SOFTWARE REQUIREMENT SPECIFICATION

Assignment No: 1

Title:
Write Software Requirement Specification (SRS) for Garage Management System (GMS).

Date: 2/08/2022

Customer: General Garages

Instructor: Dr. Dhanashree Wategoankar

Remark:

Name: Shashwat Singhal

Roll:

Batch: C1

Panel: C

ASSIGNMENT NO: 1

Aim:

The aim of SRS is to specify the software product in details. In other words, it contains all necessary and important information that the product team should be aware of in order to create the software.

Problem Statement:

The garage is for different types of four wheelers. The advanced booking/appointment is done on phone. On the day of appointment as soon as a customer arrives, a job card is created to not all the problems, requirements for the vehicle. An engineer is assigned based on availability to service a vehicle. On completion of the repair/maintenance/service the engineer prepares a report based on which a bill is created. The payment is accepted in cash against the bill. Make suitable assumptions about scope and working of your Garage (write down the scope too).

Objective:

- To understand different sections of Software Requirement Specification (SRS).
- To understand functional requirements of the system
- To understand performance requirements of the system
- To apply design constraints and appropriate validation on the system

Theory:

Software Requirement Specification (SRS) document usually contains a software vendor's understanding of a customer's software requirements. This document ensures that the software vendor and the customer are in agreement as to the features required in the software system being built.

SRS is created after the initial requirement elicitation phase in which software vendor interacts with the customer to understand the software needs. Usually, SRS documentation is prepared by a business analyst who has some technical background.

1. Introduction

1.1 Purpose

The main objective of this document is to illustrate the requirements of the project Garage Management System. The document gives the detailed description of the both functional and non-functional requirements proposed by the client.

The purpose of this project is to provide a friendly environment to maintain the details of customers and their cars. The main purpose of this project is to maintain easy Customer Service Management with a provision for adding multiple vehicles using computers and to provide different reports. This project describes the hardware and software interface requirements using DFD, ER diagrams and UML diagrams.

1.2 Scope of Development Project

Garage Management System is basically updating the manual garage system into an internet-based application so that the users can know the details of their accounts, availability of mechanics and maximum limit of cars in garage.

The project is specifically designed for handling vehicle admissions, repairs, spare part sales and all other aspects of running a motor vehicle garage. The product will work as a complete user interface for garage management process and garage usage for customers. Garage management software helps to achieve auto shop goals by automating procedures, tracking customer vehicle history, and maintaining a database of other dealers and customers.

The project can be easily implemented under various situations. We can add new features as and when we require, making reusability possible as there is flexibility in all the modules.

1.3 Definitions, Acronyms and Abbreviations

GMS	GMS (garage management system)
SRS	SRS (software requirements specification) is a document of the software system to be developed, that discusses the functional and nonfunctional requirements.
PC	PC stands for personal computer.
HDD	HDD stands for Hard Disc Drive.
RAM	RAM (Random Access Memory). Computer memory in which data can be changed or removed and can be looked at in any order
IE	IE stands for Microsoft Internet Explorer.
JAVA	Java is a programming language and computing platform first released by Sun Microsystems in 1995. It has evolved from humble beginnings to power a large share of today’s digital world, by providing the reliable platform upon which many services and applications are built.

	New, innovative products and digital services designed for the future continue to rely on Java, as well.
SQL	SQL stands for Structured query Language. It is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e., data incorporating relations among entities and variables.
ER	Am ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model, that defines a data or information structure which can be implemented in a database, typically a relational database.
UML	The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.
IDE	An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger.

IEEE	IEEE stands for Institute of Electrical and Electronics Engineers. IEEE, an organization dedicated to advancing innovation and technological excellence for the benefit of humanity, is the world's largest technical professional society. This is designed to serve professionals involved in all aspects of the electrical, electronic, and computing fields and related areas of science and technology that underlie modern civilization.
HTML	The Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. HTML can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.
CSS	Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). It is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

1.4 List of Stakeholders

- Staff, Customers, Professional Engineers
- Project Manager
- Requirements Engineer
- System Administrator

1.5 References

Books

- Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson.
- Software Requirements (Microsoft) Second Edition by Karl E. Wiegers.
- Software Engineering: A Practitioner’s Approach Fifth Edition by Roger S. Pressman.

Websites

- https://en.wikipedia.org/wiki/Software_requirements_specification
- https://www.w3schools.com/
- https://developer.mozilla.org/en-US/
- https://getbootstrap.com/

2. Overall Descriptions

2.1 Product Perspective

This project is designed to be used by garage employees of several automotive plants to record and track services provided in the garage. There are several constraints the system will need to operate under. First, the end user system will need to run on a Windows computer, while the backend will need to run on appropriate server hardware. The end user system will need to present a user interface that allows a user to select the name, type of vehicle, and types of services. In addition, users will be able to query server data by running reports. The frontend and backend will communicate using HTTP requests to both send and receive data from the database.

