**Design Document**

The user interface consists of different menus through which users interact with the application. These menus include “Calendar”, “All Tasks”, “Shared Shopping List”, “Apartment”, “Feedback”, and a “Logout” menu.

First time users will see the following screens:

* Login/Registration

Returning users will see the following screens:

* Calendar
* All Tasks
* Shared Shopping List
* Apartment
* Feedback
* Logout

**Terminology**

The application has been developed using React which is a JavaScript library for building user interfaces. We have used Material-UI which is a set of React components that implement Google’s Material Design specification. The system is composed of a client-side and a server-side application. All coding is done using JavaScript. The client-side application is developed using React and the server-side uses Node.js and MongoDB. Node.js is a runtime environment that executes JavaScript code server-side. MongoDB is a NoSQL database program. All environments and programs are open-source and cross-platform.

**Color/Style**

We will be using light blue background with white cards and black text. The font used will be Roboto which is the default font of Material-UI.

**Text Elements**

* Text is displayed as HTML text. Text elements include lists and columns.
* All ASCII characters and symbols are available.
* Main features are displayed in bold while other text is regular.

**GUI Elements**

* Windows/Cards: Windows/Cards are used to display all GUI components.
* Icon: The application has a navigation bar at the left of the screen. It has calendar, tasks, shopping list, apartment members and feedback icons.
* Form: The application has a login form for returning users and a signup form for new users.
* Button: Users will interact with the system using buttons. Human-computer interaction will be handled by the application through different kind of buttons.

**Core Features**

**User Registration and Login**

* This is the welcome screen which allows the user to make a one-time registration.
* Users can login if they are already registered.

**Apartment Management**

* Users can create an apartment or join an existing apartment.
* Only one apartment per user keeps things simple.

**Upcoming Tasks**

* The calendar page will be the home screen of the application.
* This feature allows the user to view upcoming tasks.

**Overdue Tasks**

* Tasks which are still pending after the deadline will appear here.
* The oldest tasks will appear first in a calendar view.

**Completed Tasks**

* All completed tasks will appear under this feature.
* Once the user has marked a task as complete, it will appear here.

**Task Management**

* Apartment members can add and remove tasks.
* Tasks can be assigned to different members of the apartment.

**Shared Shopping List**

* Enables apartment members to create a shopping list.
* Offers the option to add and remove items.

**Features not implemented**

**Notifications**

* Notifications are sent to users when a task is created and when a task is due.
* Users will receive notifications on their registered email-id when a task is created.
* Users will also receive an email reminding them to complete a task which is due.

Due to time constraints and a steep learning curve for React which caused a few technical challenges as all the apartment members were working on React for the first time, the email notification feature has not been implemented but a workaround is available. The task list is arranged in such a manner that the user will see a calendar view of the tasks. The tasks will be displayed in three different types viz. “Overdue”, “Upcoming” and “Completed” which will serve as a reminder to the user about the pending tasks.

**Additional features**

The beta version or the initial release may not have these additional features, but the core functionality of the application will not be affected.

**Help and Settings Menu**

* Provides information about the different features of LazyRoomies.
* Users can set and customize their preferences.

**Feedback**

* A feedback button is provided which directs the users to a short survey page.
* Users can provide feedback on the application and report any errors or bugs found in the application.

**Logout**

* The logout feature will end the user’s session.
* User will be redirected to login page.

The interface has been designed according to Shneiderman's "Eight Golden Rules of Interface Design" [1]

1. Strive for consistency

All features in LazyRoomies have consistent elements.

1. Enable frequent users to use shortcuts

A quick add button is provided to users which will enable the users to quickly add a task or edit a shopping list.

1. Offer informative feedback

Users will be notified when the application is down or is not able to synchronize with the server.

1. Design dialog to yield closure

Users will be notified when a task has been successfully completed.

1. Offer simple error handling

Suggestions are provided to users in case they provide an incorrect email, or they do not meet the password criteria.

1. Permit easy reversal of actions

Users will be able to remove an incorrect task or reassign a task if it has been incorrectly assigned.

1. Support internal locus of control

The apartment members are in control of all the tasks and shopping lists. They can control each activity.

1. Reduce short-term memory load

The functions of each menu are singular which will allow the user to easily understand them.

**Percentage of project completed**

All the core functionalities of the application have been implemented. Some of the additional features have also been implemented. From the different types of tests, we conducted and due to time constraints, it would be safe to say that 85% of the project has been completed. According to user feedback, users do not prefer push notifications as they are bombarded with pop-ups and they must allow or block them repeatedly. Users instead prefer email notifications as they check their inbox daily and will not miss any reminders this way.

**References**

[1] https://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html