Software Requirements Specification

for

Furniture Rental Store System

Version 1.0 approved

Prepared by :-

Vinod Kumar (20CS10074) dhyawanavinod2020@gmail.com

Sonu Kumar Yadav (20CS10061) sonukumaryadav18199@gmail.com

Nirbhay Kumar (20CS10040) nirbhay.jnvj@gmail.com

IIT Kharagpur

18th March 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

A website where users may look for and rent furniture. When it comes to furniture rental activities, the FRSS keeps track of everything. It saves information on the furniture, such as the type, related photographs, and so on. It is very beneficial for: rental price, cost price, age, and so on. those who are looking to renovate their home following a move, and I don't want to spend a lot of money on furnishings.

1.2 Document Conventions

This SRS Document has been written using Free and Open Source writing tools such as LibreOffice typed in a monospace font Source Code Pro. The font size used is 12 for text and 14 for headings. All headings are highlighted appropriately in bold. The document is prepared using US English convention.

1.3 Intended Audience and Reading Suggestions

This paper details all of the software's technical and non-technical features. Its purpose is to help developers and other end users understand the software's motive and purpose. It has a lot of complexities in terms of execution. Anyone who wishes to make use of the The software can read the relevant sections of the paper, as well as a list of keywords. This is listed on Page 2 in the Table of Contents.

1.4 Product Scope

FRSS is a desktop application designed for those who want to rent furniture but don't want to pay a lot of money for it. The application would be built on a client-server basis, with users interacting with a graphical user interface (GUI) that is supplied by a backend server that contains furniture and user data.

1.5 References

This document is based on the IEEE Computer Society's IEEE Recommended Practice for Software Requirements Specification (IEEE Std 830-1998).

2. Overall Description

2.1 Product Perspective

Product Perspective It's a brand-new self-contained product that was inspired by people who spend exorbitant amounts for furniture while relocating to a new location and only need it for a short time. This project addresses this issue by allowing people to rent furniture at reasonable rates for as long as they need it.

2.2 Product Functions

Product Functions Give furniture on a low-cost rental basis or on a loan basis with cheap interest rates.

2.3 User Classes and Characteristics

- Admin
    - The person who owns the furnishings in the inventory.
    - When a new product is released, the administrator can add it to a catalogue.
    - On his dashboard, the administrator may check the overall earnings and investment.
    - If the number of items in the inventory goes below a certain threshold, an alert is issued to the administrator.
    - At any moment, the administrator can insert or adjust the price of any item.
- Customer
    - Customers can borrow furniture at a certain interest rate, which is determined by whether he has previously borrowed furniture or not.
    - Customers may view their order history as well as the amount owed to them.
    - Customers can provide product reviews and comments.
    - The consumer has the option of taking any of the furnishings in the catalogue as well as the inventory.

2.4 Operating Environment

The program includes a client-server approach, with users interacting with a GUI web application as clients. Tkinter / pyqt5 is used to create the user interface, while SQL is used to manage the database. Because Python and SQL are cross-platform software, the app will run on Windows, Mac OS X, and Linux.

2.5 Design and Implementation Constraints

There will be four major constraints in developing this product:

● The user interface must be intuitive enough so that no training is required by customers, members, or store personnel.

● A good form of integration between the database and the application.

● Making real-time updates to the database and allowing access at the same time.

● Memory requirements should not be high, a limited amount of memory can cause issues if the database is too large.

- The software and database will be password protected so that no unauthorized person cannot *access it*.

2.6 User Documentation

The programme will come with a user manual that will allow users to understand all of the application's components and operate it as intended.
All APIs, classes, and methods will be documented and offered online for developers and other technical users.

2.7 Assumptions and Dependencies

Python 3.8 or above , Tkinter , PIL , MySQL , Matplotlib.

3. External Interface Requirements

3.1 User Interfaces

The user will interface with the product via a desktop application. Before they may see their own dashboards, the client and admin must first log in. The client may rent or borrow furniture, evaluate or provide comments on the things, and then return them. The administrator, on the other hand, has the ability to establish and remove client accounts, add new goods to the inventory, evaluate returned furniture, and review the company's overall earnings and investments.

3.2 Hardware Interfaces

To support multiple threads for worker processes, the backend server will need a competent processing unit. There are no special hardware requirements for the user if he or she has access to adequate internet connectivity via a suitable PC or laptop.

