California State University Long Beach

College of Engineering

Computer Engineering Computer Science Department

Tuesday March 24, 2020

Sixfold Development - Business Requirements Document V4.0 (Descoped)

Group Members:

Michell Kuang (Team Lead)	013421094
Joshua McDaniel	014542786
Jingyan Du	014436615
Peter Park	002948398
Jacen Tan	012393782
Daniel Gione	016513144

Version History

Version	Date	Author(s)	Changes/Comments
V1.0	10/10/2019	All members	Original document
V1.1	10/10/2019	Jacen	Drastic Organization Overhaul
V2.0	11/10/2019	Jacen	Revision of Requirements
V3.0	2/27/2020	Michell	Descope features
V3.1	3/5/2020	Michell	Further descoping
V4.0	3/13/2020	Michell	Removing supply request management

Table of Contents

1. Introduction

1.1 Project Goal

Our goal at Sixfold Development is to create a single-page web application by May 12, 2020, that allows users to better manage a cohabitational lifestyle. Our application will provide users the ability to search for available households and better manage their expenses, tasks, and supplies. This will be Sixfold Development's debut project as a technological company.

1.2 Project Need

Sixfold development has identified that managing a household's needs is difficult when coordinating multiple people. The difficulty often lies with ineffective communication amongst household parties and lack of access to essential documents and information. These difficulties ultimately lead to a monetary loss, often for all parties, which our project aims to resolve.

1.3 Project Scope

Our project will offer users above the age of 18 in CA, U.S.A. access to a single-page web application that will provide features to recruit tenants, manage expenses, and manage tasks. Sixfold Development will develop the website, software, and databases required to provide these features. The application will initially work on google chrome and operate under a broadband connection.

2. Business Requirements

The list of requirements for our project include:

- Core Requirements - The necessary features for enforcing security and management of our system.
- User Requirements - The features users expect to have access to when using our application.
- Tenant Requirements - The features that tenants of a household expect to have access to when using our application

---

2.1 Core Requirements

Error Handling

- Errors and exceptions shall be handled by notifying the user and providing appropriate error codes and information.
- Required Input: Any operation that occurs within the system and is unsuccessful.
- Success Postconditions:
  - An error will provide users information on how to resolve the error.
  - If the same operation results in an error three times in succession:
    - The system administrator is notified.
    - The user is notified to try again at a later time.
- Additional Conditions:
  - Any operation that requires greater than or equal to 3.0 seconds to complete shall be considered an error.
  - Error messages shall be user friendly.
  - Critical errors and exceptions shall alert the system administrator.
  - The system shall not halt or crash.

Data Store Access Error Handling

- Error handling shall attempt to handle the following data store errors:
  - Resource unavailability:
    - Shall attempt to retrieve information up to 3 times.
    - If all three attempts end in failure, notify the user to try again later, and notify the system administrator.
  - Resource does not exist
  - Unauthorized access
  - Data type mismatches
  - Data Store reaching max capacity.
    - Notify system administrator

System Logging

- Every successful and unsuccessful operation that occurs on the web application shall be logged to both formats except for the following operations:
    - Successful Log Creation
    - Archiving Logs
- Logs shall be stored in both of the following formats:
    - Data Store Logs
    - Flat File Logs
- A log shall include the following details:
    - Timestamp in UTC
    - Calling class name
    - Calling method name
    - Logging level
    - User system ID
    - Session ID
    - Log description

Data Store Logs

- Logs shall be recorded in an existing data store separate from the web application.
- Success Postconditions:
    - Log is stored in the designated data store in a table with the naming convention YYYYMMDD.csv
- Additional Conditions:
    - Logs should be sortable by timestamp in ascending or descending order.
    - Flat file logs and data store logs must be synced.
- Failure Postconditions:
    - If an error occurs where the log is unable to be created:
        - Log creation must be attempted a maximum of three times.
        - If all three attempts fail, notify the system administrator and cancel the logging attempt.

Flat File Logs

- Logs shall be recorded in a flat file.
- Success Postconditions:
    - Log is stored in a flat file with the naming convention YYYYMMDD.csv
- Additional Conditions:
    - Logs should be sorted by timestamp in ascending order.
    - Data store logs and flat file logs must be synced.
- Failure Postconditions:
    - If an error occurs where the log is unable to be created:

        - Log creation must be attempted a maximum of three times.
        - If all three attempts fail, notify the system administrator and cancel the logging attempt.

