islington college
(इस्लिङ्टन कलेज)

LONDON
METROPOLITAN
UNIVERSITY

CS5002NI Software Engineering

35% Individual Coursework

Milestone number: 3

AY 2024-2025

Credit: 30

Student Name: Srijan Bhandari

London met ID: 23056358

College ID: NP01CP4S40076

Assignment Due Date: April 2025

Assignment Submission Date: 28th April 2025

Submitted to: Rubin Thapa

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

Table of Figures 3

Table of Tables 0

1. Introduction 1

2. Planning Diagrams 2

2.1 WBS(Work Breakdown Structure) 2

2.2 Gantt Chart 3

3. OOAD Analysis 5

3.1 Use Case Diagram 5

3.2 High Level Description 6

3.3 Expanded Level Description 8

3.4 Communication Diagram 11

3.5 Activity Diagram 14

4. Class Diagram and Domain Class 16

4.1 Domain Class 16

4.2 Class diagram 19

5. Further Development 21

5.1. Architectural Choice 22

5.2. Design Pattern 24

5.3. Development Plan 26

5.4. Testing Plan 27

5.5. Maintenance Plan 29

6. Prototype 30

1. Register Account 30

2. Login Account 31

3. Admin Dashboard 32

4. Customer Dashboard 33

5. Sales Report 34

6. Purchase Order 35

7. Create Purchase Order 36

8. User Management 37

9. Add User(admin/customer/staff) 38

10. Inventory Management 39

11. New Product Listing 40
12. Product Page 41
13. Product Detail 42
14. Cart... 43
15. Payment Page.. 44
References.. 45

Table of Figures

Figure 1: WBS 2
Figure 2: Gantt Chart 3
Figure 3: Gantt Chart 2 4
Figure 4: Use Case Diagram 5
Figure 5: Communication Diagram Part 1 11
Figure 6: Communication Diagram Part 2 12
Figure 7: Full Communication Diagram 13
Figure 8: Activity Diagram 14
Figure 9: Class Diagram 19
Figure 10: Layer Architecture 23
Figure 11: Creational Pattern 24
Figure 12: Structural Pattern 25
Figure 13: Behavioral Pattern 25
Figure 14: Register Account 30
Figure 15: Login Account 31
Figure 16: Admin Dashboard 32
Figure 17: Customer Dashboard 33
Figure 18: Sales Report 34
Figure 19: Purchase Order 35
Figure 20: Create Purchase Orde 36
Figure 21: User Management 37
Figure 22:Add User 38
Figure 23: Inventory Management 39
Figure 24: New Product Listing 40
Figure 25: Product Page 41
Figure 26: Product Detail 42
Figure 27: Cart 43
Figure 28: Payment Page 44

Table of Tables

Table 1: Tasks Start & End Date 4
Table 2: Payment Expanded Level Description 8
Table 3: Access User System Expanded Level Description 9
Table 4: Domain Class Table 16
Table 5: Cardinal Types 19

1. Introduction

This coursework is about developing and designing the OOAD (Object-Oriented Analysis and Design). The designing and developing of OOAD is done to streamline and improve the operation of the Inventory Management System (IMS) of Global Tech Corporation. Before developing OOAD we also developed the planning diagrams Like (WBS and Gantt Chart) whereas OOAD like use case diagram, communication diagram and activity diagram are developed.

This coursework highlights the importance of system planning, requirement analysis and OOAD diagrams in developing a successful system. The Planning diagram like WBS and Gantt Chart helps in project management. It helps to divide work between teams and helps with tracking work progress as well. This coursework gives hand-on experience of developing and designing well-mannered structure to ensure proper system development.

2. Planning Diagrams

2.1 WBS(Work Breakdown Structure)

A WBS (Work Breakdown Structure) is a project handling method that breaks down the projects into multiple smaller, more manageable tasks and group of tasks. It is a visual diagram that breaks down the entire project deliverables to make it easier to plan, organize, and track progress. (Adobe Experience Cloud Team, 2023)

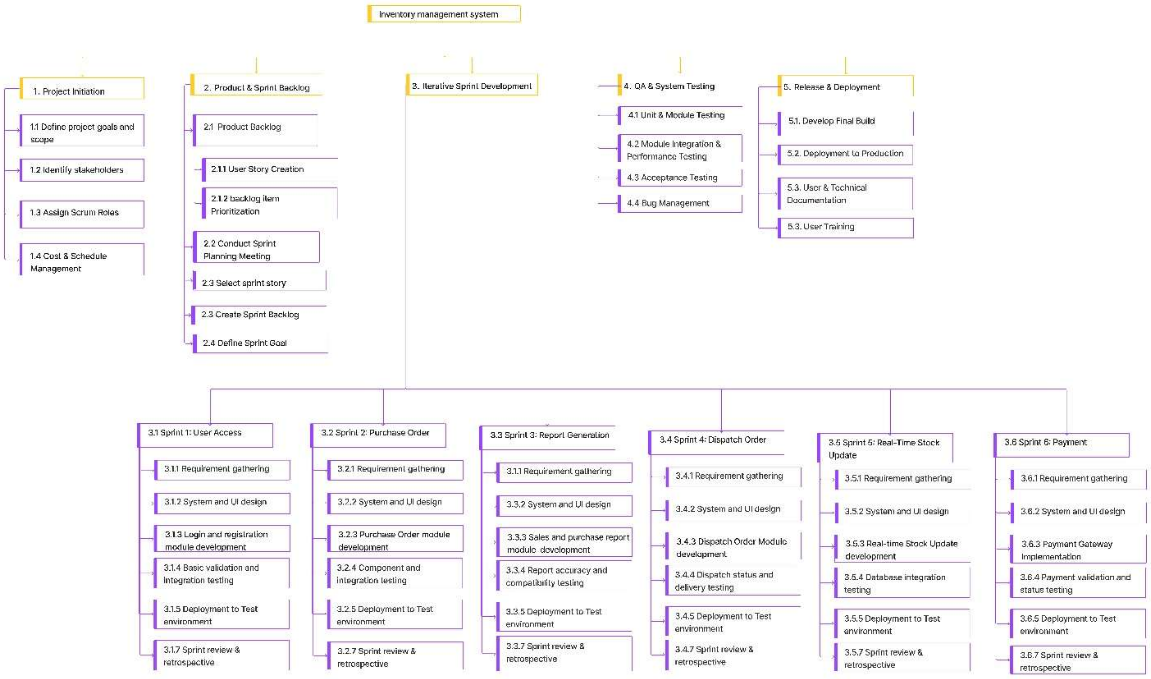
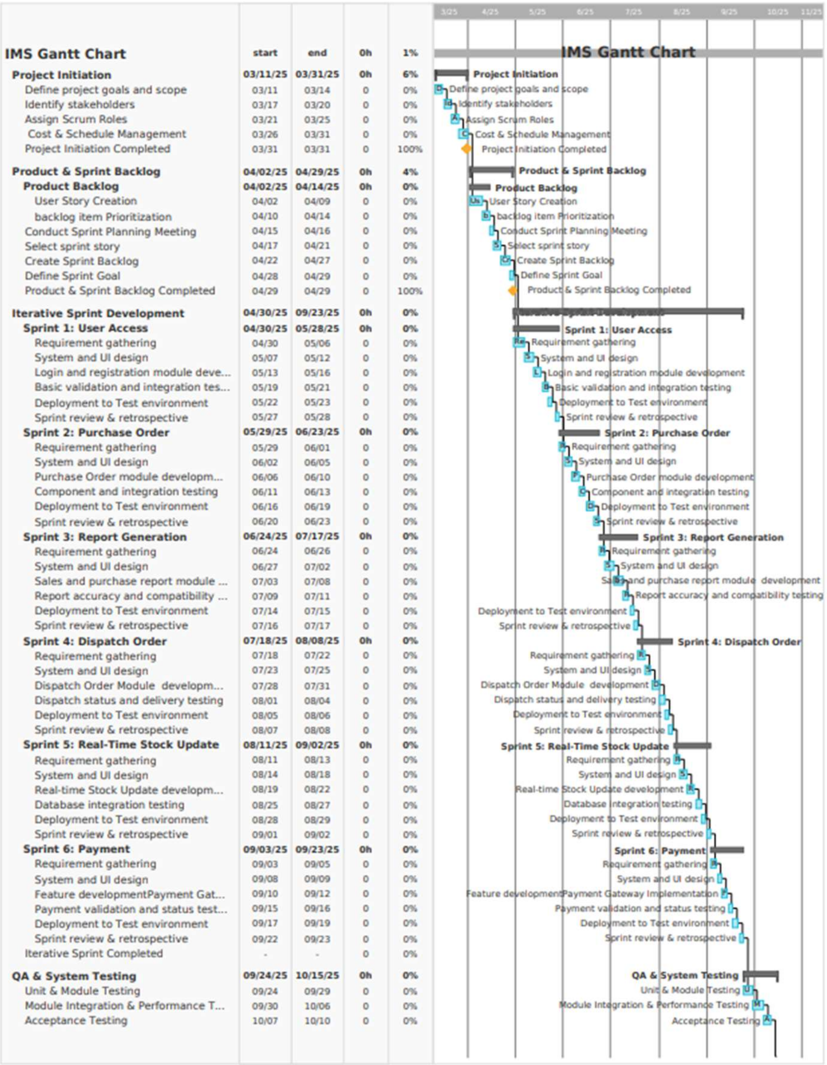

Inventory management system
1. Project Initiation
1.1 Define project goals and scope
1.2 Identify stakeholders
1.3 Assign Scrum Roles
1.4 Cost & Schedule Management
2. Product & Sprint Backlog
2.1 Product Backlog
2.1.1 User Story Creation
2.1.2 backlog item Prioritization
2.2 Conduct Sprint Planning Meeting
2.3 Select sprint story
2.3 Create Sprint Backlog
2.4 Define Sprint Goal
3. Iterative Sprint Development
4. QA & System Testing
4.1 Unit & Module Testing
4.2 Module Integration & Performance Testing
4.3 Acceptance Testing
4.4 Bug Management
5. Release & Deployment
5.1. Develop Final Build
5.2. Deployment to Production
5.3. User & Technical Documentation
5.3. User Training
3.1 Sprint 1: User Access
3.1.1 Requirement gathering
3.1.2 System and UI design
3.1.3 Login and registration module development
3.1.4 Basic validation and Integration testing
3.1.5 Deployment to Test environment
3.1.7 Sprint review & retrospective
3.2 Sprint 2: Purchase Order
3.2.1 Requirement gathering
3.2.2 System and UI design
3.2.3 Purchase Order module development
3.2.4 Component and integration testing
3.2.5 Deployment to Test environment
3.2.7 Sprint review & retrospective
3.3 Sprint 3: Report Generation
3.1.1 Requirement gathering
3.3.2 System and UI design
3.3.3 Sales and purchase report module development
3.3.4 Report accuracy and compatibility testing
3.3.5 Deployment to Test environment
3.3.7 Sprint review & retrospective
3.4 Sprint 4: Dispatch Order
3.4.1 Requirement gathering
3.4.2 System and UI design
3.4.3 Dispatch Order Module development
3.4.4 Dispatch status and delivery testing
3.4.5 Deployment to Test environment
3.4.7 Sprint review & retrospective
3.5 Sprint 5: Real-Time Stock Update
3.5.1 Requirement gathering
3.5.2 System and UI design
3.5.3 Real-time Stock Update development
3.5.4 Database integration testing
3.5.5 Deployment to Test environment
3.5.7 Sprint review & retrospective
3.6 Sprint 6: Payment
3.6.1 Requirement gathering
3.6.2 System and UI design
3.6.3 Payment Gateway Implementation
3.6.4 Payment validation and status testing
3.6.5 Deployment to Test environment
3.6.7 Sprint review & retrospective

Figure 1: WBS

In the above WBS diagram, there are five main activities that are Project Initiation, product & sprint backlog, QA & system testing and finally release & deployment. The above work breakdown structure follows the agile scrum method so there are multiple iterative sprints from sprint1 to sprint6. The sprint consists of requirement gathering, system and UI design, feature development, unit testing, sprint review and sprint retrospective.

