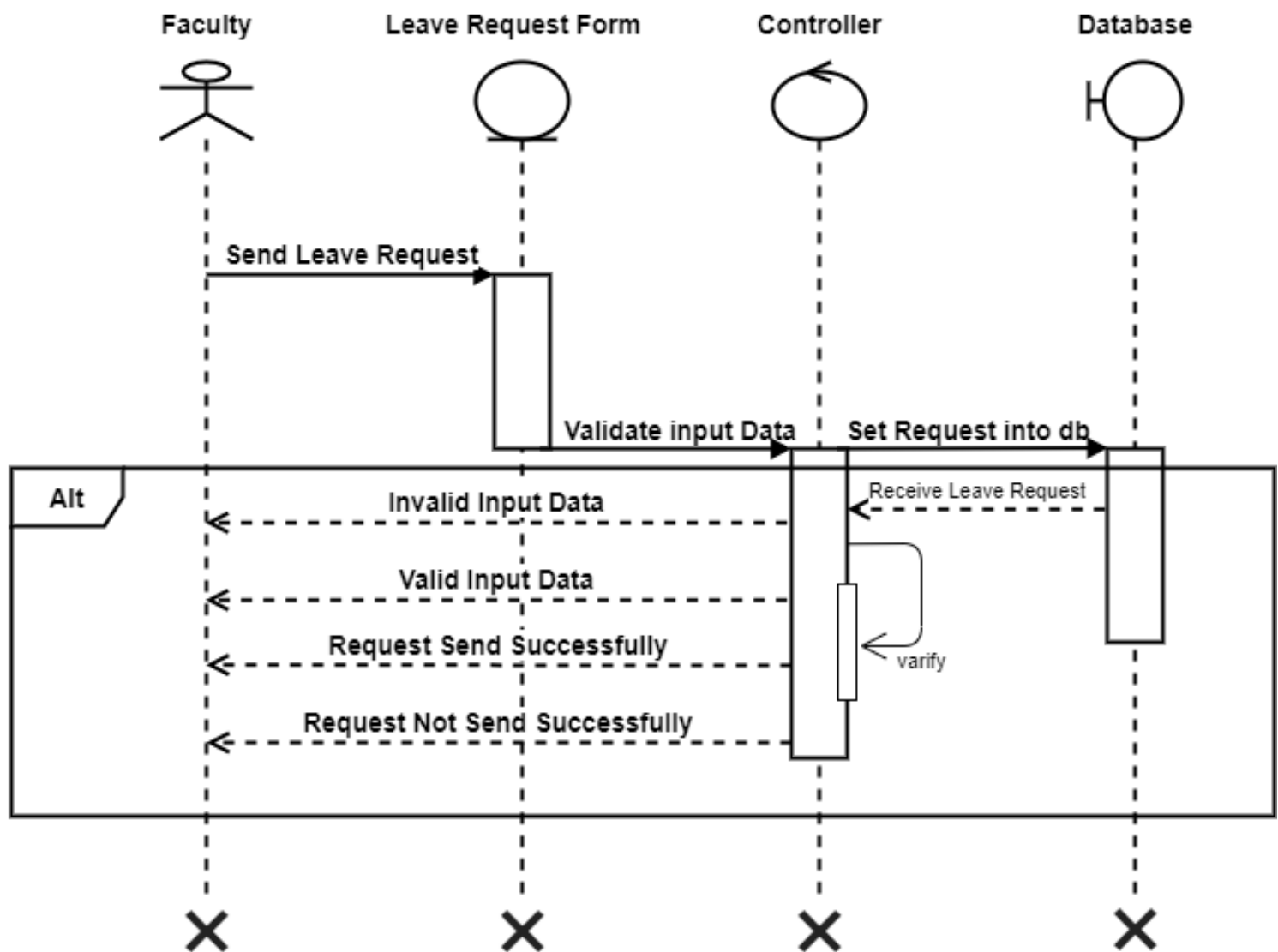
### 5.3.5 EVENT SEQUENCE



### 5.3.6 RESULT SEQUENCE



### 5.3.7 LEAVE SEQUENCE



## 5.4 CLASS DIAGRAM

- A class diagram is a picture for describing generic descriptions of possible systems.
- Class diagrams and collaboration diagrams are alternate representations of object models.
- Class diagrams contain classes and object diagrams contain objects, but it is possible to mix classes and objects when dealing with various kinds of metadata, so the separation is not rigid we applied that concepts over here.
- Class diagrams contain icons representing classes, interfaces, and their relationships.