Flat File Archiving

- System shall compress flat file logs that are at least thirty days old into a compressed file on the first day of every month at 00:00:000 UTC
- Preconditions:
    - System has flat file logs that are at least thirty days old.
    - System time has reached 00:00:000 UTC.
    - Storage in a separate location to the web service exists to store the archived logs.
- Inputs:
    - All existing flat file logs that are at least thirty days old.
    - Storage Location
- Success Postconditions:
    - Flat file logs older than thirty days are compressed and saved to the designated storage location.
    - Archived logs must have a naming convention of YYYYMM01.
- Additional Conditions:
    - Designated storage location is separate from the system.
    - Archive names must correspond to the date of creation.
    - Archived logs shall be sortable by date in ascending or descending order.
    - Archived logs must not be editable.
- Failure Postconditions:
    - If the archival fails, the system administrator is notified.

User Account Registration/Creation

- A **non-registered user** shall be able to create a user account.
- Preconditions:
    - User is not logged in to the system under an account.
    - Admin and System Admin accounts cannot be created.
- Inputs:
    - First Name
        - Can be up to 200 characters.
        - Can be alphanumeric with special characters.
    - Last Name
        - Can be up to 200 characters.
        - Can be alphanumeric with special characters.
    - Email

        - Can be up to 200 characters.
        - Can be alphanumeric with special characters.
        - Must be in email format: name@domain
        - Should be unique amongst users.
    - Date of Birth (MM/DD/YYYY)
        - Must be 18 years or older.
    - Password Entry #1
        - Can be up to 2000 characters.
        - Must have a minimum of 12 characters
        - Can be alphanumeric with special characters
            - Special characters include every special character on a standard US keyboard except for < and >.
    - Password Entry #2
        - Must match Password Entry #1
- Success Postconditions:
    - Account information is stored in persistent storage with the following details:
        - First Name
        - Last Name
        - Age calculated from DOB
        - Password
    - User is able to login to the system.
- Additional Conditions:
    - Email address must be verified before user is able to login.
    - Email verification should take a maximum of 5 minutes to send to the provided email address.
    - Email verification should be valid for 15 minutes after being sent.
    - Password must be checked to see if they have been compromised.
    - User registering must be using a browser within the current scope location (CA).
- Failure Postconditions
    - If email address is not verified, account creation should not finalize.
    - The user must be notified if:
        - Any of the input fields are invalid.
        - Their account could not be created.
        - They are accessing the site from a location outside of CA.

Admin Account Registration/Creation

- A system administrator shall be able to create an admin account.
- Precondition: A system administrator is requesting this operation.
- Inputs:
    - First Name
        - Can be up to 200 characters.
        - Can be alphanumeric with special characters.

    - Last Name
        - Can be up to 200 characters.
        - Can be alphanumeric with special characters.
    - Email
        - Can be up to 200 characters
        - Can be alphanumeric with special characters
        - Must be in email format: name@domain
        - Should be unique amongst users
    - Date of Birth (MM/DD/YYYY)
        - Must be 18 years or older
    - Password Entry #1
        - Can be up to 2000 characters
        - Must have a minimum of 12 characters
        - Can be alphanumeric with special characters
            - Special characters include every special character on a standard US keyboard except for < and >
    - Password Entry #2
        - Must match Password Entry #1
- Success Postconditions:
    - Account information is stored in persistent storage with the following details:
        - First Name
        - Last Name
        - Age calculated from DOB
        - Password
    - User is able to login to the system.
- Additional Conditions:
    - Admin must have the same functionality as system administrators except:
        - Admin accounts cannot create/register new admin accounts.
        - Admin accounts cannot remove system administrator accounts.
    - Passwords must be checked to see if they have been compromised.
- Failure Postconditions
    - System Administrator will be notified.

User Authentication

- Login: A **user** that possess account credentials and is currently not logged in shall be able to login to the system upon authentication.
    - Inputs:
        - Email Address of an enabled account.
        - Password
    - Success Postconditions:
        - **User** is authorized to access registered user functionality.

    - If three consecutive authentication attempts fail, the account is disabled indefinitely.
      - Must notify the user to contact an admin or system admin to re-enable their account.
  - Additional Conditions:
    - Authentication must take a maximum of 2 seconds.
  - Failure Postconditions:
    - Must notify the **user** if the account they are attempting to authenticate is disabled.
    - Must notify the user that one or more their credentials are invalid.
- Logout: **Users** logged into the system shall be able to be logged out of the system.
  - Preconditions:
    - **Use**r selects the option to log out of their account.
    - **OR user** has maintained 20 minutes of inactivity (time since the user's system last contacted the server).
    - **OR** the **user's** authorization attributes have been lowered.
  - Inputs: User account info.
  - Success Postconditions:
    - **User** is logged out of the system and can no longer access functions.
  - Additional Conditions:
    - Logging out shall take a maximum of 2 seconds to complete.
    - If the user manually logs out, they shall be returned to the login page.
    - An option to log in shall be offered and the user returned to where they left off if they were logged out due to:
      - Maintaining inactivity
      - Having authorization attributes lowered.
        - If they no longer have access to the view they were previously at, return the user to the home page.
  - Failure Postconditions:
    - Notify the user that they were unable to be logged out.
- Password Reset: A **user** shall be able to reset their password.
  - Preconditions:
    - **User** account is not disabled.
    - User is logged in.
  - Password Entry #1
    - Can be up to 2000 characters.
    - Must have a minimum of 12 characters
    - Can be alphanumeric with special characters
      - Special characters include every special character on a standard US keyboard except for < and >.
  - Password Entry #2
    - Must match Password Entry #1
  - Success Postconditions

