SOFTWARE REQUIREMENTS SPECIFICATION

for

Zulu - A Motor Part Shop Software

Version 1.0 approved

Prepared by -

Ashutosh Kumar Singh (19CS30008)

Vanshita Garg (19CS10064)

Suhas Jain (19CS30048)

Indian Institute of Technology, Kharagpur

March 26, 2021

Contents

Revision History

Name	Date	Reason For Changes	Version

1 Introduction

1.1 Purpose

Zulu is a software that manages the inventory and streamlines the sales and supply ordering for an automobile spare parts shop. This can be used by a motor parts shop owner. The software contains features to keep track of the quantity of each part in the inventory, order items when required and generate sales reports on a daily and monthly basis.
The purpose of this document is to explain the features, purpose and constraints under which the software would be built.

1.2 Document Conventions

This SRS document has been written in Latex using KOMA Script. The font used in the document is Computer Modern. Each chapter of the document begins from a new page and the chapter heading is written in a bold large font. Headings of sections within a chapter are also written in bold but with a slightly smaller font.

1.3 Intended Audience and Reading Suggestions

This document lists all technical and non-technical aspects of the software. It is intended to assist developers, project managers, marketing staff, users, testers, documentation writers and other end users to understand the motivation behind the software and understand implementation intricacies in it.
A brief summary of the whole SRS document can be encapsulated under following sections.
Users are suggested to follow this sequence in order to have a better understanding of the document.
Chapter 1 : Basic Introduction
Chapter 2 : Overall Description of the software giving information about functions, user classes, operating environment, assumptions and dependencies
Chapter 3 : External Interface Requirements giving a brief introduction to user, hardware, software and communication interfaces
Chapter 4 : Various system features
Chapter 5 : Non-functional requirements
Chapter 6 : Use Case Diagram and Class Diagram

1.4 Product Scope

Managing the inventory of any shop with a large number of products and items has always been a tedious task involving a lot of people, and an automobile spare parts shop is no exception. Doing this task manually consumes a lot of time, energy, money and manpower. Zulu automates this process of keeping track of the inventory. Moreover, the software automatically generates a list of items to be ordered so as to maintain sufficient stock for one week. This relieves the shop owner of the burden of calculating which items are in deficit. It also gives him/her an advantage over his/her competitors as customers will prefer a shop where parts are readily available. The software also generates daily and monthly sales reports which help the owner analyze his/her business strategy and make suitable decisions to ensure that the business flourishes in the future.

1.5 References

IEEE Standard 830 - 1998 IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

Slides from the NPTEL course Object Oriented Analaysis and Design by Prof. Partha Pratim Das, IIT Kharagpur.
https://nptel.ac.in/courses/106/105/106105153/

2 Overall Description

2.1 Product Perspective

Although there may be many similar products available on the internet that work on similar ideas, but hardly any one of them is designed specifically for an automobile spare parts shop. Moreover, they charge for their services and are hence not very popular among small scale users. Zulu is a self-contained product that aims to render its services to the users at a negligible one time cost. This product aims to break the complicated barrier by introducing a very user friendly experience.

2.2 Product Functions

This product contains a variety of small-scale features which can be listed as follows:

- Add new motor spare parts to the inventory
- Remove motor spare parts from the inventory
- Record a sale and update the quantity of any item in the inventory
- Generate a list of items to be ordered
- Generate daily sales revenue
- Generate a graph showing the daily sales for a month

2.3 User Classes and Characteristics

The software is intended to be used by the shop owner of an automobile spare parts shop. The shop owner himself/herself will use the software for all its functionalities, be it recording a sale, generating the list of items to be ordered or analyzing the daily and monthly revenue trends.

2.4 Operating Environment

The software will run on both Windows and Linux, however the setup and usage instructions will vary. The system should have Java installed in it and should also support MySQL. It is preferable to have Apache Netbeans installed as that will facilitate the execution, however it is not a necessity. There are no specific hardware requirements.

