SOFTWARE ENGINEERING LAB

PROJECT REPORT

On

Motor Part Shop Software

Submitted By:

Suwesh Prasad Sah (120CS1069)

Awantika Mallick (120CS1108)

Akash Poudel (120CS1109)

Prabesh Kumar Sah (120CS1110)

Submitted To:

Dr. Puneet Kumar Jain

Department of Computer Science and Engineering

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

APRIL, 2023

COPYRIGHT

ABSTRACT

Write your abstract here

Keywords: a, b, c

TABLE OF CONTENTS

List of Figures

List of Tables

1. LIST OF ABBREVIATIONS

RAM	Random Access Memory
GB	GigaByte
SQL	Structured Query Language
GUI	Graphical User Interface
ID	Identifier UML
Unified Modeling Language	

2. INTRODUCTION

2.1. Background

As we're closer to a completely digital era, we depend more and more on software-based solutions for our tasks. However, we could not find any relevant software in the market currently for the niche i.e, Motor Parts Shop/Automobile spare parts automation. So, to address the particular problem, we decided to build a new software highly customized and targeted towards the Automobile parts shop.

2.2. Motivation

We wanted to build software targeting the huge market of smaller businesses like Motor Part Shops. The motivation was to make businesses grow by making everything a seamless experience. Every tiresome work related to the business can be made with this within just a few clicks or touches. This would highly impact the positive growth of their business.

2.3. Objectives

1. To automate the tasks for a Motor Parts Shop.
2. To help the shops grow in revenue by providing a seamless experience.

2.4. Problem statement

The motor parts shop deals with large number of motor parts of various manufacturers and various vehicle types. Some of the motor parts are very small and some are of reasonably large size. The shop owner maintains different parts in wall mounted and numbered racks. The shop owner maintains as few inventory for each item as reasonable, to reduce inventory overheads after being inspired by the just-in-time (JIT) philosophy. Thus, one important problem the shop owner faces is to be able to order items as soon as the number of items in the inventory reduces below a threshold value. The shop owner wants to maintain parts to be able to sustain selling for about one week. To calculate the threshold value for each item, the software must be able to calculate the average number of parts sales for one week for each part. At the end of each day, the shop owner would request the computer to generate

the items to be ordered. The computer should print out the part number, the amount required and the address of the vendor supplying the part. The computer should also generate the revenue for each day and at the end of the month, the computer should generate a graph showing the sales for each day of the month.

2.5. Scope of Project

The scope of this project is to develop a software system for a motor parts shop that will enable the shop owner to manage the inventory efficiently, minimize inventory overheads, and sustain selling for about one week. The software should be able to calculate the average number of parts sales for one week for each item based on historical sales data. The system should also be able to generate the items to be ordered, the amount required, and the address of the vendor supplying the part at the end of each day.

3. LITERATURE REVIEW

to be added later

4. REQUIREMENT ANALYSIS

4.1. Introduction

4.1.1. Purpose

The purpose of the document is to serve as a guide to designers, developers, and testers who are responsible for the engineering of the Motor Part Shop project. It should give the engineers all of the information necessary to design, develop and test the software.

4.1.2. Intended Audience and Reading Suggestions

This SRS is for developers, project managers, users, and testers. Further, the discussion will provide all the internal, external, functional, and also non-functional information about motor parts shop software (MPSS).

4.1.3. Project Scope

A small automobile spare parts shop sells spare parts for vehicles of several models. Also, each part is typically manufactured by several small industries. To streamline the sales and supply ordering, the shop owner has asked us to develop the following motor part shop software. The motor part shop deals with large no. of motor parts from various manufacturers and various vehicle types. Some of the motor parts are very small, and some are very large. The owner maintains different parts in wall-mounted and numbered racks. The shop owner maintains as little inventory for each item as reasonable, to reduce inventory overheads after being inspired by the"just in time (JIT) philosophy". Now to facilitate the sales and stocking up the motor parts, we are required to develop the required software.

4.1.4. Environment Characterstics

The software is compatible with Windows version 7 and above for greater flexibility across all systems. The hardware requirements involve RAM greater than 512 MB, with Pentium core and above. The hard disk capacity needs to exceed 20 GB for this software to function smoothly.

4.2. Overall Description

4.2.1. Product Perspective

The Motor Part Shop Software will be a newly developed and self contained product. It will help in restocking of motor parts for the intended owner's shop as well as it will show required sales statistics to the owner. The Software can be modified in further versions to add complex functionalities.