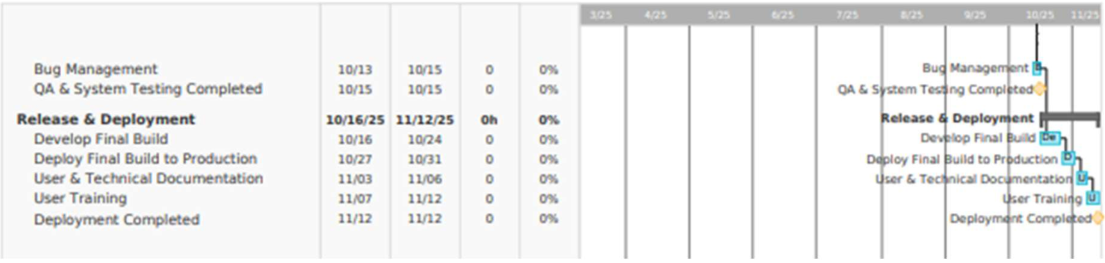
2.2 Gantt Chart

A Gantt Chart is a visual representation of project schedule. It is a graphical chart/diagram which shows the start and end dates of the project's tasks, dependencies of tasks, milestone, etc. The bars inside the tasks represent their timeline which consists of start date , end date, task sequences and duration. (Grant, 2024)



IMS Gantt Chart	start	end	0h	1%
Project Initiation	**03/11/25**	**03/31/25**	**0h**	**6%**
Define project goals and scope	03/11	03/14	0	0%
Identify stakeholders	03/17	03/20	0	0%
Assign Scrum Roles	03/21	03/25	0	0%
Cost & Schedule Management	03/26	03/31	0	0%
Project Initiation Completed	03/31	03/31	0	100%
Product & Sprint Backlog	**04/02/25**	**04/29/25**	**0h**	**4%**
Product Backlog	**04/02/25**	**04/14/25**	**0h**	**0%**
User Story Creation	04/02	04/09	0	0%
backlog item Prioritization	04/10	04/14	0	0%
Conduct Sprint Planning Meeting	04/15	04/16	0	0%
Select sprint story	04/17	04/21	0	0%
Create Sprint Backlog	04/22	04/27	0	0%
Define Sprint Goal	04/28	04/29	0	0%
Product & Sprint Backlog Completed	04/29	04/29	0	100%
Iterative Sprint Development	**04/30/25**	**09/23/25**	**0h**	**0%**
Sprint 1: User Access	**04/30/25**	**05/28/25**	**0h**	**0%**
Requirement gathering	04/30	05/06	0	0%
System and UI design	05/07	05/12	0	0%
Login and registration module deve...	05/13	05/16	0	0%
Basic validation and integration tes...	05/19	05/21	0	0%
Deployment to Test environment	05/22	05/23	0	0%
Sprint review & retrospective	05/27	05/28	0	0%
Sprint 2: Purchase Order	**05/29/25**	**06/23/25**	**0h**	**0%**
Requirement gathering	05/29	06/01	0	0%
System and UI design	06/02	06/05	0	0%
Purchase Order module developm...	06/06	06/10	0	0%
Component and integration testing	06/11	06/13	0	0%
Deployment to Test environment	06/16	06/19	0	0%
Sprint review & retrospective	06/20	06/23	0	0%
Sprint 3: Report Generation	**06/24/25**	**07/17/25**	**0h**	**0%**
Requirement gathering	06/24	06/26	0	0%
System and UI design	06/27	07/02	0	0%
Sales and purchase report module ...	07/03	07/08	0	0%
Report accuracy and compatibility ...	07/09	07/11	0	0%
Deployment to Test environment	07/14	07/15	0	0%
Sprint review & retrospective	07/16	07/17	0	0%
Sprint 4: Dispatch Order	**07/18/25**	**08/08/25**	**0h**	**0%**
Requirement gathering	07/18	07/22	0	0%
System and UI design	07/23	07/25	0	0%
Dispatch Order Module developm...	07/28	07/31	0	0%
Dispatch status and delivery testing	08/01	08/04	0	0%
Deployment to Test environment	08/05	08/06	0	0%
Sprint review & retrospective	08/07	08/08	0	0%
Sprint 5: Real-Time Stock Update	**08/11/25**	**09/02/25**	**0h**	**0%**
Requirement gathering	08/11	08/13	0	0%
System and UI design	08/14	08/18	0	0%
Real-time Stock Update developm...	08/19	08/22	0	0%
Database integration testing	08/25	08/27	0	0%
Deployment to Test environment	08/28	08/29	0	0%
Sprint review & retrospective	09/01	09/02	0	0%
Sprint 6: Payment	**09/03/25**	**09/23/25**	**0h**	**0%**
Requirement gathering	09/03	09/05	0	0%
System and UI design	09/08	09/09	0	0%
Feature developmentPayment Gat...	09/10	09/12	0	0%
Payment validation and status test...	09/15	09/16	0	0%
Deployment to Test environment	09/17	09/19	0	0%
Sprint review & retrospective	09/22	09/23	0	0%
Iterative Sprint Completed	-	-	0	0%
QA & System Testing	**09/24/25**	**10/15/25**	**0h**	**0%**
Unit & Module Testing	09/24	09/29	0	0%
Module Integration & Performance T...	09/30	10/06	0	0%
Acceptance Testing	10/07	10/10	0	0%

Figure 2: Gantt Chart



Bug Management	10/13	10/15	0	0%
QA & System Testing Completed	10/15	10/15	0	0%
Release & Deployment	**10/16/25**	**11/12/25**	**0h**	**0%**
Develop Final Build	10/16	10/24	0	0%
Deploy Final Build to Production	10/27	10/31	0	0%
User & Technical Documentation	11/03	11/06	0	0%
User Training	11/07	11/12	0	0%
Deployment Completed	11/12	11/12	0	0%

Figure 3: Gantt Chart 2

There above Gantt Chart shows the progress and deadline of the system development activities and tasks. The Gantt Chart is made by following agile scrum methodology. The Gantt chart shows name of multiple tasks and milestone on the left side of the diagram. The diagrams also show created date, end date and progress of all the tasks.

The start date and close date of the main activity/tasks are listed below.

Table 1: Tasks Start & End Date

Activity/Tasks Name	Start Date	End Date
Project Initiation	03/11/25	03/31/25
Product & Sprint Backlog	04/02/25	04/29/25
Iterative Sprint Development	04/30/25	09/23/25
QA & System Testing	09/24/25	10/15/25
Release & Deployment	10/16/25	11/12/25

3. OOAD Analysis

3.1 Use Case Diagram

The use case diagram is a visual representation of how the user interacts with the system. This is done by developer with the help of customer, to know the users and basic system features of the system. The helps developer to know how the users will interact with multiple features of the system. Use case is made considering the users perspective. (Wong, 2024)

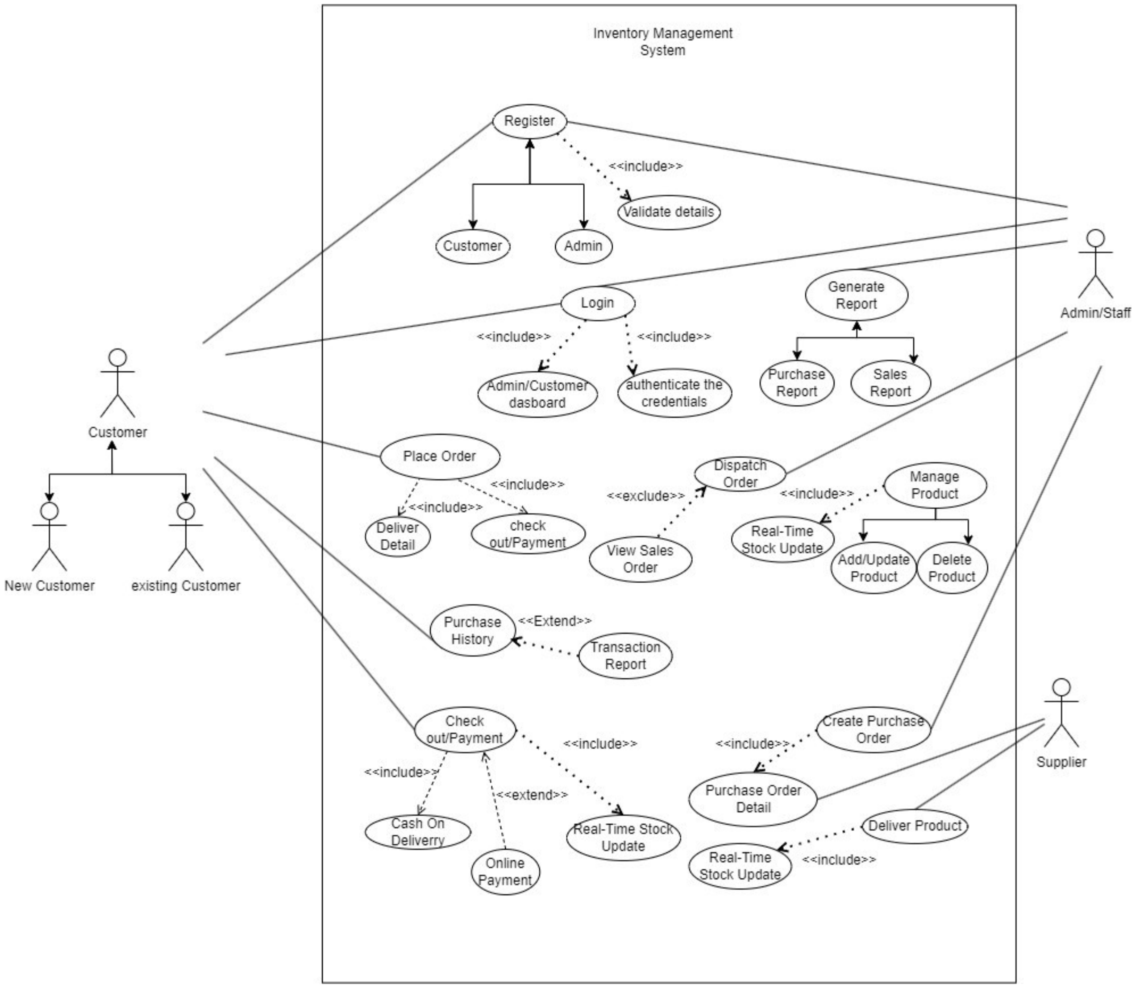


Figure 4: Use Case Diagram

3.2 High Level Description

1. Access users to system

- **Name:** Access users to system
- **Actors:** Admin, Customer
- **Description:** The users(Admin or Customers) need to register with credentials like username, password and role to create a new account. Then the user can use those registered credentials to login to the system. The system will redirect users to their respective dashboard according to the user’s role.

2. Create Purchase Order

- **Name:** Create Purchase Order
- **Actors:** Admin
- **Description:** Admin creates purchase order consisting of product details to restock the products. The purchase order details will be sent to the selected suppliers .

3. Generate Report

- **Name:** Generate Report
- **Actors:** Admin, Customer
- **Description:** This feature allows admin and customers to generate reports like purchase report, sales report and purchase report.

4. Dispatch Order

- **Name:** Dispatch Order

- **Actors:** Admin, Staff
- **Description:** After the payment and delivery address is confirmed the admin will monitor the Dispatch order process and staff will deliver the order to the designated address.

5. Real Time Stock Update

- **Name:** Real Time Stock Update
- **Actors:** Admin
- **Description:** When purchase order and sales order are placed, the system will update the stock accordingly. The admin can monitor and manage this process through an admin dashboard as well.

6. Payment

- **Name:** Payment
- **Actors:** Customer
- **Description:** After selecting items and while checking out, customers can choose between different payment methos i.e. COD and Online Payment. The system processes the payment and updates the order status and sales order.

