SOFTWARE REQUIREMENT SPECIFICATION

Software Requirements Specification(SRS)

- SRS is a document that describes what the software will do and how it will be expected to perform.

- A typical SRS includes:
  - ✓ A purpose
  - ✓ An overall description
  - ✓ Specific requirements

Format of SRS

1. Introduction
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, Acronyms, and Abbreviations
   - 1.4 References
   - 1.5 Overview
2. Overall Description
   - 2.1 Product Perspective
   - 2.2 Product Functions
   - 2.3 User Characteristics
   - 2.4 General Constraints
   - 2.5 Assumptions and Dependencies

3. Specific Requirements
   - 3.1 External Interface Requirements
     - 3.1.1 User Interfaces
     - 3.1.2 Hardware Interfaces
     - 3.1.3 Software Interfaces
     - 3.1.4 Communication Interfaces
   - 3.2. Functional Requirements
     - 3.2.1 Mode 1
       - 3.2.1.1 Functional Requirement 1.1
       - :
       - 3.2.1.*n* Functional Requirement 1.*n*
     - :
     - 3.2.m Mode *m*
       - 3.2.*m*.1 Functional Requirement *m*.1
       - :
       - 3.2.*m*.*n* Functional Requirement *m*.*n*
   - 3.3 Performance Requirements
   - 3.4 Design Constraints
   - 3.5 Attributes
   - 3.6 Other Requirements

Introduction

- **Purpose**
  - ✓ delineate the purpose of the particular SRS
  - ✓ specify the intended audience for the SRS
  - **Examples:**
  - This document describes the software requirements specification (SRS) for the ……System that provides the access and……
  - The purpose of this document is to retrieve and analyze the ideas that define the product and requirements that the user needs. This document describes the details of our product, its parameter, and its goals.

Introduction – Contd…

- **Scope**
  - ✓ **Boundary of your project**
  - ✓ Explain what the product will and will not do
  - ✓ Describe application of the software - goals and benefits
- **Definition, Acronyms, Abbreviations**
- **References – books with author name, websites, YouTube links**
- **Overview**
  - ✓ describe what the rest of the SRS contains
  - ✓ how the SRS is organized

Overall Description

- **Product Perspective**

✓ State whether the product is independent and totally self contained

✓ Describe relation with other products if any.

✓ If the product is component of a larger system then: – describe the functions of each component of the larger system and identify interfaces

•

✓ Define how the product contributes to fulfilling user needs.

✓ Define the product's functional behavior, qualities, and constraints in response to user needs.

- **Product Function**

❖ Provide a summary of functions the SW will perform

❖ The functions should be organized in such a way that they are understandable by the user

❖ Block diagrams can be used

User Characteristics

- Describe those general **characteristics** of the intended groups of users of the product including **characteristics** that may influence usability, such as
  - ✓ educational level
  - ✓ experience
  - ✓ technical expertise
  - Example: The user is expected to be Internet literate and to be able to use email with attachments.

General Constraints

- Describe any **general** constraints imposed on the product.
- **Example -**

✓ The application must execute on an IBM-PC or 100% compatible computer

✓ there must be enough RAM, disk storage, etc

✓ The OS must be DOS 3.3 or later.

Assumptions and dependencies

- Example:
  - ✓ User has an active internet connection
  - ✓ User runs an operating system which supports internet browsing
  - ✓ Website will not be violating any internet ethnic rules and won't be blocked by the telecom companies.

Specific Requirements

- **External Interface Requirement**
  - ✓ **User Interface** – name the front end , back end . The application GUI provides menus, toolbars, buttons, panes, containers, grids allowing for easy control by a keyboard and a mouse.
  - ✓ **Hardware Interface** – OS , GPS, Android device
  - ✓ **Software Interface** – s/w details, database, UPI services if any
  - ✓ **Communication Interface** – local network protocols, bluetooth, i/o interfaces etc.

- **Functional Requirement**
- Format

3.2 Functional Requirements
- 3.2.1 Module1
  - 3.2.1.1 Functional Requirement for Module1
- 3.2.2 Module 2
  - 3.2.2.2 Functional Requirement for Module 2

- Examp

3.2.1 Upload Resume	
3.2.1.1 First Name	varchar2
3.2.1.2 Last Name	varchar2
3.2.1.3 E.mail Address	varchar2
3.2.1.4 password	password
3.2.1.5 current location	varchar2
3.2.1.6 mobile number	long int
3.2.1.7 total experience	integer
3.2.1.8 Key Skills	varchar2
3.2.1.9 Industry	varchar2
3.2.1.10 Function	varchar2
3.2.1.11 Resume	file

- **Performance Requirements**
  - **Static Requirement –whether it is scalable or supports multi users at a time**
  - **Dynamic Requirement – execution behavior**
  - **Fault Tolerance**
  - **Security**
- **Examples:**
  - The application should load and be usable within 3 seconds
  - The application should update the interface on interaction within 2 seconds
  - The database should be normalized to prevent redundant data and improve performance
  - The database should be distributed to prevent outages

- **Design Constraints**
  - Specify all the constraints imposed on design.
  - It includes environmental hardware limitations, security fault tolerance etc.
  - Describe any items or issues that will limit the options available to the developers.
- Example:
  - ✓ corporate or regulatory policies
  - ✓ hardware limitations (timing requirements, memory requirements)
  - ✓ interfaces to other applications
  - ✓ specific technologies, tools, and databases to be used
  - ✓ parallel operations
  - ✓ language requirements
  - ✓ communications protocols
  - ✓ security considerations
  - ✓ design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software)

Attributes

- The attributes section specifies some of the overall attributes that a system should have.

- **Reliability**
- **Examples:**
  - ✓ **The reliability of the system is the back up of the database which is continuously maintained and updated to reflect the recent changes.**
  - ✓ **The reliability of the system depends on the reliability of each component of the system**

Attributes - Contd…

- **Availability**
- **Examples:**
  - ✓ **The system should be available all the time only restricted by the down time of the server on which the system runs.**
  - ✓ **In case of hardware failure or database corruption a replacement page should be shown.**
  - ✓ **Also in case of database corruption backup of database should be retrieved from the server and saved by the administrator.**

Attributes - Contd…

- **Security**

- **Examples:**
  - ✓ The system must automatically logout all customers after a period of inactivity.
  - ✓ The system's backend servers shall only be accessible to authenticated administrators.
  - ✓ Sensitive data will be encrypted before sending over the internet.

Attributes - Contd…

- **Maintainability**

- **Examples:**
  - ✓ In case of failure, re-initialization of the program will be done.
  - ✓ Software design is done with modularity in mind so that maintainability can be done easily.

Attributes - Contd…

- **Portability**

- **Examples:**
  - ✓ The application is HTML and script based. So the end user part is fully portable.
  - ✓ Any web browser should be able to run the features of the system
  - ✓ The system can be used on any platform.

Other Requirements

- **Safety Requirements**
- **Examples:**
  - ✓ Databases should use sharding to be redundant to prevent loss of data.
  - ✓ Backups of the databases should be done hourly and be kept for one week
- **Security Requirements**
- **Examples:**
  - ✓ Any keys used for the API should be stored securely.
  - ✓ Only the ….. API should be able to connect to the databases.
  - ✓ Databases should be behind a firewall.