2.5 Design and Implementation Constraints

Some of the constraints or difficulties that may be faced during the course of the software development process are :

- A waterfall model of software development has been followed in this project. So, there is a lack of feedback from the user during the entire software development process.
- The testing phase begins after the development phase is over, hence it is quite difficult to make amends or changes to the product.
- The software will not work on mobile operating systems like Android etc.
- The storage of data can pose a problem if the total data size becomes too large.

2.6 User Documentation

The software will come with a README file which will act as the user manual. It can also be given to the user in a printed format. It will contain all the necessary instructions to setup and use the software to avail all the functionalities. The user manual (README file) will be written maintaining proper standards and conventions.

2.7 Assumptions and Dependencies

The software assumes that only a single owner will be using it so there is no option for creating multiple owner accounts. It also assumes only a single item is entered while recording a sale (quantity can be more than one), and if more than one type of item is being sold the owner can report them multiple times as different sales. Another assumption this software makes is that the owner generates the list of items to be ordered at the end of each day. Also, the owner always orders the suggested quantity of these items. Other than this, the software just assumes a working computer, installed dependencies and enough space to store the data.

3 External Interface Requirements

3.1 User Interfaces

Zulu will be a desktop application with which the shop owner will interact. The owner will initially see a login page. After logging in to the system, the owner will see a dashboard with options to add/remove items to/from the inventory, view the current items in the inventory or record a sale. The owner will also see a list of items to be ordered at the end of each day and a graph to see the daily sales for a month.

3.2 Hardware Interfaces

The only hardware component being used is the desktop/laptop on which the application will run. A good processor will help speed up the execution and queries from the database. Apart from this, there are no specific hardware interactions or components.

3.3 Software Interfaces

The backend part is written in Java. The frontend of the application is designed using the Java Swing GUI package. MySQL has been used for database management. The application will interact with the database by means of JDBC (Java Database Connector). The database will be hosted on the owner's system itself, and will be secured with a username-password combination. The database will contain the metadata for setting up the software and will also store all the information about each item in the inventory at any point of time.

3.4 Communications Interfaces

Other than the MySQL database hosted on a local server, the software does not communicate with any other entity like the internet or any other software.

4 System Features

4.1 Add an Item

4.1.1 Description and Priority

This feature enables the owner to add a new item to the inventory. Although this feature is not of the highest priority, yet it helps the functioning of a real world automobile spare parts shop as new items, parts and tools arrive in the market very frequently.

4.1.2 Stimulus/Response Sequences

After logging into the system, the owner will select an option to add a new item, in response to which, the system will ask for information in the data fields like item type, manufacturer, vehicle type, price and initial quantity. After confirmation of these details, the new item will be added in the inventory database and a success message will be displayed.

4.1.3 Functional Requirements

REQ-1 : The software should be able to perform sanity checks to ensure that the item is not already present in the inventory database.
REQ-2 : It should be able to detect if any data field has been left empty, and also ensure that the quantity of the item entered is positive.
REQ-3 : The software should add the details of the item in the inventory database.

4.2 View Inventory

4.2.1 Description and Priority

This feature enables the owner to view the items present in the inventory along with their details like item type, manufacturer, vehicle type and quantity, at any moment of time. Viewing the items in the inventory is a high priority feature in any inventory management software, as this helps the owner to get an estimate of the current stock.

4.2.2 Stimulus/Response Sequences

The owner will have an option to view the items in the inventory, on his/her home screen. After clicking this option, a table of all the items currently present in the inventory will be displayed on the screen.

4.2.3 Functional Requirements

REQ-1 : The software should be able to access the details of all the items and display them in a tabular form on the screen.
REQ-2 : If there are a large number of items, then there should be a scrollable table to view the details of all the items.

4.3 Record a Sale

4.3.1 Description and Priority

This feature allows the owner to record the details of a sale made at any point of time in a day. This feature facilitates the process of updating the quantity of each item in the inventory and hence is quite an important feature.