3.3 Expanded Level Description

1. Payment Expanded Level Description

- **Name:** Payment
- **Actors:** Customer
- **Purpose:** To give customers options to make payments for their purchase
- **Overview:** After selecting items and while checking out, customers can choose between different payment methos i.e. COD and Online Payment. The system processes the payment and updates the order status and sales order.
- **Type:** Primary, Essential
- **Action Steps:**

Table 2: Payment Expanded Level Description

Actor Action	**System Response**
1. Customer selects items and proceeds to checkout.	2. The system calculates the total price of the selected items.
3. Customers procced to make payment.	4. System will display the price and payment option (COD or Online payment).
5. Customers choose the payment option(COD or Online Payment)	6. If online payment is selected the system redirects user to payment gateway.
7. User will enter the payment credentials in the payment portal.	8. The system will validate the payment. If successful displays message “payment successful ” to customer.

- **Alternative Course of Action:**

- **Line-3:** If online payment options fail, the customer can choose Cash on Delivery option.
- **Line-2:** If customers choose Cash on Delivery the order is placed successfully, and the payment is marked as pending.

2. Access User to Systems Expanded Level Description

- **Name:** Access User to Systems
- **Actors:** Customer, Admin
- **Purpose:** Allow Admins and customers to register and login into the system
- **Overview:** The users(Admin and Customers) first must register as customers or admin to create an account. Then the users can login with appropriate login credentials to access their individual dashboard or portal.
- **Type:** Primary, Essential
- **Action Steps:**

Table 3: Access User System Expanded Level Description

Actor Action	**System Response**
1. User will click on account setting.	2. The system will redirect user to login page and ask user to login into the account.
3. If the user does not have an account, the user will click on the signup/register button.	4. The system will redirect user to the registration page.
5. Users will register with the required details like username, password and role.	6. The system validates the details and checks if the username is available.

7. If the username is available, the user submits the registration form.	8. The system will create a new account and notify user.
9. After registering, the user will login to the system with the registered username and password.	10. The system authenticates the user entered credential. If the entered credential is valid the system will grant the user with Admin/Customer dashboard.

- **Alternative Course of Action:**
  - **Line-2:** If an account with entered username already exists, the system will prompt user to enter different username.
  - **Line-2:** If entered account details are invalid , the system will prompt users to register again with proper details.
  - **Line-**6: After the user enters the login credentials , if the credential is invalid the system will prompt the user to reenter the credentials.
  - **Line-6:** If the entered username does not exist, the system will prompt user to register an account.

3.4 Communication Diagram

The communication diagram is also known as collaboration diagram. Communication diagram is the graphical representation of how the objects interact with each other. It has elements like objects, actors, links and messages. This diagram is suitable for showcasing the simple interaction between few numbers of objects (Lewis, 2023)

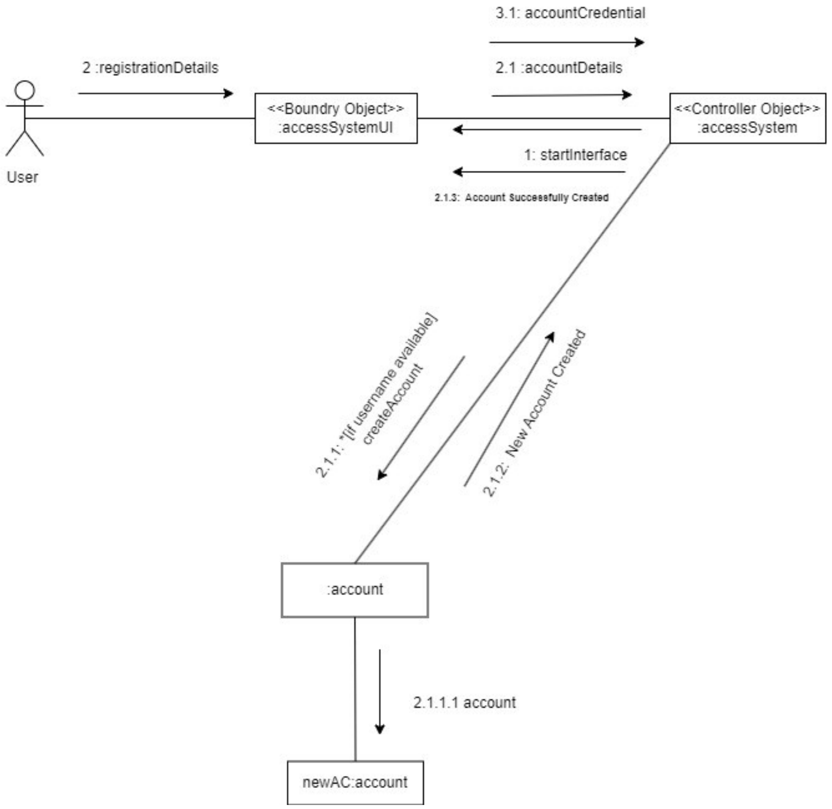


Figure 5: Communication Diagram Part 1

First, the controller starts the interface. Then the user will enter the registration details by filling in the registration form. The boundary object will then forward these details to the controller object. The controller object will then send the details to validate the account object. The new account will create a new account for the user if the details are unique(username, email). The account object will send a message to the controller object and the controller will forward that message to the boundary object.

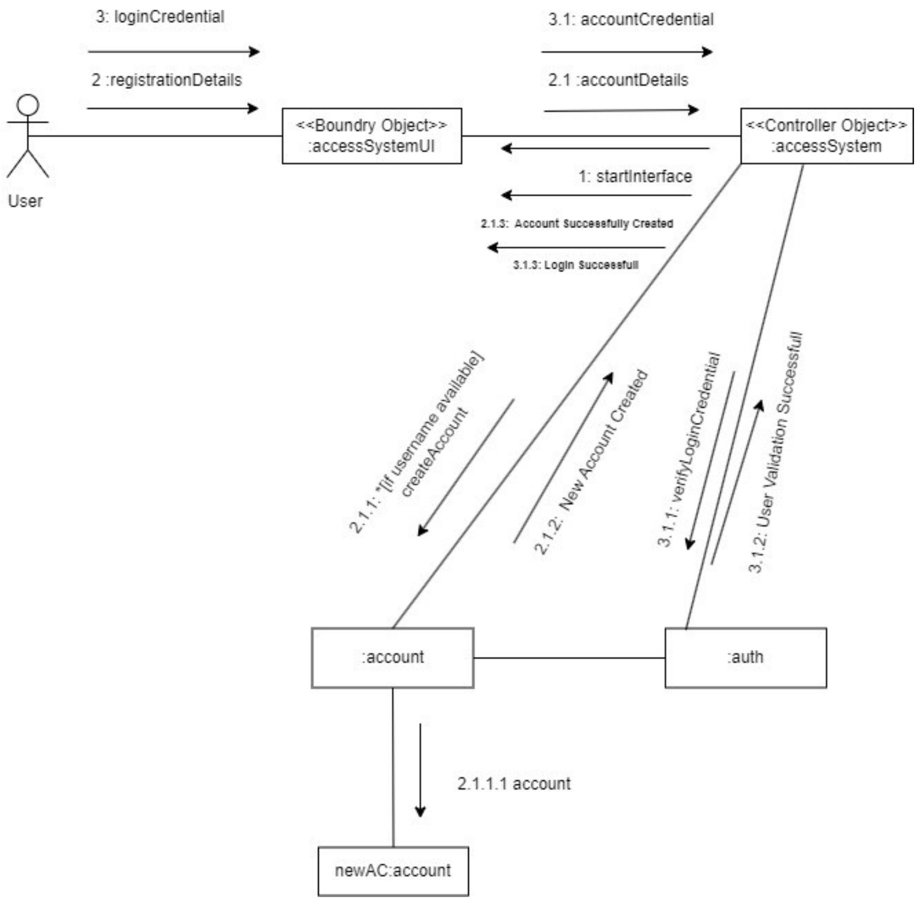


Figure 6: Communication Diagram Part 2

After creating the account, the user will then proceed to log into the system by providing the login credential. The boundary object will forward the credentials to the controller and the controller forward it to the auth object to authenticate the login credentials. If the login credentials are valid the auth object will send a message to the controller object.

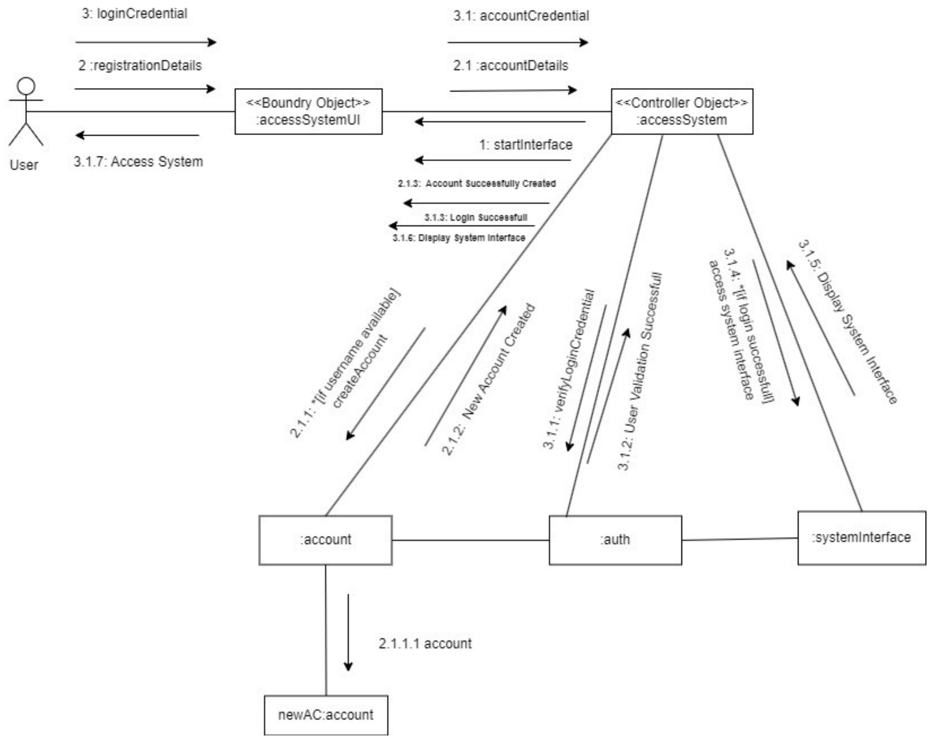


Figure 7: Full Communication Diagram

Now, the controller will send the login successful message to the boundary object and ask access of system interface from the systemInterface object. The system interface object will then give access to the system to the user.

3.5 Activity Diagram

Activity diagram is a Unified Modeling Language flowchart, that visually displays the sequential flow of the process from one activity to another. It shows how the system function will work in a sequential manner. The diagram consists of various elements like initial node, activity state, action, control flow, decision node, merge node, fork node, merging, comment and final node. (mindmanager, 2025)

