SRS

In order to form a good SRS, here we will see some points which can be used and should be considered to form a structure of good SRS. These are as follows:

1. Introduction
    - **(i)** Purpose of this document
    - **(ii)** Scope of this document
    - **(iii)** Overview
2. General description
3. Functional Requirements
4. Interface Requirements
5. Performance Requirements
6. Design Constraints
7. Non-Functional Attributes
8. Preliminary Schedule and Budget
9. Appendices

Software Requirement Specification (SRS) Format as name suggests, is complete specification and description of requirements of software that needs to be fulfilled for successful development of software system. These requirements can be functional as well as non-requirements depending upon type of requirement. The interaction between different customers and contractor is done because it's necessary to fully understand needs of customers.

Depending upon information gathered after interaction, SRS is developed which describes requirements of software that may include changes and modifications that is needed to be done to increase quality of product and to satisfy customer's demand.

1. **Introduction:**

- **(i) Purpose of this Document –**
  At first, main aim of why this document is necessary and what's purpose of document is explained and described.
- **(ii) Scope of this document –**
  In this, overall working and main objective of document and what value it will provide to customer is described and explained. It also includes a description of development cost and time required.
- **(iii) Overview –**
  In this, description of product is explained. It's simply summary or overall review of product.

2. **General description:**
   In this, general functions of product which includes objective of user, a user characteristic, features, benefits, about why its importance is mentioned. It also describes features of user community.

3. **Functional Requirements:**
   In this, possible outcome of software system which includes effects due to operation of program is fully explained. All functional requirements which may include calculations, data processing, etc. are placed in a ranked order.

4. **Interface Requirements:**
   In this, software interfaces which mean how software program communicates with each other or users either in form of any language, code, or message are fully described and explained. Examples can be shared memory, data streams, etc.

5. **Performance Requirements:**
   In this, how a software system performs desired functions under specific condition is explained. It also explains required time, required memory, maximum error rate, etc.

6. **Design Constraints:**
   In this, constraints which simply means limitation or restriction are specified and explained for design team. Examples may include use of a particular algorithm, hardware and software limitations, etc.

7. **Non-Functional Attributes:**
   In this, non-functional attributes are explained that are required by software system for better performance. An example may include Security, Portability, Reliability, Reusability, Application compatibility, Data integrity, Scalability capacity, etc.

8. **Preliminary Schedule and Budget:**
   In this, initial version and budget of project plan are explained which include overall time duration required and overall cost required for development of project.

9. **Appendices:**
   In this, additional information like references from where information is gathered, definitions of some specific terms, acronyms, abbreviations, etc. are given and explained.

Absents: 2,
12,13,16,17,18,21,28,31,32,33,35,36,44,45,51,55,56,60,L5,