4.2.2. Product Features

The Product Feature includes:

- Login Authentication.
- Search for a motor part
- Checking current stock of motor parts in the store
- Setting the threshold for a motor part
- Billing a customer
- Checking the sales statistics for the shop
- Getting the list of parts that needs to be restocked by the end of the day
- Updating the database when new stocks are added or when new parts are bought for the shop.
- Adding a new motor part to the inventory

4.2.3. User Classes and Characteristics

The various user classes that are expected to use this software as follows:

- Owner: Can use the software the look into the sales statistics and the current stock of the motor parts.
- Shopkeeper: Can use the software for billing purposes.

- Salesman: Can update the current stock of the motor parts after getting the list from the software.

4.2.4. Operating Environment

The software is compatible with the operating system windows version 7 and above and MAC OS. The database used is MySQL. Python is used for the back end, and Python's GUI Library Tkinter is used for the front end of the software. The hardware requirements involve RAM greater than 512 MB, with Pentium core and above. The hard disc capacity needs to exceed 20 GB for this software to function smoothly.

4.2.5. Design and Implementation Constraints

The information on all the motor parts must be stored in a database that is accessible by the motor part shop software. No internet is required for the functioning of the motor part shop software, so the motor part shop software should be able to run all 24 hours a day. The billing system is connected to the motor part shop software, and the database used by the billing system must be compatible with the interface of the motor part shop software. The users must have their correct usernames and passwords to enter into the motor part shop software.

4.2.6. User Documentation

A user Manual of the motor part shop software would be provided at the time of delivery of the software to the required authority, which can be used to solve any troubleshooting issues regarding this software. If the user manual does not help, the developers are always ready to help with any troubleshoot who can be contacted by the number given in the manual.

4.3. Functional Requirements

4.3.1. Login Authentication

This is the first functionality that the user encounters when it uses the software. There are three classes of users, which include owner, shop keepers and the sales executive. Users are required to enter their username and password to login into the software.

- Input: Username and Password.
- Authentication: Username and Password are matched in the database.
- Output: A successful message or failure message is displayed, once the credentials are checked.

4.3.2. Search for a motor part

This feature can be used to search for a motor part that is present in the inventory. This feature returns the location of the motor part and the number of parts left in the inventory. This feature can be accessed by the owner, the shop keeper and the sales executive

- Input: Motor Part ID
- Output: The rack location where the motor part is stored, and the number of parts stored there. If no parts are present, a warning message is to be displayed.

4.3.3. Checking current stock of motor parts in the store

This feature is kept to check the current stock of any motor part within the store. A list of all the existing motor parts is displayed when the user wishes to check the stock showing the part's number, part's name and the number of items in the shop currently

- Input: User asks for the current stock by clicking on the "Check current stock" button.
- Process: The current stock is fetched from the database in ascending order of the part ID.
- Output: he current stock is displayed in a list showing the part ID, part name and the number of parts in the inventory currently.

4.3.4. Setting the Threshold for a motor part

This functionality is required to calculate the threshold of each motor part based on the number of parts sold previously so that the shop does not run out of the specified part. The average of number of parts previously sold is calculated to get the threshold.

- Input: The stock that is sold in recent weeks.
- Process: Average of the stocks is calculated for each part.
- Output: The threshold for each motor part, if the current stock falls below the threshold for a motor part, the motor part is restocked.

4.3.5. Billing a Customer

This feature is used to bill a customer when it wishes to buy some required parts from the motor part shop. After the billing is done, the appropriate change is done in the database, which is reflected over the sales statistics.

- Input: Parts bought by user.
- Output: The stock is reduced in the database

4.3.6. Checking the sales statistics for the shop

The functionality has got two parts, as follows;

- The owner can check the sale of motor parts from the shop in the current day. This includes the parts that were sold beginning from the day itself. The user can also check the revenue of the day with the help of this feature.
  - Input: Owner asks for sale of motor parts of the current day
  - Output: Revenue for the day is calculated and displayed along with the sold parts details.
- The owner can check the sales graph which shows the daily sales of the shop over a period of a month. This includes all the parts in the shop. The graph would also show the revenue generated every day in the current month.
  - Input: Owner asks for sale of motor parts of the current month.
  - Output: A graph is displayed which shows the sale of motor parts for each day of the month.

4.3.7. Getting the list of parts that needs to be restocked

The user can get a list of all the motor parts that need to be restocked using the software. The software compares the current stock of an item with the average number of sales of the item over a week. If the current stock is low, the user needs to restock the motor part to carry on the functioning of the shop in a smooth manner.

- Input: Sales executive wants a list of parts that need to be restocked
- Output: A list is displayed showing the ID, name, vendor address and the number of the motor parts that need to be restocked.

4.3.8. Updating the Current Stock

The user can increase the existing number of a motor part after it has been restocked using the software. The change would be reflected in the database, from which the upcoming purchases would be carried on.

- Input:The sales executive adds the number of parts bought for the shop against the ID of the motor part.
- Output: The number of motor parts is increased in the database and is reflected while checking the current stock.