4.3.2 Stimulus/Response Sequences

The owner will have an option on the home screen to record a sale. He/she will then be redirected to a new page where the item which is being sold will have to be selected. First, the owner will be asked to choose the item type from a drop-down menu. Then a new drop-down menu will appear showing a list of all the manufacturers that make that item. After selecting a particular manufacturer, a new drop-down menu showing the vehicle types for the selected item type and manufacturer will appear. On choosing a vehicle type, the item selection process will be complete. The owner will also have to enter the quantity of the item being sold. On confirming the sale, it will be processed and the inventory database will be appropriately updated.

4.3.3 Functional Requirements

REQ-1 : The three-level drop down list should be such that the items appearing in the next drop-down menu change dynamically according to the item selected in the previous drop-down menu.
REQ-2 : The software should be able to ensure that the quantity being sold is positive.
REQ-3 : The software should be capable of updating the quantity of the item sold in the inventory database.

4.4 Remove Item

4.4.1 Description and Priority

This feature allows the user to remove the existence of an item from the inventory database. This is not a high priority feature, but it helps in mimicking the processes and working of a real world motor parts shop.

4.4.2 Stimulus/Response Sequences

The owner will have an option to remove a specific item from the inventory, on his/her home screen. After clicking this option, the owner will be asked to select an item in a manner similar to that described in the *Record a Sale* feature. After selecting the item and confirming, if it is present in the inventory database, then it will be removed.

4.4.3 Functional Requirements

REQ-1 : The software should be able to check that the item which is being removed exists in the inventory or not.
REQ-2 : The software should be able to delete the entry of the item from the database, and if required, it should also be able to delete the entry of the manufacturer supplying that item, provided the same manufacturer does not supply any other item.

4.5 Generate and Save Order List

4.5.1 Description and Priority

This feature enables the owner to generate a list of all the items at the end of a day, that have fallen below their threshold value and needs to be ordered. This is the feature with the highest priority as this is the task which when automated, saves a lot of time and energy.

4.5.2 Stimulus/Response Sequences

To indicate that a particular sale was the last sale of a day, the owner will select a button to indicate that the current day has ended. After this, the software will automatically find out which items are in deficit (fallen below the threshold in quantity) and hence, need to be ordered. It will generate a list of such items in the form of a table and the owner will have an option to save the list.

4.5.3 Functional Requirements

REQ-1 : The software should be able to calculate the threshold for each item at the end of a day by computing the average sale of that item.

REQ-2 : The software should display the list of items to be ordered along with their quantity and vendor address in a tabular format and provide an option to the user to save that list.

4.6 Generate Revenue and Graph

4.6.1 Description and Priority

This feature helps the owner to perform various analytics tasks like generating the revenue at the end of each day and at the end of each month, and also plotting a graph showing the sales for each day of the month. This can be considered as a secondary feature.

4.6.2 Stimulus/Response Sequences

At the end of each day, after the order list has been generated, the owner will have an option to see the revenue generated on that day. On selecting that option, the appropriate revenue will be displayed. Also, there will be an option to generate a graph showing the sales of each day of the previous month, which will also be displayed by selecting the appropriate option.

4.6.3 Functional Requirements

REQ-1 : The software should be capable of storing the revenue for each day of the last one month.

REQ-2 : The software should be able to display the daily sales for a month in a graphical format on the screen.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

The software does not require a high-end CPU, GPU or internet connection. As long as all the dependencies are installed and the system is decent enough, the software should run without any issues.

5.2 Safety Requirements

The application operates above multiple abstraction layers over the hardware and there is little possibility of damage to the owner's device. However, it is the responsibility of the owner, if he/she incurs any business losses.

5.3 Security Requirements

To prevent anyone else apart from the shop owner to use the software, the software displays a login page whenever it is loaded. Only when the owner enters the correct username and password, he/she is allowed to move to the next page where he/she can access the inventory or make any kind of changes to the data.

