—---

Software Requirements Specification

for

Furniture Rental Store System

Version 1.0 approved
Prepared by

Banoth Dinesh Karthik 20CS30009
Saikat Moi 20CS10050
Saurabh Jaiswal 20CS30047

Department of Computer Science and Engineering
Indian Institute of Technology ,Kharagpur

16th March 2022

Table of Contents

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 was created in a monospace font Source Code Pro using free and open source writing tools such as LibreOffice. The text font size is 11 and the heading font size is 14. All headings are bolded to make them stand out. The paper is written in the American English style.

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 inspiration behind the programme as well as the implementation complexities. Anyone interested in using the software can read the relevant sections of the text, which are listed in the Table of Contents on Page 2.

1.4 Product Scope:

FRSS is a desktop application designed for people 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:

IEEE.IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements

Specification. IEEE Computer Society, 1998.

2. Overall Description

2.1 Project 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:

● FRSS gives furniture on rent at low prices. And also gives furniture on loan based at low interest rates.

2.3 User Classes and Characteristics:

Customer

- Customers may borrow furniture at a certain interest rate, which is determined by whether or not he has previously borrowed furniture.
- Customers can view their previous order history as well as the amount due.
- Customers can leave product reviews and feedback.
- The consumer has the option of taking any furniture from the catalogue as well as the inventory.

Administrator

- The person who owns the furnishings in the inventory.
- When a new product is released, the administrator can add it to the database.

a catalog of sorts

- On his dashboard, the administrator can check the overall earnings and investment.
- If the number of items in the inventory goes below a certain threshold

When a certain threshold is reached, a notice is delivered to the administrator.
need.

- At any time, the administrator can insert or adjust the price of any item.

2.4 Operating Environment:

The application uses a client-server approach, with users interacting with a GUI web application as clients. Tkinter / pyqt5 is used to create the user interface, and 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:

● Unless the administrator changes it, a department's furniture costs must remain consistent. The software will adjust the pricing of furniture after it has been returned, using a dynamic pricing algorithm.

● To have access to the system's functionality, all users must first log in.

● The customer's user ID and password will be required at the time of login.

● To secure the database, advanced security measures have not been deployed.

● Notifications about changed furniture cannot yet be provided to the user via any forms of communication.

2.6 User Documentation:

License and User Manual will be provided along with the software.

2.7 Assumptions and Dependencies:

- To use the software, the user must be familiar with the internet and must be connected to the internet.
- Python 3.8 or above , Tkinter , PIL , MySQL , Matplotlib.

3. External Interface requirements

3.1 User Interfaces:

The product will be a desktop programme that the user may interact with. Before they may see their dashboards, both the client and the administrator must first log in. The consumer can rent or borrow furniture, review or provide feedback on the things, and return them. The administrator, on the other hand, can create/delete client accounts, add new goods to the inventory, evaluate returned furniture, and review the company's total profits 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 specific hardware requirements for the user assuming 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.

4. System Features

This subsection provides details about the identified functional requirements of the software.

4.1 Sign up, Login, and Forgot Password

4.1.1 Description and Priority

Takes the user's information to establish an account and allows them to log in using their user id and password.

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.

When a new user registers, the system sends a verification email to verify that no fraudulent users gain access.

4.1.3 Functional Requirement

Signup: A new user is generated in the database as soon as the user inputs information and the user's identity is confirmed by email. The user can then carry out the desired activities.

Existing users can log in using existing authentication credentials once the system has verified that they are correct.

4.2 Stock Availability

4.2.1 Description and Priority:

When a new product is introduced, the admin can enter it into the inventory.

4.2.2 Stimulus/Response Sequences:

If a new piece of furniture is released in the market, the admin can add it into the inventory.

4.2.3 Functional Requirements:

The admin makes an insert query into the furniture database.

4.3 Payment

4.3.1 Priority and Description

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 Requirements

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 Cancellation and Return

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

If an item is available in the inventory, the customer goes to the furniture department, looks for it, and leases it if it is available.

4.4.3 Functional Requirements

Renting Furniture: When a customer wishes to rent furniture, a query is run against the database to determine 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 Administrator's page Login

4.5.1 Description and Priority

When the count of any item falls below a certain threshold, the admin is notified.

4.5.2 Stimulus/Response Sequences

A notification is sent 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 Requirements

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

.

5. Other Non-Functional Requirements

5.1 Performance Requirements

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

5.2 Safety and Security Requirements

The application 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 various versions should be simple to maintain. It should be simple to add code to an existing system for development, as well as to upgrade for new features and technologies as they become available. Maintenance should be inexpensive and simple.

