**LAB:01**

**INTRODUCTION:**

A Gantt chart is a type of bar chart that represents a project schedule. It is a powerful project management tool that provides a visual timeline for tasks and activities within a project. Each task is represented by a bar, with the length of the bar indicating the duration of the task. The position of the bar along the timeline shows the start and end dates of the task.

To create a Gantt chart showing project phases duration and any overloops ensuring clarity in time line and dependency for the project progress from start to finish. We have following data,

* Requirement: Gathering 1 month (Jan 1st 2024 – 31st Jan 2024)
* Planning: 15 days (1st Feb 2024 – 15th Feb 2024)
* Design: 1 month (15th Feb 2024 – 15th March 2024)
* Development: 2 months (16th March 2024 – 15th May 2024)
* Testing: 1 month (16th May 2024 – 15th June 2024)
* Deployment & Maintenance: Starting from 16th June 2024 – ongoing

**OBJECTIVES:**

1. Visualize the project timeline using a Gantt chart.
2. Show the duration and dependencies of each project phase.
3. Ensure clarity in the timeline and highlight any overlaps between phases.

**GANTT CHART:**

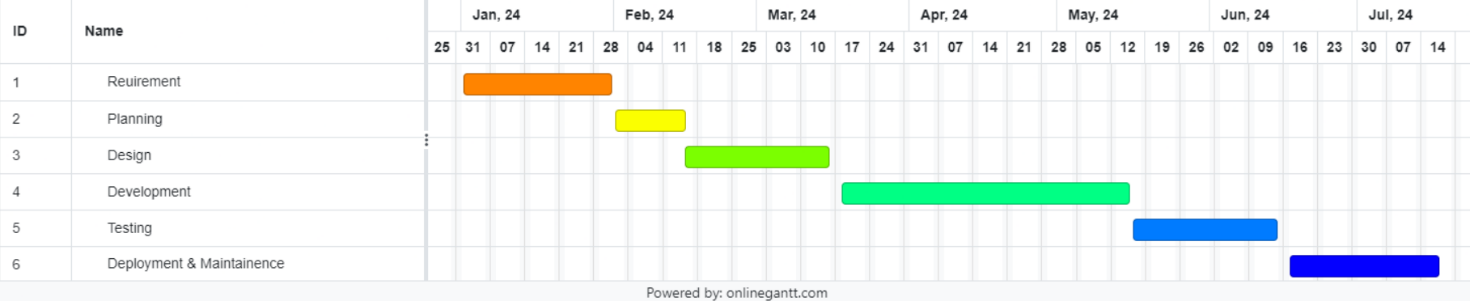
****

Fig: Gantt chart of project phase duration

**CONCLUSION:**

From the above Gantt chart, we can clearly illustrate the project phases, their durations, and dependencies, providing a comprehensive overview of the project timeline. This visualization aids in project management by highlighting overlaps and ensuring that each phase is completed in a timely manner.

**LAB:02**

**INTRODUCTION:**

An Entity-Relationship (ER) diagram is a visual representation of the entities and relationships within a database schema. It helps to illustrate the structure of a database, including the entities (such as tables), attributes (columns), and relationships (links between entities).

To create an ERD of an online shopping system (E-commerce) by showing the associated entities, attributes and relationship between them along with cardinality mapping, we use following notations:

**1. Entity**

An entity may be any object, class, person, or place. In the ER diagram, an entity can be represented as rectangles. Consider an organization as an example—a manager, product, employee, department, etc., can be taken as entities.

**a. Weak Entity**

An entity that depends on another entity is called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.

**2. Attribute**

The attribute is used to describe the property of an entity. An ellipse is used to represent an attribute. For example, ID, age, contact number, name, etc., can be attributes of a student.

**a. Key Attribute**

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.

**b. Composite Attribute**

An attribute that is composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.

**c. Multivalued Attribute**

An attribute can have more than one value. These attributes are known as multivalued attributes. The double oval is used to represent multivalued attributes. For example, a student can have more than one phone number.



**d. Derived Attribute**

An attribute that can be derived from another attribute is known as a derived attribute. It can be represented by a dashed ellipse. For example, a person's age changes over time and can be derived from another attribute like date of birth.

**3. Relationship**

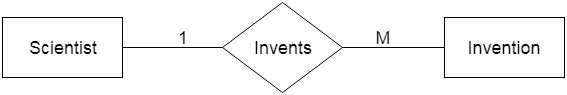
A relationship is used to describe the relation between entities. A diamond or rhombus is used to represent the relationship.