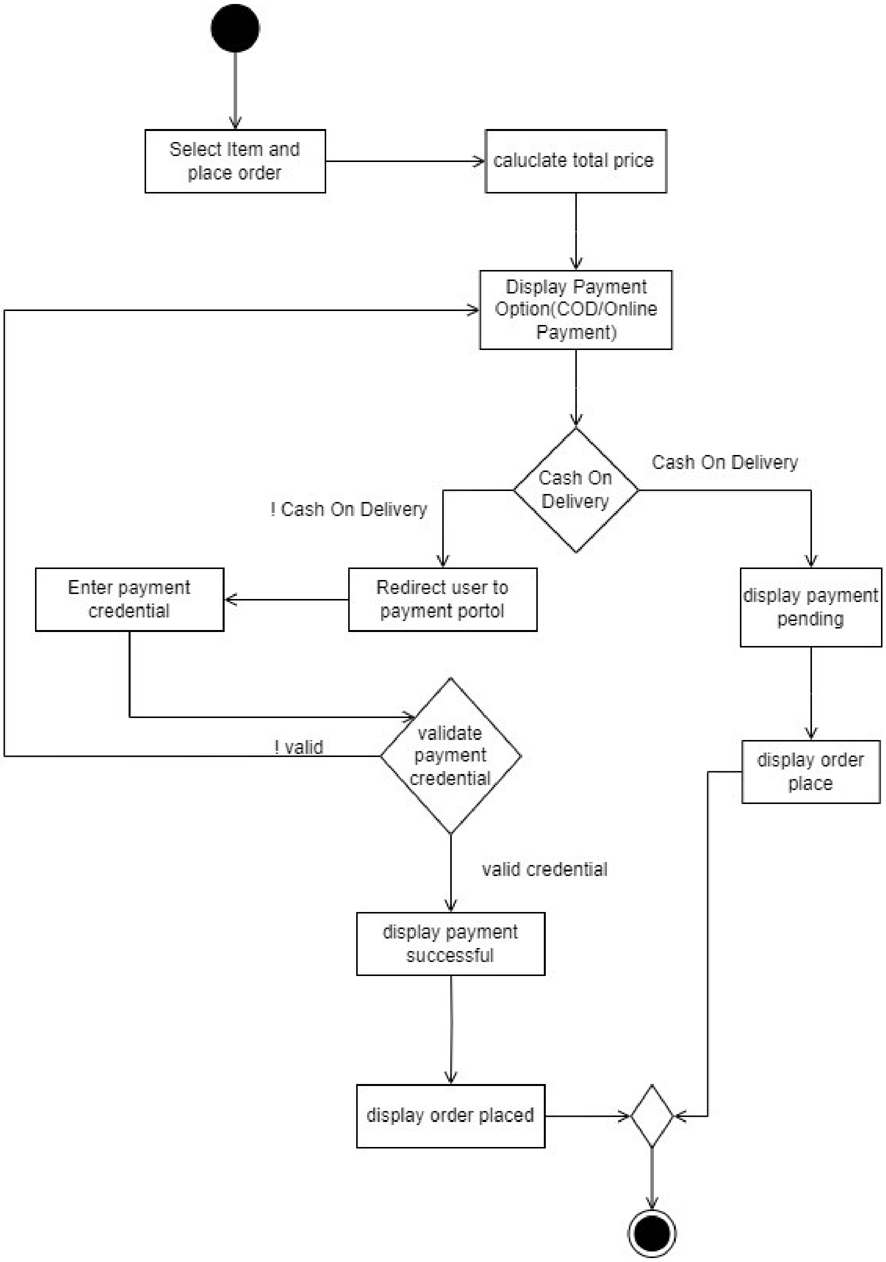


Figure 8: Activity Diagram

The above diagram is the activity diagram for the payment function. The user will first select the item and place order. The total price of the order will be calculated, and the system will then forward the user to a payment page where the user can select any payment method. If the user selects the Cash on Delivery option the payment will be marked as pending and the order is placed ending the process. But, if the user selected online payment the system will redirect user to the online payment portal where user will enter the account credentials. If the account credential is valid the system will display payment successfully and place order ending the process. If the account credential is not valid then the system will redirect user to the payment option page.

4. Class Diagram and Domain Class

4.1 Domain Class

Table 4: Domain Class Table

Use Case	Domain Classes
Admin	Manage User Accounts Manage Products Add Product Update Product Delete Product Manage Delivery Dispatch Delivery
Customer	Manage User Accounts View Product Details Manage Cart Add Product to Cart Checkout Place Order Make Payment
Cart	View Product Details Manage Cart Add Product to Cart Update Cart Quantity Checkout

Order	Checkout Place Order View Order Details Confirm Order Manage Delivery Make Payment
Order Detail	Calculate Order Price Update Order Item
Product	Manage Products Add Products Update Product Delete Product View Product Details Manage Cart Add product to Cart Calculate Order Price
Supplier	Manage Products Add Product
Delivery	Manage Delivery Track Delivery Dispatch Delivery Update Delivery Address

Payment	Making Payment Process Payment

4.2 Class diagram

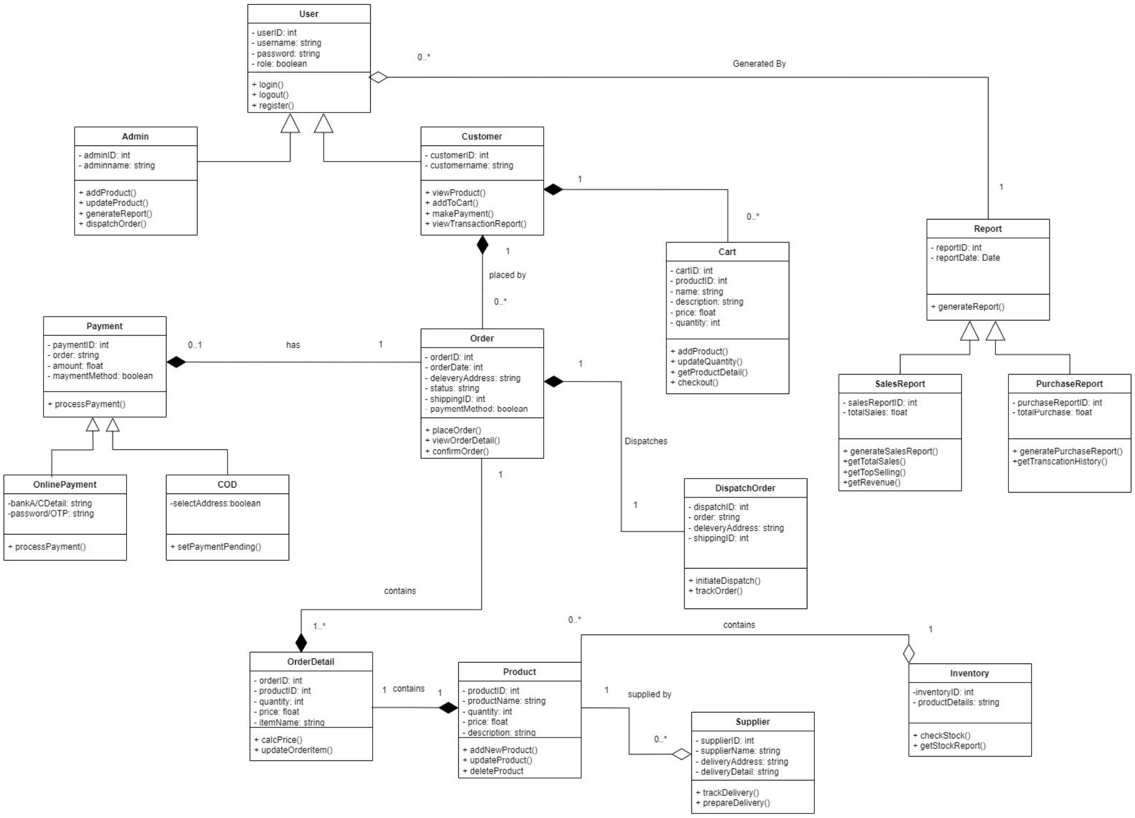


Figure 9: Class Diagram

Class Diagram is a UML diagram that shows all the classes of the system, their relationship, cardinality, attributes and operations. It is used as a general conceptual model of system coding. Class diagram has seven types of cardinalities, they are :

Table 5: Cardinal Types

Cardinal Name	Cardinal Number
Exactly One	1
Zero or More	0..*
One or More	1..*

Zero or One	0..1
Specified Range	2..4
Multiple Disjoint	2,4…6,8
One or Other	4,6

Relationship among classes:

- A User has two child class Admin and Customer.
- A customer generates zero or more Cart items.
- An Order contains one or more Order details.
- An Order has zero or one payment.
- A Dispatch Order dispatches one order.
- An Order details contains Product.
- A Product is supplied by zero or more suppliers.
- An inventory contains one or more products.
- A User generates zero or more Report.
- A Report has two child class Sales Report and Purchase Report.
- Payment has two child class Online Payment And COD.

5. Further Development

To move forward with the development of the Inventory Management System (IMS), a structured approach is essential. I propose using the Agile Scrum methodology, for its iterative and adaptive nature, which aligns well with the evolving requirements of the system. Agile Scrum is used when there is need of continuous new adaptation of changes in system. I have used agile scrum to develop the WBS (Work Breakdown Structure) and Gantt Chart of the Inventory Management System. It is one of the trending methodologies used to complete the complex software development. This software development methodology is needed because of IMS's changing and evolving requirements.

5.1. Architectural Choice

IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” To develop software, we need to choose an architecture design. The main types of architecture design choice are listed below. (geeksforgeeks, 2025)

- Monolithic Architecture
- Client-server Architecture
- Layered Architecture
- Peer-to-Peer Architecture
- Blockchain Architecture
- Serverless Architecture

I will be using Layered Architecture, since this layered Architecture is aligned with the Agile Scrum method where the development works are done in layered and modular way. This architecture organizes the system into different layers:

- **Presentation Layer:** Handles the user interface (UI) and user interactions.
- **Application Layer:** Contains the application functional logic and coordinates interactions between the UI and the domain layer.
- **Domain Layer:** Implements the core business logic and data management.
- **Infrastructure Layer:** Provides an interface for external system like databases.

The advantages of choosing this architecture are:

- **Modularity:** The whole software functions are divided into multiple modules, which makes the software easy to develop and manage.
- **Scalability:** It is easier to scale the software since all the different modules can be scaled independently without affecting others.

- **Reusability:** The different layers of software can be used in different software, which makes the software code reusable. (systemdesignschool, 2025)

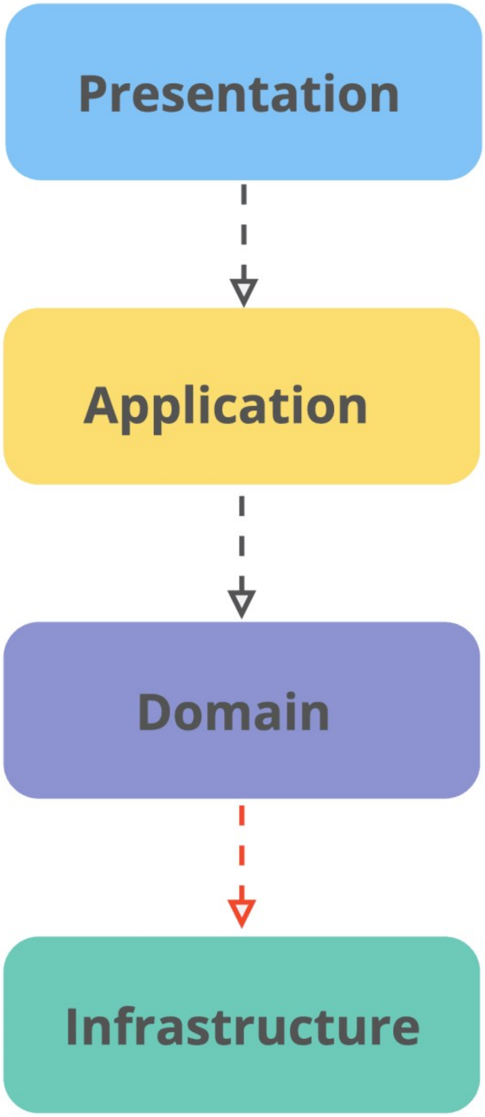


Figure 10: Layer Architecture

5.2. Design Pattern

Design Pattern are the ready-made customizable pattern or blueprint that can be used to solve the commonly occurring problems in software development. The main types of design patterns are explained below.

- **Creational patterns:** It is a design pattern which provides us with an object-based mechanism which helps in increasing the flexibility and reusability of the code. The creational pattern consists of many methods like Factory method and Singleton method.

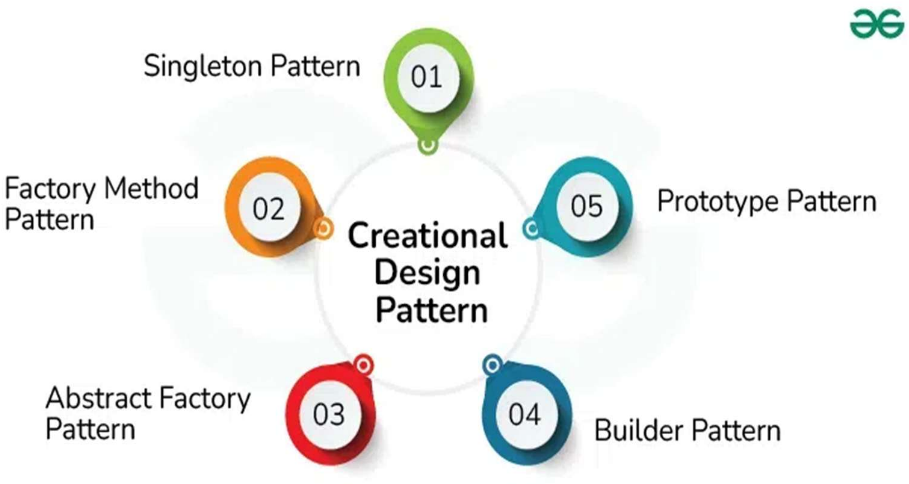


Figure 11: Creational Pattern

- **Structural patterns:** It is a type of design pattern which gathers the objects and classes to form a large structure. The structure is flexible and well organized. The method used in this pattern are adapter, bridge, composite, decoder, facade, flyweight and proxy.