4.3.9. Adding a new motor part to the inventory

This functionality enables the sales executive to add a new motor part into the inventory. To add a new part, the sales executive needs to enter the part ID, part name, the address of the vendor who currently sells the motor part. The initial count for a new part is set to 0, which can be incremented accordingly by the sales executive.

- Input: Sales Executive adds the id of the new part, name of the part, address of the vendor from which the part is bought along with the number of the parts.
- Output: New part is added in the database which can be checked by the current stock.

4.4. External Interface Requirements

4.4.1. User Interfaces

When the user will open the software, a welcome page asking the user id and password will be displayed. The software would entertain three categories of users, as it has been mentioned in the user classes. Upon logging in, the software will display four buttons along with a billing button and an exit button. Upon the clicking the exit button, the current user will be logged out and the login page will be displayed again. The billing button will direct the user to the billing page. The four functionalities include, Stocks, Sales, Requires Parts, Add parts. The stocks button will show the list of current stock of motor parts. The sales button will display the sales statistics of the shop. The requires parts button will display the motor parts that need to be restocked by the end of the day. The add parts page will help in adding the recent number of restocked motor parts. The UI is kept simple and understandable for the user, so that it can work with it without hassle.

4.4.2. Hardware Interfaces

Since the software is supposed to be run on a single system, we do not require cloud based hosting solution here. If more than one system needs to be connected, cloud based solution is required to store the database.

4.4.3. Software Interfaces

A firewall will be used with the server to prevent unauthorized access to the system

4.4.4. Communication Interfaces

The software is to be kept connected to the internet for continuous use in multiple systems.

4.5. Other Non-Functional Requirements

4.5.1. Maintainability Requirements

The software must be robust enough to require as lesser maintenance as possible. Given a glitch in the software, the administrators must be capable enough to sort out the bug quickly to prevent delay in the shop's functionality.

4.5.2. Portability Requirements

The software should be able to be deployed in any machine.

4.5.3. Performance Requirements

The software must be developed using an object oriented model. The performance of every existing module in this software must be robust. This software should be able to run on various operating systems steadily as it as been specified before. Overall the performance of the software must be reliable and the data kept must be safe in case of a power failure.

4.5.4. Safety Requirements

Passwords must be kept different from the student id's for safety purpose. A mail must be provided for emergency queries regarding the software, so that the software can be used without concerns. The mail must be replied by the admins of the software for quick responses.

4.5.5. Security Requirements

Passwords must be stored in hashed format with some salt to prevent any data leak. The authentication used in the login page should work without fault. And finally the whole software is completely secured from outside accessing.

5. SYSTEM ARCHITECTURE AND METHODOLOGY

The whole system architecture is divided into various parts. We have represented the design of the system in following UML diagrams:

5.1. Use Case Diagram

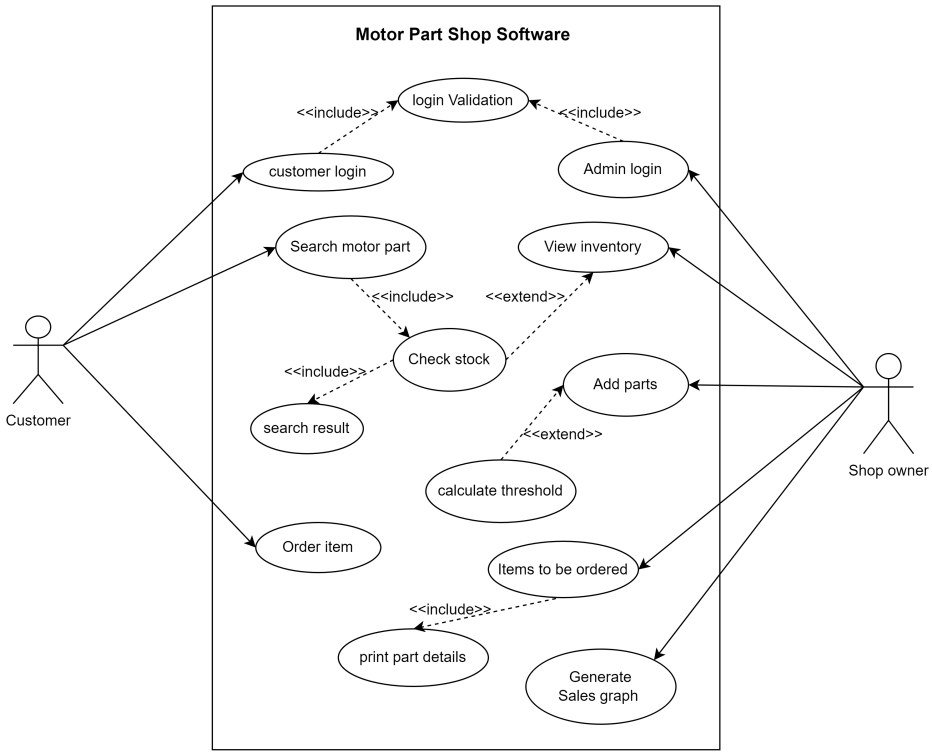


Figure: Use Case Diagram for Motor Part Shop Software

5.2. Activity Diagram

