Project Proposal Final

Group 4

Calm Chores

Software Engineering
Fall 2024

CS673

Keshav Saraogi
Chun Qin Huang
Anup Sindagi
Xiaoyue Zhang
Fernanda Nano

Title

Calm Chores

Purpose

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. It will help rotate the chores distributed amongst roommates in a routine manner by which no one house member will be overburdened with multiple tasks. It will also feature various options to schedule chores on time, have vacation mode to alter schedules, have a to-do list for a particular room or house, and much more. The app will also feature a public page for all users, which will display ratings about the user's task completion score, this can be used as a recommendation for future references. Calm chores will help simplify chore management of shared household tasks and responsibilities.

Benefits

1. **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.
2. **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.
3. **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.
4. **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.
5. **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.
6. **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.
7. **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.

Features and Functionality

The main purpose of this application is to distribute house chores amongst roommates while allowing potential roommates to understand each other. This enables a variety of features available to the users that serve this purpose:

1. Splitting Housework Amongst Roommates
    a. Chore Creation and Assignment
        i. Users can create Chores that can be assigned to specific roommate/roommates to be completed within a certain deadline.
    b. Chore Rotation
        i. Implement a chore rotation system to ensure that the chores are being distributed over time for timely completion.
    c. Vacation Mode
        i. Implement a feature that allows roommates to swap tasks in case of unavailability of the roommates, e.g. schoolwork, health, etc.
    d. Task Completion Verification
        i. Once the tasks are completed, the tasks can be verified by another roommate and assigned to be complete.
2. Notifications for Housework
    a. Real-Time Notifications:
        i. Users will be able to receive real-time notifications for upcoming tasks and task completion.
        ii. Notifications will be sent for late completion and allow operations to extend the deadline.
        iii. Notifications will be sent to roommates regarding the timely payment of rent, utilities, and other necessary bills.
3. Profile for Roommates
        i. Users as roommates will be able to include their detailed information, including their daily routine, availability, daily habits, and exceptions.
        ii. Users as “Potential Roommates” can also create their own profile that could allow them to see profiles of users who would be interested in leasing the apartment with.
4. Public Ratings:
    a. User Profiles with Ratings:
        i. Users as Roommates will have a profile with their ratings publicly visible.
        ii. After a defined time, roommates can rate other roommates based on metrics, such as chores completion, punctuality, and cleanliness.

5. Point-based Reward System
    a. Points Allocation with Rewards:
        i. Every chore completed by a deadline gets the reward (e.g. 5 points added) while completing after the deadline results in a deduction (e.g. 3 points deducted). If the chores are completed within a specified time after the deadline (e.g. 2 hours after the deadline) points will not be deducted as there could be a variety of reasons for small delays.
        ii. Negative Points can affect the user's standing on the profile but alert other roommates in the public profile, altering future roommates about their behavior.
6. Maintenance
    a. Maintenance Request
        i. Tenants can request maintenance to the Landlord to help them fix the issue.
    b. Maintenance History
        i. A record of all the maintenance requests and solutions will be provided, ensuring transparency between the tenants and the landlords.
7. Dashboard
    a. Chores Overview
        i. Implement a dashboard that would allow roommates to see an overview of the house, upcoming tasks, pending requests, maintenance requests, and task completion.

Target Users

This application targets a wide range of individuals who need to maintain the living environment of rental housing.

1. Our primary users are individual house tenants, especially those who share a house with others and share household chores.

2. Our potential users are rental landlords and tenants who expect to rent. They may not use this app directly but can benefit from the features this app offers.

Technologies

We are planning to use web development technologies which include but are not limited to the following:

- **Backend** - Java
- **Frontend** - React Framework, HTML, CSS, JavaScript
- **Database** - FireBase
- **Frontend Design** - Figma
- **Quality Assurance and Testing** - JUnit

Team Members, Roles, and Responsibilities:

The project's tasks will be equally divided between all the team members to ensure quality and on-time completion of the app. The following table describes the roles and responsibilities assigned to the team members:

Members	**Roles**	**Responsibilities**
Keshav Saraogi	Project Manager, Backend Developer, Q/A	Schedule Team Meetings. Manage Jira tasks, and maintain team progress. Write robust backend API, Quality Assurance, and Testing.
Chun Qin Huang	Full-Stack Developer	Develops the product features and functionalities in Java and creates a responsive web interface.
Fernanda Nano	Frontend Developer, QA	Implement front-end interfaces and functionality. Test the front-end code block
Xiaoyue Zhang	Full-Stack Developer, UI/UX	Front-end interface visualization design Front-end and back-end functionality implementation and testing.
Anup Sindagi	Frontend Developer, UI/UX	Create designs for user interface and design, test user interaction, and improve designs. Create React components/UI for the finalized design.

Software Configuration Management Plan(SCMP):

Our Software Configuration Management Plan will define and outline the processes involved with the project development such as communication, task tracking, version control, and documentation. We will be using Discord, Trello, Google Docs, and GitHub as our primary tools to carry out the project development.

Tools Overview

- **Discord:** This will be our primary communication platform to arrange meetings.
- **Trello:** For task tracking and assignment and managing weekly sprint objectives.
- **Google Docs:** It will be used for collaborative note-taking and drafts related to project development meetings and documentation.
- **GitHub:** This will be our primary tool for code version control, to track issues and manage code reviews.

Version Control Strategy

- **Branching Strategy:** There would be production-ready code in the main branch and all the individual contributors would be required to use development branches for working on various aspects of the project.
- **Code review:** All the development branches and feature additions would go through pull requests and code reviews before being pushed to the main branch.
- **Update Log:** There would be a changelog file documenting all the minor and major changes between versions to notify the contributors.

Project Management and Task Tracking

- **Trello Boards**: There would be different boards to manage backlogs, in-progress tasks completed tasks. The boards will be further divided into different sections based on frontend and backend development progression.
- **Task Assignment:** The tasks will be assigned to contributors by the project manager and task assignments will be discussed beforehand in weekly meetings.
- **Sprint Planning:** There will be weekly sprints and everyone will be updating their progress and update on their respective Trello Boards

Communication Protocol:

- **Discord Channels:** There would be different channels to carry communication about different aspects of the project, like meetings channel, development channel for development-related issues and discussion.
- **Meetings:** There will be weekly meetings carried out by the project manager to discuss weekly progress and issues that need to be addressed, meeting minutes will be shared by all the contributors in a shared Google Document.

Risk Identification

1. **Development Delays:** Technical issues or complex tasks may cause delays.
2. **Feature Expansion:** Adding additional features during the development process may cause delays.
3. **Team Member Unavailability:** A team member's unavailability due to any circumstances might affect the progress or the deliverable of the project.
4. **Data Loss:** Loss of code or data.
5. **Deployment Process Inefficiencies:** Issues with the inconsistency of the project deployment.

Risk Mitigation strategies

1. **Developmental Delay:** This risk can be avoided through thorough research and understanding.
2. **Feature Expansion:** This can be avoided by listing out the features and sticking with them before starting deployment.
3. **Team Member Unavailability:** This risk can be avoided by arranging for appropriate replacements as soon as possible.
4. **Data Loss:** This risk can be avoided by having configured code and data backups.
5. **Deployment Process Inefficiencies:** This risk can be avoided by using automated deployment scripts.

Conclusion:

Throughout the course of this project, our team intends to complete the project deliverables and deliver a high-end product Calm Chores. Should there be any changes, we will ensure all issues are resolved in a timely manner. We look forward to completing this project and are completely transparent with all the communication, deliverables while maintaining the learnings from the course.