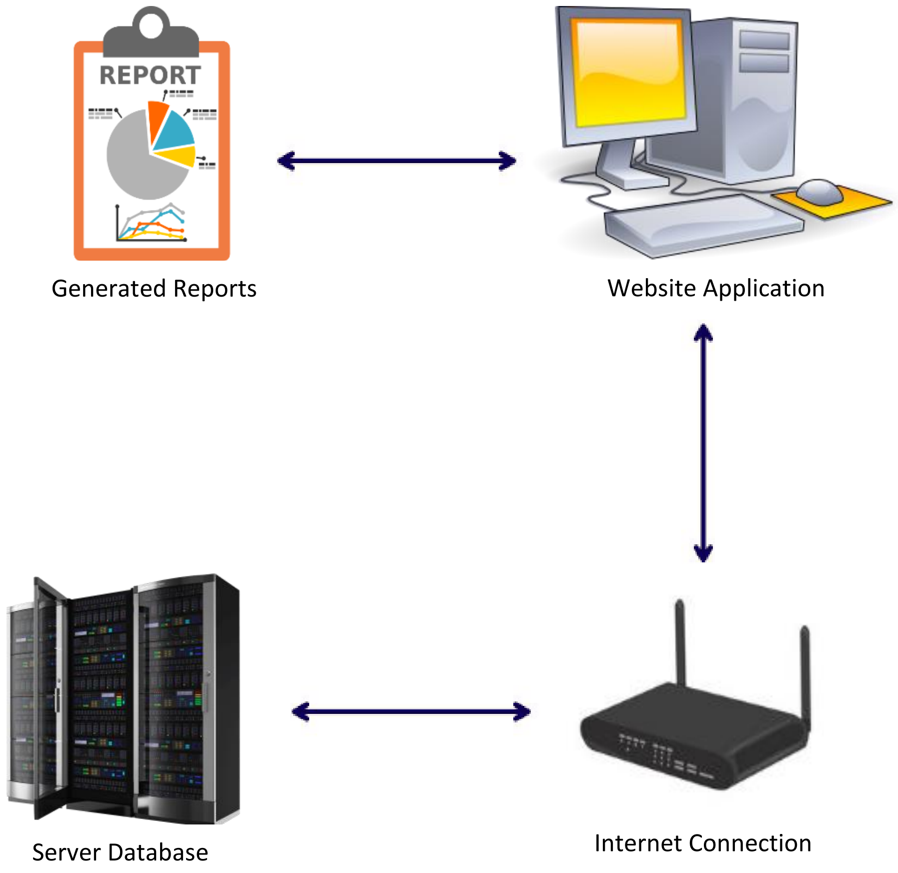

Fig 1: Pictorial representation of the bigger picture

2.2 Product Function

The Online Garage System provides online real time information about the cars available in the Garage and the user information. The main purpose of this project is to reduce the manual work. This software is capable of managing **Accounting, Billing and invoicing, Customer database, Inventory control, Labor rates, Maintenance scheduling, Parts management, Quotes or estimates, Scheduling, Service history, Work order management, Push notifications, Payment options.** The member's status of issuing/return of vehicle is maintained in the garage's database. The valid members are also allowed to view their account information.

2.3 Operating Environment

The product will be operating in windows environment. The basic input devices required are keyboard, mouse and output devices are monitor, printer, versatile device that can be carried when a greater number of vehicles are in queue for servicing at a time etc.

2.4 Development Environment

Software Configuration: -

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows XP, Windows 10
Language: Java Runtime Environment, Net beans 7.0.1 (front end)
Database: MS SQL Server (back end)

Hardware Configuration: -

Processor: Core(R), 11th Generation Quad-core CPU
Hard Disk: 1TB
RAM: 256 MB or more

2.5 Data Requirement

The inputs consist of the query to the database and the output consists of the solutions for the query. The output also includes the user receiving the details of their accounts. In this project the inputs will be the queries as fired by the users like create an account, name of vehicle, type of vehicle etc. Maintaining an accurate database of all members, vehicles, spare parts purchased and sold, is necessary to improve the overall of productivity of the garage.

3. External Interface Requirement

The purpose of this section is to identify and document interfaces and interaction of the software with external entities in detail.

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing the details of the book. The system should be able to interact with the user management module and external hardware devices like printer, barcode reader etc.

3.1 Functional requirements

Functional requirements are the following:

1. The GMS should store all information about vehicle details, garage managers and members.
2. The GMS allow searching items by Name, Type or keywords.
3. The GMS should allow garage managers to add, delete and modify items in database, and check availability of the items.
4. The GMS should allow members to view or update their personal information
5. The GMS should generate request's reports for garage managers every day, on base of which managers could make decisions about buying or selling the spare parts (Stock Check).