        - **User's** password has been changed to the provided password.
    - Additional Conditions:
        - Must send the **user** a verification code before changing their password.
        - Verification code should take a maximum of 5 minutes to be sent.
        - Verification code should remain valid for 15 minutes after it is sent.
        - Password Inputs must be checked to see if they have been compromised.
    - Failure Postconditions
        - If verification code expires, the user shall be notified and the password shall remain unchanged.

Authorization

- System shall check the authorization privileges of a requesting **user**.
- Preconditions:
    - An operation is being requested by any **user.**
- Inputs:
    - User authorization attributes
- Success Postconditions:
    - User's authorization attributes match the authorization requirements of the operation.
    - Operation shall proceed as normal.
- Additional Conditions:
    - Authorization shall take a maximum of 1 second to complete.
- Failure Postconditions:
    - Operation shall terminate and the user shall be notified they lack the required authorization.

Documentation

- User Manual
    - Shall provide new and current users information on their authorizations and how to use all their features.
    - Shall be accessible to non-registered users and users.
- FAQ
    - Shall provide questions and answers to the systems most common questions.
    - Shall be accessible to non-registered users and users.

System Analytics

- Website Usage Statistics
- Preconditions:
    - An Admin account is accessing this operation.

- Successful Postconditions
    - Admin shall have access to the following statistics:
        - The number of non-registered users that visit the site per month.
        - The number of successful login attempts per month.
        - The type and number of transactions made per month.
- Additional Conditions:
    - Statistics shall be sortable by ascending or descending date.

Data Access Layer

- Shall allow records and logs used by our system to be created, read, updated, and deleted
- Shall be accessible through the provided features
- Shall be accessed through transactions that either complete or don't do anything

---

2.2 User Requirements

User Account Features

- Account Editing: A **user** shall be able to edit the following details regarding their account:
    - Preconditions:
        - User is currently logged into their account.
    - Input:
        - First Name
            - Can be up to 200 characters
            - Can be alphanumeric with special characters
        - Last Name
            - Can be up to 200 characters
            - Can be alphanumeric with special characters
        - Email
            - Can be up to 200 characters
            - Can be alphanumeric with special characters
            - Must be in email format: name@domain
            - Should be unique amongst users
        - Age
            - A positive integer greater than 18
    - Success Postconditions:
        - User account details have been changed to reflect the new input inside persistent data storage.
        - If any input is invalid, the account details shall remain unchanged.

    - Additional Conditions:
        - Shall require user confirmation before editing account details.
    - Failure Conditions:
        - Account details shall not be changed.
        - If any input field is invalid, notify the user that one or more fields were invalid.
        - If user does not provide confirmation, account details sh
- Account Deletion: A **user** shall be able to delete their account and all information associated with it.
    - To delete their account, a user must select the action to delete and confirm this action by entering their password in the subsequent view.
    - Preconditions:
        - User is logged into their account.
        - User must not be a host or tenant in a household.
    - Success Postconditions:
        - Account details are deleted from persistent storage.
        - System retains the associated system ID in relevant log files, expenses, and tasks.
    - Additional Condition:
        - User must provide their password to delete their account.
        - Must take a maximum of 3 seconds to complete.
    - Failure PostConditions:
        - Account shall not remain inside persistent storage
        - User shall be notified their account failed to be deleted.
        - If password provided is incorrect, the user shall be notified.