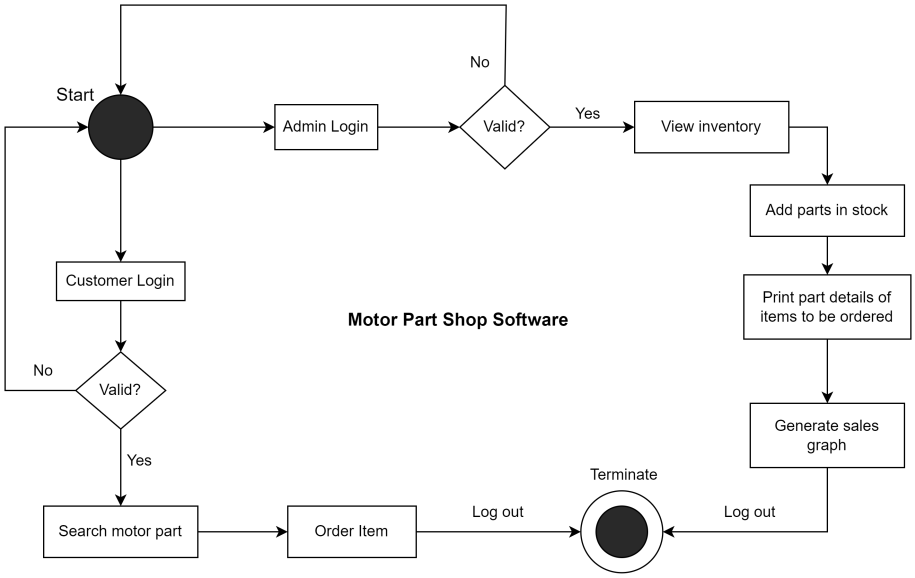


Figure: Activity Diagram for Motor Part Shop Software

5.3. Class Diagram

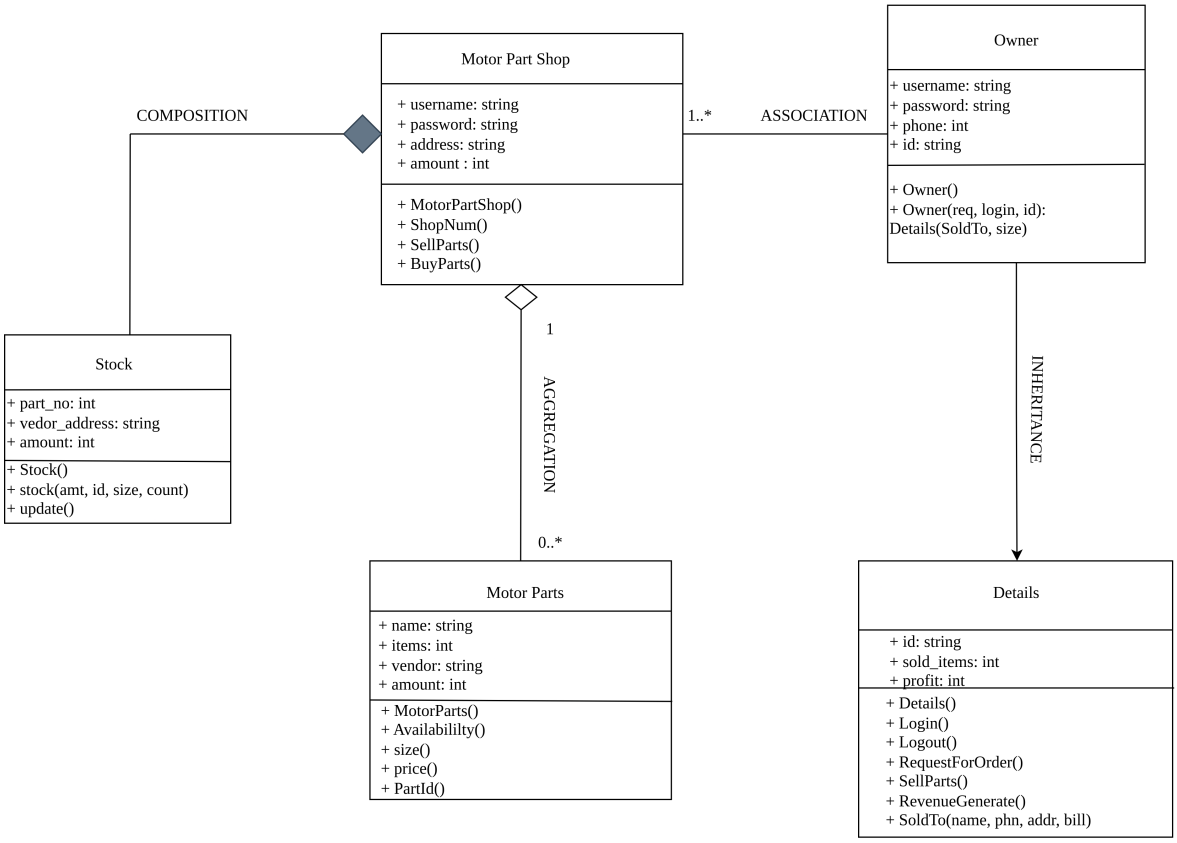


