UE22CS352B - Object Oriented Analysis & Design

Mini Project Report

Title
"Laptop Service & Inventory Management System"

Submitted by:

Name :

PES1UG22CS562 Shivarakshit p Valamannavar

PES1UG22CS558 Sherwin Allen

PES1UG22CS563 Shiven Singh

PES1UG22CS592 Siddhanth

Semester : 6 Section : J

Faculty :
Bhargavi Mokashi

January - May 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

Problem Statement: Laptop Service&Inventory Management System

Key Features:

User Registration & Authentication:

The system supports different types of user registrations—Customer,Service Technician,Vendor. Users can register, log in, and log out. Each role has a dedicated authentication flow that directs them to their respective home pages.

Role-Based Dashboards:

After logging in, users are routed to home screens tailored to their roles:

Customers can view the available laptops buy them ,Make a mortgage Request , log a Service Request and Log a find my Device request.

Service Technician can view the pending servicing requests and act upon them

Vendor has access to Restock the laptops , Update the status of Find my Device Request , Accept/Reject the mortgage Request made by the Customer, add a new laptop

Customer can Also see the status of his requests in his home page .

Raising Requests :

Customer can log Different type of request which later are seen by the authorized person respective to the requests logged.

Restock Mechanism:

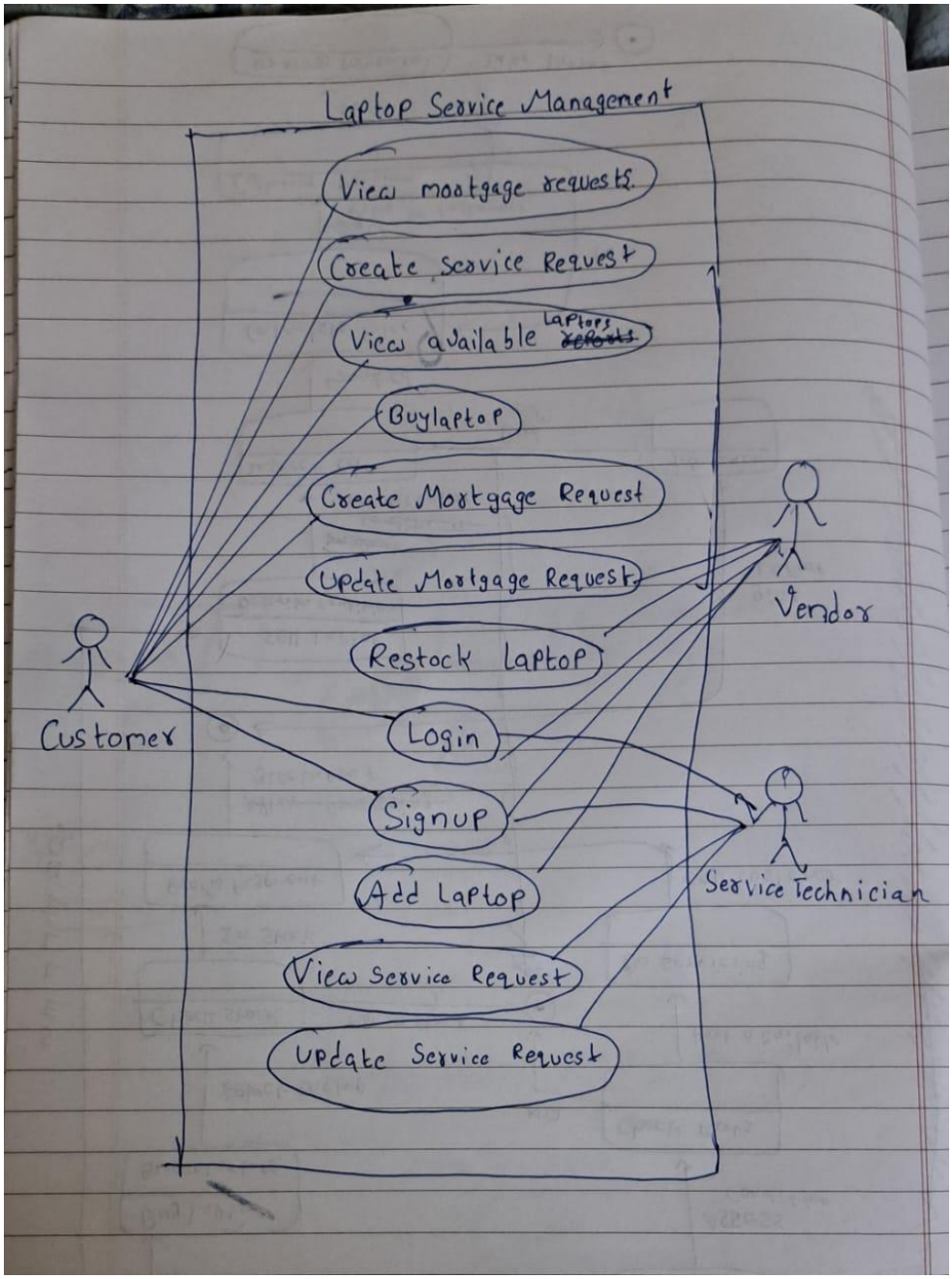
By default whatever the laptops go out of stock they are then itself shown to the vendor so that he can restock those laptops as soon as possible