**Types of Relationships:**

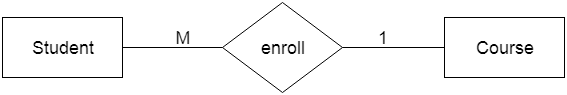
**a. One-to-One Relationship**

When only one instance of an entity is associated with the relationship, it is known as a one-to-one relationship. For example, a female can marry only one male, and a male can marry only one female.

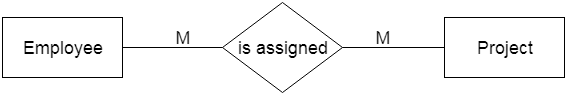
**b. One-to-Many Relationship**

When only one instance of the entity on the left and more than one instance of an entity on the right associate with the relationship, it is known as a one-to-many relationship. For example, a scientist can invent many inventions, but the invention is done by only a specific scientist.

**c. Many-to-One Relationship**

When more than one instance of the entity on the left and only one instance of an entity on the right associate with the relationship, it is known as a many-to-one relationship. For example, a student enrolls in only one course, but a course can have many students.

**d. Many-to-Many Relationship**

When more than one instance of the entity on the left and more than one instance of an entity on the right associate with the relationship, it is known as a many-to-many relationship. For example, an employee can be assigned to many projects, and a project can have many employees.

**OBJECTIVES:**

1. To show different users and systems interaction with the system.
2. To illustrate the functionality or services the system provides
3. To provide clear outline for system design and development