User Profile

- Profile Creation: A user profile is created with relevant account information.
    - Preconditions:
        - User successfully created an account in the system.
    - Inputs: User account information
    - Success Postconditions
        - A user profile is created in persistent storage containing the following information:
            - Their first and last name
            - Their age calculated from provided date of birth
            - A profile description.
                - Shall have a maximum of 300 characters
                - Characters shall be alphanumeric and special characters.
                - Shall be empty by default.
    - Additional Conditions:

        - Profile must be created at the same time as account creation.
- Profile Viewing: A user shall be able to view another user’s profile by selecting a user’s name from their household list or from their messages.
    - Preconditions: User is currently logged into the system.
    - Success Postconditions:
        - Shall display the following user profile information based on the corresponding information in persistent storage.
            - First Name
            - Last Name
            - Age
            - Gender
            - Profile Description
    - Additional Conditions:
        - Shall display user profile information in a user friendly manner.
        - Shall take a maximum of 3 seconds to perform.
    - Failure Postconditions:
        - User shall be notified that their request failed if:
            - The profile information was unable to be retrieved.
- Profile Editing: Shall allow a **user** to edit their profile description.
    - Preconditions:
        - User is currently logged into the system.
        - User is editing their own profile description.
    - Input:
        - A profile description.
            - Shall have a maximum of 2000 characters
            - Characters shall be alphanumeric and special characters.
            - Shall be empty by default.
    - Successful Postcondition:
        - Profile description shall be updated in persistent storage.
    - Additional Conditions:
        - Shall notify the user the remaining characters they have left in their profile description.
        - Shall prevent the user from providing more than 2000 characters in their profile description.
    - Failure Postcondition:
        - Profile description shall remain unchanged in persistent storage.
        - User shall be notified that their profile description failed to be edited.

Messaging

- User Message Creation and Sending: A user shall be able to send a message to the host of a household that they find through a household search.

    - Preconditions:
        - User is logged in.
        - User has selected the option to message a household found via the household search.
    - Inputs:
        - **Message content**
            - Shall have a 2000 character limit
            - Shall include alphanumeric and special characters.
            - Shall be empty by default.
    - Success Postconditions:
        - Message is created and sent to the host of the household host.
        - User message contains:
            - Sending user’s name
            - Current timestamp in UTC
            - Message content
    - Additional Conditions:
        - User messages can only be created and sent as part of a household search or as a reply.
        - User shall be notified of their current message content character count.
        - User shall not be able to input more than 2000 characters in their message content.
    - Failure Conditions:
        - If a message is empty when the user attempts to send, the user shall be notified that their message is empty and be prompted to provide input.
- System Messages
    - Message creation
        - A **system message** can only be created by system.
    - Success Postcondition:
        - System Message contains:
            - System name
            - Receiving user’s email
            - Current timestamp in UTC
            - System Message content
- Message Inbox
    - **Messages** shall be stored in the message inbox.
        - Users shall be able to find their messages by navigating to the “Messages” subview of the message inbox
    - A message must be automatically stored in the receiver’s message inbox and contain the following information:
        - Message sender
        - Message content
        - Timestamp
    - For each **user**, the maximum storage for message inbox is 2000 messages.

        - If the quantity of messages is more than 2000, messages with the earliest timestamp will be deleted.
- Message View: Shall display a list of messages.
    - Preconditions:
        - **User** is logged into their account
        - **User** is requests access to their Message inbox
    - Success Postconditions:
        - Shall provide a list of received messages containing the following details:
            - Sending user’s name
            - Timestamp
        - **User** shall be able to click a message in the list to access:
            - Sending user’s name
            - Timestamp
            - Message content
            - Option to delete this message
            - Option to reply this message
    - Additional condition:
        - List shall be sorted by timestamp in descending order.
    - Failure Postconditions:
        - Failure to retrieve a user’s messages will notify the user to attempt again later.
- Message Deletion: Shall delete a selected message.
    - Preconditions:
        - User is viewing a specific message in the message list.
    - Shall display a warning with confirm and cancel
        - If the **user** selects confirm, this message shall be deleted from the message inbox, and the **use**r shall return to the message list.
        - If the **user** selects cancel, no messages shall be deleted and the **user** shall return to the start of the message list.
- Message Reply: A user shall be able to reply to a message in their message inbox.
    - Preconditions:
        - **User** is logged in
        - **User** selected a user message to view.
    - Input:
        - **Message content**
            - Shall have a 2000 character limit
            - Shall include alphanumeric and special characters
            - Shall be empty by default
    - Success Postconditions:
        - Message shall be sent as a reply with the following information:
            - Sending user’s name
            - Current timestamp in UTC
            - Message content

    - A message must be automatically stored in the receiver's message inbox after replying is finished