3.3 Software Interfaces

The CRM will be used to interface with the database (SQL server). A username-password combination will be used to protect the database. There will be three databases – customer information, admin information, and furniture information – to increase software quality.

3.4 Communications Interfaces

4. System Features

4.1 User Authentication – Sign Up & login

4.1.1 Description and Priority

To utilise FRSS, a user must first create an account by signing up for the application. This operation is critical since it creates the user class that is required for all other tasks.

4.1.2 Stimulus/Response Sequences

To get started, the user must fill out a form with basic information such as their name, Email address, username, and password. The system will send a verification email to a new user as soon as they sign up to guarantee that no fraudulent users get access to the system.

4.1.3 Functional Requirement

Signup: As soon as the user provides information and the user's validity is validated by email, the user becomes a member. In the database, a new user is created. Then user can perform the desired actions.

Login: Existing users can log in using their credentials once the system judges it to be useful.

4.2 Addition of new products to inventory

4.2.1 Description and Priority

The admin can add a new product to the inventory when it is launched

4.2.2 Stimulus/Response Sequences

If a new piece of furniture is released, the administrator can add it to the inventory.

4.2.3 Functional Requirement

The administrator enters a query into the furniture database to insert a new item

4.3 Price Flexibility

4.3.1 Description and Priority

Any item's price can be entered or changed by the administrator at any moment

4.3.2 Stimulus/Response Sequences

If the market price fluctuates, the admin can change the price of the furniture.

4.3.3 Functional Requirement

Price change: The administrator searches the furniture database for the item whose price has to be modified and edits or updates the item's pricing.

4.4 Searching and Renting Furniture

4.4.1 Description and Priority

Any furniture from the catalogue, as well as inventory, can be taken by the purchaser.

4.4.2 Stimulus/Response Sequences

Customer walks to the furniture department, looks for anything they like, and leases it if it's in stock.

4.4.3 Functional Requirement

Renting Furniture: When a client wishes to rent furniture, a query is run against the database to see whether the item is available for rent. If it is, the customer can rent it, and the furniture is then added to the customer database for that user.

4.5 Notification on less amount of item

4.5.1 Description and Priority

When the count of any item goes below a certain threshold, the admin is alerted.

4.5.2 Stimulus/Response Sequences

A message is issued to all administrators on the problem, and specific actions may be taken as needed. For example, the item might be completely removed from the database or new items of that type could be added.

4.5.3 Functional Requirement

Information to be sent: The system notifies all administrators about the problem.

4.6 Loan on furniture

4.6.1 Description and Priority

The consumer can borrow furniture at a certain interest rate, depending on whether he has previously borrowed furniture.

4.6.2 Stimulus/Response Sequences

The consumer can only borrow furniture if he has already borrowed furniture.

4.6.3 Functional Requirement

Loan on the following items: The consumer has the option of taking out a loan on furnishings.
Interest rate calculation: If more furniture has been taken in the past, the interest rate is low.

4.7 Viewability of past order history and amount due

4.7.1 Description and Priority

The customer may view the history of previous orders as well as the outstanding balance.

4.7.2 Stimulus/Response Sequences

The customer may access his profile to view his previous orders and history, as well as the amount.

4.7.3 Functional Requirement

History of the Profile: When a client accesses his profile, a query is sent to the customer database, and information about his previous orders is gathered.

Due Amount: When a client accesses his profile, a query is sent to the customer database, and information about his or her outstanding balance is collected.

4.8 Verification/readdition of products on return

4.2.1 Description and Priority

The administrator can inspect the condition of returned items and take appropriate action.

4.8.2 Stimulus/Response Sequences

The administrator inspects the merchandise upon return and, depending on the condition, may seek full payment from the consumer if necessary. If not, the goods may be returned to the inventory with a ten percent rental fee decrease.

4.8.3 Functional Requirement

Payment must be demanded: If the product is not in excellent working order, the admin may demand full payment before it is removed from the database.

Re-inventorying: If the product is discovered to be in good condition, the admin may re-inventory it.

Price reduction: After a consumer has utilised the product for a year, the administrator reduces the price by 10%.

4.9 Review/Feedback

4.9.1 Description and Priority

The consumer has the option to provide a comment or feedback on the item he has leased.

