Date

Page No.

Practical No :- 06

Aim:- To understand and prepare ER diagram for given problem statement.

Theory:- Entity Relationship Model :-

Entity relationship model is used to represents logical design of a database to be created. In ER model real world objects are abstracted as entities and different possible associations among them are modeled as relationships.

For eg:- Student and college, they are two entities. Students study in college. So, these two entities are associated with relation "Studies in".

* Entity Relationship set :-

An entity is a collection of all similar entities. For example, "students" is an entity set that abstracts all students. Ram, John, are specific entities belongs to this set.

* Attribute of Entity.

Attributes are the characteristics describing any entity belongs to an entity set. Any entity in a set can be

Date

Page No.

described by zero or more attributes.

For example, any student has got a name, age, an address. At any given time a student can study only at one school.

★ Keys :- One or more attributes of an entity set can be used to define following keys.

- Super key :- One or more attribute, or which when taken together, helps to uniquely identify an entity is an of entity set.

- Candidate key :- A minimal subset of a super key. In other words, a super key might contain extraneous attributes, which do not help in identifying an object uniquely. When such attributes are removed, the key formed so is called a candidate key.

- Primary key :- A database might have more than one candidate key. Any candidate key is chosen for a particular

Date
Page No.

implementation of the database is called as primary key.

- Prime attribute :- Any attribute taking part in a super key.

* Weak Entity :-
An entity set is said to be weak if it is dependent upon another entity set. A weak entity can't be uniquely identified only by its attributes. In other words, it doesn't have a super key.

* ~~Four~~ ~~exam~~ Entity Generalization And Specialization :-

Once we have identified the entity sets, we might find some similarities among them. For example, multiple person interact with a banking system. Most of the them are customers, and rest employees or other service providers.

ER model use the "ISA" hierarchy to depict specilization and thus, generalization.

Date
Page No.

★ Mapping Cardinalities :-

One of the main tasks of ER modeling is to associate different entity sets. Let's consider two entity sets E1 and E2 associated by a relation set R. Therefore there are four type of mappings:

- One to One : An entity in E1 is related to at most a single entity in E2, and vice versa.
- One to many :- An entity in E1 could be related to zero or more entities in E2.
- Many to One :- Zero or more of entities in E1 could be associated to a single entity in E2.
- Many to Many : Any no. of entities could be related to any no. of entities in E2, including zero and vice versa.

Date
Page No.

~~Importance of ER diagram~~ :-

* ER diagram for our given problem statement.

→ Design a e-commerce website

For making ER diagram, the first step is to identify the entities, attributes and relationships. Then we can represents them in ER diagram.

1) Entities in our ER diagram for e-commerce website :-

→ User / Admin :- Represents both user and admins.

→ Login :- Contains user credentials.

→ Product :- A key entity ~~representing~~ products available for sale.

→ Category :- categorize product with attributes.

→ Cart :- Stores product added into with attributes.

→ Orders :- Represents customer orders with attributes.

→ Customer Product :- Represents product that user can customize.

→ User Upload :- Manages images uploaded by users with attributes.

→ Views : Tracks the no. of product is viewed.

Date
Page No.

- Attributes

→ User / Admin :- ⇒ - username, email, contact, role
→ Login :- - username, password
→ Product :- - product_id, product_name, product_desc, price, color, brand, image1, image2, image3.
→ Category :- - cat_id, cat_name, cat_description.
→ Cart :- - quantity, size, custom
→ Order :- order_id, address, city, state, zip_code, contact
→ Custom_Product :- - title, image, cost, uid, cid
→ User_Upload :- uid, image.
→ Views :- views.

- Relationships :-
→ Login - User / Admin
→ Admin - Category
→ Category - Product
→ Product - Cart
→ Product - Order
→ Order - Customer
→ Custom-product - cart
→ User - User Upload
→ Product - Views

P	T	D	K	Total
3M	3M	3M	6M	15M
3	3	3	4	13
Sign With Date				

Result :- Hence, we studied ER diagram for our given problem statement.

Practical No :- 07

Date

Page No.

Aim :- To study and illustrate the use of class diagram as per the given problem statement.

Theory :- Class Diagram :-
It is a graphical representation for describing a system in context of its static construction.

* Elements in class Diagram :-
class diagram contains the system classes with its class members, operations and relationships between classes.

* Class :- A set of objects containing a similar class member function is describe by a class. In UML syntax, class is identified by solid outline rectangle with three compartments which contain.
  - class name
  - Attributes
  - Operations.
  - Generalization and Specialization.

Date

Page No.

Operation eg. :-

Course
Course name : string Course ID : string.
addCourse () : string removeCourse () : string.

* Relationships :-

~~• Association~~ Existing relationships in a system describe legitimates connections between the classes in that system.