---

Invitation

- Invitation Creation and Sending
    - Invitation by Profile Selection
        - A **host** must select the **invite** action on a **user's** profile, and then confirm this action on the subsequent view**.**
            - The **user** must not be a current **tenant** of a **household.**
    - Invitation by Email Entry
        - A **host** must select the **invite** action on their household home view, input a valid **user's** email, and then confirm this action on the subsequent view.
            - The input shall be at least 1-30 alphanumeric ASCII characters.
            - The input email must exist in the system and must not belong to a user that is currently a **tenant** of a **household**.
    - If a **host** fails to confirm the above action, no **invitations** shall be created or sent.
    - If the action is confirmed, an **invitation** shall be created and sent to the receiving **user's** message inbox with the following details:
        - Sender's name
        - Current timestamp in UTC
        - An option to accept the invitation
        - An option to decline the invitation
    - Upon successful sending of an invitation, a **host** shall be prompted with an option to make the **household unavailable**.
- Invitation View
    - **Invitations** shall be stored in the message inbox.
        - Users shall be able to find their **invitations** by navigating to the "Invitations" subview of the message inbox
    - **Invitations** shall be displayed as a list.
        - The list will be sorted by date and time.
        - A **user** shall be able to select an **invitation** in the list to view invitation details (see above).
        - An **invitation** shall be deleted from the receiver's invitation inbox after the receiver accepts or declines the **invitation**.
        - If a **user** accepts an **invitation**, this **user** shall automatically join the inviting **household** as a **tenant**.
        - If a **user** declines an **invitation**, this **user** shall return to the invitation list view.

Household Creation

- Preconditions:
    - User is logged into the system.
    - User is not a tenant of another household.
- Inputs:
    - A valid physical address
        - A street address with a character range of 1-200 characters
        - A city name with a character range of 1-200 characters
        - A numeric ZIP code
        - Optional: Apartment number and suite number
        - The entered address must exist in CA.
    - The monthly rent
        - Shall be a nonnegative number with 2 decimal places.
    - An availability indication that is either **available** or **unavailable**
        - If **available**, shall appear in household searches
        - If **unavailable**, shall not appear in household searches
        - Shall be **unavailable** by default
- Successful Postcondition:
    - A household must be created in persistent storage with the following information:
        - Internal household Id
        - Physical address
        - Monthly rent
        - Availability
- Additional Conditions:
    - The creating user shall gain the host authorization attribute.
    - A household listing shall be created.

Household Listing

- Making a Household Listing Available
    - Preconditions:
        - User has created a household
        - User has selected to make their household available
    - Inputs:
        - Estimated monthly rent of available room(s)
        - A household description
            - Shall have a 2000 character limit
            - Shall allow alphanumeric and special characters
            - Shall be empty by default
        - A household type

        - Household images (optional)
    - Successful Postconditions:
        - A **household listing** is **available** and appears in household searches with the following information:
            - The household host's name
            - A zip code retrieved from the household physical address
            - A monthly rent
            - A household type
            - A household description
                - Shall have a 2000 character limit
                - Shall allow alphanumeric and special characters
                - Shall be empty by default
            - An availability indication that is either **available** or **unavailable**.
        - A **household listing** may additionally have the following details:
            - Images of the household
- Household Listing Viewing: A user shall be able to view a household listing by selecting it from the list of households generated from a household search.
    - Preconditions:
        - **User** selects a household listing
    - Success Postconditions:
        - Shall display the following details to a non-tenant user:
            - The zip code
            - The estimated monthly rent of available room(s)
            - The household host's name
            - The availability
            - The household description
            - Images of the household
    - Additional Conditions:
        - The correct details must be shown depending on the **user's** authorizations.
    - Failure Postconditions:
        - If the household becomes **unavailable** and a **user** attempts to view the household listing through a household search, notify the **user** that the household is no longer available.
- Household Listing Editing: A user shall be able to edit a household listing's information.
    - Preconditions:
        - **User** is the **host** of the household they are attempting to edit
    - Inputs:
        - A valid physical address
            - A street address with a character range of 1-200 characters
            - A city name with a character range of 1-200 characters
            - A numeric ZIP code
            - Optional: Apartment number and suite number

      - The entered address must exist in CA, USA
    - The estimated monthly rent of available room(s)
      - Shall be a nonnegative number with 2 decimal places.
    - The availability as either **available** or **unavailable**
    - The household description
      - Shall have a limit of 2000 characters
      - Shall allow alphanumeric and special characters
      - Shall be empty by default
  - Success Postconditions:
    - The following information regarding the household shall be edited in persistent storage if changes were made.
      - Physical address
    - The following information regarding the household listing shall be edited in persistent storage if changes were made:
      - Availability
      - Estimated monthly rent
      - Household Description
      - Household images
      - Household type
  - Additional condition:
    - Current values for the household listing must be provided as default input
  - Failure Postconditions:
    - If any of the above inputs are invalid, notify the user that one or more fields were invalid

Household Search

- Search by City
  - Preconditions:
    - User is logged into the system.
  - Inputs:
    - Valid city in CA, USA
  - Success Postconditions:
    - The search shall return a list of household names belonging to **households** with a matching city name.
      - Shall only contain **available** households
      - Shall be ordered by availability
      - Shall be paginated with 20 entries per page
    - If a household is selected, the search shall:
      - Allow the **user** to view the household listing's details
      - Offer an option to message the **host** of the **household**
    - The **user** shall be notified that no **households** were found if no **available households** have a matching city

