**SC2006 Software Engineering**

****

**Lab#1 Deliverables**

Lab Group SSP1

Group Name: Depressed Elderly

Lab Supervisor: Liu Yang

|  |  |
| --- | --- |
| Group Members | |
| Name | Matriculation No. |
| Johnny | U2121697B |
| Bertranne Lee Zi - Wen | U2120209L |
| Yap Shen Hwei | U2140630E |
| Koh Jia Wei | U2121270D |
| Zhi Guang | U2120504C |
| Chong Yue Hong | U2120181D |

Table of Contents

[1.0 Project Introduction 3](#_Toc113437442)

[**1.1 Purpose** 3](#_Toc113437443)

[**1.2 Intended Audience** 3](#_Toc113437444)

[1.0 Documentation of Functional and Non-functional requirements 4](#_Toc113437445)

[1.1 Functional Requirements 4](#_Toc113437446)

[1. Account 4](#_Toc113437447)

[2. Job 5](#_Toc113437448)

[1.2 Non-functional requirements 7](#_Toc113437449)

[2.0 Data Dictionary 8](#_Toc113437450)

[3.0 Initial Use Case Model 9](#_Toc113437451)

[3.1 Use Case Diagram 9](#_Toc113437452)

[3.2 Use Case Description 10](#_Toc113437453)

[4.0 UI Mock-Ups 23](#_Toc113437454)

[4.1 Login Page 23](#_Toc113437455)

[4.1.1 Login Page (Error) 24](#_Toc113437456)

[4.1.2 Login Page (Forgotten Password) 25](#_Toc113437457)

[4.2 Registration Page 25](#_Toc113437458)

[4.3 Landing Page 26](#_Toc113437459)

[4.3.1 Landing Page (Selecting Job) 26](#_Toc113437460)

[4.3.2 Landing Page (Accepting Job) 27](#_Toc113437461)

[4.3.3 Landing Page (Job Details - Servicer) 27](#_Toc113437462)

[4.3.4 Landing Page (Messaging - Chat Logs) 28](#_Toc113437463)

[4.3.5 Landing Page (Messaging - User to User) 29](#_Toc113437464)

[4.3.6 Landing Page (Job - Details - Requester) 30](#_Toc113437465)

[4.4 Account Page 30](#_Toc113437466)

[4.4.1 Account Page (Edit Details) 31](#_Toc113437467)

# 1.0 Project Introduction

## 1.1 Purpose

Our group, Depressed Elderly, aims to create a web application which will help enable people who are searching for help to find people to assist them in odd jobs that they might have. Our web application will serve as a platform for people who are seeking help and those offering help to meet and exchange their services.

## 1.2 Intended Audience

Our intended audience for the application is those who are seeking a platform to find help for odd jobs that they might have and for people who are able to offer help for those finding it.

# 2.0 Documentation of Functional and Non-functional requirements

## 

## 2.1 Functional Requirements

### Account

* 1. Users must be able to register for a new account.
     1. System must create a new account for the user once all requirements are fulfilled.
        1. System must prompt the user to change the username if it is taken.
        2. System must prompt the user to change the passwords if the password didn’t adhere to requirements: at least 1 special character, letters in mixed cases, at least 8 characters.
        3. System must prompt the user to change the phone number if there is a phone number linked to an existing account.
        4. System must prompt the user to change the email address if there is an email address linked to an existing account.
     2. System must send a confirmation email to the user when a new account is successfully created.
  2. Users must be able to login to the account with username and password.
     1. After the user submits details, the system must validate the user’s username and password.
     2. System shall show the main landing page.
     3. Users must be able to view their profile after clicking the profile tab.
     4. Users must be able to modify their profile if they want
        1. Any changes made by users must fulfilled the requirements according to 1.1.1
        2. System must be able to update the profile picture after edit the profile picture
        3. System must be able to update the email after email has been changed by user.
        4. System must be able to update the bio after bio has been changed by user.
        5. System must be able to update the phone number after phone number has been changed by user.
  3. Users must be able to reset the password if they had forgotten their passwords.
     1. System must send an email to users to reset the password.
     2. System must prompt the user to change the password if the new password didn’t adhere to requirements: at least 1 special character, letters in mixed cases, at least 8 characters.
     3. System must prompt the user to change the password if the new password is used before.

### Job

* 1. Requesters must be able to create job requests.
     1. System must ensure the job request has a job name.
     2. System must ensure the job request has a job description.
     3. System must ensure the job request has a location specified.
     4. System must ensure the job request has date and time specified.
     5. System must ensure the job request has a salary specified.
     6. System must ensure the requester specifies the payment method.
     7. System must add the job request to the existing job listing after the job request is successfully created.
  2. Requesters must be able to edit job requests if the job request hasn’t been accepted.
     1. System must update the changes to the job.
  3. Requesters must be able to delete job requests
     1. System must remove the job request from the job listing.
     2. System must penalise requesters after the servicers have accepted for 3 minutes
  4. Servicers must be able to filter through the available jobs.
     1. System shall allow servicers to filter by keywords, salary, and location.
  5. Servicers must be able to accept job requests.
     1. System must create a job contract with all details on it.
     2. System must add the job to the servicer’s “Job List”.
        1. System must set the job status to active.
     3. System must remove the job request from the job listing.
     4. System must ensure both servicers and requesters are able to contact each other after the job contract is created.
     5. System must allow both servicers and requesters to contact through the app.
  6. Servicers must be able to retract services
     1. System must add the job request to the job listing.
     2. System must penalise servicers after the servicers have accepted the job for 3 minutes.
  7. Requesters must be able to report if any emergency occurred.
     1. System must immediately alert customer service.
  8. Requesters must be able to end the job after it is completed.
     1. Servicers must be able to check the “I have received payment”.
     2. System must change the job status to history.
     3. Requesters must be able to rate and review the servicers.
        1. System must update any comments and ratings in the servicer’s profile.
     4. Servicers must be able to rate and review the requesters.
        1. System must update any comments and ratings in the requester’s profile.

