

# Test Plan: Eden Fresh

Provided by Team Web Crawlers

Alexander Dung

Devarsh Patel

Justin Calma (Team Leader)

Kayla Chu

Sze Man Tang

Submission Date: December 15, 2021

# 1. Introduction

The objective of the test plan is to ensure that the web application, Eden Fresh, is going to work properly before being released. It outlines what component needs to be tested, how it's going to be tested, along with the testing schedule and deadlines. This document also outlines what our strategy is and the identification and mitigation of potential risks. The test plan will be used as a blueprint to ensure all the function testing will be done in an orderly and timely fashion.

## 2. Scope

### 2.1 In Scope

The following areas are in scope for this test plan:

- Be able to evaluate the main priorities of the project
- Test features to ensure that the project will work correctly
- Ensure that individual features will be able to function when grouped together
- Know what the risks are and what to do to avoid it
- State the timeframe that each component should be tested during

### 2.2 Out of Scope

The following areas are out of scope for this test plan:

- Assigning responsibilities and testing requirements to team members
- Going into low-level details

## 3. Objective

The objective of the test plan is to ensure all the components of the web application are working correctly. The goal is to test the user interface, security, usability, and functionality of the application. For the user interface portion, we want to ensure that the user can only see what is made visible to them and access/edit pages that are pertinent to them. We also want to make sure that the buttons are correctly functioning. For the security aspect, we want to make certain that all the user's private information can't be accessed by anyone other than themselves unless they give us permission otherwise. Usability is tested by making sure the website is user-friendly and easy to understand. Lastly, the functionality will be tested by ensuring that the user will be able to login/out of their account and the user will be able to interact with their account like how it is made to function (ie. create posts, edit posts, view posts, write reviews, chat, etc.).

## 4. Strategy

To make sure that the website will perform as best as it can, the test plan will be set up so that all the tests will be assigned a number according to its importance. In the case that we are not able to test everything we want to due to time constraints, the most important task will be done first.

## 5. Risks Identification and Risk Mitigation

Risks that are susceptible to occur within our web application are due to poorly executed test scenarios and overlooked requirements. Some risks that could be identified before the start of the implementation of each feature are from 3rd party APIs, requirements changes within a feature, and changes to dependencies that will affect the rest of the system. For example, we are planning to use a 3rd party maps and directions service, if this service were to fail then the feature that depends on the service and its dependencies will fail. This is an inevitable major risk to our web application since we do not own the 3rd party service. In order to avoid this risk, we would need to find backup solutions for the service or constantly be testing it throughout development. Requirement changes within all of the system requirements and unique features are a risk. If we were to add or remove user requirements in a given feature then it would affect the tests we plan on conducting. To mitigate this, we have identified and stated all of the requirements that the user should be able to do and did not include requirements that the user should not do. Similar to the previous risk statement, changes to dependencies can drastically affect the rest of the system. In order to mitigate this, we have already identified and included the requirements of the user for all the features. There should not be any changes to dependencies; however, if there were to be changed then we would make that test scenario have high priority and begin testing it.

## 6. Resource Requirements

The number of testers that we are delegating to each functionality in order to perform each test scenario within the schedule is 3. We would delegate 3 testers per functionality since there are at most 5 test scenarios per functionality. This would ideally result in 1 tester performing each test scenario in a particular functionality. Then the leftover test scenarios would be tested by a tester who has completed their test scenario first or the tester who has tested the dependency of the remaining test scenarios. However, some functionalities will only require 1 tester in order to test each test scenario i.e Miles Radius Search, Cuisine Filter, and Rating and Review.

## 7. Testing For Application Specific Features

### 7.1 Set-up Shop

#### Timeframe

2/22/2022 - 2/26/2022

#### Total Effort For All Test Cases

11 Hours

#### 7.1.1 Post Current Products For Sale With All Product Information

##### *Verification*

This functionality is verified when logged in users are given the ability to:

- Post current products for sale with all the product information
  - The current number of stock, price per pound or piece, images of the product, and a description of the product.

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Set-up Shop view

##### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - The user is able to post current products for sale with all the product information
  - User inputs valid data to the required fields and selects to post the created product post. The user is automatically directed to the view of the created post.
- Invalid Cases:
  - The user is unable to post the product, the current number of stock, price per pound or piece, images of the product, or a description of the product.
  - User inputs invalid data to the required fields and attempts to post the created product post. Access is denied and a system message will be displayed, "Invalid Information within Required Fields."