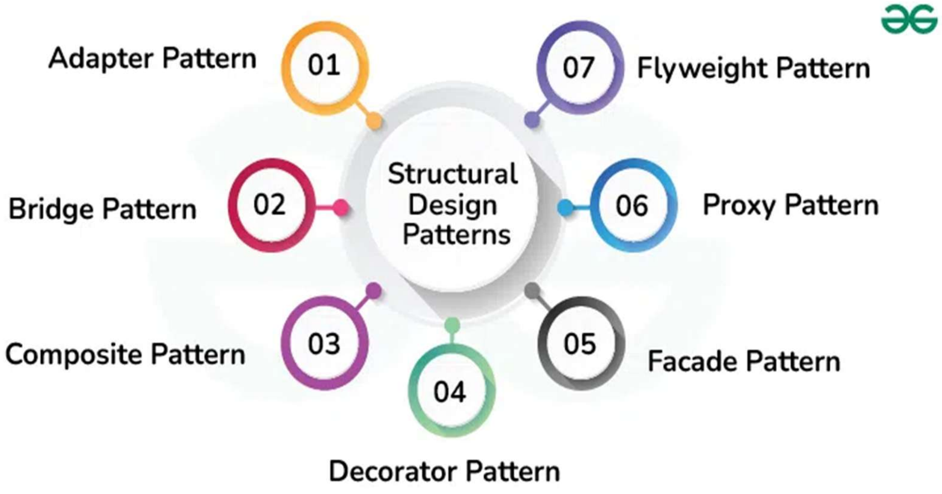


Figure 12: Structural Pattern

- **Behavioral patterns:** It is a type of design pattern that emphasizes the way objects communicate and assign responsibilities between them. The methods of behavioral patterns are command, chain of responsibility, memento, mediator, iterator, observer, strategy, state, template and visitor.

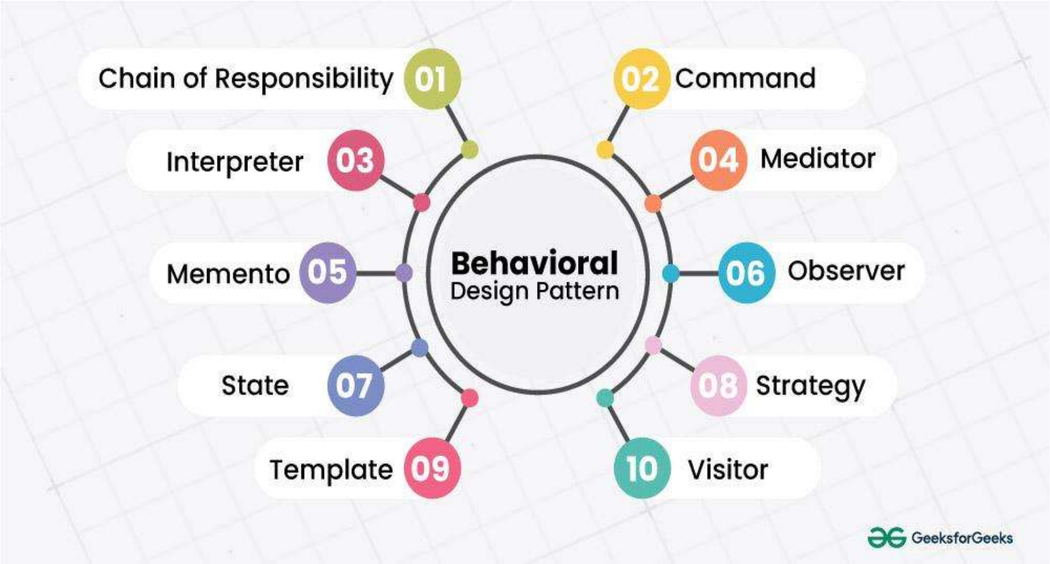


Figure 13: Behavioral Pattern

5.3. Development Plan

The tools/resources/programming language used in the development of inventory management system are:

- **Programming Language:** NodeJS, JavaScript for backend and frontend
- **Database:** MongoDB, Firebase
- **IDE:** Visual Studio
- **Version Control**: Git
- **Testing Frameworks:** Jest/Mocha

The priority order of features is listed below:

Must Have	Should Have	Could Have
Login & Register Purchase Order Creation Dispatch Order	Real-time stock update Payment Gateway Generating sales and purchase Report Role based account	Supplier Portal Sales and Trend Analytics AI assistance

5.4. Testing Plan

There are two types of testing in software development, they are functional and non-functional testing.

1. Functional Testing

The functional testing is the testing done to check if the system function is working as intended. This test focuses mainly on the applications features and what the system does.

Types:

- **Unit Testing:** Testing individual components of the system
- **Integration Testing:** Testing interactions between different modules.
- **System Testing:** Testing the entire system requirements and functions.
- **UI Testing:** Testing the system interface or user experience with the system.
-

2. Non-functional Testing

The non-functional testing verifies the way the system does its work. This test evaluates the system performance, security, scalability, reliability, usability, etc.

Types:

- **Performance Testing:** Testing stability and how fast the system does its tasks.
- **Load Testing:** Testing how the system work on the expected user load.
- **Stress Testing:** Testing how the system function on the beyond expected user's load.
- **Usability Testing:** Testing how easy or hard it is to interact and use the features of the system.
-

3. Box Testing

The three main types of box testing are explained below.

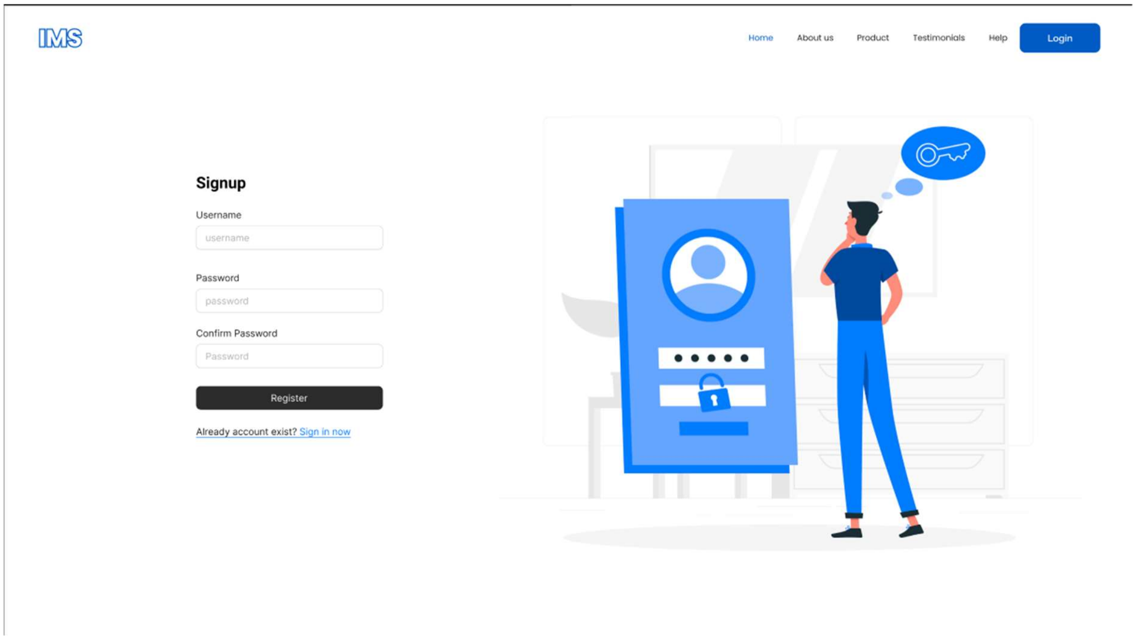
- **White Box Testing:** The testing where the tester has knowledge and access to the internal code. In this test, the tester evaluates the modules, codes and their functions.
- **Black Box Testing:** The testing where the tester has no knowledge and access to the internal code. In this test, the tester evaluates the output given by the system after giving the output. Example: testing login/register with credentials, testing adding to cart function, etc.
- **Grey Box Testing:** The testing where the tester has partial knowledge of the code. It is a combination of both white box and black box testing. Example: API testing.

5.5. Maintenance Plan

- **Corrective Maintenance:** The maintenance where the bugs and error of the system are fixed.
- **Adaptive Maintenance:** The maintenance where the system is modified to facilitate the new requirement and changes.
- **Perfective Maintenance:** The maintenance, which is done to improve performance, usability, scalability, etc. of the system.
- **Preventive Maintenance:** The maintenance where the preventive measures are made to prevent future harm or problem.