## 2.2 Non-functional requirements

1. Performance
   1. The application shall take within 3 seconds to load in
   2. The sign up and sign in page shall take within 1 second to load.
   3. The validation email for sign up shall take within 1 min to arrive.
   4. Resetting password shall take 1 second to update in the system.
   5. Each function module shall take within 1 seconds to load in.
   6. System must add the new job request to the job listing in 1 minute.
   7. Details of job shall take within 2 seconds to load.
   8. The application shall take within 3 seconds to exit.
   9. All reminding messages shall take within 0.5 second to load in.
   10. Accepting and posting jobs shall take within 1 second to be updated in the system.
   11. Map must be displayed within 1 second.
   12. Information of jobs must be displayed within 1 second.
2. Safety Requirements
   1. Users shall add friends via this application and chat.
   2. Team shall maintain the system
   3. Team shall test and debugging the application beforehand.
3. Security Requirements
   1. If the user enters in the wrong password more than 3 times, the account will be locked
   2. Users must log in their account to post or accept jobs.
   3. The system shall handle exceptions.
      1. Reminding message shall display when the user enters two different passwords when setting or resetting the password.
      2. Reminding message shall display when the user inputs the wrong verification code.
   4. The application must not leak user information.
      1. User information must not be seen or used by others.
      2. User information must be deleted if the user account is deleted.

# 3.0 Data Dictionary

|  |  |
| --- | --- |
| Term | Definition |
| User | A person who has a registered account and is using the services provided by the application. |
| Account | The profile of the user in our system which contains details (Name, Username, Email, Phone Number, Rating) associated with them. |
| Username | A unique name that is associated with the user. |
| Password | A string of characters (letters, numbers, special symbols) that allow the user to access their account |
| Job | An instance of an ad hoc request created by a requester which details such as location, time, description, number of people needed and salary.  Jobs are completed by the servicer. |
| Contract | An agreement which is created when a requester has accepted the help from a Servicer(s). |
| Requester | A user who creates a job request in the system |
| Servicer | A user who does the job created by a requester |
| Rating | A numerical way to indicate how well a Servicer or Requester worked when the job has been completed. They will be rated on a 1 - 5 system (1 = worst, 5 = best) |
| Review | A short description which allows both users to talk about their experience doing the job. |

## 

# 4.0 Initial Use Case Model

## 4.1 Use Case Diagram

<https://drive.google.com/file/d/1ZwOVDVyyZb0XPN66p2pxzKTQuC_Ie6qz/view?usp=sharing>

Diagram

Description automatically generated

## 4.2 Use Case Description

Key:

Frequency 1-5, 1 = Low 5 = Very Frequent

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **USER\_UC\_1** | | |
| Use Case Name: | **Create Account** | | |
| Created By: | **Shen Hwei (Nicole)** | Last Updated By: | **Shen Hwei (Nicole)** |
| Date Created: | **02.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User** |
| Description: | **Create Account allows new users to create an account in our system database. To create an account, users must input Name, Username, Email, Phone Number, Password.** |
| Preconditions: | 1. **Username must not be taken.** 2. **Passwords must adhere to requirements: at least 1 special character, letters in mixed cases, at least 8 characters.** 3. **Phone number must not link to an existing account.** 4. **Email must not link to an existing account.** |
| Postconditions: | 1. **System sends confirmation link to user’s email account.** 2. **System displays main landing page.** |
| Priority: |  |
| Frequency of Use: | **5** |
| Flow of Events: | 1. **User inputs required information: Name, Username, Email, Phone Number, Password.** 2. **System validates the information.** 3. **System creates the account.** 4. **System updates account information in database.** 5. **System directs the user to main landing page.** |
| Alternative Flows: | **AF-S1-a. User account already exists: Phone number/ email taken.**   1. **System displays “User account already exists”** 2. **System prompts user to create account or log in.** 3. **If create account, then continue from main flow step 1.** 4. **If log in, then system directs to log in page to continue from Login (USER\_UC\_2)**   **AF-S1-b. Username already taken**   1. **System displays “Username already taken”** 2. **System prompts user to input a different username.** 3. **Continue from main flow step 1.**   **AF-S1-c. Password does not meet requirements.**   1. **System displays “Please input a password of at least 1 special character, mixed case and at least 8 characters long” message.** 2. **Continue from main flow step 1.** |
| Exceptions: |  |
| Includes: | **Update Account Information (S4)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **USER\_UC\_2** | | |
| Use Case Name: | **Login** | | |
| Created By: | **Shen Hwei (Nicole)** | Last Updated By: | **Shen Hwei (Nicole)** |
| Date Created: | **02.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User** |
| Description: | **Login allows users to enter the app to use features catered for registered users of the app. To login to the system, the user must input a unique username and the corresponding password. There will be a maximum of 5 attempts, after which, the user will have to wait for 30 minutes before another attempt.** |
| Preconditions: | **User must have an existing account in the application database** |
| Postconditions: | **System displays the main landing page** |
| Priority: |  |
| Frequency of Use: | **5** |
| Flow of Events: | 1. **User enters username and password on login form.** 2. **User submits details.** 3. **System validates the user’s username and password.** 4. **System shows the main landing page.** |
| Alternative Flows: | **AF-S2-a. Invalid Username**   1. **System cannot find the username in the database.** 2. **System displays an error message.** 3. **System prompts user to input username and password or Create Account.** 4. **If the “Create Account” prompt is ignored, use case resumes at main flow step 1.** 5. **If the user wants to “Create Account”, then System resumes from Create Account step 1.**   **AF-S2-b. Invalid Password**   1. **Password does not match for unique username.** 2. **System displays an error message.** 3. **System prompts user to input username and password.** 4. **Use case resumes at main flow step 1.**   **AF-S2-c. Missing Username/ Password**   1. **System displays an error message.** 2. **System prompts user to input username and password.** 3. **Use case resumes at main flow step 1.**   **AF-S2-d. Maximum 5 attempts exceeded**   1. **System displays “maximum attempts exceeded, try again in 30 minutes” message.** 2. **System locks the user account for 30 minutes.** |
| Exceptions: |  |
| Includes: | **Validate Account, Verify Account** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **USER\_UC\_3** | | |
| Use Case Name: | **Update Account Information** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User (primary), Bank Account (secondary)** |
| Description: | **Update Account Information allows user to edit account information such as username, name, email, phone number, bank account details, profile picture, password.** |
| Preconditions: | **User must have an existing account.** |
| Postconditions: | **User will have updated account information.** |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. **User selects field to update.** 2. **User inputs new desired update.** 3. **System updates information in the database.** |
| Alternative Flows: | **AF-S1-a.**   1. **If user selects the password field to update, user must input old password and new password.** |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **SERVICER\_UC\_1** | | |
| Use Case Name: | **Accept Job** | | |
| Created By: | **Shen Hwei (Nicole)** | Last Updated By: | **Shen Hwei (Nicole)** |
| Date Created: | **02.09.2022** | Date Last Updated: | **02.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User (Servicer), User (Requester)** |
| Description: | **When a user is on Servicer mode, they can browse the listing of ad hoc jobs to accept jobs that they wish to take up. Upon accepting the job, the servicer (primary actor) will receive details of the requester (secondary actor). The servicer will only receive pseudo information of the requester. For example, the name is not displayed as the requester’s full name; phone number will be hidden, any communication between the servicer and requester must be done through the application.** |
| Preconditions: | 1. **User must be logged in to accept a job.** 2. **User must have GPS location enabled.** 3. **User must be in “Servicer” mode.** 4. **User must have a verified phone number linked to the account.** |
| Postconditions: | 1. **System creates a chat box between the Servicer and Requester.** |
| Priority: |  |
| Frequency of Use: | **5** |
| Flow of Events: | 1. **System verifies user phone number.** 2. **System creates a contract between the servicer and the requester to indicate agreement.** 3. **System sets a timer of 3 minutes.** 4. **Precise location information of requester displayed to Servicer.**   **System creates a chat box with call functions for the servicer and requester.** |
| Alternative Flows: | **AF-S1-a. Phone number not verified.**   1. **System prompts user to verify phone number in order to accept a job.**   **AF-S1-b. User is not logged in**   1. **System prompts user to log in/ create an account.** 2. **If User chooses Login, use case resumes from Login (S1).** 3. **If User chooses Create Account, use case resumes from Create Account (S1).**   **AF-S3-a. Servicer retracts service**   1. **Use case continues from Retract Service (S1).** |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **USER\_UC\_3** | | |
| Use Case Name: | **View Listing** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User** |
| Description: | **View Listing allows users, regardless of logged in or not, to view the ad-hoc jobs available. Users can choose to accept jobs but must be logged in to do so.** |
| Preconditions: | **User must be on main landing page.** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **5** |
| Flow of Events: | 1. **System displays the main landing page to user.** |
| Alternative Flows: | **AF-S1-a.**   1. **User can accept jobs on main landing page.** 2. **Use case continues from Accept Job (S1).** |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **USER\_UC\_4** | | |
| Use Case Name: | **Update Status** | | |
| Created By: | **Koh Jia Wei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User** |
| Description: | **When a user is in “Requester” mode, they can modify the ad-hoc jobs if the Servicers have not taken up the job. If the job is set to “Finished” by the servicer, the requester confirms the job has been successfully serviced, the requester can update the status to “Done”.**  **When a user is in “Servicer” mode and is willing to take up the job, the servicer can update the status to “Servicing”.**  **When the servicer has finished the job, the servicer can update the status to “Awaiting confirmation” and wait for the requester’s approval.** |