• Association

It is an instance level relationship that allows exchanging messages among the objects of both ends of association. A simple straight line connecting two class boxes represent beginner association.

• Aggregation :-

It is a special form of association which describes a part-whole relationship between a pair of classes. It means in a relationship, when a class holds some instances of related classes, then that relationship can be designed as an aggregation.

Date
Page No.

- Multiplicity :-
It describes how many no. of instances of one class is related to the no. of instances of another classes in an association.

Notations for different types of multiplicity :-

Single Instance - 1
Zero or one Instance - 0..1
Zero or more Instance - 0...*
One or more instance - 1...*
Particular range (two to six) - 2...6

★ Class Diagram for our given problem statement :-

Functionalities :-

→ Customer :- ◦ Attributes :- customer-id, customer-name, email, mobile, address.
◦ Method :- register(), login(), updateProfile()

→ Seller :- ◦ Attributes :- id, address, email, mobile, seller-rating
◦ Method :- register(), login(), updateProfile().

→ Product :- ◦ Attribute :- product-id, product-name,

Date
Page No.

, qty-available , product-cost , seller-id , posted-date.

• Methods :- addItemCount() , SellProduct() , getProductDetail() , buyProduct().

→ Cart :- • Attributes :- cart-id , product-id , quantity

• Methods :- addItem() , updateQuantity() , cartDetails() , PlaceOrder() , Checkout().

→ Order :- • Attribute :- order-id , order-date , seller-id , cust-customer-id , product-id , total-amount , delivery date , delivery-status.

• Methods :- • orderDetails().

→ Payment :- • Attribute :- id , order-id , paid , total , details.

• Methods :- sendOTP() , confirmTransaction() , getPaymentDetails().

→ Admin :- • Attribute :- admin-id , admin-password

• Method :- register() , login() , updateCatalog().

→ Reviews :- • Attributes : review-id , customer-id , review-content , rating , product-id

• Methods :- addReview() , deleteReview() , editReview().

Result :- Hence , we have successfully illustrated Class diagram for problem statement.

Rubric	Timely	Book	Total
3M	3M	9M	15M
3	3	4	12

Sign With Date

Date

Page No.

Practical No: 08

Aim:- To study and illustrate use case diagram for your given problem statement.

Theory: ✱ Use Case Diagram :-

Use case diagram belongs to the category of the behavioural diagram of uml design. Use case diagrams aim to present a graphical overview of the functionality provide by the system. It consits of a set of actions that the concerned system can perform, one or more actors dependecies among them.

- **Actors :-** An actor can be defined as an object external to the system, which interacts with the system to get some meaningful work done. Actors could be human, devices, or even other systems.

- **Use case :-** A use case is simply a functionality provided by a system. continuing with the example of ATM, withdraw cash is functionality of that the ATM provides. Therefore, this is a use case.

Date
Page No.

the cases interacts or includes both successful and unsuccessful scenarios of user interactions with the system.

★ Subject :- Subject is simply the system under consideration. Use cases apply to a subject. For example, ATM is a subject, having multiple use cases and multiple actors interact with it.

★ Graphical Representation :-
An actor is represented by a stick figure and name of the actor is written below. A use case is depicted by the an ellipse and name of the use case written inside it. The subject is shown by a rectangle, label for the system could be put inside it. Use cases are drawn inside the rectangle, and actors are drawn outside the rectangle, as shown in figure.

★ Association between Actors and Use Cases
A use case is triggered by an actor. Actors can be used cases are connected through binary associations indicating that the two communicates through message passing.

Date
Page No.

* An actor must be associated with at least one use case.

* Use case Relationships :
There are three types of Relationships.
→ Include Relationship
→ Extend Relationship
→ Use case Generalization.

* Identifying Actors

For a given problem statement, the actors could be identify by asking the questions:

→ Who gets most of the benefit from system?

→ who keeps the system working?

→ what other software / hardware does the system interact with.

→ Any Interface between concerned system and any other system.

* Identifying Use Cases :-
Once the primary and secondary actors are being identified, we have to find out their

Date
Page No.

goals i.e what our functionality they can obtain from the system.

* Guidelines for drawing use cases diagrams

Following general guidelines would be kept in mind while trying to draw a use case diagram:-

→ Determine the system boundary.
→ Ensure that individual actors have well-defined purpose.
→ Use cases identified should let some meaningful work done by actors.
→ Use include relationship to encapsulate common behaviour among use cases, if any.

* Use case diagram for our problem statement:-

This use cases diagrams shows:-

1.) Actors : - Customers
- Admin / Managers

2.) Use Cases :- - Create Accounts
- Log In

Date

Page No.