Find My Device :

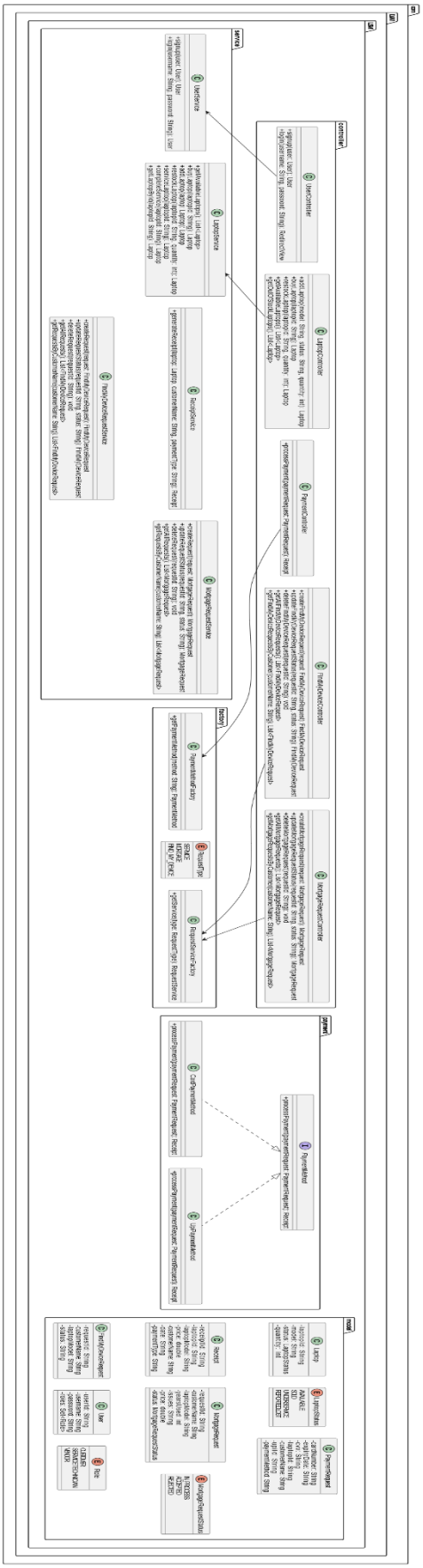
Vendors actually process the find my device by checking the laptops they have collected if the laptop model and the owner's name matches with the person who has logged the request he will be given the laptop back if not the status will be marked as not_found

Models:

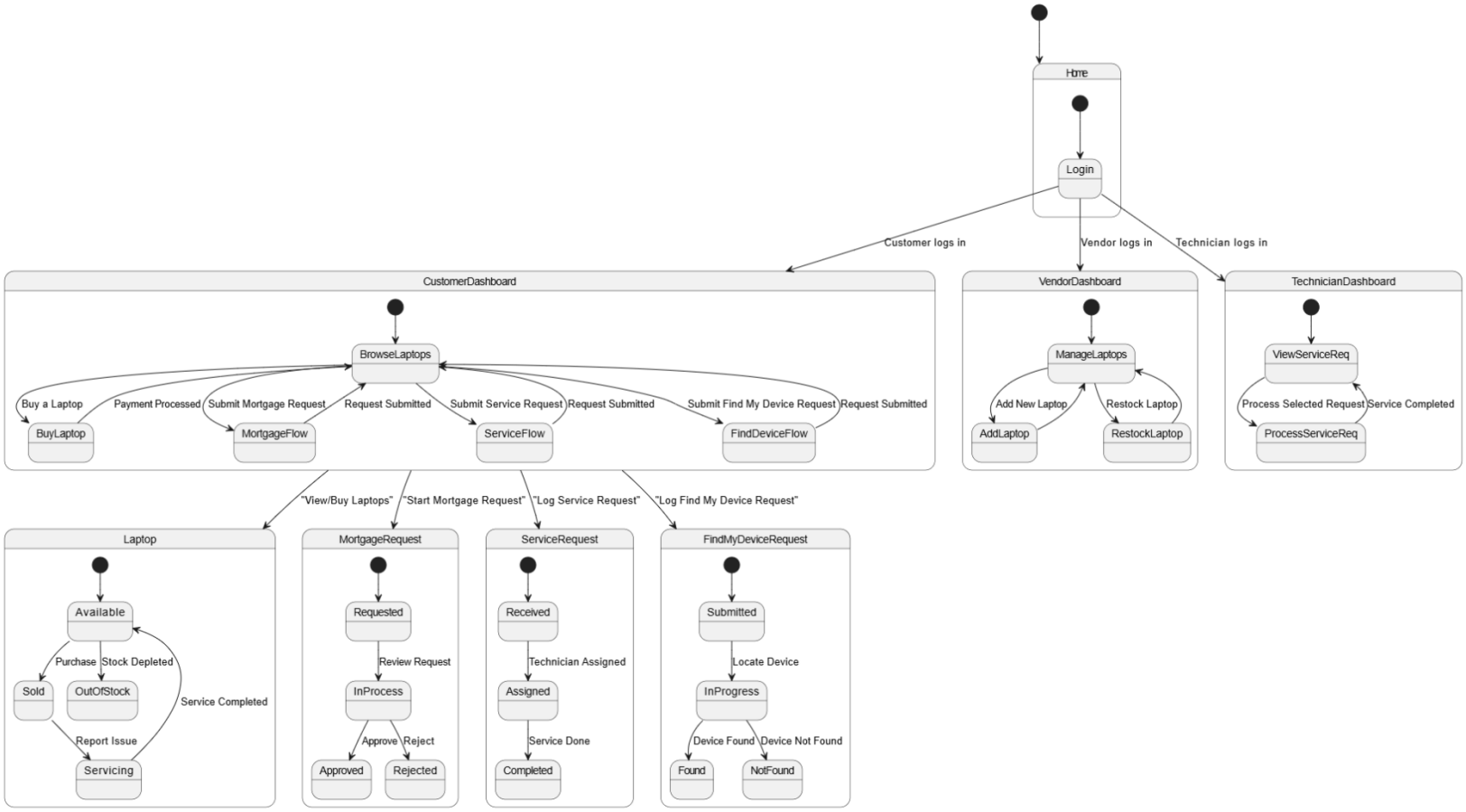
Use Case Diagram:



Class Diagram:

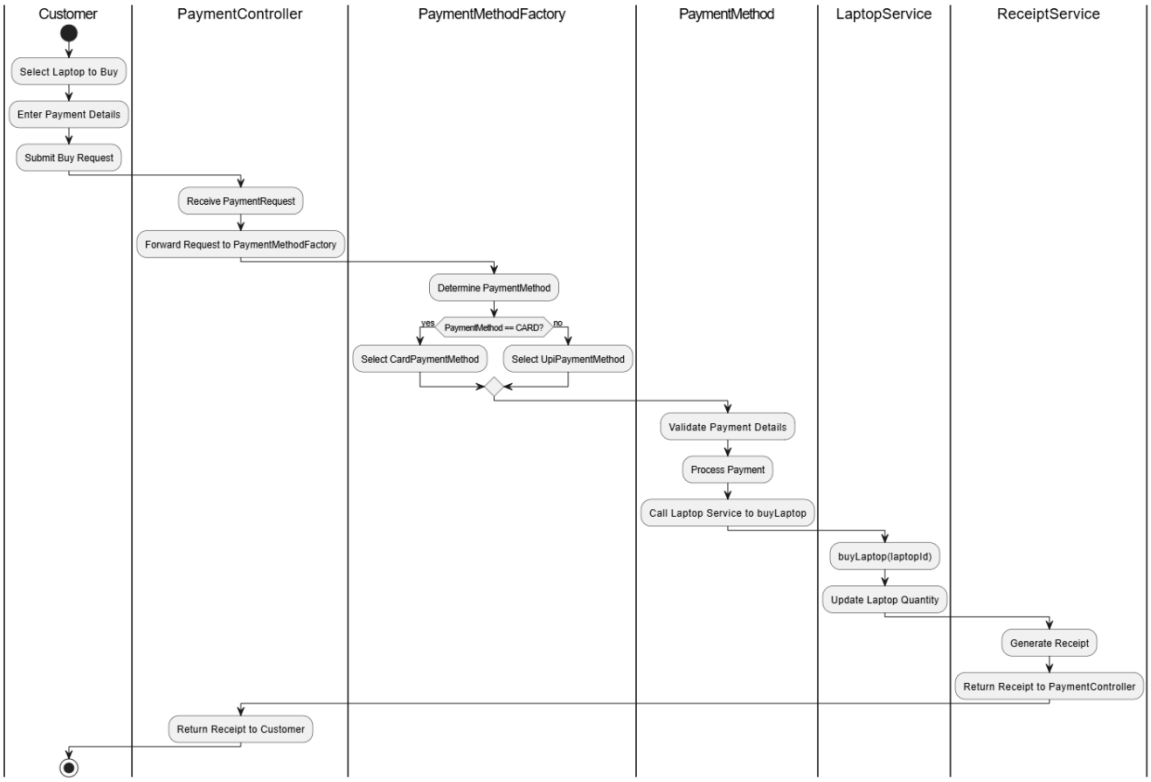


State Diagram:

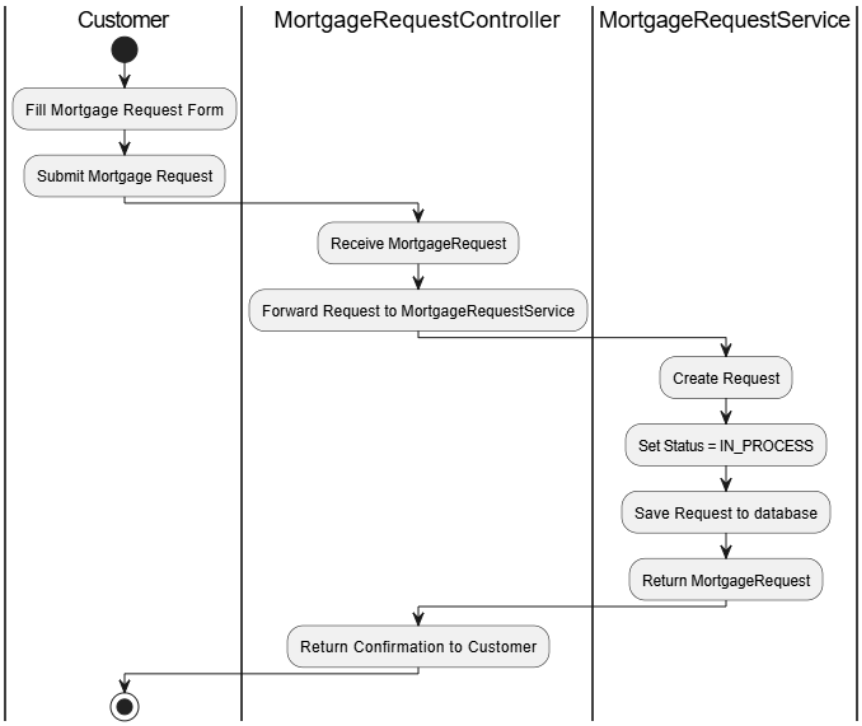


Activity Diagrams:

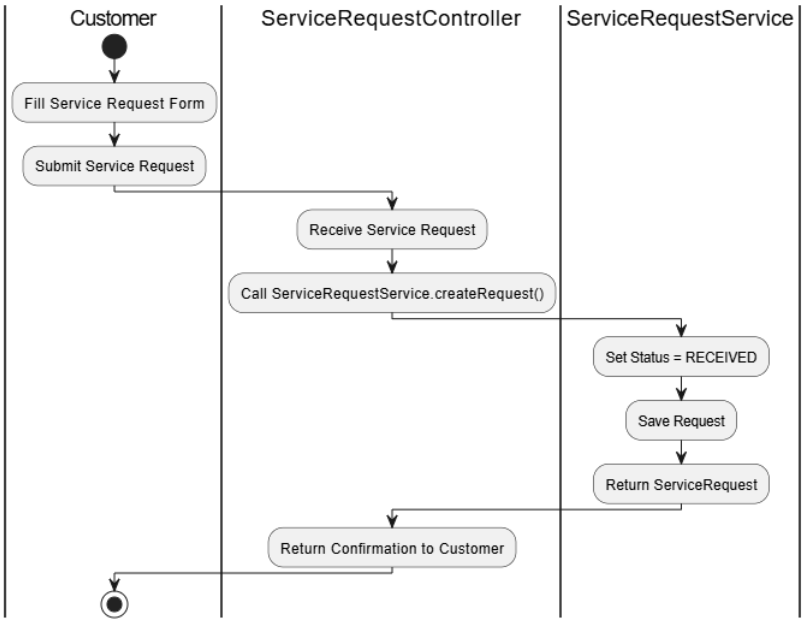
1. Major Usecase



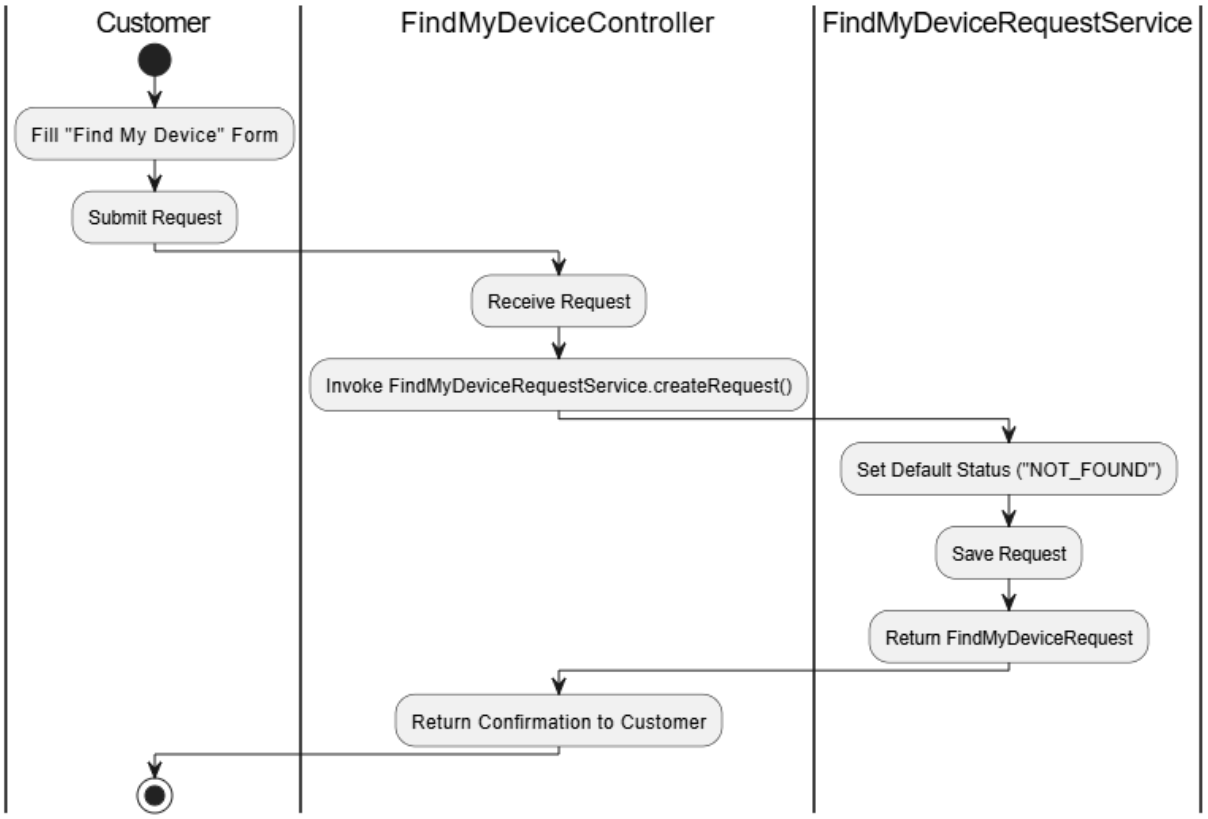
2. Major Usecase



3. Major Usecase



4. Major Usecase



Architecture Patterns, Design Principles, and Design Patterns:

Architecture Patterns

Model – View – Controller Pattern (MVC)

Design Principles

The project uses 4 of the SOLID principles
Single Responsibility Principle
Open/Closed Principle
Liskov Substitution Principle
Interface Segregation Principle

Design Patterns

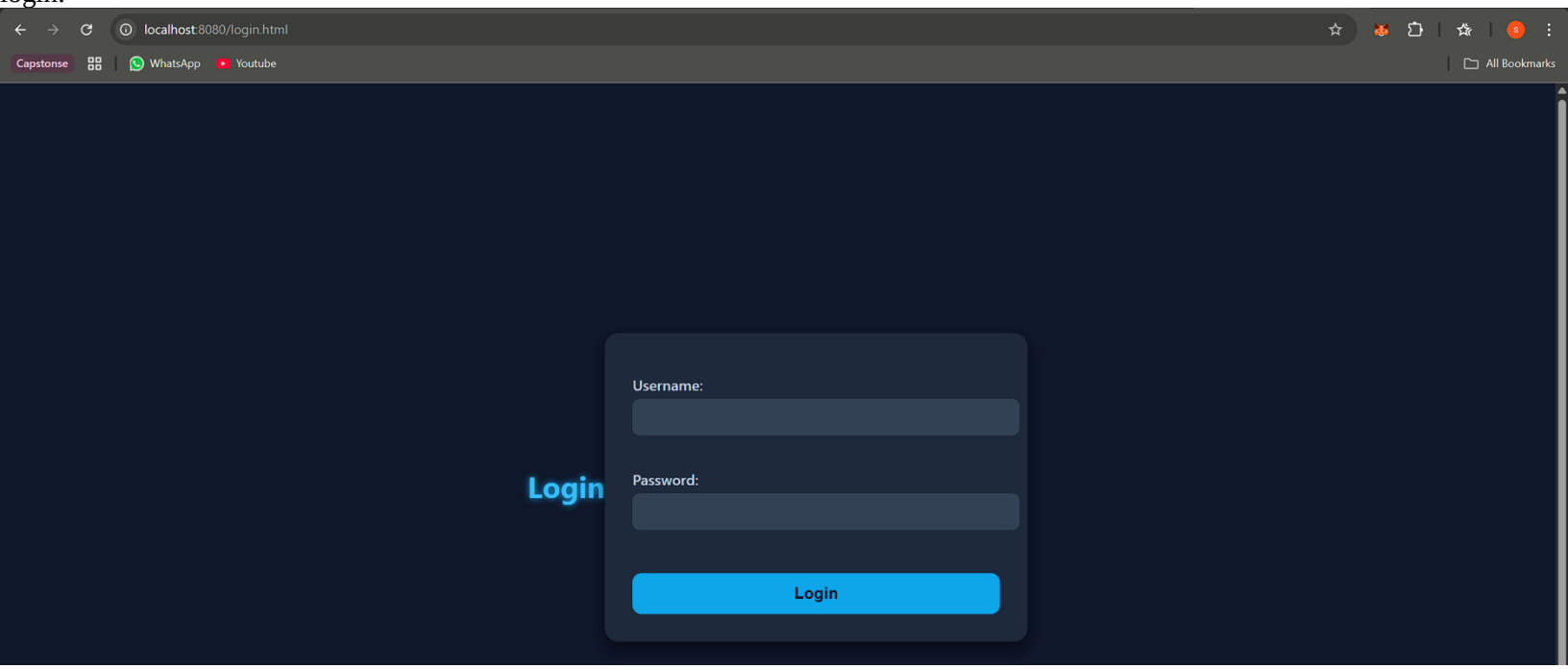
Factory for consistent Payment creation (e.g., PaymentMethodFactory).
Factory for consistent Service-request Creation(e.g.,ServiceRequestFactory).
Builder for constructing receipt objects (e.g., ReceiptBuilder).
Singleton for single instance of an object (e.g., LaptopID Generator).