### 5.4.1 SYMBOLS OF CLASS DIAGRAM

#### ❖ Aggregation



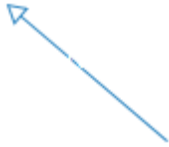
**Aggregation** is a special type of association in which objects are assembled or configured together to create a more complex object. An aggregation describes a group of objects and how you interact with them.

#### ❖ Dependency



**Dependency** relationship is a relationship in which one element, the client, uses or depends on another element, the supplier.

#### ❖ Composition



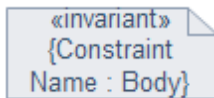
**Composition** represents whole-part relationships and is a form of aggregation.

### ❖ Generalization



**Generalization** is a relationship in which one model element (the child) is based on another model element (the parent).

### ❖ Association



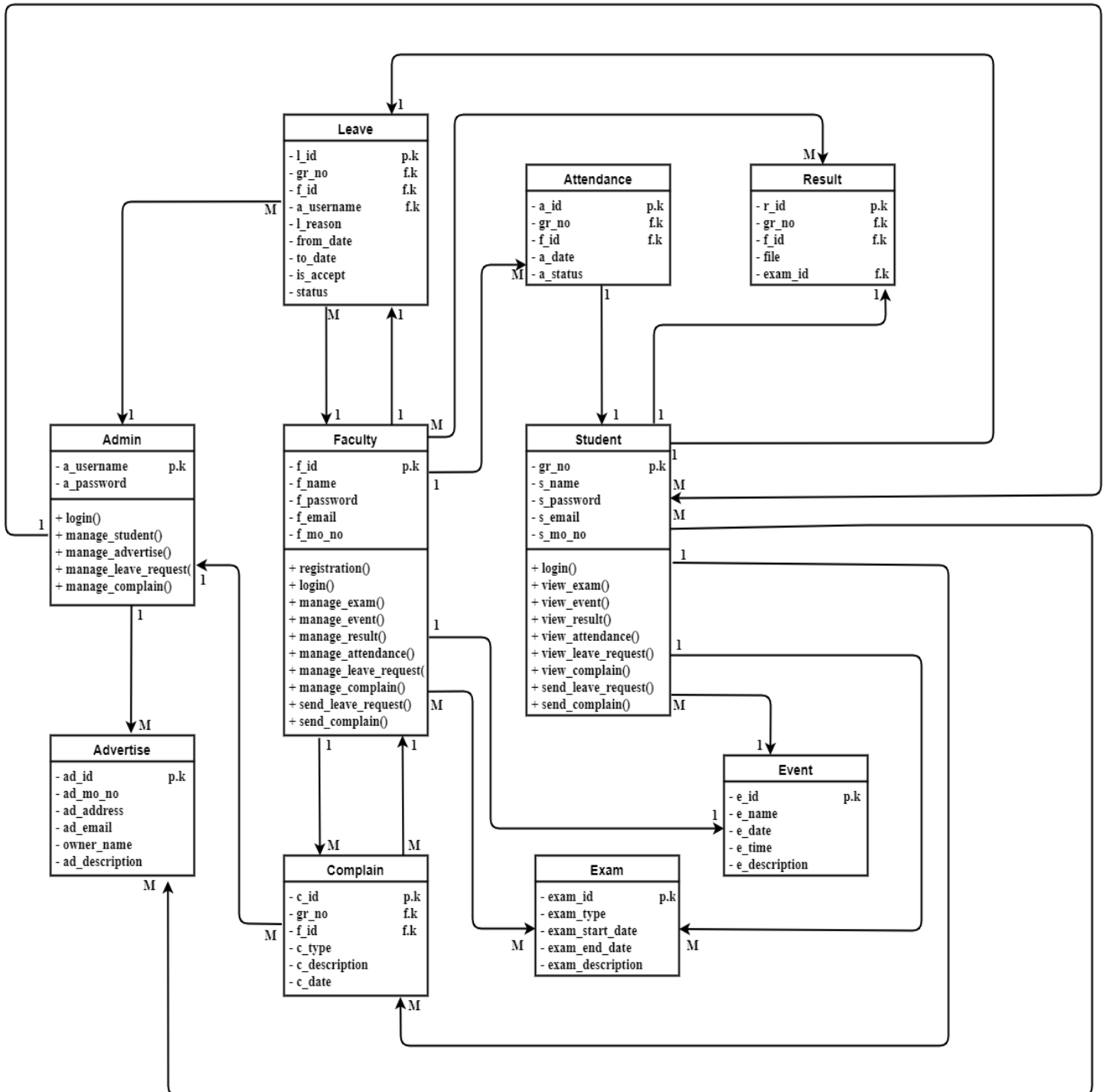
**Association** is a relationship between two classifiers, such as classes or use cases, that describes the reasons for the relationship and the rules that govern the relationship.