    - Failure Postconditions:
        - If a household is disbanded or set to **unavailable** after the search completes, notify the **user** that the household no longer exists when attempting to view the household listing or message the **host**.
- Filter Search: A user shall be able to filter their search based on certain criteria.
    - Preconditions:
        - A user is making a household search
    - Filter by Price
        - Inputs:
            - A price range between $0-$10,000
    - Filter by Household Type
        - Inputs:
            - A household type from the following selection:
                - House
                - Apartment
                - Townhouse
    - Successful Postconditions:
        - The current view of household listings is updated to reflect the selected filters.

2.3 Tenant Requirements

Tenant Management

- Tenant List
    - Preconditions:
        - User is a **host** or **tenant** of the household.
    - Inputs: None
    - Successful Postconditions:
        - Shall provide a list of the names of all **tenants** in the **household** and their role.
            - Display order must be alphabetical ascending order by first name, then last name.
        - Upon selecting a user, the **current user** shall view that user's profile.
    - Failure Postconditions:
        - If the names are unable to be retrieved, the **current user** shall be notified and the process shall be cancelled.

- Promote Tenant: An authorized user shall be able to promote a tenant by selecting the option to promote a role.

    - Preconditions:
        - **User** is the **host** of the household.
        - **User** selects a **tenant** from the tenant list.
    - Success Postconditions:
        - Shall **promote** a selected **tenant** to a **host**.
    - Additional Conditions:
        - User must provide their password to promote a **user**.
        - Upon **promoting** a selected **tenant** to a **host**, the current **host** shall be demoted to a **tenant**.
        - The **promoted user** shall be notified of their **promotion**.
    - Failure Postconditions:
        - If failure occurs while promoting a **tenant**, the **current user** must be notified and the **promotion** shall be cancelled.
- Remove Tenant: An authorized user shall be able to remove a tenant by selecting the option to remove a user from the household.
    - Preconditions:
        - **User** is the **host** of the household
        - **User** selects a **tenant** from the tenant list.
    - Successful Postconditions:
        - Shall **remove** a selected **tenant** from the **household**.
            - Shall not be able to **remove** the **host** from the **household**.
    - Additional Conditions
        - Tenant Removal shall not alter past and current **household records**.
            - Expenses and tasks shall retain the name of the removed user. Details must be edited manually.
        - User must provide their password to remove a **user**.
        - The **removed user** shall be notified of their **removal**.
    - If failure occurs while removing a **tenant**, the **current user** must be notified and the removal shall be cancelled.
- Leave Household
    - Preconditions:
        - User is a **tenant** of the household
        - User selects the leave household option.
    - Inputs: None
    - Successful Postconditions:
        - Shall **remove** the **current user** from the **household**.
            - Upon leaving, the **user** shall lose access to all **household** related features.
            - Shall require the **user** to log back into their account.
        - Leaving a household shall not alter past and current **household records**.
            - Expenses and tasks shall retain the name of the removed user. Details must be edited manually.
        - **User** must provide their password to leave the household.

    - Failure Postconditions:
        - If failure occurs while a **tenant** is attempting to **leave** a household, the **current user** must be notified and the **user** shall remain a part of the household.
- Disband Household
    - Preconditions:
        - User is the host of the household.
        - User selected the disband household option.
    - Success Postconditions:
        - Shall **remove** all **tenants** of the **household** and delete the **household**.
            - All **tenants** shall be notified that their **household** is being disbanded.
    - Additional Conditions:
        - User must provide their password to disband the household.
    - Failure Postconditions:
        - If failure occurs while **disbanding** the **household**, the **current user** must be notified and the **household** shall remain.

Expense Management

- Expense Creation:
    - Preconditions:
        - **User** is logged in and is a **host**
    - Inputs:
        - Expense Name
            - Shall be within 1-200 characters.
            - Shall include alphanumeric characters
        - Expense Type
            - Shall be within 1-50 characters.
            - Shall include alphanumeric characters.
            - Shall be able to choose from predefined and user-defined types.
        - Payment Date
            - Shall be formatted as MM/DD/YYYY
            - Must be after the current date.
        - A Total Payment Amount
            - Must be provided in USD
            - Total amount to pay is $0.00 by default.
            - If the **expense** is a **rent** type, the total amount to pay is equal to the **household's monthly rent**.
        - Expense Splits
            - **Tenants** shall be added by the **current user** to an **expense** and assigned a **split percentage**.