Figure: Class Diagram

5.4. Object Diagram

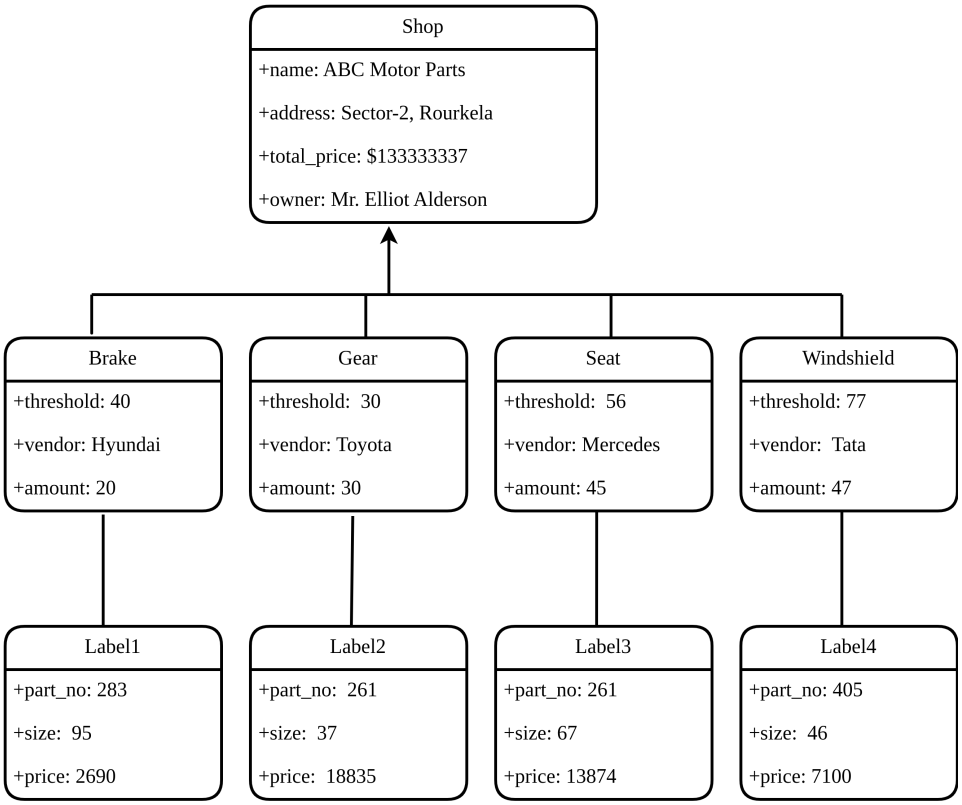


Figure: Object Diagram for Motor Part Shop Software

5.5. Database Schema