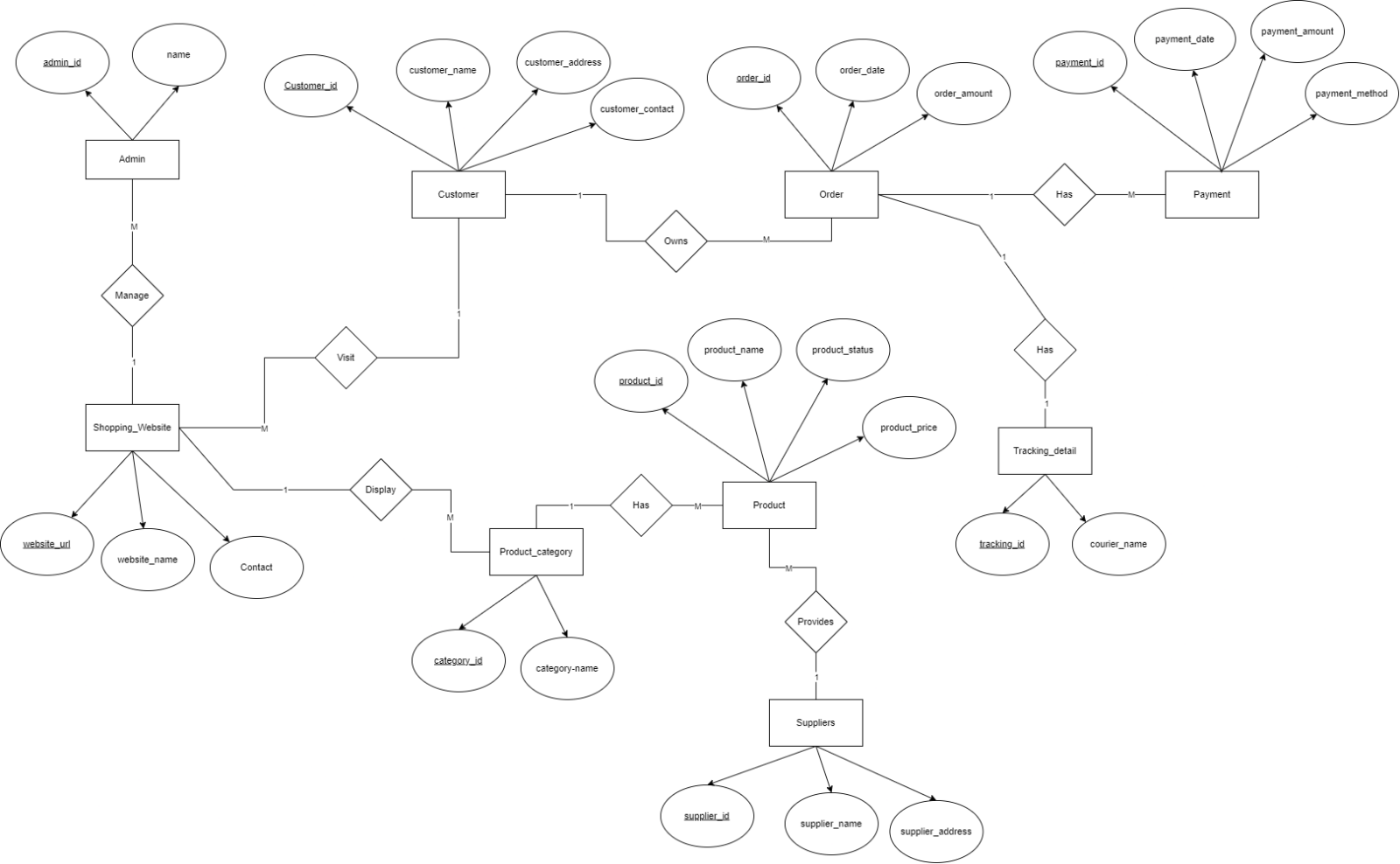
**ER DIAGRAM:**

fig: ER diagram of Online shopping system

**CONCLUSION:**

From the above use case diagram, we can get clear overview of the interactions between users and the system. It serves as a foundational tool for understanding and meeting the requirements of the ATM system.

**LAB:03**

**INTRODUCTION:**

A Use Case Diagram is a vital tool in system design, it provides a visual representation of how users interact with a system. It serves as a blueprint for understanding the functional requirements of a system from a user’s perspective, aiding in the communication between stakeholders and guiding the development process.

To create a use case diagram that shows the ATM system along with its function/operations by an actor, we use following notations:

1. Actors:

Actors are external entities that interact with the system. These can include users, other systems, or hardware devices.

Actors

1. Use Cases:

Use cases are like scenes in the play. They represent specific things your system can do. Use cases are represented by ovals.

Use case

1. System Boundary:

The system boundary is a visual representation of the scope or limits of the system you are modeling. It defines what is inside the system and what is outside.

System Boundary

1. Association Relationship:

TheAssociation Relationship represents a communication or interaction between an actor and a use case. It is depicted by a line connecting the actor to the use case.

Relationship

**OBJECTIVES:**

1. To show different users and systems interaction with the system.
2. To illustrate the functionality or services the system provides
3. To provide clear outline for system design and development

**USE CASE:**

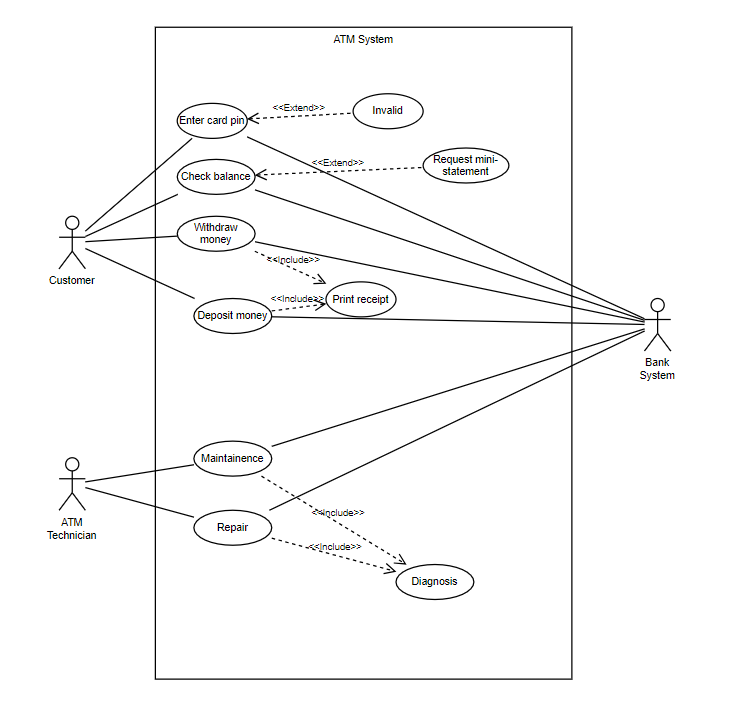


fig: Use case diagram of ATM system

**CONCLUSION:**

From the above use case diagram, we can get clear overview of the interactions between users and the system. It serves as a foundational tool for understanding and meeting the requirements of the ATM system.