##### *Type of Test to be Conducted*

System Testing, Interface testing

*Priority: (High) 1 - 10 (Low)*

1 - This requirement has high priority since the other requirements depend on it in order to fully function.

*Effort Estimate:*

5 Hours

*Test Execution Steps:*

- Inputs:
  - Product listing with all of the needed information:
    - Product name, price, current stock, images, description
  - Inputs for a Normal Post
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Images - Two 40 MB images showing the product
    - Description - Freshly grown, non-GMO lychee
  - Inputs for a post that Exceeds the Constraints (Image size / Description length)
    - Product Name - Daikon Radish
    - Price - \$2.00 per piece
    - Current Stock - 10
    - Images - 1 50 MB Image, 2 60 MB Image
    - Description - (1000+) UTC-8 Characters
  - Inputs for a post that does not have the required information
    - Product Name - EMPTY FIELD
    - Price - \$5.00 per piece
    - Current Stock - 10
    - Images - 1 40 MB Image
    - Description - EMPTY FIELD
- Expected Outputs:
  - A posted product listing containing all of the inputted information
- Process:
  - User inputs all of the information required for posting the product
    - Product name, price, current stock, images, description
  - The user publishes their posting
- Testing Steps
  - A normal post
    - 1) Make a test post and ensure that it can be posted
    - 2) Once posted check if the post is viewable to the public
  - A post that isn't in the required limits
    - 1) Make a test post that exceeds the word limits and the image limits
    - 2) The system should not allow the user to do so

- a) If the user attempts to write more than the word limit, the excess would be truncated
- b) If the user attempts to upload a file that is too large, the system will not attempt to upload it and an error should pop up
- A post that does not have the required information
  - 1) Make a test post that does not have each of the required information (ie one without the price, one without the product name, etc)
  - 2) Ensure that the system does not post the unfinished post and gives an error message

### 7.1.2 Delete a Posted Product

#### *Verification*

This functionality is verified when logged in users are given the ability to:

- Delete a post regarding a product currently on sale which they have posted

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Set-up Shop view
- User must have an active product post

#### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - The user is able to delete the post that they have posted and it is deleted within 3 seconds
  - The user chooses to delete a product post that they have posted and confirms to delete it. All data saved for the product post will be permanently deleted from the system. Upon successful deletion, a system message will be displayed stating "Successful Deletion of Product Post." The user is automatically redirected to the Set-up Shop View.
- Invalid Cases:
  - The user is unable to delete the post that they have posted or it is not deleted within 3 seconds
  - The user chooses to delete the product post and confirms to delete it. However, the system does not delete the data for the post.
  - A system message is not shown upon deletion or an incorrect message is displayed.
  - The user is unable to determine whether the product post has successfully been deleted.

#### *Type of Test to be Conducted*

System Testing, Interface Testing

*Priority: (High) 1 - 10 (Low)*

2 - This requirement has the medium priority out of the three because it first requires requirement 7.1.1 to be implemented and functional first.

*Effort Estimate:*

3 Hours

*Test Execution Steps:*

- Inputs:
  - Product listing with all of the needed information:
    - Product name, price, current stock, images, description
  - Inputs for the post to be deleted
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Images - Two 40 MB images showing the product
    - Description - Freshly grown, non-GMO lychee
- Expected Outputs:
  - The selected product listing has been deleted and cannot be viewed
- Process:
  - User selects the product posting they want to delete
  - User selects delete
- Testing Steps
  - Delete a test post
    - 1) Delete a test post
    - 2) Ensure that the system removes the post from the user's profile
    - 3) Ensure that the post is no longer visible to the public

### 7.1.3 Post Preferred Meet-up Location

*Verification*

This functionality is verified when logged in users are given the ability to:

- Post their preferred meet-up location

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Set-up Shop view

*Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Users are able to post a valid meet-up location

- The user inputs a valid meet-up location in the specified field. A message will be displayed stating “Valid meet-up location has been posted.” The user is automatically redirected to the Set-up Shop view.
- Invalid Cases:
  - Users are unable to post a valid meet-up location
  - The user inputs an invalid meet-up location in the specified field. A message will be displayed stating “Invalid meet-up location has been posted.”
  - The user inputs a valid meet-up location but the system considers it invalid
  - The user is not automatically redirected to the Set-up Shop view

### *Type of Test to be Conducted*

System Testing, Interface Testing

### *Priority: (High) 1 - 10 (Low)*

3 - This requirement has the lowest priority out of the three because it does not rely on the other requirements.