| Preconditions: | 1. **Requester has already submitted a request** 2. **Servicer has done the ad hoc job** 3. **Requester confirms the ad hoc job** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **5** |
| Flow of Events: | **1. Requester submits a request for any potential servicer to take on the ad hoc job.**  **2. The ad hoc job is taken by a servicer.**  **3. The servicer updates the status of the job to “Servicing”, and this is reflected in the system for all to see.**  **4. The job is successfully done by the servicer.**  **5. The servicer updates that the job is done, the status of job is now “Awaiting confirmation”, and this is reflected in the system.**  **6. The requester is satisfied with the job and the status is now updated to “Done”** |
| Alternative Flows: | **AF-S6 The requester is unsatisfied with the job and the status is updated to “Incomplete” and the job is listed on the listings page again. The servicer and requester are given the option to rate each other.** |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **USER\_UC\_5** | | |
| Use Case Name: | **Rate User** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User** |
| Description: | **Rate User allows a user to rate either the Servicer or the Requester after a job/service completion. If it’s service completion, implying that user is a Servicer, then the user rates the Requester. Otherwise, if it’s a job completion, implying that the user is a Requester, then the user rates the Servicer.** |
| Preconditions: | 1. **User must have a history of completed Job/ completed Service.** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. **System prompts user to rate the other user.** 2. **User performs rating.** 3. **System records the rating for the user for the specified Job/Service.** 4. **System calculates the average rating for user.** 5. **System updates account information.** 6. **System updates database for user account.** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: | **Update Account Information (S5)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **USER\_UC\_6** | | |
| Use Case Name: | **Report Emergency** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User** |
| Description: | **Report Emergency allows users who are participants of a contract to contact Police directly through the application.** |
| Preconditions: | 1. **Reporters must be users who are participants of a contract.** 2. **Users must be logged in.** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **1** |
| Flow of Events: | 1. **User clicks onto a “Report Emergency” interface on the application.** 2. **System initiates a call to the police authority immediately.** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **REQUESTER\_UC\_1** | | |
| Use Case Name: | **Manage Job** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **Requester** |
| Description: | **Encapsulates the following functions**   1. **Post Job** 2. **Delete Job** 3. **Update Status** 4. **Check Status** 5. **Complete Job** 6. **Rate User** |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **REQUESTER\_UC\_2** | | |
| Use Case Name: | **Post Job** | | |
| Created By: | **Shen Hwei (Nicole)** | Last Updated By: | **Shen Hwei (Nicole)** |
| Date Created: | **02.09.2022** | Date Last Updated: | **02.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User (Requester), User (Servicer)** |
| Description: | **When a user is in “Requester” mode, they can post ad-hoc jobs for Servicers to take up. To post a job, the Requester will have to input the location, title of job, description of job, date and time of job and salary. Requester must also have a verified phone number linked to the Requester’s account.** |
| Preconditions: | 1. **User must be logged in to post a job.** 2. **User must have GPS location enabled.** 3. **User must be in “Requester” mode.** 4. **User must have a verified phone number linked to the account.** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: | **5** |
| Flow of Events: | 1. **Requester fills up job information: exact location of job, title of job, description of job, date and time of job and salary.** 2. **System identifies the type of job based on schedule (now or future).** 3. **If job is now, System sets a timer of 3 minutes.** 4. **System records job into database.** 5. **System adds the job to the job listing page.** |
| Alternative Flows: | **AF-S3-a. Requester deletes job, no assigned servicer yet.**   1. **System resumes to Delete Job**   **AF-S3-b. Requester deletes job, already have an assigned servicer.**   1. **If the assigned servicer accepted the job for more than 3 minutes, then Use Case will extend to Impose Penalty on the Requester.** |
| Exceptions: |  |
| Includes: | **Validate Account, Verify Account, Schedule Job** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **REQUESTER\_UC\_3** | | |
| Use Case Name: | **Delete Job** | | |
| Created By: | **Shen Hwei (Nicole)** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User (Requester)** |
| Description: | **Delete Job allows Requester to remove the job that has been posted into the database.** |
| Preconditions: | 1. **User must be in Requester mode.** 2. **Job must exist.** |
| Postconditions: | **Job deleted from database.** |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. **Requester selects the job for deletion.** 2. **System removes job from database.** |
| Alternative Flows: |  |
| Exceptions: | 1. **If Requester deletes 3 minutes after a Servicer has been assigned to the job, use case continues from Impose Penalty (S1).** 2. **System removes job from database.** |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: | **Normal flow assumes that no Servicer has taken up the job yet, or that Servicer was assigned but Requester deletes job within 3 minutes of assignment.** |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **SERVICER\_UC\_1** | | |
| Use Case Name: | **Manage Service** | | |
| Created By: | **Jia Wei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **User** |
| Description: | **Encapsulates the following functions**   1. **Accept Job** 2. **Retract Service** 3. **Update Status** 4. **Check Status** 5. **Rate User** |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **SERVICER\_UC\_2** | | |
| Use Case Name: | **Retract Service** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **Servicer** |
| Description: | **Retract Service allows Servicer to remove from being assigned to a particular job.** |
| Preconditions: | 1. **User must be in Servicer mode** |
| Postconditions: | **Job Assignment removed from database.** |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. **Servicer chooses service/job to retract from.** 2. **System removes assignment.** |
| Alternative Flows: |  |
| Exceptions: | 1. **If Servicer retracts 3 minutes after contract has been made, use case continues from Impose Penalty (S1) on Servicer.** 2. **System removes job from database.** |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **REQUESTER\_UC\_3** | | |
| Use Case Name: | **Complete Job** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **Requester** |
| Description: | **Complete Job allows Requester to confirm that the job has been accomplished by the Servicer.** |
| Preconditions: | 1. **There must exist a contract between the Requester and Servicer** |
| Postconditions: | 1. **System updates the Job history for the Requester.** 2. **System updates the Service history for the Servicer.** |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. **Requester confirms that job has been completed by Servicer.** 2. **System updates job status.** 3. **System updates job and service history for Requester and Servicer.** |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: | **Update Account Information (S3), Update Status (S2)** |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **SYSTEM\_UC\_1** | | |
| Use Case Name: | **Check Status** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

