RoomEase Use Cases

Alex Vrhel (avrhel), Sid Gorti (sgorti3) Jakob Sunde (jsunde), Omar Alsughayer (oasugher), Cheryl Wang (cwang7), Matthew Mans (mans1626), Weijia Dai (weijid)

I. Signing up with roommates

Precondition: User has downloaded the app
Trigger: User opens the app for the first time
Success condition: User has created a RoomEase group with their roommates
End failure condition: User has not created a RoomEase group, or their roommates could not all join it
Steps to success condition:

1. User authenticates their identity via facebook login
2. User indicates they want to create a new group
3. User is prompted to invite their roommates via email address
4. User's roommates download the application and perform step 1
5. User's roommates indicate they want to join the group
6. User's roommates use the contents of the email invitation to indicate they will join the user's group
7. All roommates are logged in and in a group together

Extensions (error scenarios):

1a. User does not provide correct Facebook login info
    1a.1 App returns user to login screen
    1a.2 User tries again or backs out
3a. One or more of the entered email addresses is not valid
    3a.1 User will be informed that an email did not send
    3a.2 User will attempt to enter a different email and re-send the invitation
6a. Roommate enters incorrect information when attempting to join user's group
    6a.1 App displays a message informing them the group was not found
    6a.2 Roommate tries to enter the information again or backs out

Variations:

3a. Upon creation of group, user invites their roommates via Facebook
    3a.1 Invitations to join group are sent to roommates
    3a.2 Roommates can join group by following a link in the invitation, bypassing steps 5 and 6

II. Adding an item to the fridge/pantry

Precondition: User is logged into the app
Trigger: User goes to the food management screen while putting away food
Success condition: User's food item and its relevant info is visible to their group
End failure condition: User's food item is not entered into the system
Steps to success condition:

1. User presses a button indicating they want to add a food item
2. Application prompts user to enter information about the item

    a. User is prompted for name, expiration date (in days from present), and sharing settings
3. User enters the relevant information and puts the food item away
4. Food item and its information is now visible to all roommates in the group

III. Adding an item to a shared list

Precondition: User is logged into the app
Trigger: User has the app open and wants to add an item to a list for their group to see
Success condition: The item has been added to the list, all group members can see it
End failure condition: The item was not added, group does not see the new item
Steps to success condition:

1. User navigates to one of the shared lists screen on the application
2. User selects the list that they would like to add to, from a selection of all existing lists for their group
3. User presses a + button to add a new item to the list
4. User enters the content of the item they would like to add in a pop-up
5. User indicates they are done, and the item is added to the specified list, now visible to all group members

Variations:

1. “A shared list” may refer to many of the modules in the application (which follow this use case with only slight differences in the details of the UI)

    1a. User navigates to the chore schedule screen to add a new chore
    1b. User navigates to the shared list screen to add something to a list
    1c. User navigates to the scheduler screen to reserve a resource

2a. User may add a new list rather than selecting an existing list

    2a.1 User presses a + button before selecting any existing list
    2a.2 User enters a name for the new list

IV. Editing an item on a shared list

Precondition: User is logged into the app, at least one list has an item on it
Trigger: User wishes to change an item (either notification settings or content) of an item on a list
Success condition: The item has been modified on the list, all group members can see the change
End failure condition: The item was not added, group does not see the changes for the item
Steps to success condition:

1. User selects the “shared lists” screen from the menu at the landing screen
2. User clicks the list that they would like to edit
3. User clicks the item that they want to change
4. System prompts user whether they will be deleting or changing the item
5. User indicates they want to change the item
6. User edits the content or notification settings of the item as desired
7. User confirms their changes to the item

Extensions (error scenarios):

7a. User attempts to delete/edit an item while their roommate also edits/deletes the same item

    7a.1 System will reflect the changes that were made and confirmed most recently

7a.2 User whose changes were overwritten backs out, accepting their roommate's edit, or tries again

Variations:

3a. User may change the list as a whole rather than a single item

3a.1 User interacts with the title of the list

3a.2 Use case continues as above, but for the list rather than a single item

5a. User indicates they want to delete the item

5a.1 System prompts user to ensure deletion is intentional

5a.2 User confirms and the item is deleted from the list, bypassing steps 6,7

For the use cases above, the main failure condition would be caused by an error in the system (the changes made on a user's app aren't properly transmitted to the data that our system stores), which will ideally be ironed out as we design our front end and back end, or by an issue in the user interface (such as confusion in how to proceed with their desired action). Since most of the use cases are simple sequences of operations within the UI, we will make it a main goal of our project that the UI flows logically and is very responsive, so that users don't fail these use cases due to confusion or frustration.

We feel that the use cases listed above cover the important scenarios, because adding items to lists and editing/removing those items are the core action behind many of the different modules of the application (including the fridge management, shared lists, chore schedules, and reservation portions of the app). Although the cases may vary slightly, such as what options need to be entered in or the details of the UI at these different screens, the core process is very similar. Thus, outlining how to sign up and how to interact with these list modules covers the functionality of the app at a high level.