4.9.2 Stimulus/Response Sequences

After using the item, the consumer can provide comments on the service and the product.

4.9.3 Functional Requirement

Items' feedback: The consumer has the option of providing feedback on the item.

4.10 Profit and Investments

4.2.1 Description and Priority

In the form of a graph, the admin may see the company's entire profit and investments.

4.10.2 Stimulus/Response Sequences

When an item is rented, the rental price is added to the total earnings, and when an item is purchased or added to the inventory, the cost is added to the investments.

4.10.3 Functional Requirement

Profit calculation: The overall profit is determined by the entire amount of rent paid on the furniture.
Investment calculation: The entire amount of money spent by the firm on purchasing the things is used to determine the investment.
Graphing the data: The admin is presented a graph depicting the company's earnings and investment.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The user should be able to query databases fast and receive suitable responses. This may be accomplished by utilising MySQL to strike the correct balance between speed and accuracy. In-memory caching techniques will be used to optimise the databases that contain user profile information.

5.2 Safety Requirements

The programme runs on top of numerous abstraction layers over the hardware and kernel, with little chance of causing damage to the user's device.

5.3 Software Quality Attributes

Maintainability

The product's many versions should be simple to maintain. It should be simple to add code to an existing system for development, as well as to update for new features and technologies as they become available. Maintenance should be inexpensive and simple.

Usability

This may be assessed in terms of usability. The application should be simple to use. It should be simple to grasp. The navigation should be straightforward.

Flexibility

It should be adaptable enough to change. It can be adapted to work with other products with which it needs to interact. It should be simple to integrate with other common third-party components.

5.4 Business Rules

All users will be able to use the programme for free, and the source code will be freely available for usage and modification.