  - The maximum number of **tenants** per **expense** is the number of **tenants** in the **household**.
  - The sum of all split percentages of every **tenant** added shall not be outside the range of 0% and 100%.
- The **split percentage** shall take a percentage of the total amount to pay and assign it to the **tenant** as a **split payment**.
  - **Split payment** must be in the form of USD.
  - Must be updated every time the split percentage is edited.

- Success Postconditions:
  - Shall create an expense in persistent storage with the following details:
    - Expense Name
    - Expense Type
    - Creation Date
    - Payment Date
    - The name of the creator
    - Total Payment
    - Added Tenants
    - **Split percentages**
    - The **split payment** shall be calculated from the **split percentage** and the total payment amount.
- Failure Postconditions:
  - **Expenses** with the same expense name and payment date cannot be created. User will be notified.
  - If any of the above input is invalid, the user will be notified that at one more more fields are invalid.

- Expense Viewing
  - Preconditions:
    - **User** is logged in and is a **host** or **tenant**.
    - **User** has selected an **expense** that belongs to their household.
  - Success Postconditions:
    - Shall display the following details about a selected **expense**:
      - Expense Name
      - Expense Type
      - Creation Date
      - Payment Date
      - Total Payment Amount
      - For each tenant:
        - Name
        - Split Percentage
        - Split Payment
  - Failure Postconditions:
    - If any of the above details cannot be retrieved, the entire process shall be aborted and the user notified that the selected **expense** cannot be viewed.

- Expense Editing
    - Preconditions:
        - **User** is logged in and is a **host**
        - **User** has selected an **expense** that belongs to their household.
    - Inputs:
        - Expense Name
        - Expense Type
        - Payment Date
        - Total Payment Amount
            - Shall not be editable for a **rent expense.** Must be edited through household listing editing.
        - The **tenants** added to the expense:
            - Their **Split Percentage**
    - Success Postconditions:
        - Expense details that are changed will be saved to the persistent storage.
    - Additional Conditions:
        - Adding or removing a **tenant** from an **expense** shall not change the **split percentages** of other **tenants**.
            - Editing the **split percentage** will automatically update the tenant's **split payment**.
    - Failure Postconditions:
        - If any of the above details are invalid, the changes must not be made to the **expense** and the **current user** shall be notified.
            - The **current user** shall be allowed to correct their mistakes and make another attempt.
        - If any of the above details fail to be edited, the **current user** must be notified that there was an error and all changes must be undone.
- Expense Deletion
    - Preconditions:
        - **User** is logged in and is a **host**
        - **User** has selected an **expense** that belongs to their household
    - Success Postcondition:
        - Selected **expense** shall be be deleted from the system
    - Additional Condition:
        - Shall require the user to provide confirmation before deleting the expense
    - Failure Postcondition:
        - If the selected **expense** is a **rent expense**, the **current user** must be notified that the **rent expense** cannot be deleted. The deletion must be cancelled.
        - If confirmation fails, the **user** shall be notified and the deletion shall be cancelled.

Task Management

- Task Creation
    - Preconditions:
        - **User** is logged in and is a **host**
    - Inputs:
        - A task name
            - Shall be within 1-200 characters.
            - Shall include alphanumeric characters
        - A starting date and due date
            - Dates must be formatted as MM/DD/YYYY.
            - Dates must be the current date or a future date.
            - Dates are set to the current date by default.
        - **Instructions**
            - Shall Include a text field with a limit of 1000 ASCII characters
            - Shall allow alphanumeric and special characters.
        - A **Responsible Tenant List**
            - Shall select tenants that are part of the **household**
            - Shall have at least 1 **tenant**.
            - Shall have a maximum number of **tenants** equal to the number of **tenants** in the **household**.
    - Success Postconditions:
        - Shall create a task in persistent storage with the following information:
            - Task name
            - Start date
            - Due date
            - Creator’s name
            - Instructions
    - Failure Postconditions:
        - If any of the above details are invalid, the **user** shall be notified and the **task** shall not be created.
- Task Viewing
    - Preconditions:
        - **User** is a **host** or **tenant** of a household
        - **User** selected a task that belongs to their household.
    - Success Postconditions:
        - Shall display the following task information in a user friendly manner:
            - Task name
            - Starting date
            - Due Date
            - Completion Status
            - **Instructions**