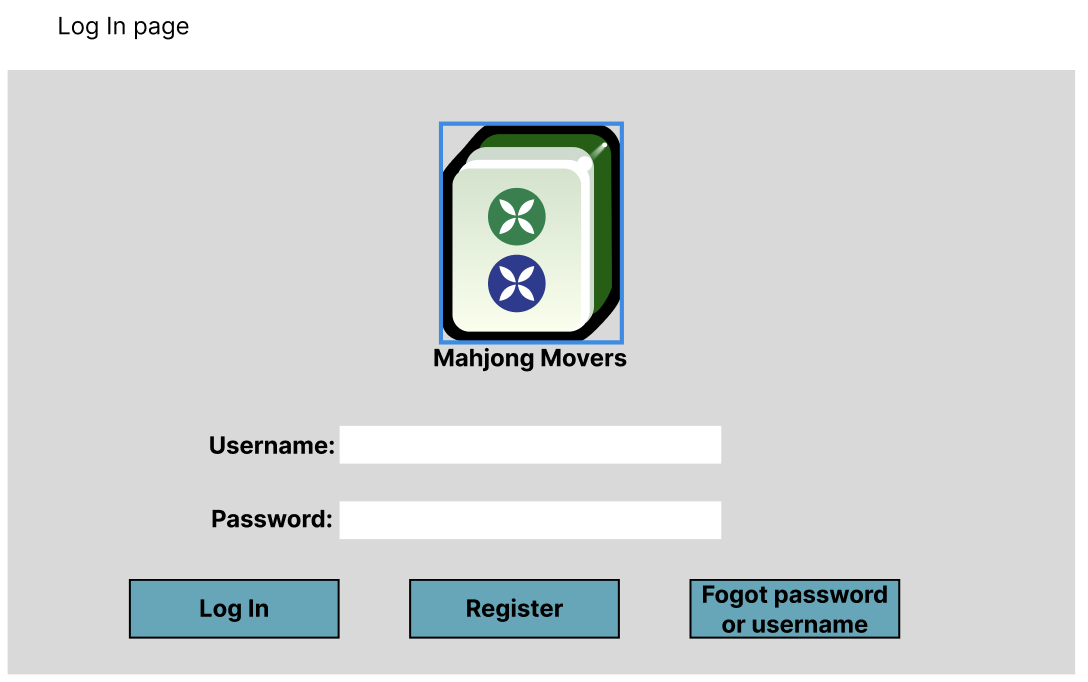
|  |  |
| --- | --- |
| Actor: | **-** |
| Description: | **Check Status confirms whether the user is logged in. If logged in, then it checks whether the user is currently in “Requester” mode or “Servicer” mode. It also checks whether the Job/ Service** |
| Preconditions: |  |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. **System checks whether user is logged in.** 2. **If logged in, System confirms the role of user.** |
| Alternative Flows: | **AF-S2-a.**   1. **If not logged in, System confirms that the user is not logged in.** |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | **SYSTEM\_UC\_2** | | |
| Use Case Name: | **Impose Penalty** | | |
| Created By: | **Shen Hwei** | Last Updated By: | **05.09.2022** |
| Date Created: | **05.09.2022** | Date Last Updated: | **05.09.2022** |