### ❖ Constraint

**Constraint** is an extension mechanism that enables you to refine the semantics of a UML model element.



## 5.4.2 CLASS DIAGRAM



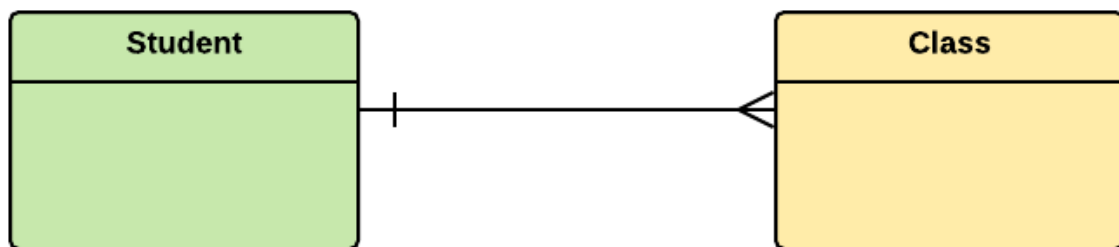
## 6. E-R DIAGRAM

**ENTITY-RELATIONSHIP DIAGRAM (ERD)** displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

### 6.1 SYMBOLS OF E R DIAGRAM

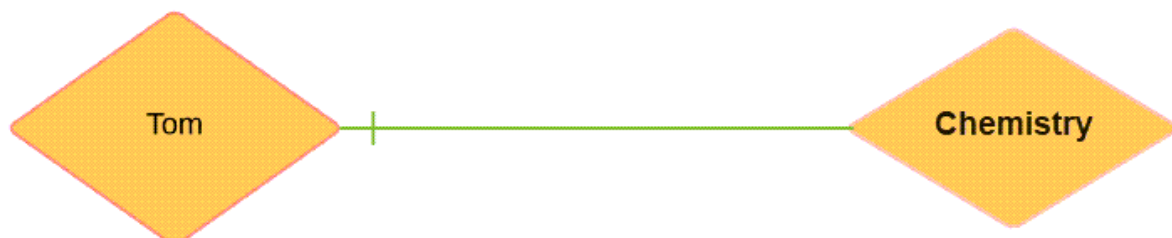
#### ❖ Entity set

An entity set is a group of similar kind of entities. It may contain entities with attribute sharing similar values. Entities are represented by their properties, which also called attributes. All attributes have their separate values. For example, a student entity may have a name, age, class, as attributes.



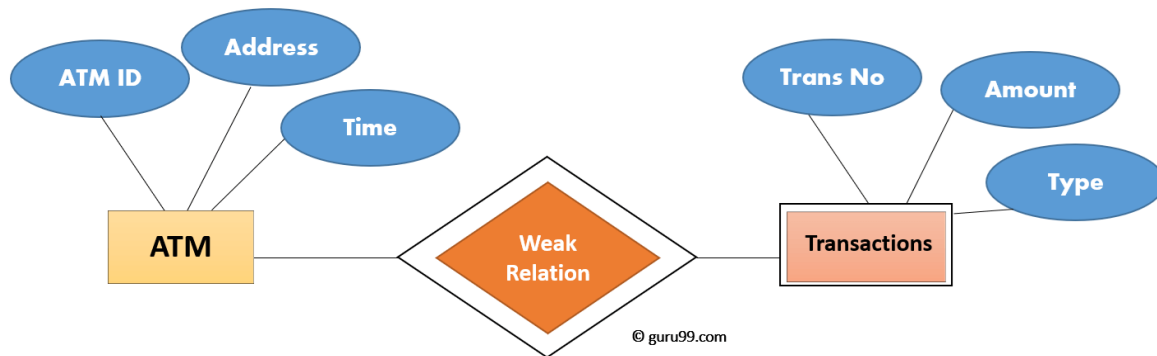
#### ❖ Relationship

Relationship is nothing but an association among two or more entities. E.g., Tom works in the Chemistry department.



### ❖ Weak Entities

A weak entity is a type of entity which doesn't have its key attribute. It can be identified uniquely by considering the primary key of another entity. For that, weak entity sets need to have participation.

