Software Requirements Specification

for

Cyber Cafe Management System

Version 2.0 approved

Vidhi Kothari

Mathew Medayil

T V K Jahnavi

National Institute of Technology

07/02/19

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
1st Srs	17-01-19	Addition of Introduction	1

1. Introduction

1.1 Purpose

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals.This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1.2 Document Conventions

The document is using simple conventions headings for different sections is size 18 cm in bold. These are numbered as well.The sub headings are numbered by heading number followed by "." and the subheading number. Sub headings are of size 14 cm and the details under it is of size 11 cm. Page number is marked with headings and subheadings in table of content which allow easy access to the material.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document is project developers and users who wish to view the project requirements and specifications. The rest of the documents has been highlighted the overall description, functional requirement, non-functional requirement, deployment strategy and tentative development schedule.

1.4 Product Scope

This project is intended to be used in Cyber Cafe . All cyber cafes have some basic needs likeable to control the systems that are being rented to the customers and are charged on timely basis.This project only have server part which runs on Admin system .The Admin should have control over the usage of the client system and keep track of what the client is doing. Also we can have a column for calculation of extra charges if the client wants to take a printout or a photocopy.

1.5 References

The references are :

1. SRS_IEEE_TEMPLATE
2. E-STORE PROJECT BY MARVEL ELECTRONICS AND ENTERTAINMENT
3. SRS BY KARL E WIEGERS

2. Overall Description

2.1 Product Perspective

Let's say someone goes to the cyber cafe and tries to access dangerous information which is intended for wrong purpose .Example can be ,if someone tries to access information like how to make a bomb at home .Other realistic examples can be if someone tries to fool girls on social media or tries to cyber attack from the cyber cafe then it is very important for the owner to keep record of the customers who visited the shop and what activities they performed.It is also difficult for the owner to keep track of the time limit of different users if there are many terminals.
Cyber Cafe management system is app which provide the ease to the owner to distribute the customer to different terminals in best possible way and to know when the session is complete .Its major advantage is that it records the type of terminals accessed by the customers and full details of the customers are recorded in the database so that if any enquiry is needed by anyone later it can be systematically done.It works on the admin's front so user cannot manipulate or access any part of the information in the system.

2.2 Product Functions

Product functions will include following:

1. *Making account and its interface*
2. *Making customer information interface*
3. *Making session end information interface*
4. *Making system information related interface*

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may 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. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

The software will run on the any operating system which can run JAVA applications .It is possible if they can run Java Virtual Machine .Popular examples would be windows installed with Java and ubuntu or any linux operating system.

2.5 Design and Implementation Constraints

The operating system must have the functionality to run java virtual machine. Design should be user friendly so that the admin can use it without much difficulty. It should be designed such that history before some time limit gets deleted automatically. Implementation will use multi-thread programming so it will use the concept of parallel processing.This can arise problems of thread crash .This should be handled carefully and implementation should be thread safe.

2.6 User Documentation

The user will be provided with be provided the help documentation which contains the way to handle the interfaces

2.7 Assumptions and Dependencies

The software will use the machine local time so it is assumed that no tampering is done with that clock time otherwise wrong data will be recorded and this can be misused by anyone

3. External Interface Requirements

3.1 User Interfaces

It tells about all the different interfaces present in the system.

The Main screen of user will contain following buttons

1. *Account Register*
2. *Login*
3. *Logout*

3.1.1 Account Register

It will contain two text boxes for entering the username and password .It will have one submit button for submitting the information .

3.1.2 Login

It will contain two text boxes for entering the username and password .It will have one submit button for submitting the information .

It will open another interface if information entered matches

1. *Enter System Info*
2. *Show System Info*
3. *Enter Customer Info*
4. *Show Customer Info*
5. *Amount Received*

3.2 Hardware Interfaces

It is just that the system must have sufficient memory for the storage of the details in the database.

NOTE: Other Interfaces which are not present are not mentioned .

4. 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, whatever makes the most logical sense for your product.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>