|  |  |
| --- | --- |
| Actor: | **-** |
| Description: | **Impose penalty reduces the points of the user** |
| Preconditions: | 1. **Job must be deleted, or Service must be retracted.** |
| Postconditions: |  |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. **System deducts 50 points, equivalent to SGD 3 from user.** 2. **System updates account information.** 3. **System notifies the user.** |
| Alternative Flows: | **AF-S1-a. Not enough points**   1. **If not enough points, System sends get request to bank.** 2. **Bank validates System.** 3. **Bank verifies System.** 4. **Bank deducts from account.** 5. **Bank updates System.** 6. **Use case continues from S2.** |
| Exceptions: |  |
| Includes: | **Update Account Information (S1).** |
| Extends: | **Delete Job, Retract Service.** |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: | **Job deleted implies that user is Requester.**  **Service Retracted implies that user is Servicer.** |

# 5.0 UI Mock-Ups

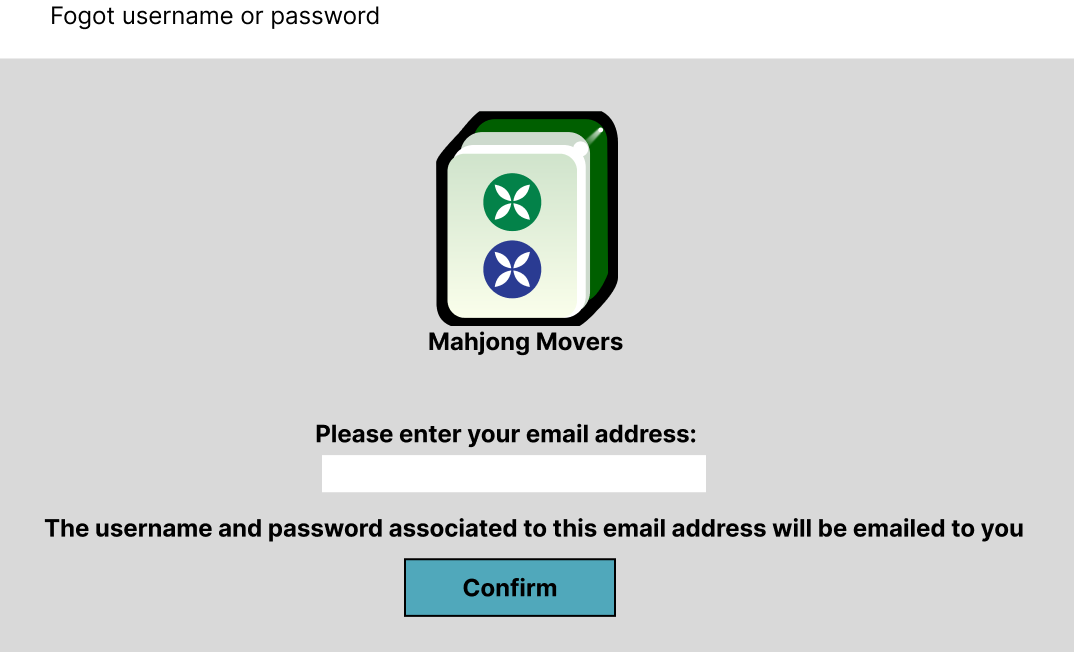
## 5.1 Login Page



## 5.1.1 Login Page (Error)



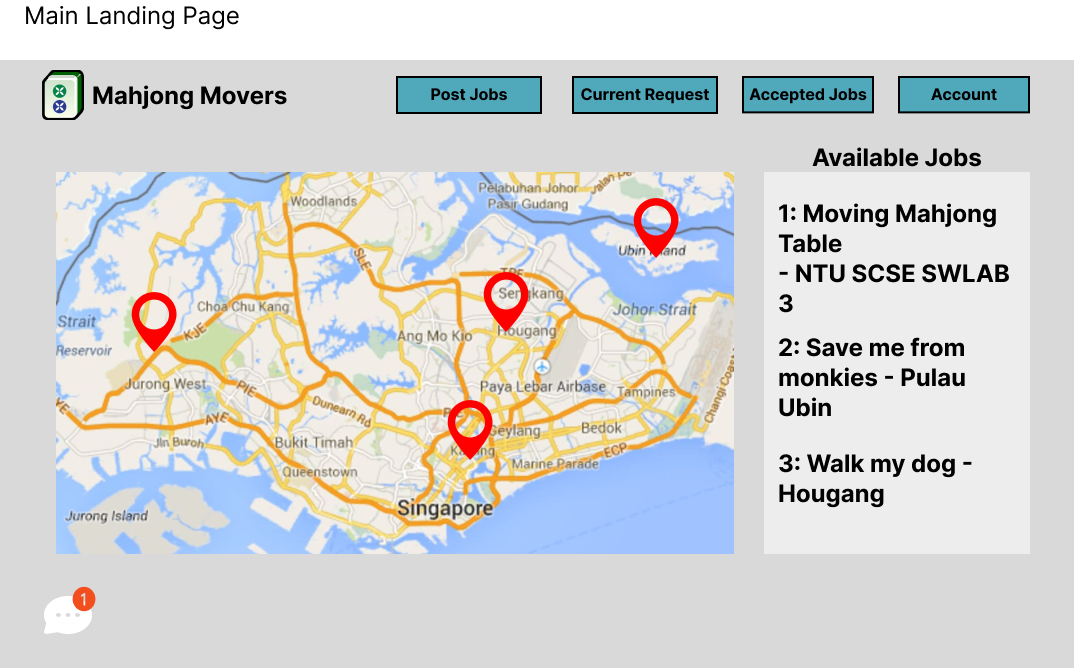
## 5.1.2 Login Page (Forgotten Password)