Usability :

Software Prerequisites Page 11 of the FRSS Specification

This can 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 goods with which it needs to interface. It should be simple to integrate with other common third-party components.

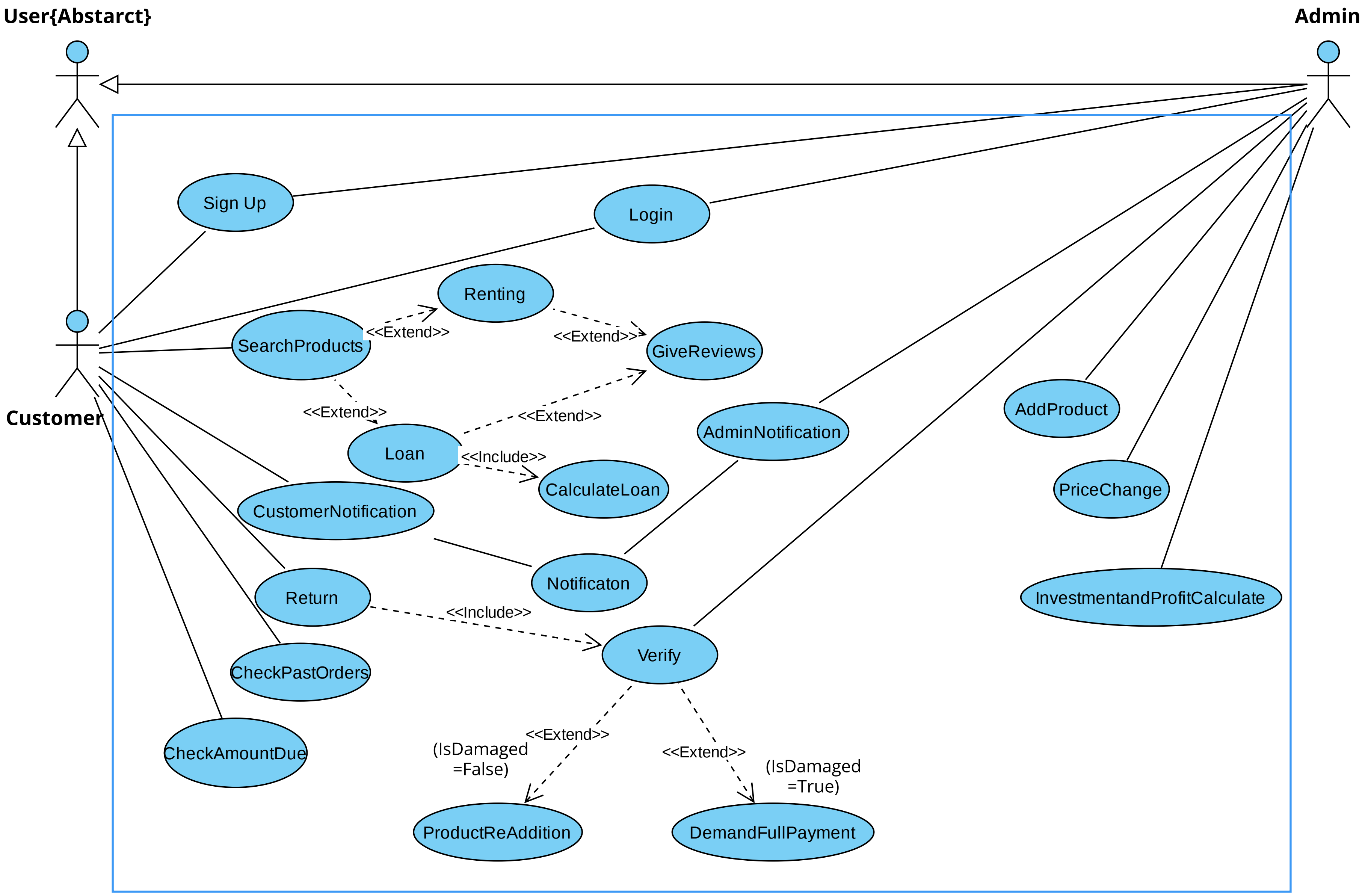
5.4.Business Rules:

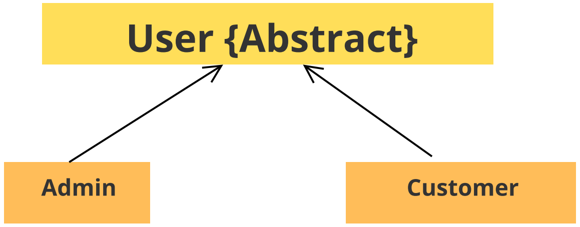
Only after a proper contract arrangement with the firm developing it should this software be utilized , and in the event of any problems, the user should immediately contact our team for clarifications.