- show orders
- Add products
- search for product
- View product details.
- Update, View Cart
- Place Order
- checkout
- Manage Accounts
- View order History.

3. Relationships :- << Include Relationship >> :-
   - Log In , checkout ,
   - Manage Accounts , Dashed arrows.

Result :- Hence successfully studied and illustrate the use diagram for our given problem statement.

P	T	D	K	Total
3M	3M	3M	6M	15M
3	3	3	5	14
Sign With Date				

Practical 09

* Aim - To study and illustrate the various UML diagram design tool

* Theory -

What is UML

Unified Modelling Language is an essential tool in software engineering for visualising, specifying, construction and documenting software system. This report covers several popular UML diagram design tool, their features and their applicability in SEPM.

Lucid Chart

Key Features -

- Web based tools for real time collab
- Supports all 14 UML types.
- Extensive template library
- Intuitive drag and drop interface
- Integration with popular platforms (G Suite, Atlassian)

SEPM Application. - Ideal for team projects due to its collaboration features. Useful for creating and sharing diagrams throughout the software development lifecycle.

2 Visual Paradigm

Key Features –

- Comprehensive UML modelling tool.
- Code generation and reverse engineering
- Database design and ORM
- Requirements Managment
- Agile project management features.

SEPM Application

Suitable for complex projects requiring extensive modelling and integration with other projects managment aspects.

3 Star UML

Key Features –

- Open source tool with a user friendly interface
- Support 11 types of UM diagram
- Model driven development approach
- Extensible through add ons
- Cross platform (windows, macOS, Linux)

SEPM application

Good for academic projects and small to medium sized teams. Its open source nature makes it cost efficient for students to use.

4 Dia

Key Features

Date
Page No.

Practical No :- 10

Aim :- To study and illustrate the various automation tools like selenium.

Theory :- There are several automation tools across different domain, each used to improve efficiency, reduce human errors, or streamline workflows. Below are the categories of automation tools with illustration of their usage.

★ **Testing aumation tools :-**
Testing automation tools helps automate the testing of software application. They ensure that software works as expected by running pre-defined test scripts without manual intervention.

→ **Selenium :-** A widely used open-source for automating web browser actions.

- Use case :- Automating browser testing across different browsers like chrome, firefox and safari.

- Example :- Automatically performs login and form submission tests on a website.

Date
Page No.

→ Junit :- A unit testing framework for java application.

- Use case :- Automating the testing of individual units of source code in java programs.
- Example :- Running automated tests for java methods in an application.

→ Appium :- A mobile application testing tool that supports automation of both android and iOS platforms.

- Use case :- Automating mobile app testing across different devices.
- Example :- Testing mobile app functionalities such as navigation, swipe gestures, and login processes.

★ Selenium Testing :-

It is an open-source automation testing tool primarily used for automating web applications. It allows testers to write scripts in multiple programming languages to automate web browsers.

Selenium supports multiple browsers such as chrome, firefox, safari and Edge.

Date
Page No.

* Components of Selenium :-

1. ~~Web~~ Selenium Web Driver :-
   → Purpose :- Automates the browsers and interacts with web elements.

→ Architecture :- • Client Libraries
• JSON Wire Protocol
• Browser Driver :
• Browsers.

2. Selenium IDE :-
   → Purpose :- A simple record and playback tool used for creating quick test scripts without coding.

→ Features :- • Allows recording interactions with a web-page.
• Test steps are recorded as commands can be played back.

→ Limitations :- It has limited control flow, cannot handle complex scenarios, and is less flexible than Web Driver.

3) Selenium Guild :-

→ Purpose :- Used for Parallel test execution across multiple environments.

→ Architecture :- • Hub
• Node

→ Use Case :- • Cross browser Testing.

★ Use Cases :-

→ Cross browser Testing :- Selenium web-driver allows tests to run on different browsers, ensuring compatibility across Chrome, Firefox, Safari, and beyond.

→ Regression Testing :- Automating repetitive test cases to check if new code breaks existing functionality.

→ Continous Integration / Continous Deployment (CI/CD) :- Automated selenium tests can be integrated with tools like jenkins, allowing automatic test execution during the build process.

Date
Page No.

★ Advantages of Selenium :-

→ Open source and free to use.

→ Support multiple languages and browsers.

→ Flexible : can be integrated with various testing frameworks and CI/CD tools.

→ Parallel execution :- Selenium grid allows you to run tests on multiple machines and browsers simultaneously.

Limitation of Selenium :-

No desktop or mobile App Automation.

Learning Curve.

Limited Support for dynamic web elements.

Result :- Hence, we studied and studied automation tools like ~~automation~~ Selenium.

P	T	D	K	Total
3M	3M	3M	6M	15M
3	3	3	9	12

Sign With Date