

Software Requirements Specification

Group 4

Calm Chores

**Software Engineering
Fall 2024**

CS673



Keshav Saraogi
Chun Qin Huang
Anup Sindagi
Xiaoyue Zhang
Fernanda Nano

1. Introduction

1.1. Overview

Calm Chores app is going to be a chores management app for homes being shared by roommates. The app will offer a one-stop platform to manage different chores, bill payment reminders, and common house tasks. This document outlines the intended features and functionalities of the system, including the design requirements and user interactions, and ensures that project developers, users, and managers have a common understanding of the project.

1.2. Benefits

Fair Task Distribution: The app will ensure that a single roommate is not overburdened with multiple chores. It will rotate tasks evenly to create a fair division of labor and responsibilities.

Accountability Through Public Ratings: The app will feature a rating page for all users which will foster transparency and accountability. It can be used as a recommendation reference for potential future roommates.

Point-Based Reward System: The app would reward points for completing tasks on time and negative points if tasks are not completed on time, which would drive better chore management and motivate users to maintain a good profile.

Vacation Mode: Users can activate vacation mode beforehand to notify all the roommates that they will be out of town and all the chores will be adjusted accordingly.

Detailed Task Management: The tasks can be scheduled ahead of time and can be set in any particular routine according to the need, tasks can be further divided into sub-tasks for convenience.

Integrated To-Do List and Bill Payment Reminders: Apart from the chores, there will be a section for a one-time to-do list to note down and track other one-time chores related to the entire house or particular room. It will feature a similar functionality to maintain due payments related to the house.

Enhanced Collaboration and Communication: The app will help improve communication regarding common chores and responsibilities related to the house and will help maintain a good track record of all its users.

1.3. Scope

The scope of this document includes the high-level and detailed functional requirements and system interfaces. This SRS will guide the development process and help in verifying that the final product meets the expected user.

2. High-level requirements

2.1. Functional Requirements

- 2.1.1. The application should allow the users to register for an account when provided with their email, password, and profile type (roommate/landlord).
- 2.1.2. The user should be able to log in the application after providing the email and password.
- 2.1.3. Users as roommates should be able to create, assign, and rotate chores presented within the same house within their respective roommates.
- 2.1.4. Each user as roommate should be able to mark a chore as complete, verified, or incomplete, and other roommates should be able to verify other tasks as well.
- 2.1.5. The application should allow landlords as users to create, or remove a listing for their house (e.g. apartment/unit).
- 2.1.6. The application should allow landlords as users to add, remove, and update the tenants of the house.
- 2.1.7. The application should allow the users to send maintenance requests to the landlord as users, filed by the roommates as users, including a description and a priority level.
- 2.1.8. The landlord should be able to mark the requests as complete, pending, or incomplete for the request on the application.
- 2.1.9. The application should notify each of the users as roommates with real-time notifications about the status of the tasks (for example, reminders about upcoming tasks, and completion of tasks).
- 2.1.10. The application should notify the users as roommates about the status of the maintenance request to each of the roommates.
- 2.1.11. The application should allow each roommate to provide ratings to other roommates based on their task completion, and punctuality on a scale of 1 - 5.

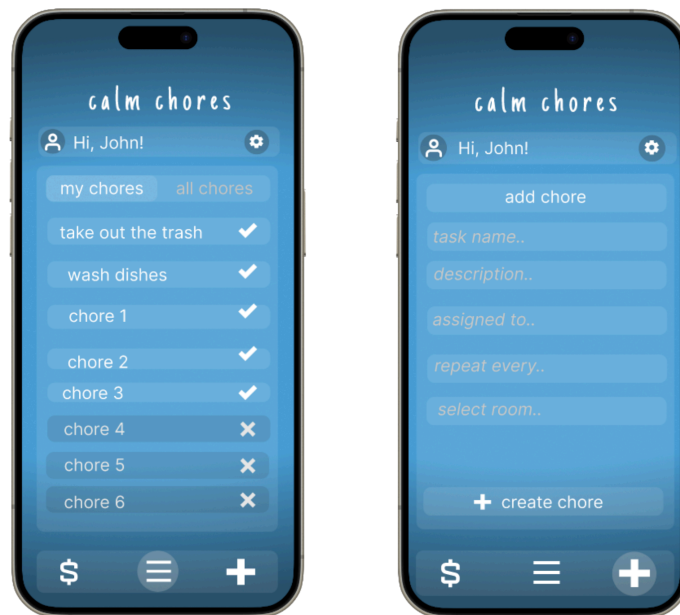
2.2. Non-functional Requirements

- 2.2.1. The system should allow the users to have a rich user interface and user experience throughout the usage of the application while not having any sense of delays or lags during any operation.
- 2.2.2. The system should allow reliable backups to prevent data loss.
- 2.2.3. The system should prevent unauthorized access to sensitive information and only provide role-based access control to roommates and landlords at all times.
- 2.2.4. The system should operate quickly and responsively for all users from the client side.
- 2.2.5. The system should fetch queries and have fast data retrieval from the back-end side.
- 2.2.6. In case of an unexpected failure causing downtime, the system should be up and running within less than 60 seconds.

2.3. User Stories

- 2.3.1. As a user, I want to be able to register for an account on the application so that I can start using the application.
- 2.3.2. As a user, I want to log in with my email and password to the application so that I can access my account.
- 2.3.3. As a tenant, I want to create, assign, and rotate chores within the house so that every roommate can manage household tasks together.
- 2.3.4. As a tenant, I want to mark the tasks as complete, verified, or incomplete so that my other roommates know if the tasks are completed or not.
- 2.3.5. As a tenant, I would like to be notified of the status of the tasks or maintenance requests so I would know the real-time status of it when it's completed.
- 2.3.6. As a tenant, I would like to request maintenance for repairing something in the household so I won't be charged if something is broken.
- 2.3.7. As a landlord, I want to be able to mark the maintenance request completed or pending so I know if I should get someone over to solve the issue.
- 2.3.8. As a landlord, I would like to list my houses on the application so that someone who wants to rent the house can see the house I have listed.
- 2.3.9. As a landlord, I want to be able to add or remove tenants from my house (unit) when they move out.
- 2.3.10. As a tenant, I would like to rate my other roommates according to their task completion so that when others are going to rent the house that person would know if they do the chores or not.

3. High-level interface



This is the UI mockup for our application dashboard, it's meant to be simple and user-friendly. The left screen shows a chore list with checkmarks, while the right screen lets you add new chores. It uses a calm blue theme, with easy navigation buttons at the bottom.