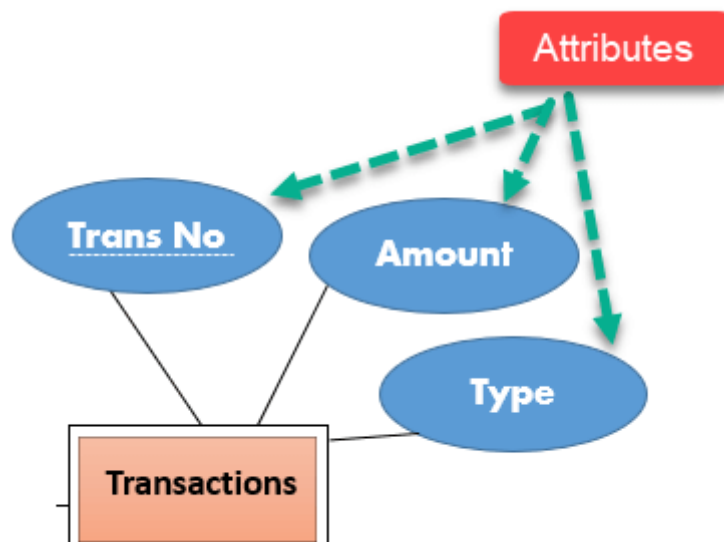


### ❖ Attributes

It is a single-valued property of either an entity-type or a relationship-type.

For example, a lecture might have attributes: time, date, duration, place, etc.

An attribute is represented by an Ellipse

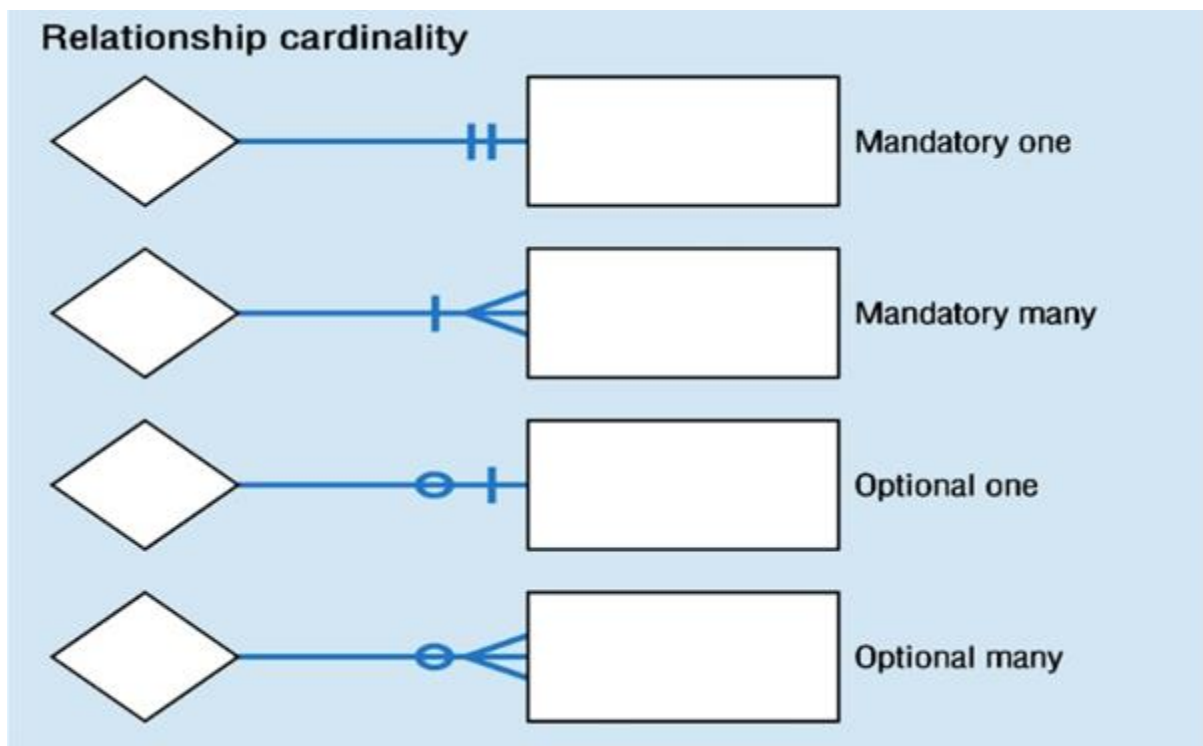


## ❖ Cardinality

Defines the numerical attributes of the relationship between two entities or entity sets.