6. Prototype

1. Register Account



Signup

Username

username

Password

password

Confirm Password

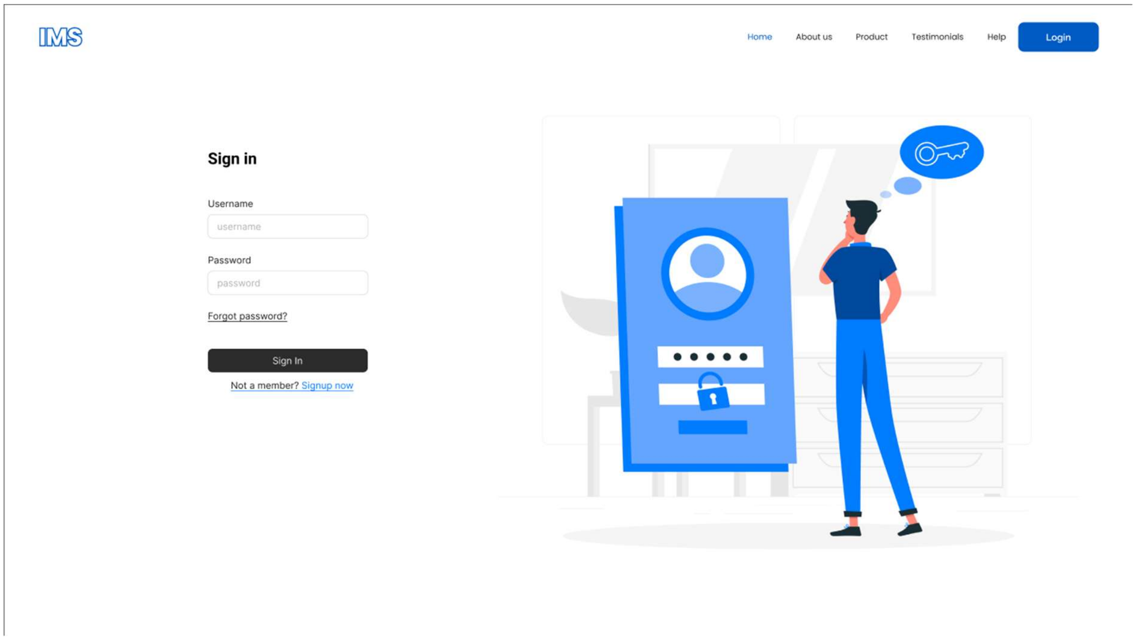
Password

Register

Already account exist? Sign in now

Figure 14: Register Account

2. Login Account



Sign in

Username

username

Password

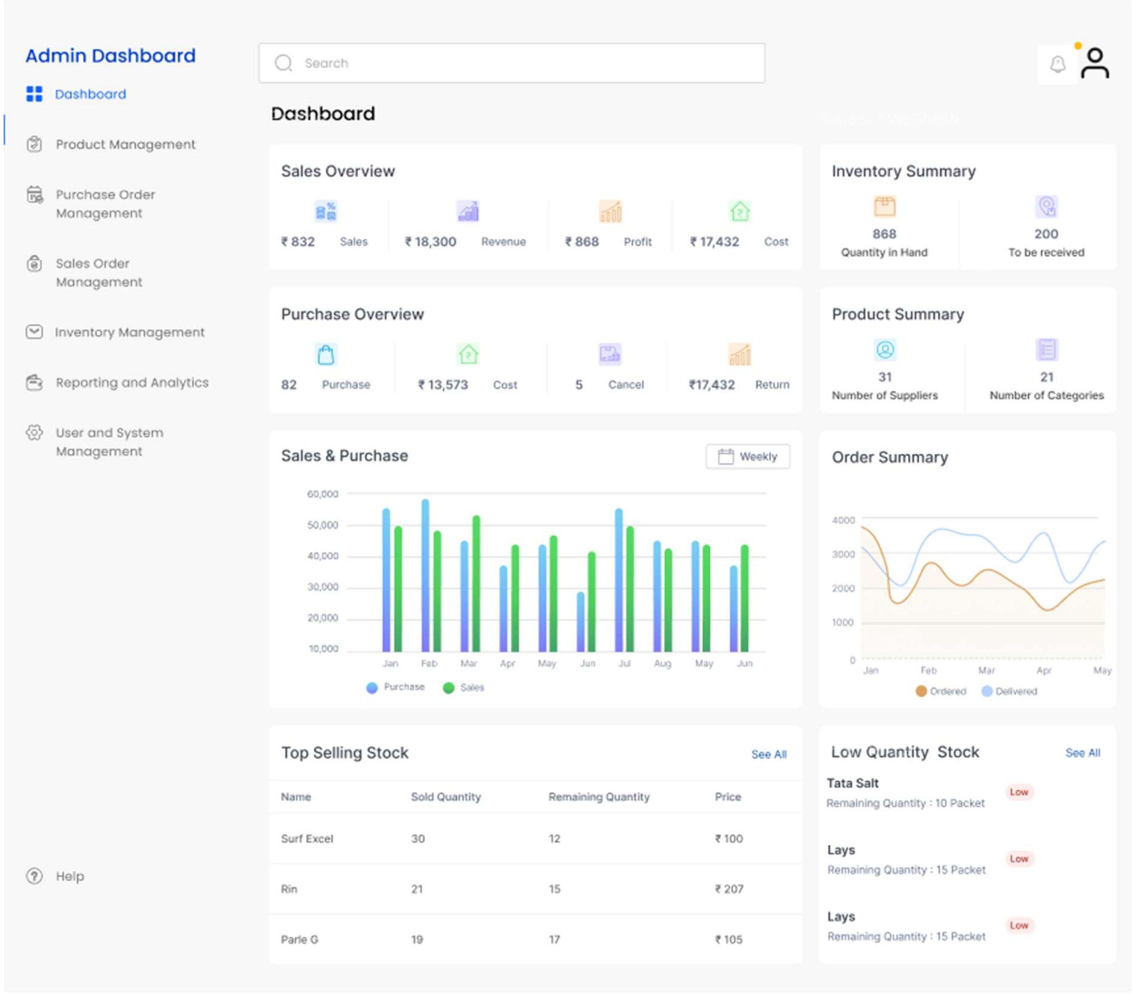
password

Forgot password?

Not a member? Signup now

Figure 15: Login Account

3. Admin Dashboard



Dashboard

Sales Overview

₹ 832 Sales | ₹ 18,300 Revenue | ₹ 868 Profit | ₹ 17,432 Cost

Inventory Summary

868 Quantity in Hand | 200 To be received

Purchase Overview

82 Purchase | ₹ 13,573 Cost | 5 Cancel | ₹17,432 Return

Product Summary

31 Number of Suppliers | 21 Number of Categories

Sales & Purchase

Order Summary

Top Selling Stock — See All

Name	Sold Quantity	Remaining Quantity	Price
Surf Excel	30	12	₹ 100
Rin	21	15	₹ 207
Parle G	19	17	₹ 105

Low Quantity Stock — See All

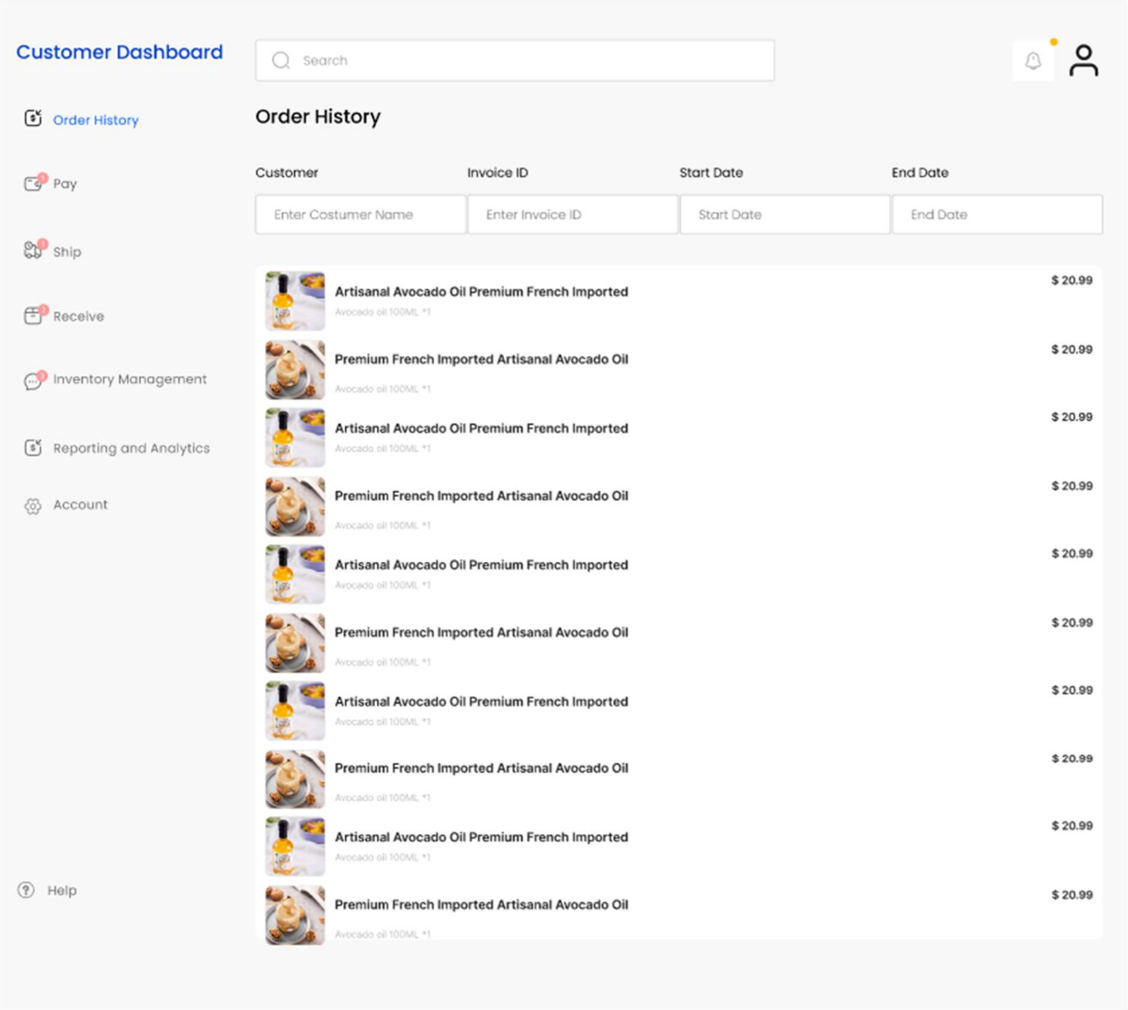
Tata Salt
Remaining Quantity : 10 Packet — Low

Lays
Remaining Quantity : 15 Packet — Low

Lays
Remaining Quantity : 15 Packet — Low

Figure 16: Admin Dashboard

4. Customer Dashboard



Order History

Customer	Invoice ID	Start Date	End Date
Enter Costumer Name	Enter Invoice ID	Start Date	End Date

Artisanal Avocado Oil Premium French Imported $ 20.99
Avocado oil 100ML *1

Premium French Imported Artisanal Avocado Oil $ 20.99
Avocado oil 100ML *1

Artisanal Avocado Oil Premium French Imported $ 20.99
Avocado oil 100ML *1

Premium French Imported Artisanal Avocado Oil $ 20.99
Avocado oil 100ML *1

Artisanal Avocado Oil Premium French Imported $ 20.99
Avocado oil 100ML *1

Premium French Imported Artisanal Avocado Oil $ 20.99
Avocado oil 100ML *1

Artisanal Avocado Oil Premium French Imported $ 20.99
Avocado oil 100ML *1

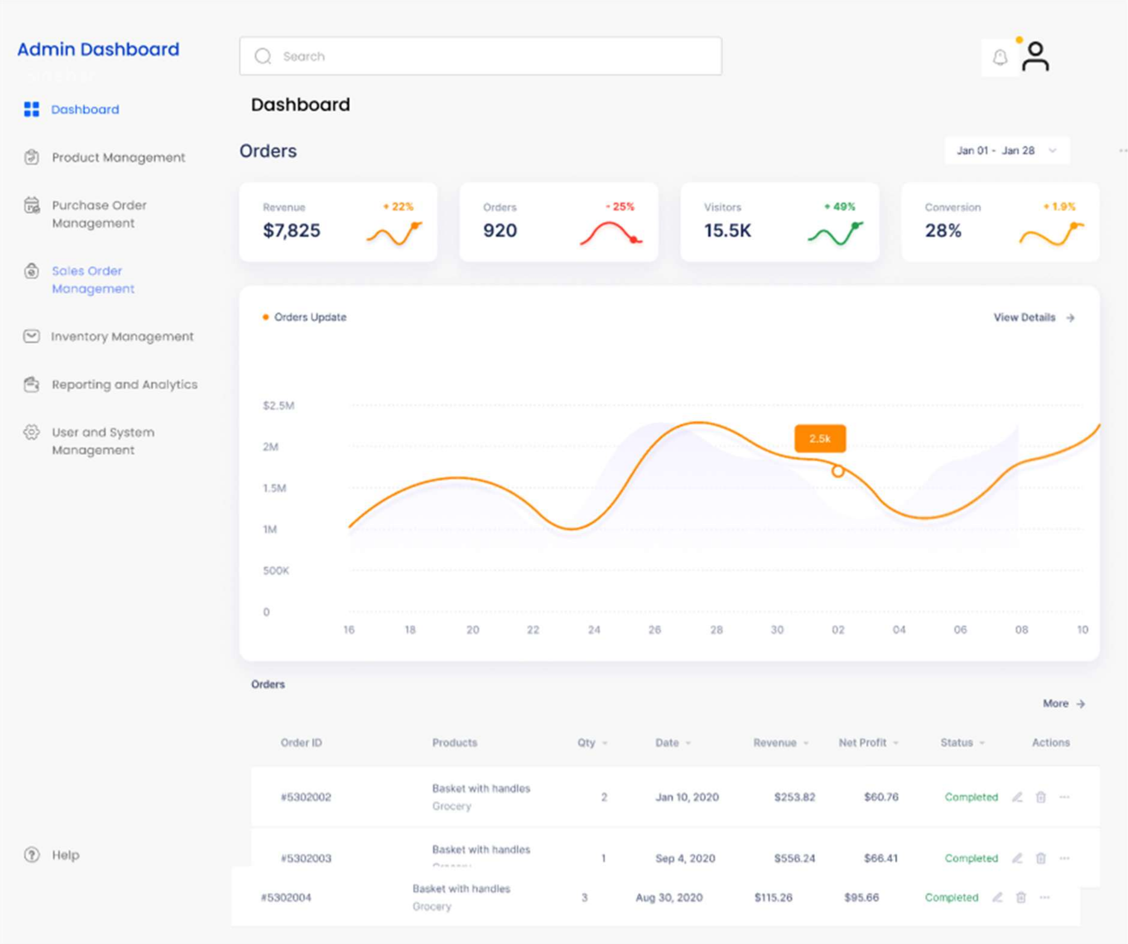
Premium French Imported Artisanal Avocado Oil $ 20.99
Avocado oil 100ML *1

