Lab 1 : Develop requirement specification for given problem.

Aim: To develop requirement specification for given problem

Description: A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do.

Procedure:

Step 1:

1. **Introduction**

Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.

2. Intended Audience and Reading Suggestions

Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. **Project Scope**

Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available. Refer to it rather than duplicating its contents here.

Step2:

1. Product Perspective

Describe the content and origin of the product being specified in this SRS. Forexample, state whether this product is a follow-on member of a

product family, areplacement for certain existing systems, or a new, self-contained product.

Features

Summarize the major features the product contains or the significant functions that itperforms or lets the user perform. Only a high level summary is needed here.Organize the functions to make them understandable to any reader of the SRS.

User Classes and Characteristics

Identify the various user classes that you anticipate will use this product. User classesmay be differentiated based on frequency of use, subset of product functions used,technical expertise, security or privilege levels, educational level, or experience.Describe the pertinent characteristics of each user class.

Operating Environment

Describe the environment in which the software will operate, including the hardwareplatform, operating system and versions, and any other software components orapplications with which it must peacefully coexist.

2. Design and Implementation Constraints

Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory poloies; 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).

Step 3:

1. System Features

This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whether makes the most logical sense for your product.

Step 4:

1. External Interface Requirements

1.1. User Interfaces

Describe the logical characteristics of each interface between the softwareproduct and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layoutconstraints, standard buttons and function (e.g. ,help)that will appear on everyscreen, keyboard shortcuts, error message display standards, and so on.

1.2. Hardware Interfaces

Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This mayinclude the supported device types, the nature of the data and controlinteractions between the software and the hardware, and communicationprotocols to be used.

1.3. Software Interfaces

Describe the connections between this product and other specific softwarecomponents (name and version), including databases, operating systems, tools,libraries, and integrated commercial components. Identify the data items ormessages coming into the system and going out and describe the purpose ofeach. Describe the services needed and the nature of communications.Communications Interfaces.

Describe the requirements associated with any communications functionsrequired by this product, including e-mail, web browser, and network savercommunications protocols. Electronic forms, and so on.

2. Nonfunctional Requirements

2.1 Performance Requirements

There are performance requirements for the product under variouscircumstances, state them here and explain their rationale, to help thedevelopers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make suchrequirements as specific as possible.

2.2 Safety Requirements

Specify thaw requirements that are concerned with possible loss, damage, orharm that could result from the use of the product. Define any safeguards oractions that must be taken, as well as actions that must be prevented.

2.3 Security Requirements

Any requirements regarding security or privacy issues surrounding use of theproduct or protection of the data used or created by the product. Define anyuser identity authentication requirements.

Software Quality Attributes :

Specify any additional quality characteristics for the product that will be important to eitherthe customers or the developers. Some to consider are adaptability, availability, correctness,flexibility, interoperability, maintainability, portability, reliability, reusability, robustness,testability, and usability.