Different types of cardinal relationships are:

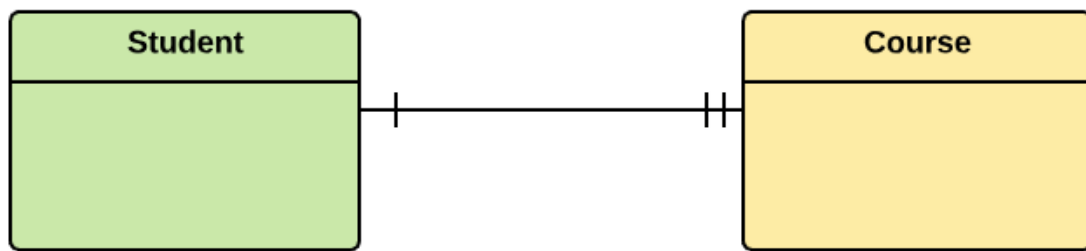
- One-to-One Relationships
- One-to-Many Relationships
- May to One Relationships
- Many-to-Many Relationships



### 1. One-to-one:

One entity from entity set X can be associated with at most one entity of entity set Y and vice versa.

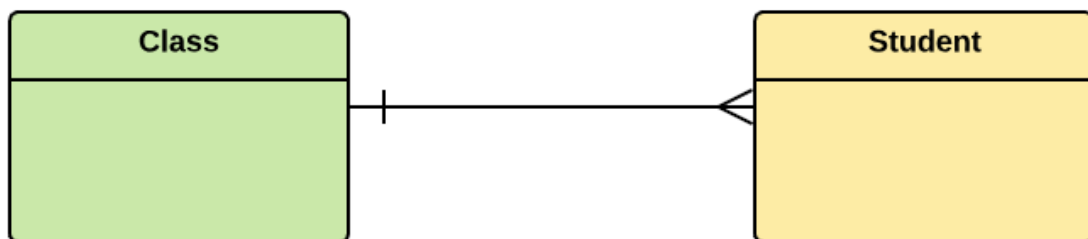
Example: One student can register for numerous courses. However, all those courses have a single line back to that one student.



## 2. One-to-many:

One entity from entity set X can be associated with multiple entities of entity set Y, but an entity from entity set Y can be associated with at least one entity.

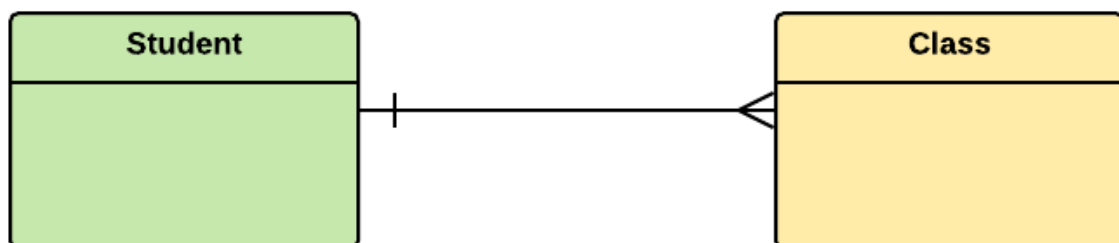
For example, one class is consisting of multiple students.



## 3. Many to One

More than one entity from entity set X can be associated with at most one entity of entity set Y. However, an entity from entity set Y may or may not be associated with more than one entity from entity set X.

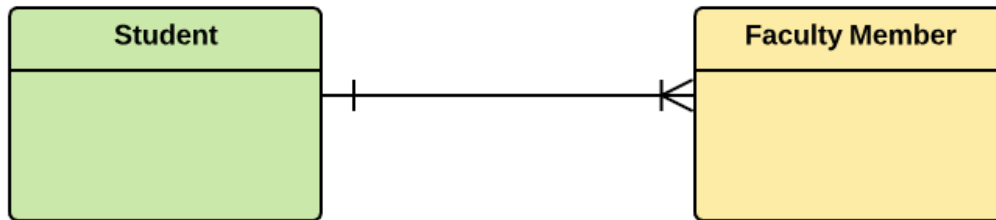
For example, many students belong to the same class.



## 4. Many to Many:

One entity from X can be associated with more than one entity from Y and vice versa.

For example, Students as a group are associated with multiple faculty members, and faculty members can be associated with multiple students.