5.6. Sequence Diagram

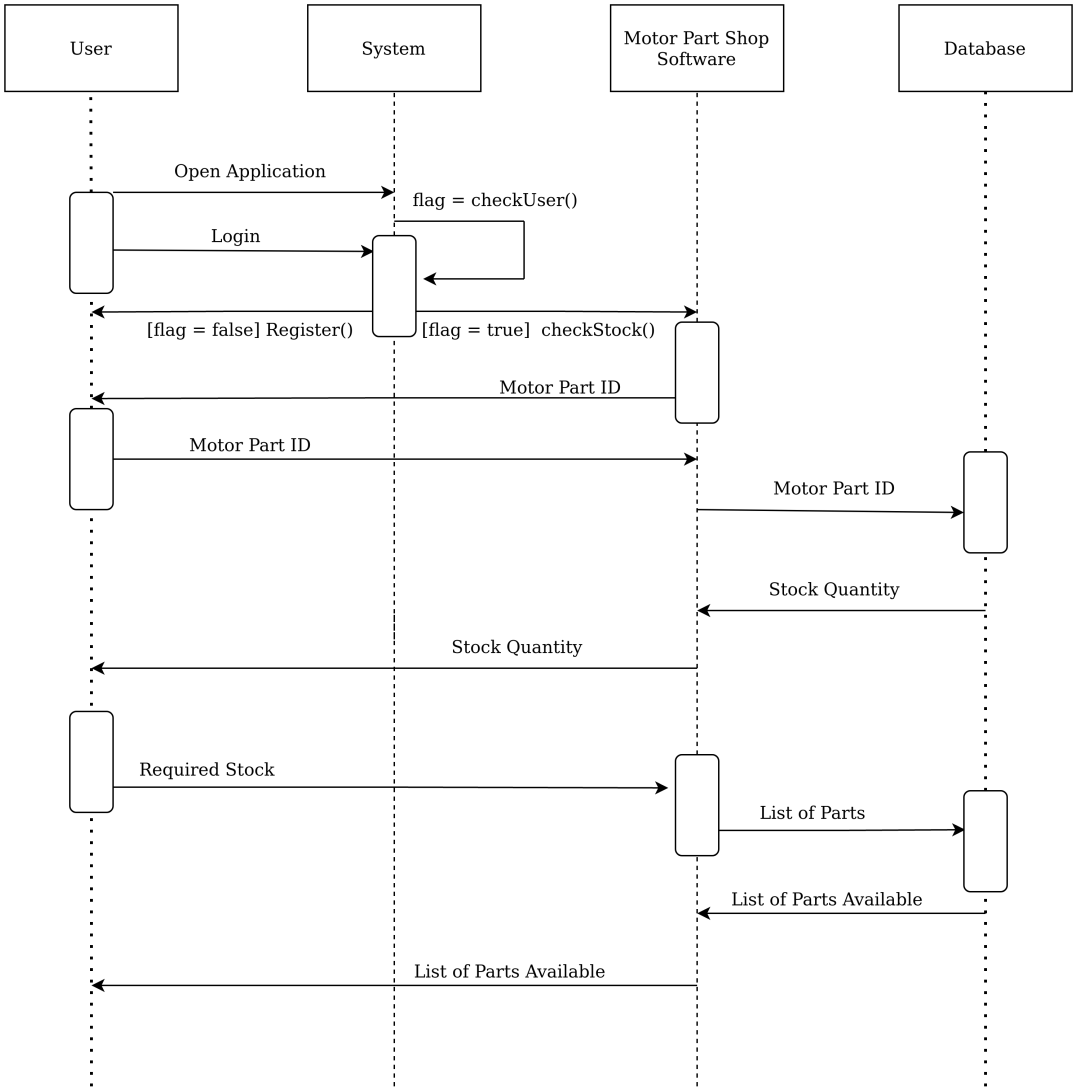
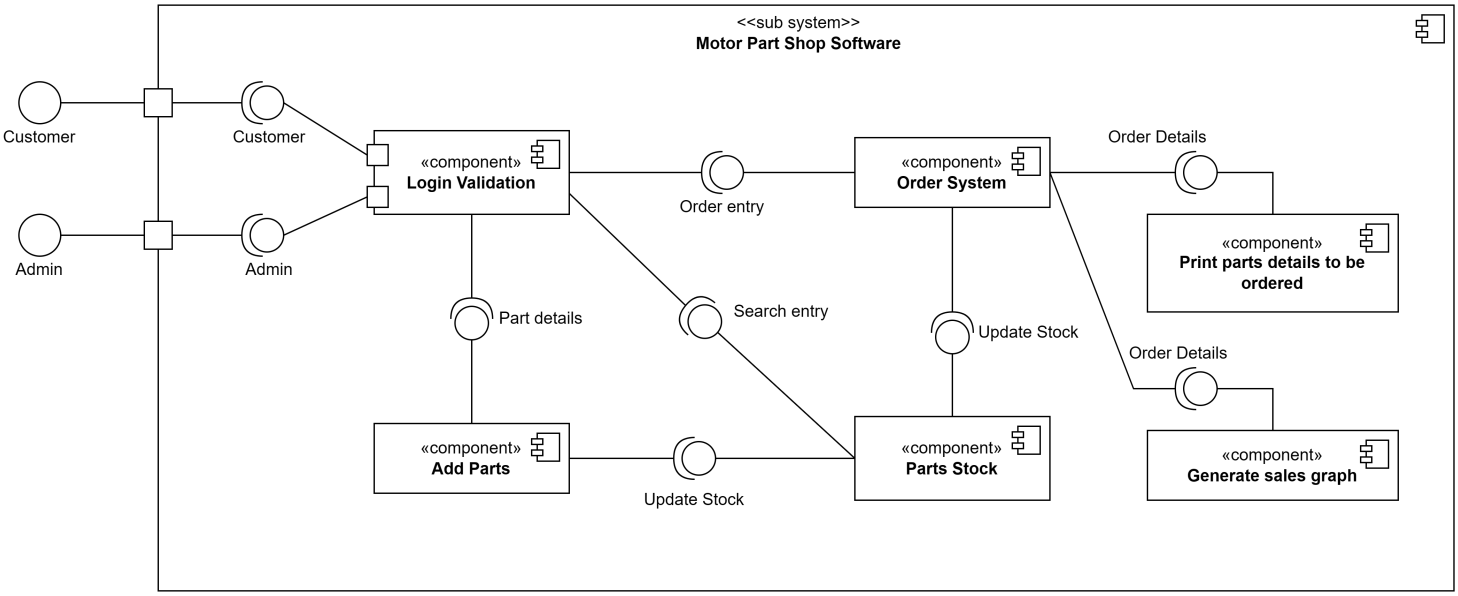
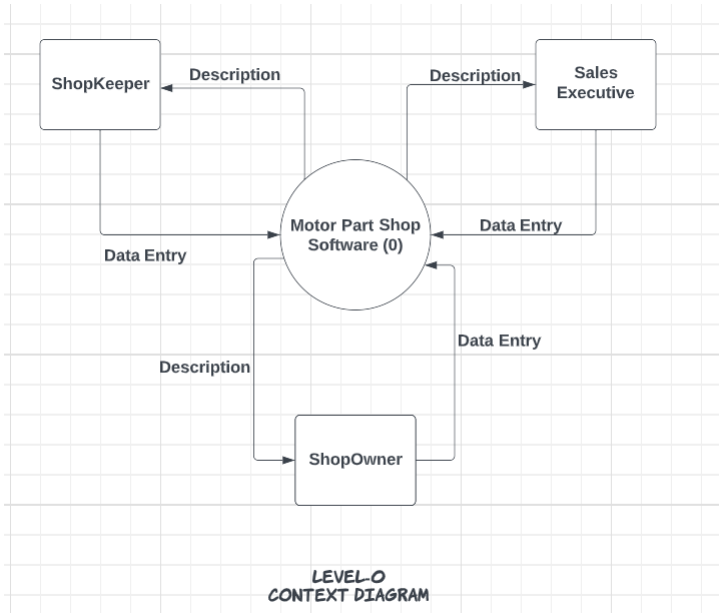