### *Effort Estimate:*

3 Hours

### *Test Execution Steps:*

- Inputs:
  - Desired Meeting-up Location
  - Inputs for a valid meet-up location
    - 1250 Bellflower Blvd, Long Beach, CA 90840
  - Inputs for an invalid meet-up location
    - 1100 S Harbor Blvd, Fullerton, CA 99999
- Expected Outputs:
  - The inputted, desired meeting-up location has been posted and is viewable to other users
- Process:
  - The user inputs their desired meet-up location
  - The user saves their edit.
- Testing steps
  - Posting a valid meet up location
    - 1) Make a test meeting location (has to be an actual address)
    - 2) Ensure that the system makes the location viewable
  - Posting an invalid meeting location
    - 1) Make an invalid test meeting location
    - 2) Ensure the system can recognize that the location is not real
    - 3) The system should no allow the user to use the location and an error should show up



## 7.2 Reservation and Transaction

### Timeframe

2/29/2022 - 3/5/2022

### Total Effort For All Test Cases

14 Hours

#### 7.2.1 Request a Product Reservation

##### *Verification*

This functionality is verified when logged in users are given the ability to:

- Request a product reservation for a currently in-stock product

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Reservation and Transaction view

##### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Users are able to send a request for a valid amount of stock for an on-sale product that is sent to the seller within 3 seconds
  - The user inputs a valid amount of stock they want to reserve and confirm the amount. A message is displayed stating "You have successfully reserved (X) amount of (Y)." The user is automatically redirected to the view of the reserved product.
- Invalid Cases:
  - Users are able to send requests for a valid amount of stock for an on-sale product or the request is not sent within 3 seconds
  - The user inputs an invalid amount of stock and they attempt to confirm their reservation. A message is displayed stating "Invalid data has been inputted." The user is redirected back to the view of the product.
  - The user inputs a valid amount of stock and the system considers the amount invalid.
  - Upon successful reservation, a system message is not shown or an incorrect message is displayed.

##### *Type of Test to be Conducted*

System Testing, Performance Testing

*Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority because the rest of the feature requirements depend on it to fully function.

*Effort Estimate:*

5 Hours

*Test Execution Steps:*

- Inputs:
  - The desired amount of stock for a given product
  - Reserving a product within the quantity
    - Current Available Stock - 10
    - Desired Inputted Amount - 5
  - Reserving a product that is exactly the stock amount
    - Current Available Stock - 5
    - Desired Inputted Amount - 5
  - Reserving a product outside the quantity
    - Current Available Stock - 1
    - Desired Inputted Amount - 15
  - Reserving zero amount of product
    - Current Available Stock - 25
    - Desired Inputted Amount - 0
- Expected Outputs:
  - A sent product reservation request to the seller
- Process:
  - The user inputs the desired amount of stock for a product
  - The user selects to send their product reservation request to the seller
- Testing Step
  - Reserving a product within the quantity
    - 1) Attempt to reserve a product and ensuring that the desired amount we want is within the amount of stock the seller has
    - 2) The system should allow us to send the reservation request
    - 3) The system should return a confirmation message
  - Reserving a product that is exactly the stock amount
    - 1) Attempt to reserve a product with the amount being exactly the stock number
    - 2) The system should allow this and send a reservation request
    - 3) The system should return a confirmation message
  - Reserving a product outside the quantity
    - 1) Attempt to reserve a product the exceeds the quantity the seller has
    - 2) The system should not allow this and send a error message which states that it is outside the seller's stock
  - Reserving zero amount of product

- 1) Attempt to reserve a product with the quantity being zero
- 2) The system should not allow this and send an error message stating the quantity is invalid

### 7.2.2 Accept or Deny a Product Reservation Request

#### *Verification*

This functionality is verified when logged in users are given the ability to:

- Users are able to view the received product reservation request, accept the request, or deny it

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Reservation and Transaction view
- User must have received a product reservation request for a product they are selling

#### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Users who have received a product reservation request are able to view the request, accept it, or deny it
  - The user selects to accept the received product reservation. A message is displayed saying "The request has been accepted." The user selects to deny the received product reservation. A message is displayed saying "The request has been denied."
- Invalid Cases:
  - Users who have received a product reservation request are unable to view the request, accept it, or deny it
  - The user selects to accept or deny the received product reservation, and an incorrect message is displayed or no message has been displayed.
  - The user selects to accept the received product reservation. The system incorrectly marks an accepted request denied and a denied request accepted.