Artisanal Avocado Oil Premium French Imported $ 20.99
Avocado oil 100ML *1

Premium French Imported Artisanal Avocado Oil $ 20.99
Avocado oil 100ML *1

Figure 17: Customer Dashboard

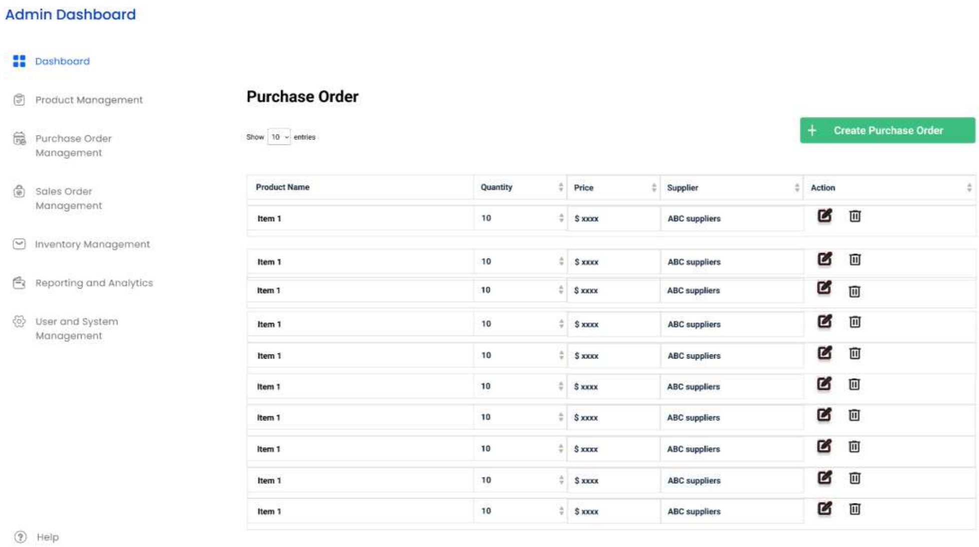
5. Sales Report



Order ID	Products	Qty	Date	Revenue	Net Profit	Status	Actions
#5302002	Basket with handles Grocery	2	Jan 10, 2020	$253.82	$60.76	Completed	
#5302003	Basket with handles	1	Sep 4, 2020	$556.24	$66.41	Completed	
#5302004	Basket with handles Grocery	3	Aug 30, 2020	$115.26	$95.66	Completed	

Figure 18: Sales Report

6. Purchase Order



Purchase Order

Product Name	Quantity	Price	Supplier	Action
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑
Item 1	10	$ xxxx	ABC suppliers	✎ 🗑

Figure 19: Purchase Order

7. Create Purchase Order

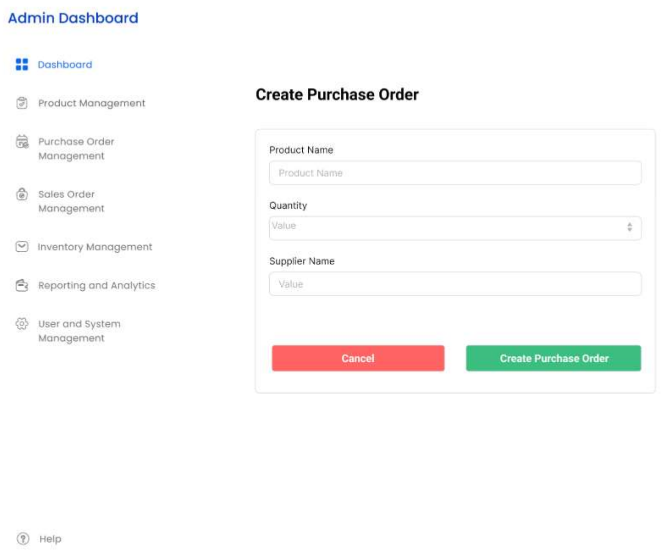
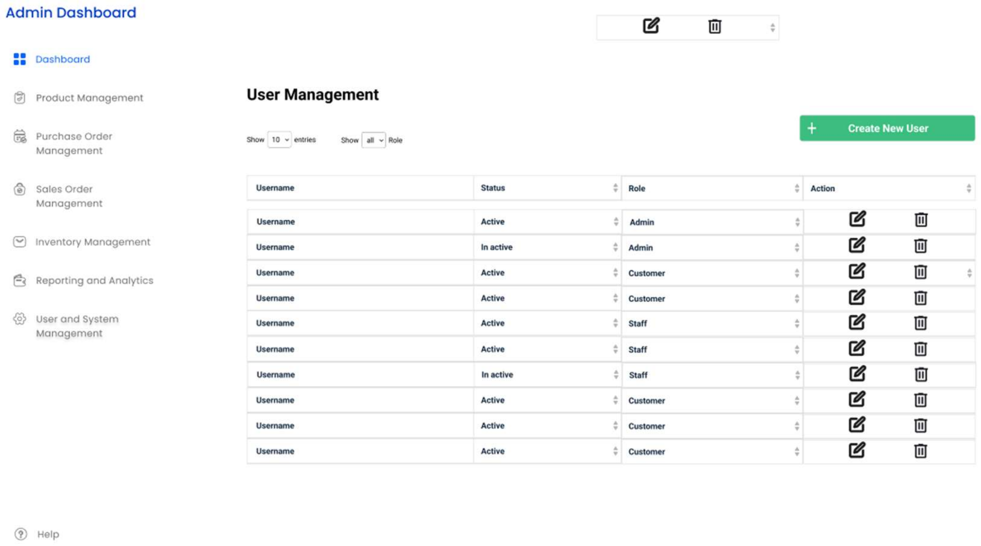


Figure 20: Create Purchase Orde

8. User Management



User Management

Show 10 entries Show all Role

Username	Status	Role	Action
Username	Active	Admin	✎ 🗑
Username	In active	Admin	✎ 🗑
Username	Active	Customer	✎ 🗑
Username	Active	Customer	✎ 🗑
Username	Active	Staff	✎ 🗑
Username	Active	Staff	✎ 🗑
Username	In active	Staff	✎ 🗑
Username	Active	Customer	✎ 🗑
Username	Active	Customer	✎ 🗑
Username	Active	Customer	✎ 🗑

Figure 21: User Management

9. Add User(admin/customer/staff)

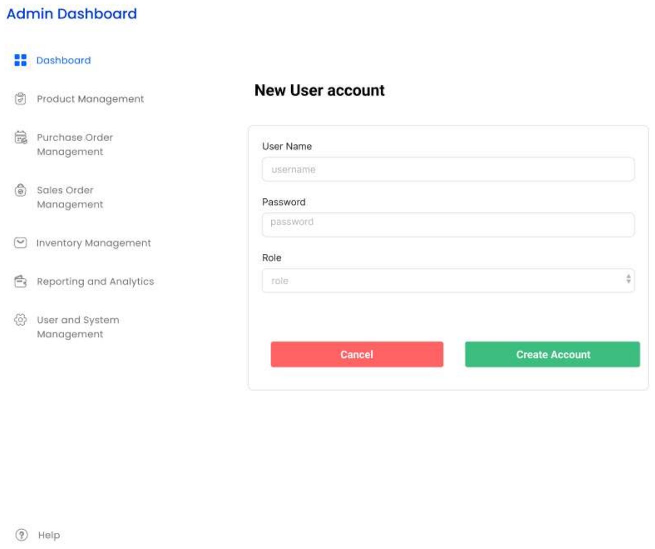
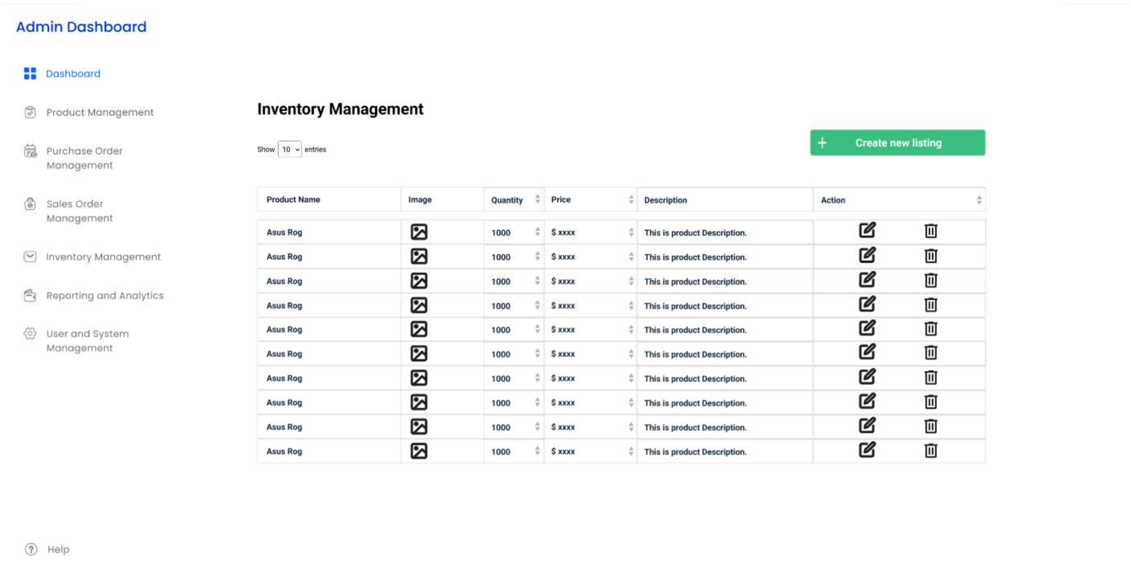


Figure 22:Add User

10. Inventory Management



Inventory Management

Product Name	Image	Quantity	Price	Description	Action
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑
Asus Rog	🖼	1000	$ xxxx	This is product Description.	✎ 🗑