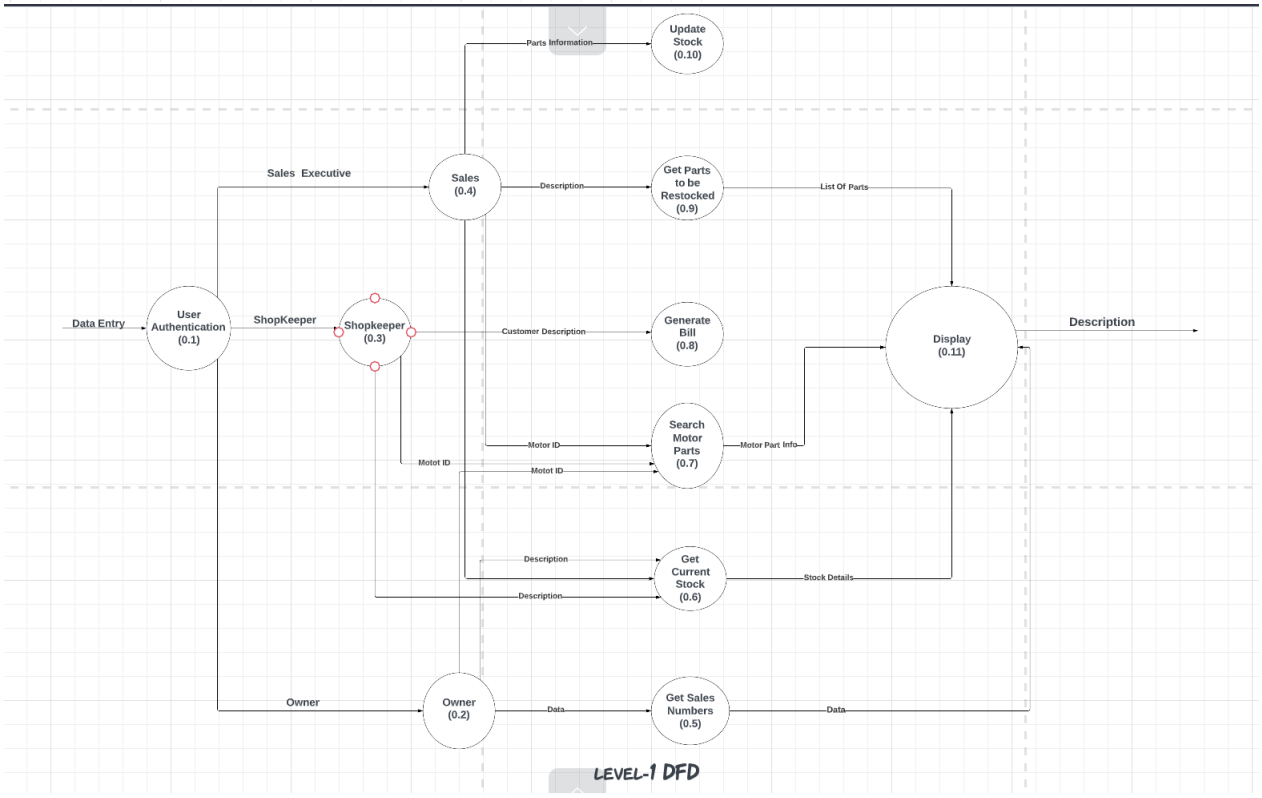
Fig: Sequence Diagram

5.7. Component Diagram



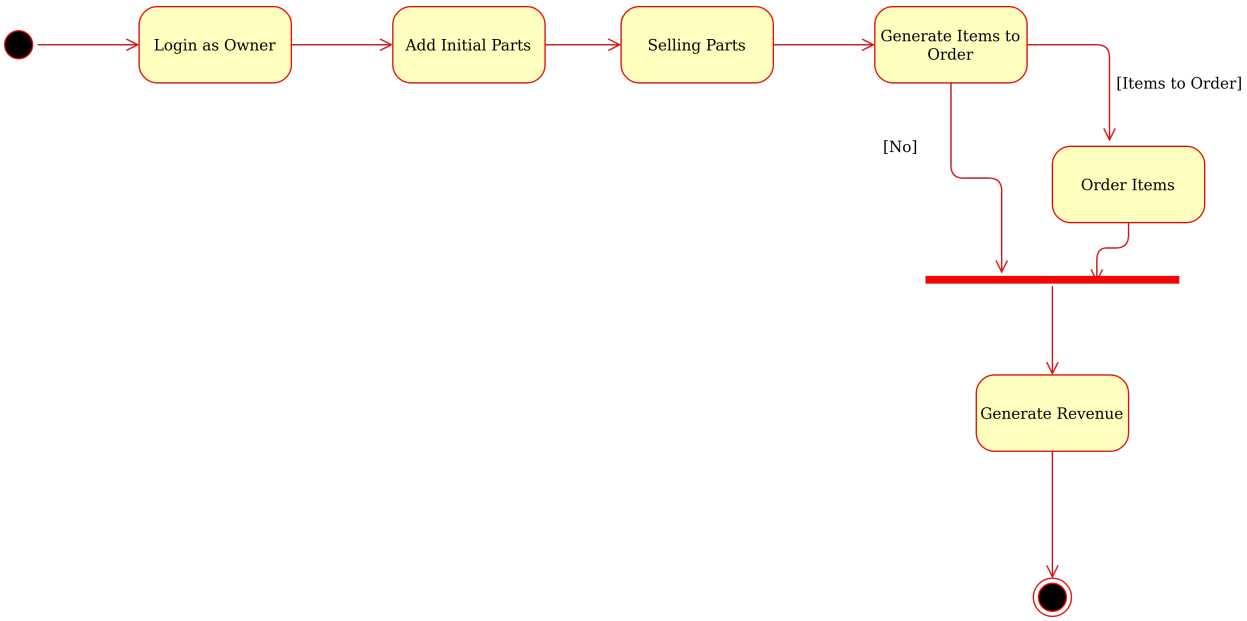
5.8. Data Flow Diagram





5.9. Deployment Diagram

5.10. Statechart Diagram



6. IMPLEMENTATION DETAILS

Describe how the hardware components / instruments & software function in your project. Describe the calibration process required for correct operation of each module. Describe the interfacing technicalities / protocol of each module used in your project. Explain in detail how all components are interconnected to make a functioning system.

7. RESULTS AND DISCUSSIONS

(It contains the results/outputs of your project. The output can be numeric or graphical based. Present the outputs of your project in the form of tables, graphs, charts, figures and explain their behavior. You can also represent or write down the results in tabular form if applicable and analyze that by using graphs or charts. Perform error analysis, comparisons (theory, simulation, practical) and validate your output. Discuss the sources of errors in your project that caused your outputs to deviate from expected values.

8. CONCLUSION

In this project

A. APPENDIX

It may contains the additional topics or data sheets or reference sheets or even user manual. Project Budget (Detailed Breakdown of Costs), Project Timeline (Gantt chart), Circuit Diagrams (Should be Clear and Legible), PCB Designs (Should be Clear and Legible), Module Specifications (Should be brief - Keep only necessary tables and figures), Details of Dataset can be included here.

References