4. System Features

The users of the system should be provided the surety that their account is secure. This is possible by providing: -

- User authentication and validation of members using their unique member ID.
- Proper monitoring by the administrator which includes updating account status, showing a popup if the member attempts to book an appointment for service that does not matches the time slot provided.
- Proper accountability which includes not allowing a member to see other member's account. Only administrator or whom so ever managing the GMS will have the rights to see and manage all member accounts.

5. Other Non-functional Requirements

5.1 Performance Requirement

The proposed system that we are going to develop will be used as the Chief performance system within the different garage, which interacts with the customers, garage managers and engineers. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the university.

- The performance of the system should be fast and accurate.
- Garage Management System shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus, it should have inbuilt error testing to identify invalid username/password.
- The response should be fast enough to avoid users' response collisions.

5.2 Safety and Security Requirement

- System will use secured database that will keep you safe from data leakages.
- Users can just read information but they cannot edit or modify anything except their personal and some other information, so that we can maintain privacy of our customers.
- System will have different types of users and every user has access constraints.
- Proper user authentication should be provided, so that we can serve you better in terms of privacy.
- Passwords must not be shared with anyone.

Conclusion:

Hence, we can conclude that we must prepare the SRS document to describe a project in most non-technical format.

FAQs: (Kindly refer for pictures of the handwritten answers at end)

1. What is the need for an SRS document?

Ans.

The Software Requirements Specification (SRS) document serves to outline the technical requirements needed by the developers working on creating the system for automating their client's manual system. This document provides specific insight such as requirements, constraints, and detailed explanations of the proposed system to combat the client's problem. The intended audience for this document is the developers working on creating the system for the client. The developers and technology experts make sure that system is "technically and economically feasible".

Every business owner aims to calculate and analyze all the nuances of implementing a project, consider all the risks, draw up a schedule, and determine a budget. Therefore, it is necessary to write software requirement documentation. This document will help to estimate the costs, ensure the timely delivery of the project, determine the testing strategy, and help create the basis for effective improvements and corrections.

2. What are the contents of an effective SRS document?

Ans.

Key components of an effective SRS document are: -

1) Introduction
2) Overall description
3) System/Product features
4) External Interface
5) Non – functional requirements
6) Appendixes

3. What are the qualities of good SRS?

Ans.

1) All requirements must be easily measurable so that all specifications can be verified, avoiding fatal errors in the future.
2) Viable - Budgets, deadlines, technologies, the current situation – all these parameters should be considered while drafting the document. Do not hope for a lucky accident like the situation on the market changing radically, analyze the realistic situation now and in the near future.
3) No implementation constraints - Software requirement documentation should be detailed so that you can calculate everything up to the completion of the product. However, it is also necessary to consider various nuances and leave some room for changes.
4) Accurate - Requirements spelled out as precisely as possible, no ambiguity and subjectivity, no vague wording like ‘probably’, ‘better’, ‘the nearest’. Specifics only.

Name: Shashwat Singhal

Panel: C

Date ____

Page ____

9/8/22

* Assignment 1 *

Q.1] What is the need for a SRS document?

Ans. The Software Requirements Specification (SRS) document serves to outline the technical requirements needed by the developers working on creating the system for automating their clients manual system. This document provides specific insight such as requirements, constraints, and detailed explanation of the proposed system to combat the clients problem. The intended audience for this document is the developers working on creating the system for the client. The developers and technology experts make sure that system is "technically and economically feasible".

Every business owner aims to calculate and analyze all the difficulties possible of implementing a project, consider all the risks, draw up a schedule, and determine a budget. Therefore, it is necessary to write software requirement documentation. This document will help to estimate the costs, ensure the timely delivery of the project, determine the testing strategy, and help create the basis forx effective improvements and corrections.

Q.2] What are the contents of an effective SRS document?

Ans. Key components of an effective SRS document are:-

1) Introduction
2) Overall description
3) System / Product features
4) External Interface
5) Non-functional requirements
6) Appendines

Q.3] What are the qualities of good SRS?

Ans. Qualities of good SRS are:-

1. All requirements must be easily measurable so that all specifications can be verified, avoiding fatal errors in the future.

2. Viable- Budgets, deadlines, technologies, the current situation- all these parameters should be considered while drafting the document. Do not hope for a lucky accident like the situation on the market changing radically, analyze the realistic situation now and in the near future.

3. No implementation constraints- Software requiremen documentation should be detailed so that you can calculate everything up to the completion of the product. However, it is also necessary to consider various nuances and leave some room for changes.

4. Accurate- Requirements spelled out as precisely as possible, no ambiguity and subjectivity, no vague wording like 'probably', 'better', 'the nearest'. Specifics only