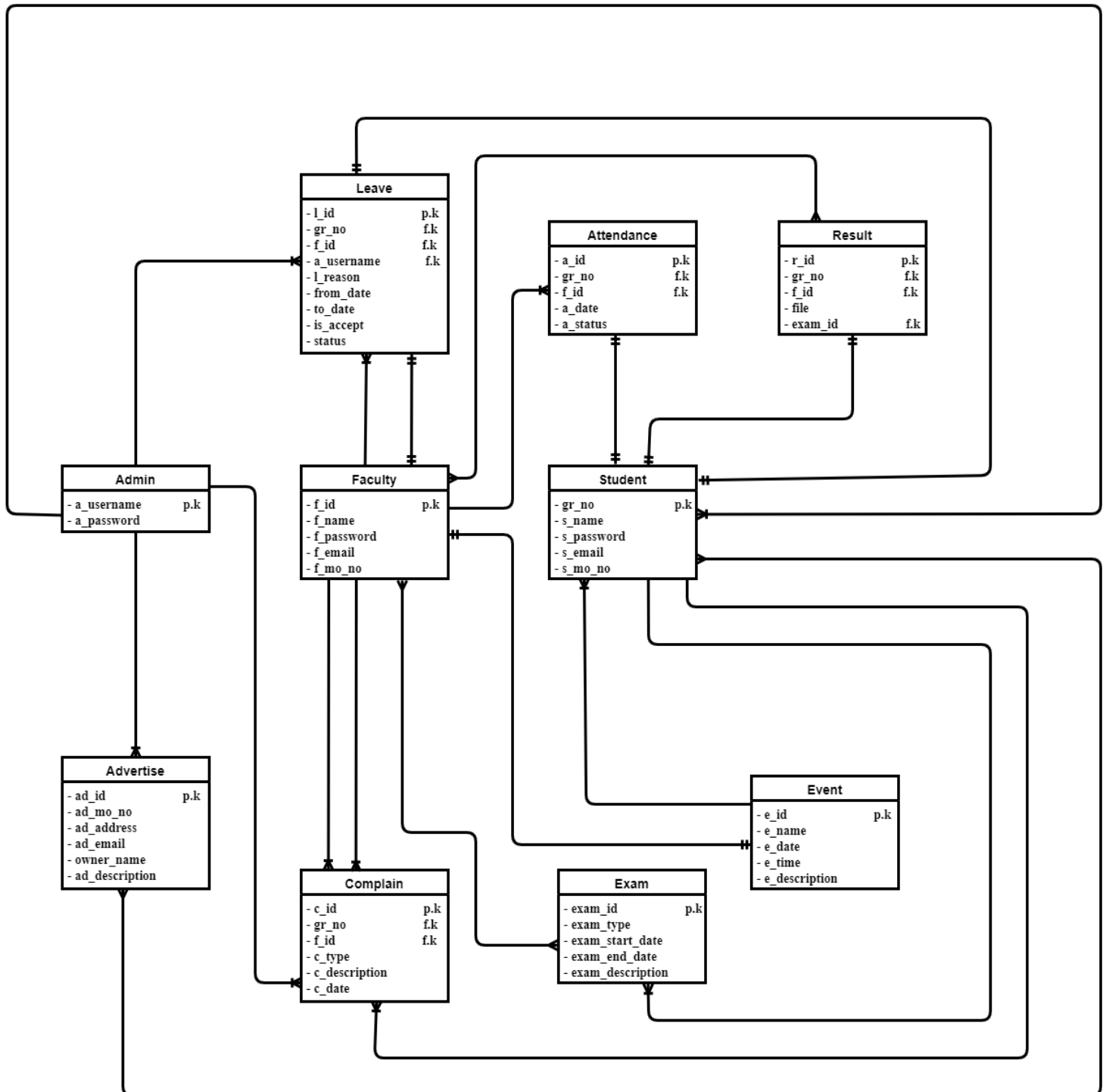
## ❖ ER- Diagram Notations

ER- Diagram is a visual representation of data that describe how data is related to each other.

- **Rectangles:** This symbol represent entity types
- **Ellipses :** Symbol represent attributes
- **Diamonds:** This symbol represents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined
- **Double Ellipses:** Represent multi-valued attributes



## 6.2 E R DIAGRAM



## 7. DATA DICTIONARY

- A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them.
- A first step in analyzing a system of objects with which users interacts is to identify each object and its relationship to other objects. This process is called data modeling and results in a picture of objectrelationships.
- After each data object or item is given a descriptive name, its relationship is described (or it becomes part of some structure that implicitly describes relationship), the type of data (such as text or image or binary value) is described, possible predefined values are listed, and a brief textual description is provided. This collection can be organized for reference into a book called a datadictionary.
- When developing programs that use the data model, a data dictionary can be consulted to understand where a data item fits in the structure, what values it may contain, and basically what the data item means in real-worldterms.