#### *Type of Test to be Conducted*

System Testing, Interface Testing

#### *Priority: (High) 1 - 10 (Low)*

2 - This requirement has a "2" priority since it helps represent what the feature is supposed to do and there are requirements that depend on this requirement.

*Effort Estimate:*

3 Hours

*Test Execution Steps:*

- Inputs:
  - A product reservation request that has been received
  - Accept Product Reservation Request
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
    - Accept / Deny - Accept
  - Deny Product Reservation Request
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
    - Accept / Deny - Deny
  - Do Nothing Product Reservation Request
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
    - Accept / Deny -
- Expected Outputs:
  - Notification sent to the buyer stating whether their request has been accepted or denied
  - Accept Product Reservation Request
    - "Your product reservation request has been accepted by the seller."
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
  - Deny Product Reservation Request
    - "Your product reservation request has been declined by the seller."
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
    - Accept / Deny - Deny
  - Do Nothing Product Reservation Request
    - "Your product reservation request has been declined by the seller."
    - Product Name - Lychee

- Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
    - Accept / Deny -
  -
- Process:
  - The seller receives the product reservation request and views it
  - The seller selects to accept or deny the request
  - The seller sends the decision to the buyer
- Testing Step
  - Accepting a Request
    - 1) Accept a request
    - 2) The system should send a confirmation to the buyer
    - 3) The system should also reduce the stock amount to the post
  - Denying a Request
    - 1) Denying a request
    - 2) The system should inform the buyer that their request has been rejected
  - Not doing anything with the request
    - 1) Leaving a request on hiatus
    - 2) The system will automatically deny the request after 48 hours
    - 3) Check that the request has been deleted after 48 hours

### 7.2.3 Cancel a Product Reservation

#### *Verification*

This functionality is verified when logged in users who have sent the request are given the ability to:

- Cancel the product reservation request that they have sent

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Reservation and Transaction view
- User must have sent a product reservation request for a product they are reserving

#### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Users who have sent the product reservation request are able to cancel it and it is no longer viewable to the receiver
  - User selects the product reservation request that they want for cancellation and confirms it. All information regarding the cancelled product reservation is permanently removed from the system. A message is displayed stating "The product reservation request has been successfully cancelled." The user is redirected to the Reservation and Transaction view.

- Invalid Cases:
  - Users who have sent the product reservation request are unable to cancel it or it is still viewable to the receiver after it has been cancelled
  - The user chooses to cancel the product reservation and confirms to cancel it. However, the product reservation request is still viewable to both parties.
  - A system message is not shown upon cancelation or an incorrect message is displayed.
  - The user is unable to determine whether the product post has successfully been canceled.

*Type of Test to be Conducted*

System Testing, Interface Testing

*Priority: (High) 1 - 10 (Low)*

4 - This requirement has a "4" priority because it relies on requirements 7.1.1 and 7.1.2 in order to work.

*Effort Estimate:*

2 Hours

*Test Execution Steps:*

- Inputs:
  - A product reservation request that has been sent to the seller
  - Product Reservation Request Inputs
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
- Expected Outputs:
  - A cancelled product reservation request that cannot be accepted or denied.
  -
- Process:
  - The buyer views their product reservation request
  - The buyer selects to cancel the request
  - The buyer saves their edit
- Testing Step
  - Deleting a product reservation
    - 1) Delete an already confirmed product reservation
    - 2) The system will inform the seller that the reservation has been cancelled
    - 3) The system will increase the stock again
    - 4) Check that the reservation had disappear from the seller's account

## 7.2.4 Change Reserved Amount

### *Verification*

This functionality is verified when logged in users who have sent the request are given the ability to:

- Change the initially desired reserved amount to a new valid desired amount

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Reservation and Transaction view
- User must have sent a product reservation request for a product they are reserving

### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - The user who has sent the product reservation request is able to change the reserved amount to a valid amount and the change is updated for the receiver
  - The user selects the product reservation request that they want to modify. The user inputs a valid amount of stock that they want to reserve. The system changes the initial reserved amount with the new amount. A message is displayed stating "The reserved amount of stock has successfully been changed." The user is redirected to the Reservation and Transaction view.
- Invalid Cases:
  - The user who has sent the product reservation request is unable to change the reserved amount to a valid amount or the change is not updated for the receiver
  - The user inputs an invalid amount of stock. A message is displayed stating "Invalid data has been inputted." The user is redirected back to the product request view.
  - The user inputs a valid amount of stock and the system considers it invalid. Or the user inputs an invalid amount of stock and the system considers it valid.
  - Upon successful modification, the system does not change the initial reserved amount to the new amount.

### *Type of Test to be Conducted*

Unit Testing, Interface Testing

### *Priority: (High) 1 - 10 (Low)*

5 - This requirement has the lowest priority because it does not rely on any of the requirements for this feature.

### *Effort Estimate:*

3 Hours

*Test Execution Steps:*

- Inputs:
  - A product reservation request that has been sent to the seller
  - Product Reservation Request Inputs (Within Stock Limit)
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Initially Desired Inputted Amount - 5
    - New Desired Inputted Amount - 10
  - Product Reservation Request Inputs (Outside Stock Limit)
    - Product Name - Daikon Radish
    - Price - \$2.00 per piece
    - Current Stock - 10
    - Initially Desired Inputted Amount - 5
    - New Desired Inputted Amount - 11
- Expected Outputs:
  - An updated product reservation request that contains the new desired amount of stock.
  - Updated Product Reservation Request (Within Stock Limit)
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 10
  - Product Reservation Request Inputs (Outside Stock Limit)
    - "The new amount that you desire is outside of the current available stock."
    - Product Name - Daikon Radish
    - Price - \$2.00 per piece
    - Current Stock - 10
    - Initially Desired Inputted Amount - 5
    - New Desired Inputted Amount - 11
- Process:
  - The buyer views their product reservation request
  - The buyer selects to update the amount of stock requested
  - The buyer inputs the amount of stock they desire
  - The buyer saves their edit
- Testing Step
  - Changing the reservation amount within the stock limit
    - 1) Change the stock amount ensuring that it is within the stock limit
    - 2) The system should change the amount and inform the seller of the change
    - 3) The seller has the ability to accept or reject the change
      - a) Accept:
        - i) The system will inform the buyer that the user has accepted the change



- ii) Ensure that the system updates the stock amount
- b) Deny:
  - i) The system will inform the buyer that the user has denied the stock change request
  - ii) Ensure that the request will revert back to the original amount
- Changing the reservation amount outside of the stock limit
  - 1) Change the reservation amount to a number that exceeds the stock limit
  - 2) The system will not allow the request to go through
  - 3) The system will show an error message informing the user that it is unable to do so

### 7.2.5 View Current State of Transaction

#### *Verification*

This functionality is verified when logged in users who have sent the request and received it are given the ability to:

- View the current state of the transaction

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Reservation and Transaction view
- User must have sent a product reservation request for a product they are reserving

#### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Both parties are able to see the current state of the transaction that is taking place
  - The user selects a product reservation request that they have. The user is directed to a view containing all information regarding the selected product reservation request.
- Invalid Cases:
  - Both parties are unable to see the current state of the transaction that is taking place
  - The current state of the transaction is not accurate
    - The current state is not being updated or has been updated to the wrong state
  - The user selects a product reservation request and they are not directed to the view containing all information regarding the selected product reservation request. Or they are directed to the view but it contains incorrect information.
  - The user has a product reservation request but they are unable to view them.

### *Type of Test to be Conducted*

#### Interface Testing

### *Priority: (High) 1 - 10 (Low)*

3 - This requirement has a medium priority because it is one of the requirements that depict what this feature represents and it does not have any dependencies.