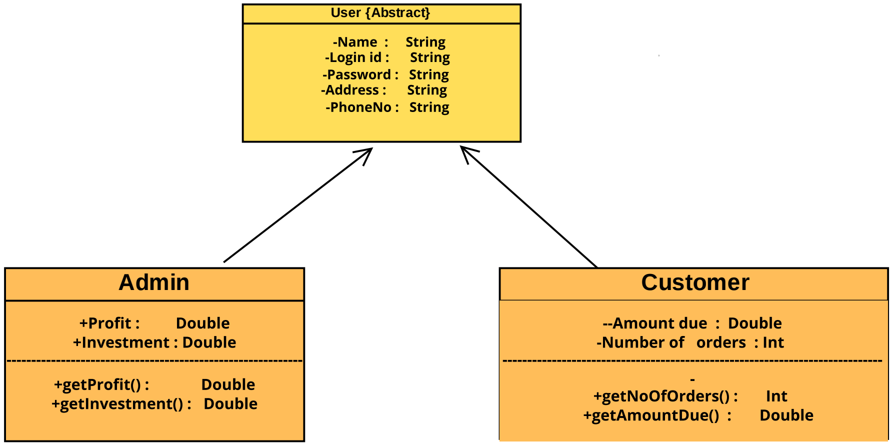
6.Other Requirements:

- Licensing requirements: Applicable
- Legal, Copyright and Other notices: All rights reserved by our team.

User Case Diagram






User {Abstract}

-Name : String
-Login id : String
-Password : String
-Address : String
-PhoneNo : String

Admin

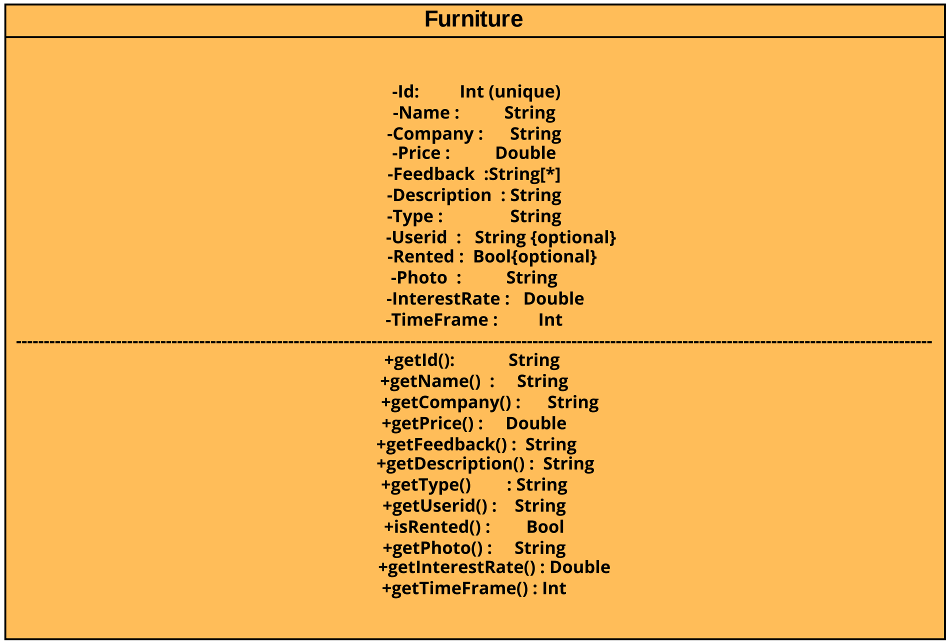
+Profit : Double
+Investment : Double

+getProfit() : Double
+getInvestment() : Double

Customer

--Amount due : Double
-Number of orders : Int

-
+getNoOfOrders() : Int
+getAmountDue() : Double



Furniture

-Id: Int (unique)
-Name : String
-Company : String
-Price : Double
-Feedback :String[*]
-Description : String
-Type : String
-Userid : String {optional}
-Rented : Bool{optional}
-Photo : String
-InterestRate : Double
-TimeFrame : Int

+getId(): String
+getName() : String
+getCompany() : String
+getPrice() : Double
+getFeedback() : String
+getDescription() : String
+getType() : String
+getUserid() : String
+isRented() : Bool
+getPhoto() : String
+getInterestRate() : Double
+getTimeFrame() : Int