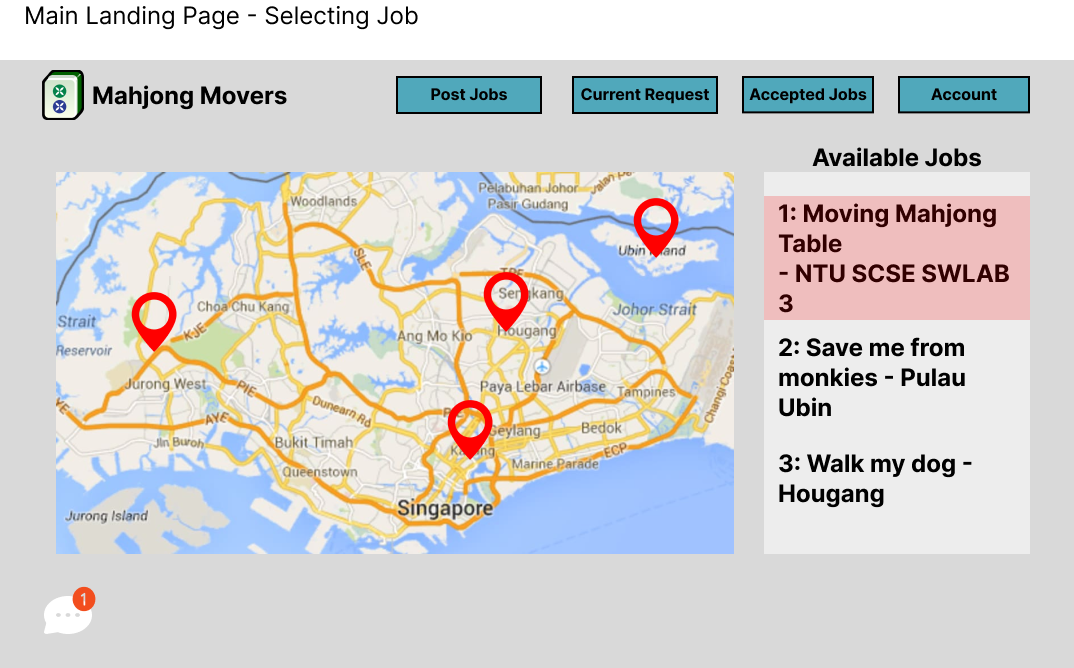
## 5.2 Registration Page



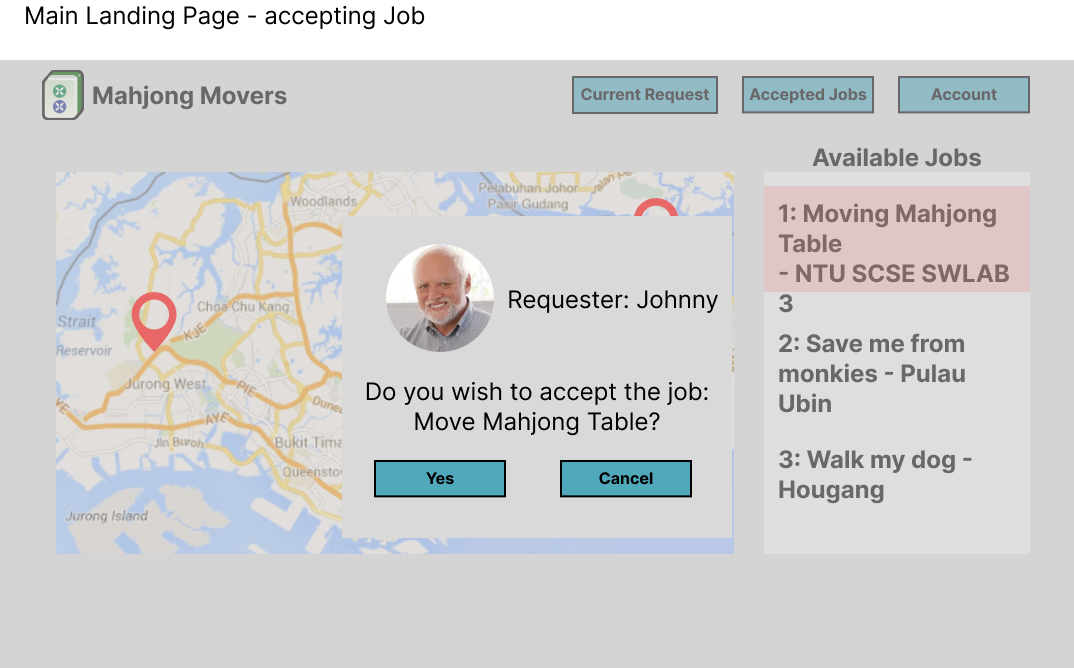
## 5.3 Landing Page



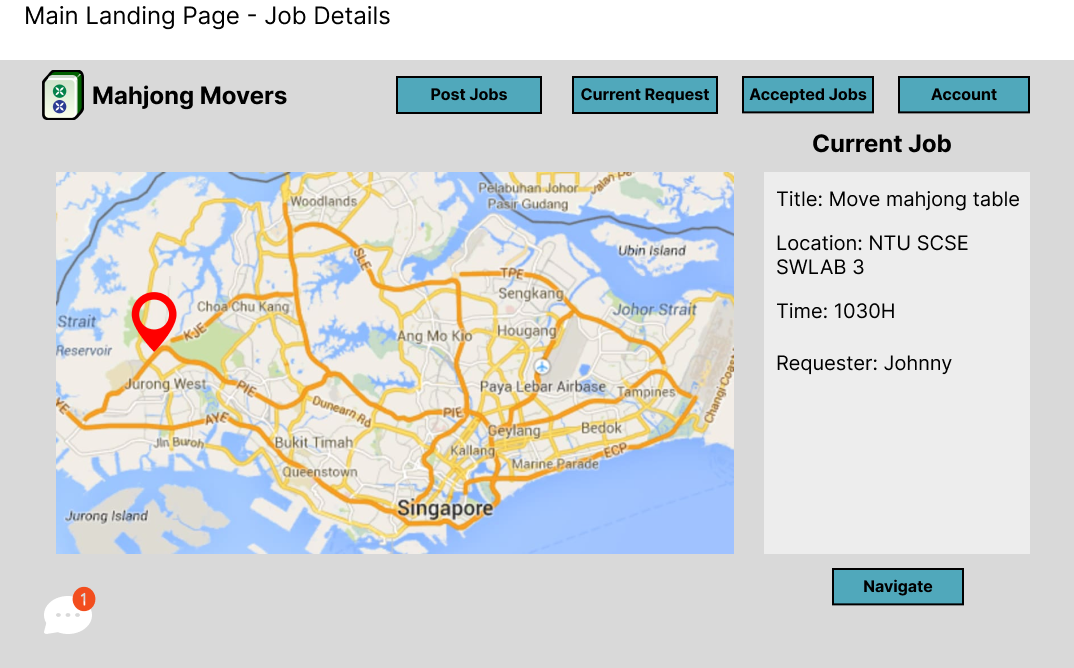
## 5.3.1 Landing Page (Selecting Job)



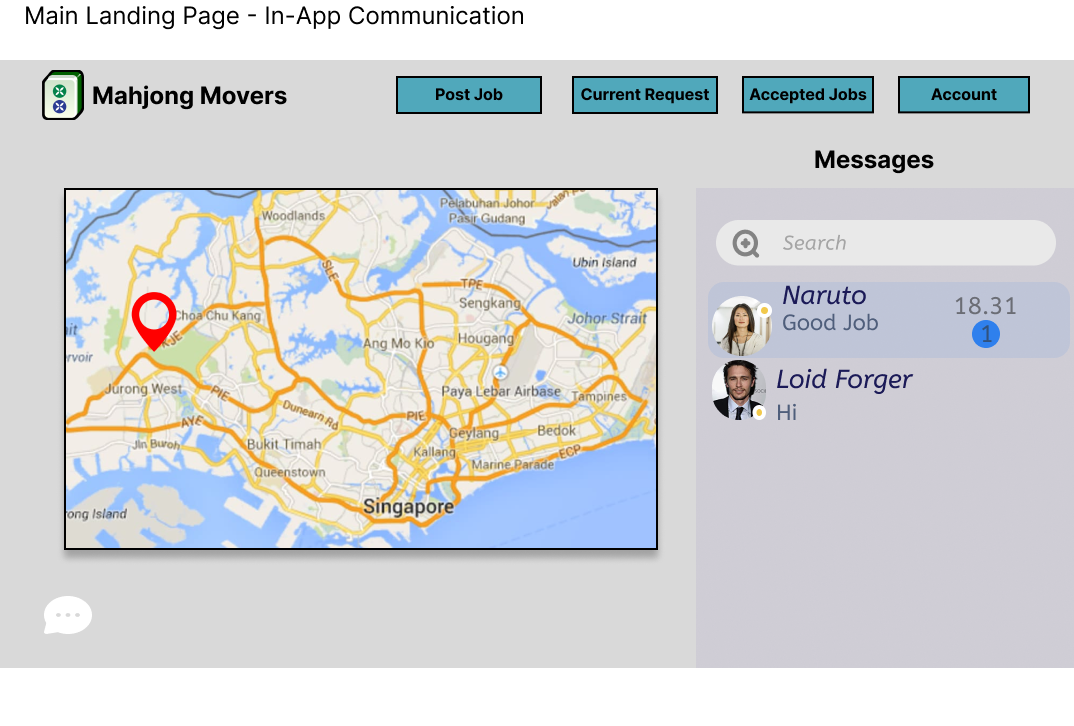
## 5.3.2 Landing Page (Accepting Job)



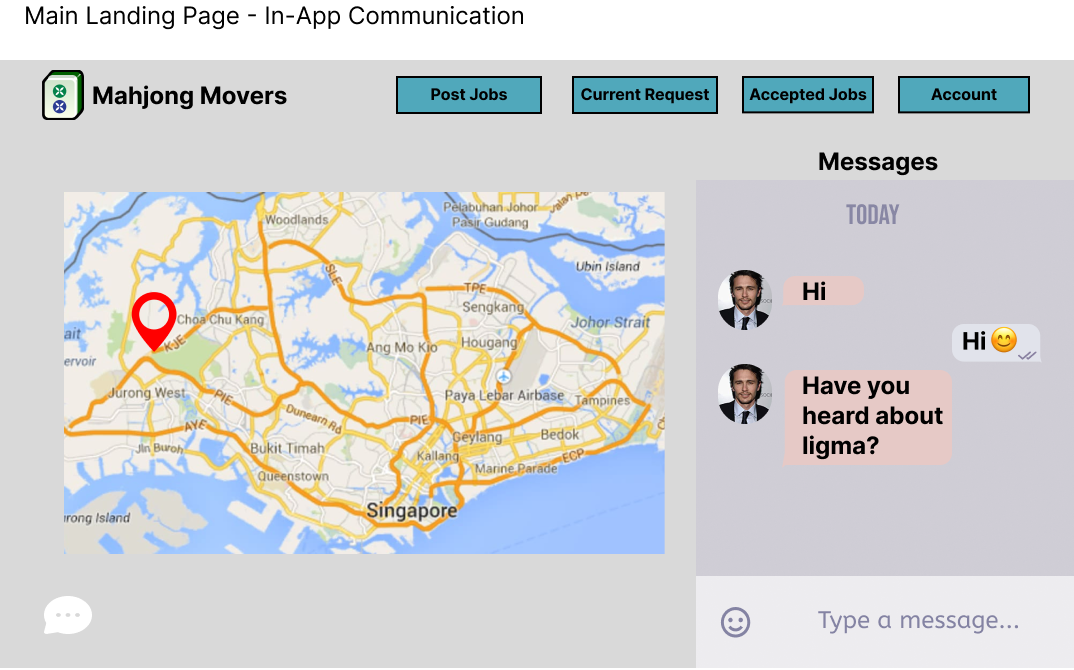
## 5.3.3 Landing Page (Job Details - Servicer)