Github link to the Codebase:

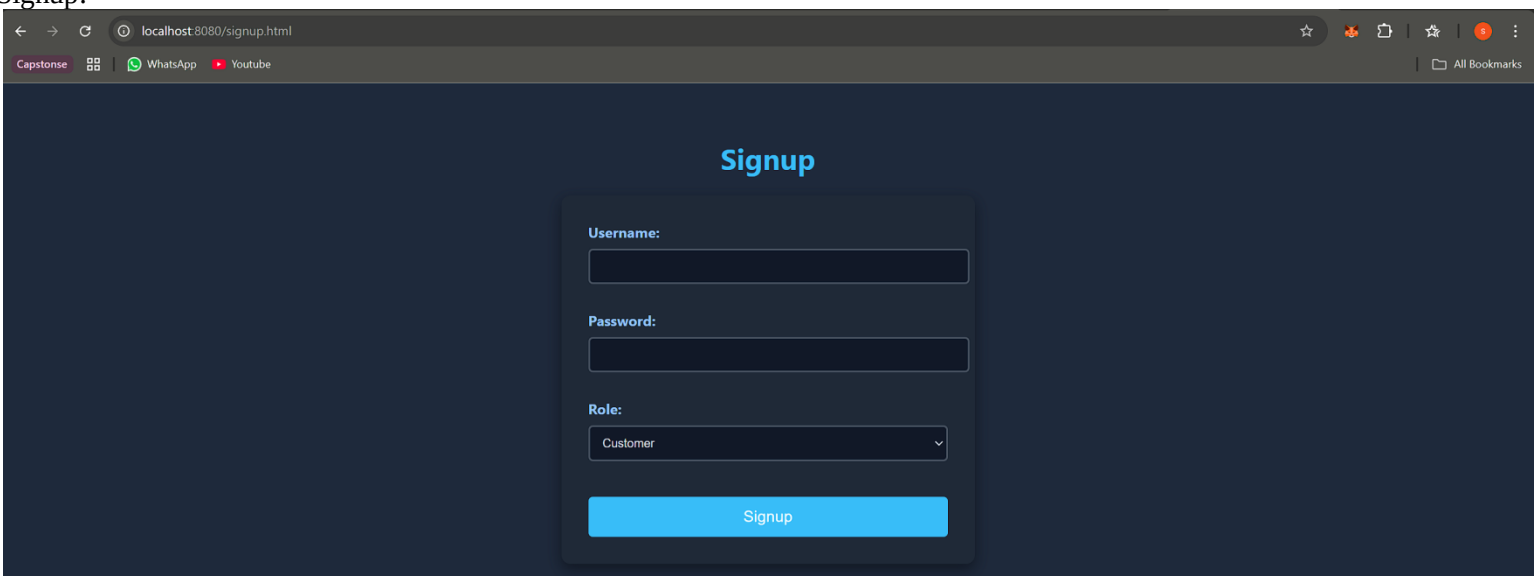
https://github.com/Shivarakshitvalamannavar/LSM

Screenshots of UI:

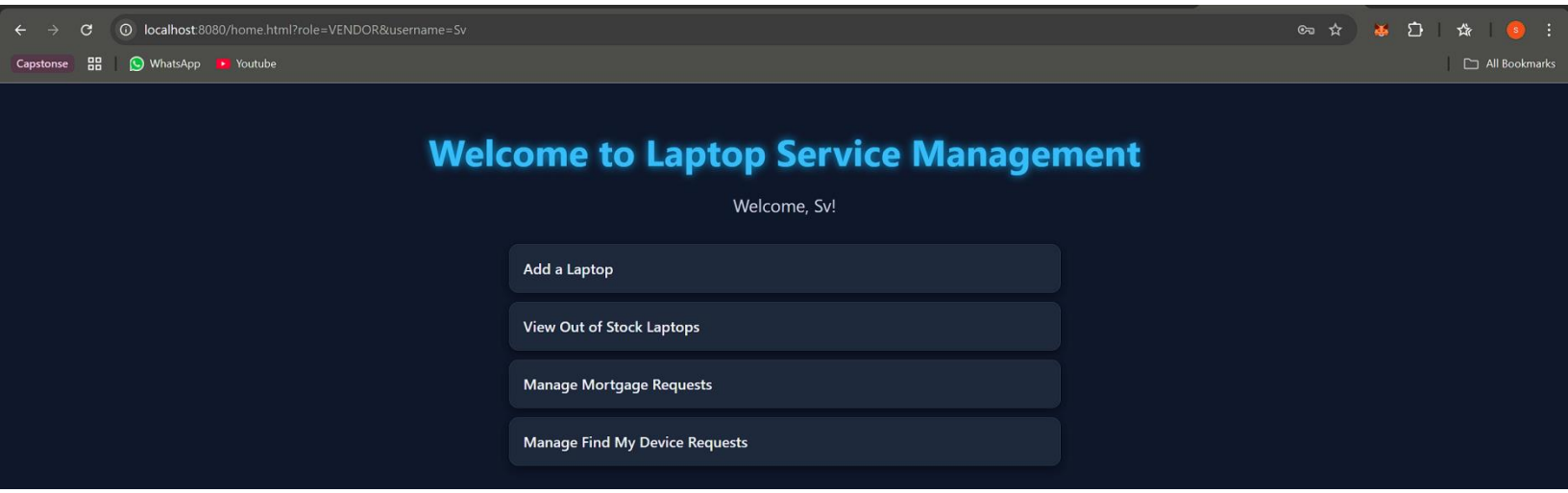
login:



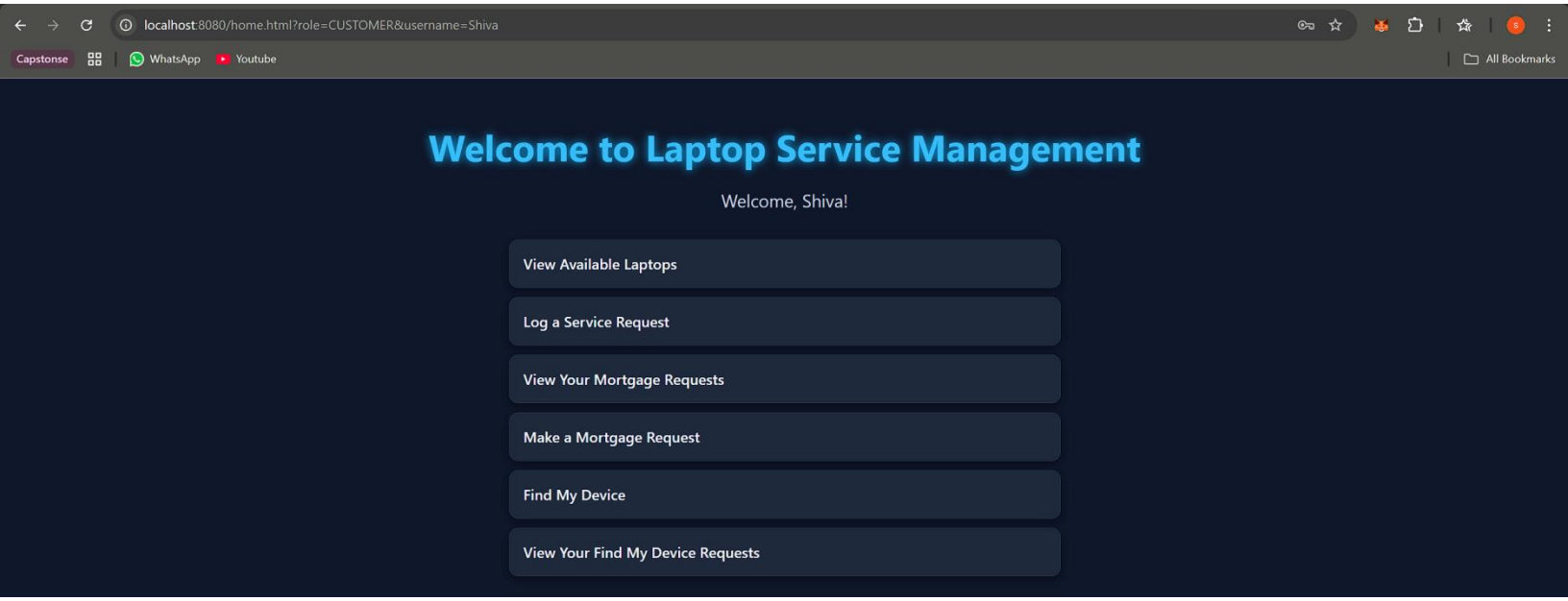
Signup:-



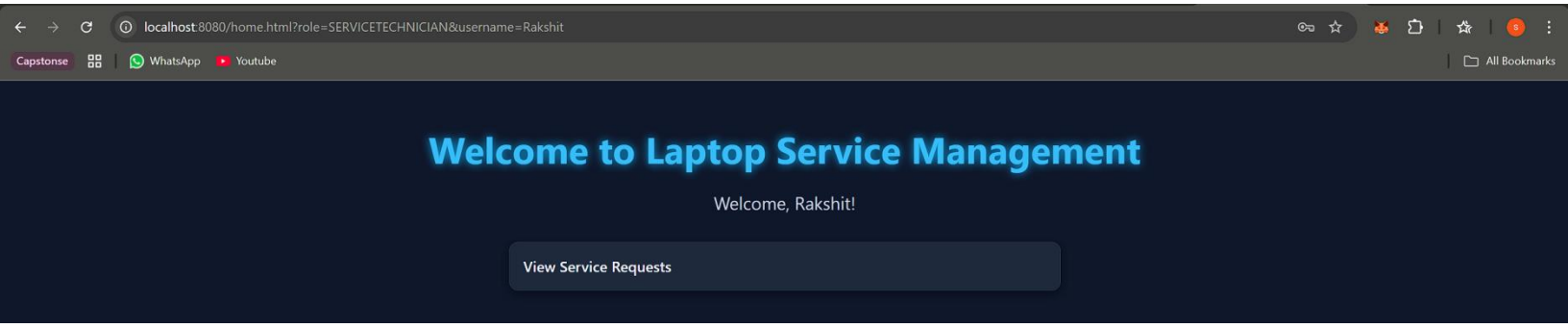
Vendor:

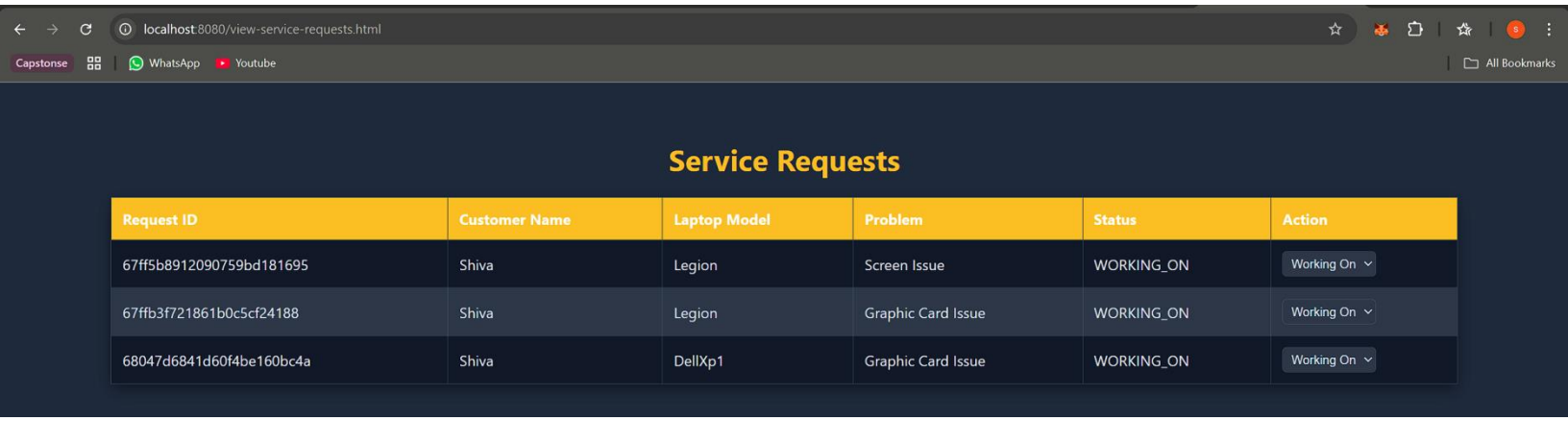


Customer:



Service Technician:-





Individual contributions of the team members:

Name	**Module worked on**
Shivarakshit	Factory
Sherwin Allen	Factory
Shiven Singh	builder
Siddhanth	Singleton