## A DATA DICTIONARY CONTAINS

- ❖ The definitions of all schema objects in the database. (Tables, Views, Indexes, Clusters, Synonyms, Sequences, Procedures, Functions, Packages, Triggers, and soon)
- ❖ How much space has been allocated for, and its currently used by, the schemaobjects.
- ❖ Integrity constraintinformation
- ❖ The names of My SQLusers.
- ❖ Privileges and roles each user has beengranted.
- ❖ Auditing information, such as who accessed or updated various schemaobjects.
- ❖ Other general database information.

<b>Table Name:</b>	<b>Standard</b>
<b>Table Description:</b>	This table contains details about standard
<b>Constraint:</b>	Primary key, Not Null

<b>Attribute</b>	<b>Data type</b>	<b>Constraint</b>	<b>Description</b>
<b>S_id</b>	Int(3)	P.k	Store the id of standard
<b>S_description</b>	Text	Not Null	Store the description of standard

<b>Table Name:</b>	<b>Student</b>
<b>Table Description:</b>	This table contains details about Student
<b>Constraint:</b>	Primary key, foreign key, Not Null

Attribute	Data type	Constraint	Description
<b>gr_no</b>	Varchar(10)	P.k	Store the gr_no of student
<b>s_name</b>	Varchar(25)	Not Null	Store the name of student
<b>s_email</b>	Varchar(40)	Not Null	Store the email of student
<b>s_mo_no</b>	Decimal(10)	Not Null	Store the mo_no of student
<b>s_gender</b>	char(1)	Not Null	Store the gender of student
<b>s_date_of_birth</b>	Date	Not Null	Store the date of birth student
<b>s_id</b>	Int(3)	F.k	Store the id of student
<b>s_address</b>	Text	Not Null	Store the address of student
<b>s_password</b>	Varchar(10)	Not Null	Store the password of student

<b>Table Name:</b>	<b>Faculty</b>
<b>Table Description:</b>	This table contains details about Faculty
<b>Constraint:</b>	Primary key, foreign key, Not Null

Attribute	Data type	Constraint	Description
<b>f_id</b>	Int(3)	P.k	Store the id of faculty
<b>f_name</b>	Varchar(20)	Not Null	Store the name of faculty
<b>f_mo_no</b>	Decimal(10)	Not Null	Store the mo_no of faculty
<b>f_email</b>	Varchar(40)	Not Null	Store the email of faculty
<b>f_gender</b>	char(1)	Not Null	Store the gender of faculty
<b>f_date_of_birth</b>	Date	Not Null	Store the date of birth of faculty
<b>f_address</b>	Text	Not Null	Store the address of faculty
<b>f_password</b>	Varchar(10)	Not Null	Store the password of faculty

<b>Table Name:</b>	<b>Admin</b>
<b>Table Description:</b>	This table contains details about Admin
<b>Constraint:</b>	Primary key, Not Null

<b>Attribute</b>	<b>Data type</b>	<b>Constraint</b>	<b>Description</b>
<b>a_username</b>	Varchar(30)	P.k	Store the username of admin
<b>a_password</b>	Varchar(10)	Not Null	Store the password of admin

<b>Table Name:</b>	<b>Attendance</b>
<b>Table Description:</b>	This table contains details about Attendance
<b>Constraint:</b>	Primary key, foreign key, Not Null

<b>Attribute</b>	<b>Data type</b>	<b>Constraint</b>	<b>Description</b>
<b>a_id</b>	Int(6)	P.k	Store the id of attendance
<b>gr_no</b>	Varchar(10)	F.k	Store the gr_no of attendance
<b>a_date</b>	Date	Not Null	Store the date of attendance
<b>f_id</b>	Int(3)	F.k	Store the id of Faculty
<b>a_status</b>	Varchar(1)	Not Null	Store the status of attendance

<b>Table Name:</b>	<b>Result</b>
<b>Table Description:</b>	This table contains details about Result
<b>Constraint:</b>	Primary key, foreign key, Not Null

Attribute	Data type	Constraint	Description
<b>r_id</b>	Int(6)	P.k	Store the id of result
<b>f_id</b>	Int(3)	F.k	Store the id of faculty
<b>gr_no</b>	Varchar(10)	F.k	Store the gr_no of result
<b>s_name</b>	Varchar(25)	Not Null	Store the name of student
<b>exam_id</b>	Int(6)	F.k	Store the id of exam
<b>File</b>	Text	Not Null	Store the file of result