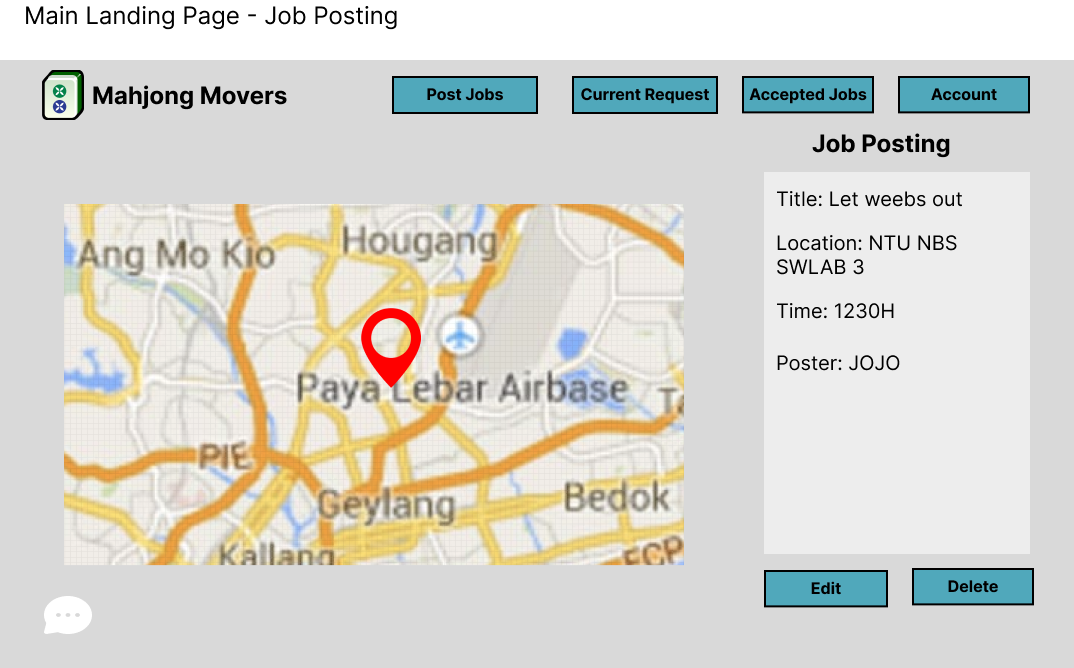
## 5.3.4 Landing Page (Messaging - Chat Logs)



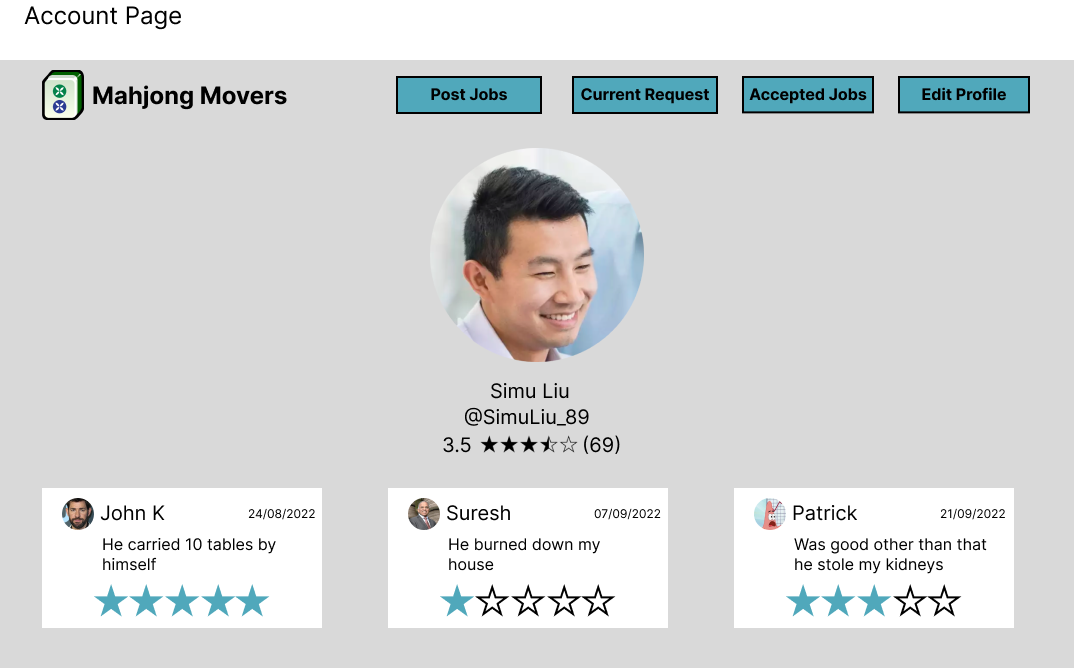
## 5.3.5 Landing Page (Messaging - User to User)



## 5.3.6 Landing Page (Job - Details - Requester)



## 5.4 Account Page



## 5.4.1 Account Page (Edit Details)