            - **Responsible Tenant List**
                - Shall display the tenant’s name(s).
    - Failure Postconditions
        - If any of the details fail to be retrieved, the entire process shall be aborted and the **user** notified that the selected task cannot be viewed.
- Task Editing
    - Preconditions:
        - **User** is a **host** of a household
        - **User** selected a task that belongs to their household.
    - Inputs for regular task:
        - A task name
            - Shall be within 1-200 characters
            - Shall include alphanumeric characters
            - Shall be unique amongst tasks
        - Due date
            - Dates must be formatted as MM/DD/YYYY.
            - Dates must be the current date or a future date.
            - Dates are set to the current date by default.
        - **Instructions**
            - Shall Include a text field with a limit of 1000 ASCII characters
            - Shall allow alphanumeric and special characters.
        - A **Responsible Tenant List**
            - Shall select tenants that are part of the **household**
            - Shall have at least 1 **tenant**.
            - Shall have a maximum number of **tenants** equal to the number of **tenants** in the **household**.
    - Success Postconditions:
        - Task in persistent storage should be changed to reflect the edits made.
        - The names of users who are either removed or leave the household shall remain as part of the task until removed via editing.
    - Failure Postconditions:
        - If any of the above details are invalid, the changes must not be made to the task and the **current user** shall be notified.
            - The **current user** shall be allowed to correct their mistakes and make another attempt.
        - If any of the above details fail to be edited, the **current user** must be notified that there was an error and all changes must be undone.
- Task Deletion:
    - Preconditions:
        - **User** is a **host** of a household
        - **User** selected a task that belongs to their household.
    - Successful Postconditions:
        - Selected **task** to be manually deleted from persistent storage.

        - Additional Conditions:
            - Shall require the user to provide confirmation before deleting the task
            - **Tasks** shall be automatically deleted if:
                - The **task's** due date has passed for 180 days.
        - Failure Postconditions:
            - If the **task** is unable to be deleted from the system, the **current user** shall be notified and the deletion shall be cancelled.
- Task Reminders
    - Preconditions:
        - Any existing task with a responsible tenant.
    - Success Postconditions:
        - Shall automatically message the **responsible tenant(s)** at most 1 day before the due date if their **task** is not complete upon logging.
            - Message shall be sent to the tenant's message inbox.
                - If the message inbox is full, the **user** must be notified to empty their mailbox upon login.
            - Message shall consist of the **task's**:
                - Task Name
                - Start Date
                - Due Date
                - Instructions
    - Additional Conditions
        - The system shall attempt to message the **responsible tenant(s)** once.
        - The system shall only send 1 message per task.

- Task Completion
    - A **responsible tenant(s)** shall be able to set their **incomplete** task as **complete** before the due date passes.
    - If an error occurs, the changes shall not be made and the **current user** shall be notified.

3. Glossary

- Available - If a household shows up when performing a household search
- Current User: The person that is using the system/feature at the moment a request is made
- Completed tasks - Tasks that have been marked as completed before the due date

- Demote - When a host’s role drops to a tenant
- Disband - When a household is deleted from the system, removing all tenants from the household
- Expenses - Bill that is assigned a name and type of bill that needs to be paid off
- Hash - The process of putting something through a hashing function for the purpose of password security
- Host - A tenant who has complete control of their respective household
- Household - A group consisting of a host and many tenants that represents a place of living, each with its own expenses
- Household Creation - The making of a household, complete with name and location
- Household Name - Name of respective household
- Household Listing - A collection of information about a household
- Household Records - Details for expenses and tasks
- Household Search - Ability for user to find a household
- Incomplete tasks - Tasks that have not been marked as complete before the due date passes
- Instructions - Specific instructions written about a task that provide additional details
- Invite - The act of a host sending an invitation to a user
- Invitation - A special form of communication where a user is given the option to join a host’s household or not
- Leave - When a tenant removes themselves from a household
- Non-registered User - a person who has not made a registered account or has not logged in that is using our system
- Promote - When a host gives host role to a tenant
- Removal - When a host removes a tenant from the household
- Rent Expenses - The total cost of rent for the household
- Responsible Tenant - A tenant who is responsible for a specific task
- Responsible Tenant List - List of tenants responsible for completing specific tasks
- Salt - A cryptographic technique that improves password security
- Task - A job created by a host that is assigned to a user in household
- Task Completion - A qualifier signifying a task is complete or incomplete.
- Task Creation - Ability of host to create and assign tasks
- Tenant - A user who is within a household, can only be in one household at a time
- Tenant List - A list of all tenants that are within the household
- Unavailable - A household that is not displayed when performing a household search
- User - Any person that has created an account within our system
- User Profile- A collection of details about their respective user

4. References

- Brown, P. (2018, August 20). Why do you need to Salt and Hash passwords? Retrieved from https://culttt.com/2013/01/21/why-do-you-need-to-salt-and-hash-passwords/.
- NIST Special Publication 800-63B. (n.d.). Retrieved from https://pages.nist.gov/800-63-3/sp800-63b.html#memsecret.

- OWASP Secure Coding Practices Checklist. (n.d.). Retrieved from https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_Checklist.