5.4 Software Quality Attributes

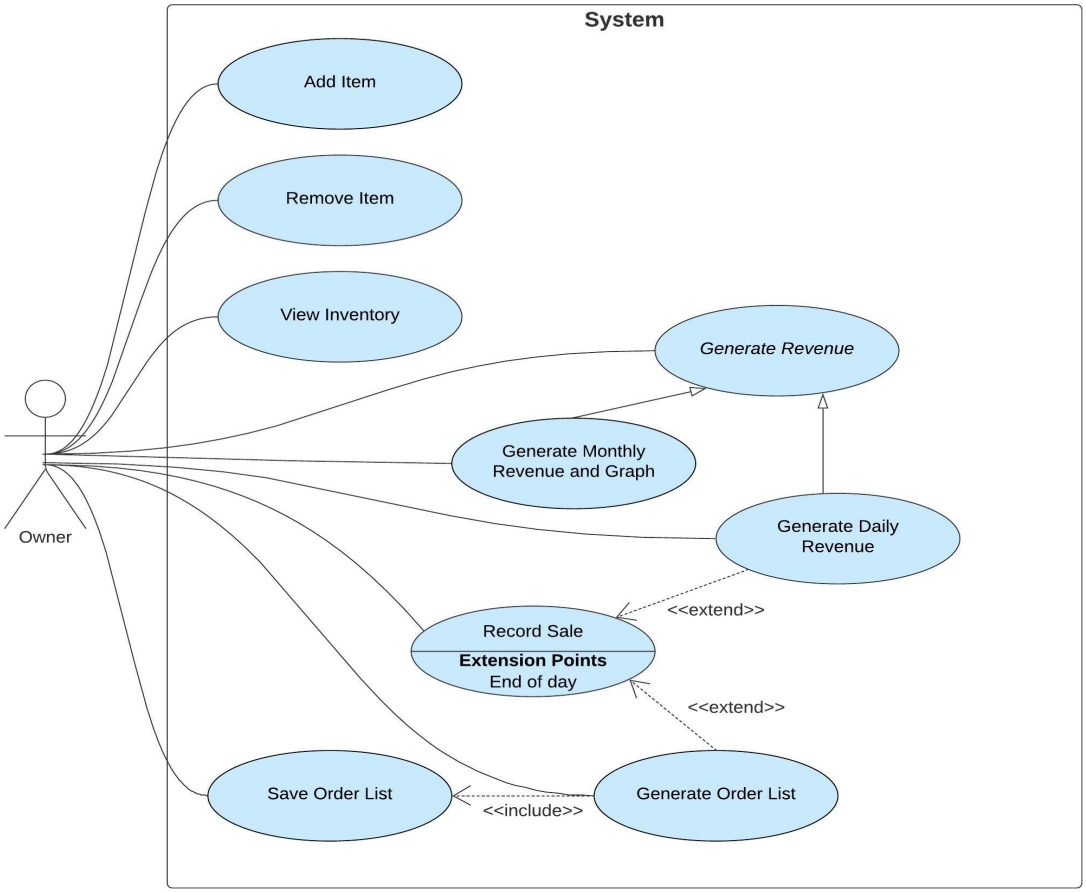
The software should be easy to maintain and user friendly. It will be able to adapt to new features by making least changes to the code. It will be easy to upgrade to later versions of the software. This will enhance the maintainability and reusability of the software. The software will support feedback once it is deployed to inculcate the newer requirements that may come ahead. The software would be flexible with any third party software that it may interact with. The user interface will be friendly and will enable easy understanding of the features and their usage.

5.5 Business Rules

The owner of the shop is allowed to use all the features of the software, as he is expected to be the sole user. Also, the software should not be outsourced to any third party without prior permission. The project holder reserves all the applicable rights of the project licenses and permissions are applicable before any industrial use.

6 Analysis Models

6.1 Use Case Diagram



6.2 Class Diagram