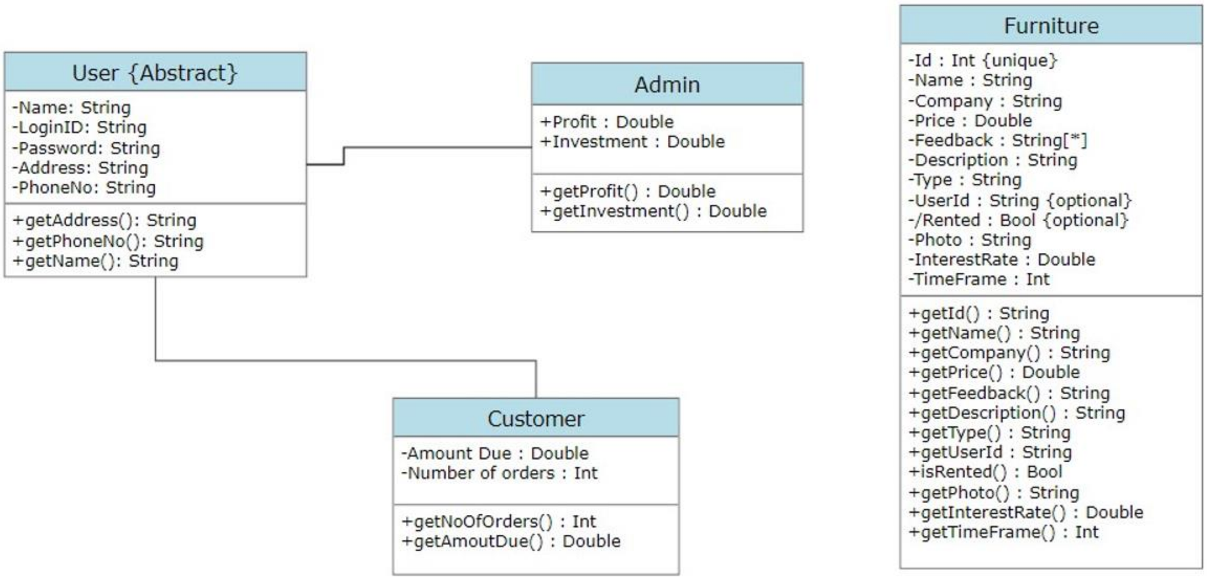
6. Other Requirements

Appendix A: Glossary

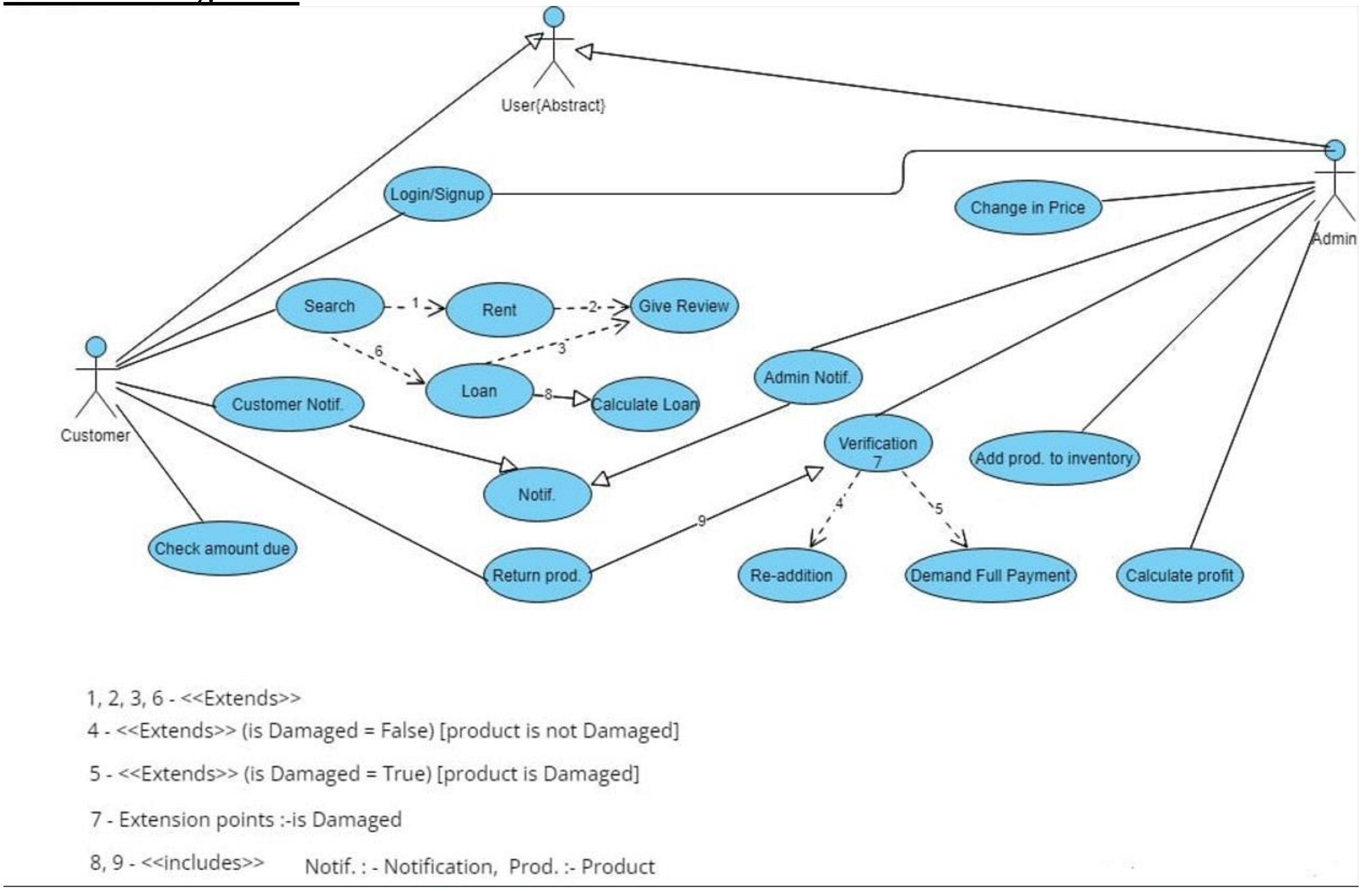
SQL:- Structured Query Language
DB:- Database

Appendix B: Analysis Models

Class Diagram :-



User Case Diagram :-



1, 2, 3, 6 - <<Extends>>

4 - <<Extends>> (is Damaged = False) [product is not Damaged]

5 - <<Extends>> (is Damaged = True) [product is Damaged]

7 - Extension points :-is Damaged

8, 9 - <<includes>> Notif. : - Notification, Prod. :- Product

Appendix C: To Be Determined List