<b>Table Name:</b>	<b>Exam</b>
<b>Table Description:</b>	This table contains details about Exam
<b>Constraint:</b>	Primary key, foreign key, Not Null

<b>Attribute</b>	<b>Data type</b>	<b>Constraint</b>	<b>Description</b>
<b>exam_id</b>	Int(6)	P.k	Store the id of exam
<b>exam_type</b>	Varchar(20)	Not Null	Store the type of exam
<b>exam_start_date</b>	Date	Not Null	Store the start date of exam
<b>exam_end_date</b>	Date	Not Null	Store the end date of exam
<b>f_id</b>	Int(3)	F.k	Store the id of faculty
<b>exam_description</b>	Text	Not Null	Store the description of exam



<b>Table Name:</b>	<b>Event</b>
<b>Table Description:</b>	This table contains details about Exam
<b>Constraint:</b>	Primary key, foreign key, Not Null

<b>Attribute</b>	<b>Data type</b>	<b>Constraint</b>	<b>Description</b>
<b>e_id</b>	Int(6)	P.k	Store the id of event
<b>e_name</b>	Varchar(20)	Not Null	Store the name of event
<b>f_id</b>	Int(3)	F.k	Store the id of faculty
<b>e_date</b>	Date	Not Null	Store the date of event
<b>e_time</b>	Time	Not Null	Store the time of event
<b>e_description</b>	Text	Not Null	Store the description of event

<b>Table Name:</b>	<b>Complain</b>
<b>Table Description:</b>	This table contains details about Complain
<b>Constraint:</b>	Primary key, foreign key, Not Null

Attribute	Data type	Constraint	Description
<b>c_id</b>	Int(6)	P.k	Store the id of complain
<b>gr_no</b>	Varchar(10)	F.k	Store the gr_no of complain
<b>f_id</b>	Int(3)	F.k	Store the id of faculty
<b>reply_des</b>	Text	Null	Store the reply of complain
<b>c_description</b>	Text	Not Null	Store the description of complain
<b>c_date</b>	Date	Not Null	Store the date of complain
<b>reply_by</b>	varchar	Null	Store the reply_by of complain
<b>reply_date</b>	Date	Null	Store the date of reply
<b>Send_to</b>	Text	Null	Store the id of person who receive complain

<b>Table Name:</b>	<b>Leave</b>
<b>Table Description:</b>	This table contains details about leave
<b>Constraint:</b>	Primary key, foreign key, Not Null

<b>Attribute</b>	<b>Data type</b>	<b>Constraint</b>	<b>Description</b>
<b>l_id</b>	Int(6)	P.k	Store the id of leave
<b>gr_no</b>	Varchar(10)	F.k	Store the gr_no of leave
<b>l_reason</b>	Text	Not Null	Store the reason of leave
<b>from_date</b>	Date	Not Null	Store the from_date of leave
<b>to_date</b>	Date	Not Null	Store the to_date of leave
<b>Isaccept</b>	Boolean	Null	Store the if data is accepert or not
<b>Status</b>	Varchar(20)	Null	Store the status of leave
<b>f_id</b>	Int(3)	F.k	Store the id of faculty
<b>reply_by</b>	Text	Null	Store the id of person who gives reply
<b>reply_date</b>	Date	Null	Store the value of reply date
<b>send_to</b>	Text	Null	Store the id of person who recive leave request

<b>Table Name:</b>	<b>Advertise</b>
<b>Table Description:</b>	This table contains details about Advertise
<b>Constraint:</b>	Primary key, Not Null

Attribute	Datatype	Constraint	Description
<b>ad_id</b>	Int(6)	P.K	Store the id of advertise
<b>ad_mo_no</b>	Decimal(10)	Not Null	Store the number of person
<b>ad_address</b>	Text	Not Null	Store the address of shop
<b>ad_email</b>	Varchar(40)	Not Null	Store the email of shop
<b>owner_name</b>	Varchar(10)	Not Null	Store the name of owner
<b>ad_description</b>	Text	Not Null	Store the description
<b>shop_name</b>	Text	Not Null	Store shop name

## 8. REFERENCES

- **BOOKS**

- System analysis and designing By Elias M. Awad
- Software Engineering By Roser S. Pearson
- Object Oriented Analysis and design By Grady Booch

- **WEBSITES**

- W3Schools.com