Figure 23: Inventory Management

11. New Product Listing

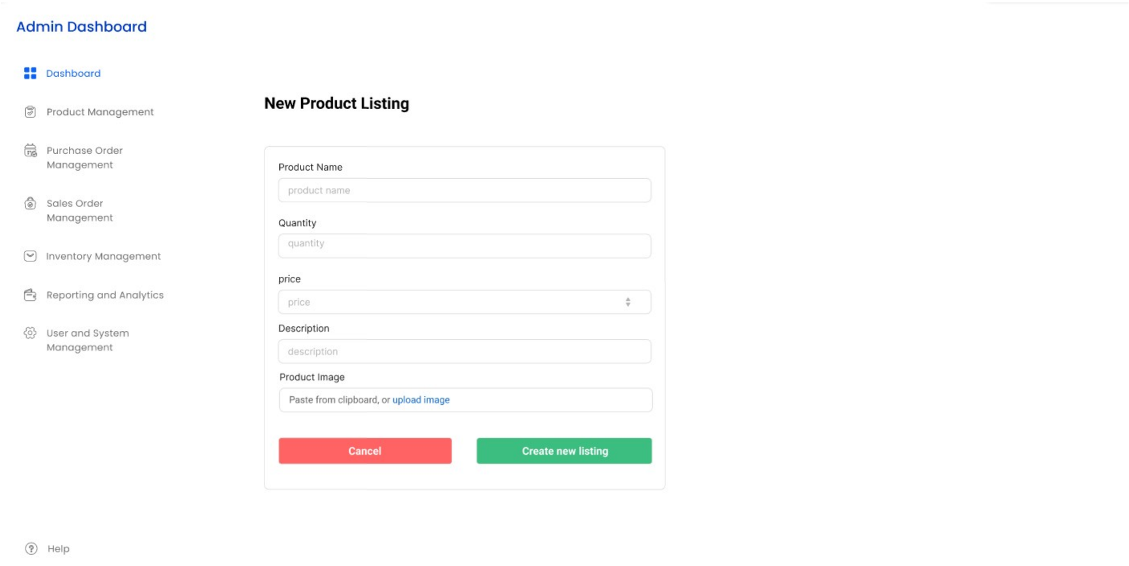


Figure 24: New Product Listing

12. Product Page

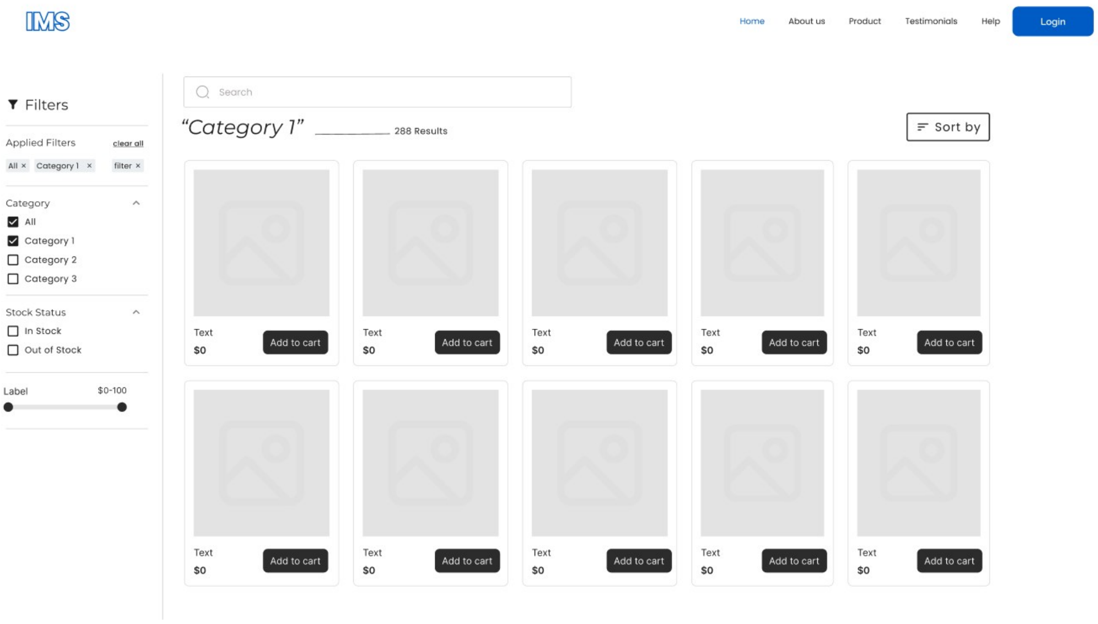
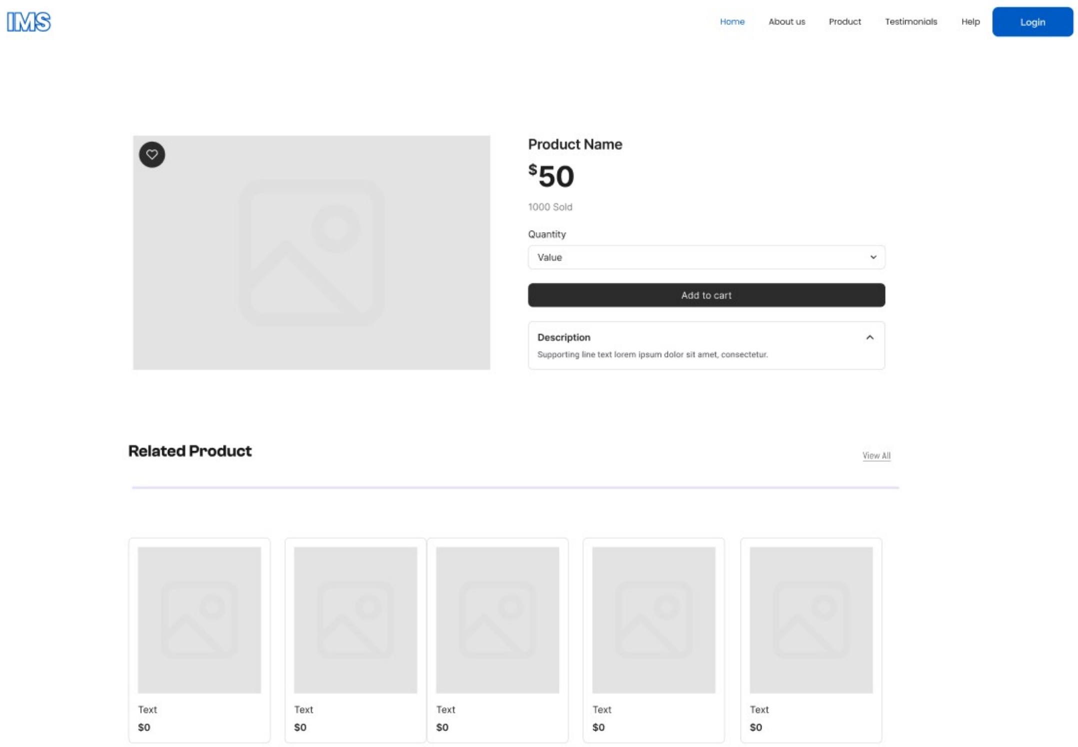


Figure 25: Product Page

13. Product Detail



Supporting line text lorem ipsum dolor sit amet, consectetur.

Figure 26: Product Detail

14. Cart

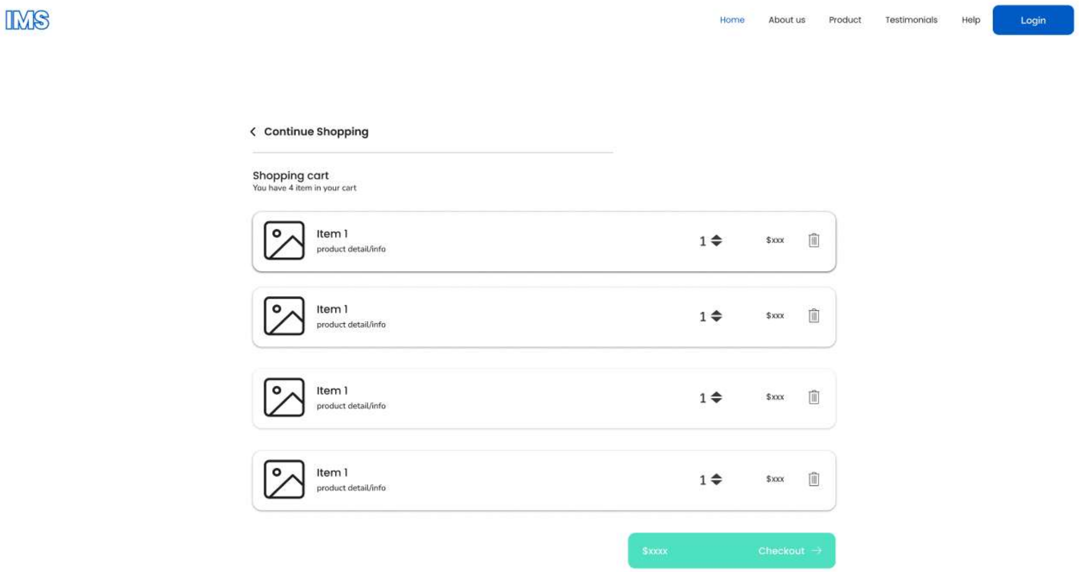
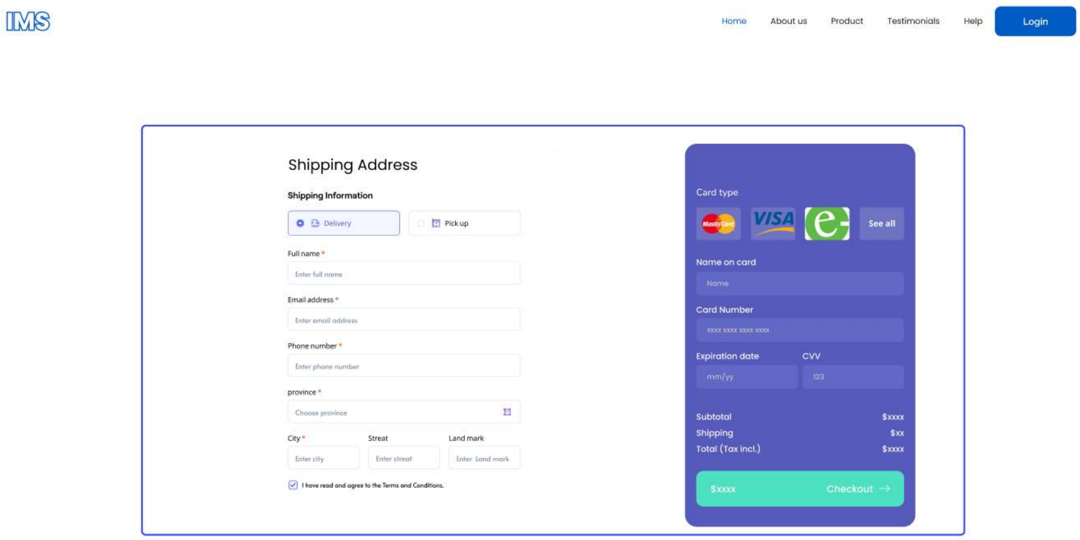


Figure 27: Cart

15. Payment Page



Shipping Address

Shipping Information

● Delivery ○ Pick up

Full name *
Enter full name

Email address *
Enter email address

Phone number *
Enter phone number

province *
Choose province

City * Enter city

Streat Enter streat

Land mark Enter Land mark

☑ I have read and agree to the Terms and Conditions.

Card type

Name on card
Name

Card Number
xxxx xxxx xxxx xxxx

Expiration date mm/yy

CVV 123

Subtotal	$xxxx
Shipping	$xx
Total (Tax incl.)	$xxxx

Figure 28: Payment Page

References

Adobe Experience Cloud Team. (2023, 04 24). *what-is-work-breakdown-structure*. Retrieved 04 01, 2025 from Adobe For Business: https://business.adobe.com/blog/basics/what-is-work-breakdown-structure

Ang, J. (2025, 02 17). *activity-diagram*. Retrieved 04 01, 2025 from venngage: https://venngage.com/blog/activity-diagram/

atlassian. (2024, 3 11). *The Workstream*. Retrieved 3 11, 2024 from atlassian: https://www.atlassian.com/work-management/project-management/work-breakdown-structure

geeksforgeeks. (2025, 01 03). *software engineering-architectural design*. From geeksforgeeks: https://www.geeksforgeeks.org/software-engineering-architectural-design/

geeksforgeeks. (2024, june 17). *software-engineering-software-maintenance*. From geeksforgeeks: https://www.geeksforgeeks.org/software-engineering-software-maintenance/

geeksforgeeks. (2025). *unified-modeling-language-uml-class-diagrams*. From geeksforgeeks: https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/

Grant, M. (2024, 08 14). *Gantt Chart: Definition, Benefits, and How It's Used*. From investopedia: https://www.investopedia.com/terms/g/gantt-chart.asp

lectera. (2025, 04 01). *gantt-chart*. From lectera: https://lectera.com/info/articles/gantt-chart

Lewis, S. (2023, 03). *collaboration diagram*. From techtarget: https://www.techtarget.com/searchsoftwarequality/definition/collaboration-diagram

mindmanager. (2025). *What's an activity diagram?* From mindmanager: https://www.mindmanager.com/en/features/activity-diagram/?srsltid=AfmBOop4vEG18cwrMFUl5XUUtlerEQg6KVi0do0FIghQsfjyat8rTuWs

Opinaldo, N. (2022, 06). *communication-diagram*. Retrieved 04 01, 2025 from gitmind: https://gitmind.com/communication-diagram.html

projectmanager. (2024, 3 11). *A Gantt Chart Guide with Definitions & Examples*. Retrieved 3 11, 2024 from projectmanager: https://www.projectmanager.com/guides/gantt-chart

systemdesignschool. (2025). *layered-software-architecture*. From systemdesignschool: https://systemdesignschool.io/blog/layered-software-architecture

Wong, S. (2024). *Use Case*. Retrieved 04 01, 2025 from Rice University: https://www.clear.rice.edu/comp310/JavaResources/usecases.html