### *Effort Estimate:*

1 Hour

### *Test Execution Steps:*

- Inputs:
  - Accepted Product Reservation Request (Processing)
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
    - Accept / Deny - Accept
    - Current State - Processing
  - Accepted Product Reservation Request (Pending Approval)
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
    - Accept / Deny - Accept
    - Current State - Pending Approval
  - Accepted Product Reservation Request (Ready)
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Desired Inputted Amount - 5
    - Accept / Deny - Accept
    - Current State - Ready
- Expected Outputs:
  - A view that shows all of the information regarding the request and the current state of the transaction.
  -
- Process:
  - The user selects the product reservation request
  - The user views the product reservation request and sees all of the information regarding the request
- Testing Step
  - Viewing the current state of a transaction

- 1) Test each state of the transaction ensuring that as the state is changed on the seller side, the state should also get updated on the user side.

## 7.3 Request Fulfillment

### Timeframe

38/2022 - 3/12/2022

### Total Effort For All Test Cases

11 Hours

#### 7.3.1 Post a Product Request Fulfillment Bid

##### *Verification*

This functionality is verified when logged in users are given the ability to:

- Post a request fulfillment bid that is viewable to other logged in users

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Request Fulfillment view

##### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Users are able to post a request fulfillment bid that is viewable to other logged in users within 3 seconds
  - User inputs valid data to the required fields and selects to post the created product request fulfillment bid. The user is automatically directed to the view of the created product request fulfillment bid.
- Invalid Cases:
  - Users are unable to post a request fulfillment bid
  - The bid is not viewable to other logged in users or it takes longer than 3 seconds to be viewable
  - User inputs invalid data to the required fields and selects to post the created product request fulfillment bid. A message is displayed stating "Invalid data has been inputted." The user is redirected to the request fulfillment view.

##### *Type of Test to be Conducted*

System Testing, Interface Testing

*Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since the other requirements for this feature depend on it to be fully functional before it can be tested.

*Effort Estimate:*

5 Hours

*Test Execution Steps:*

- Inputs:
  - All of the needed information for the requested product:
    - Product name and desired stock
  - Product Request Fulfillment Bid (Normal)
    - Product Name - Bok Choy
    - Desired Amount - 50
  - Product Request Fulfillment Bid (Zero Amount)
    - Product Name - Bok Choy
    - Desired Amount - 0
- Expected Outputs:
  - A posted product request fulfillment bid containing all of the inputted information, which can be viewed by all users
  -
- Process:
  - User inputs all of the information required for posting the product
    - Product name, desired stock
  - The user publishes their posting
- Testing Step
  - Post a normal request fulfillment bid
    - 1) Post a test request
    - 2) Check that the system posted the request
    - 3) Check that the post is viewable to everyone
  - Post a request fulfillment bid that is outside the limits
    - 1) Make a test request that exceeds the word limits and the image limits
    - 2) The system should not allow the user to do so
      - a) If the user attempts to write more than the word limit, the excess would be truncated
      - b) If the user attempts to upload a file that is too large, the system will not attempt to upload it and an error should pop up
  - Post a request fulfillment bid that doesn't have the required information (ie stock information, product name, etc)
    - 1) Make a test request fulfillment bid that does not have each of the required information (ie one without the price, one without the product name, etc)
    - 2) Ensure that the system does not post the unfinished post and gives an error message

### 7.3.2 Reply to Product Request Fulfillment Bid

#### *Verification*

This functionality is verified when logged in users are given the ability to:

- Reply to a request fulfillment bid with their proposed price offer

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Request Fulfillment view
- There must be active product request fulfillment bids

#### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Users are able to reply to a request fulfillment bid with their proposed price offer
  - The user selects a product request fulfillment bid they want to reply to. They input valid information and select to reply to the post. A message is displayed stating “The offer has been successfully posted.” The user is directed back to the product request fulfillment bid view.
- Invalid Cases:
  - Users are unable to reply to a request fulfillment bid with their proposed price offer
  - User input valid information and select to reply to the post. A message is displayed stating “Invalid data has been inputted.” The user is redirected to the product request fulfillment bid view.

#### *Type of Test to be Conducted*

System Testing, Performance Testing

#### *Priority: (High) 1 - 10 (Low)*

2 - This requirement has a “2” priority because it is the requirement that depends on the previously stated requirement. This requirement also has dependencies (7.3.3).

#### *Effort Estimate:*

2 Hours

#### *Test Execution Steps:*

- Inputs:
  - Product listing with all of the needed information:
    - Product name, desired stock
  - The price offer for the product request fulfillment bid
  - Product Request Fulfillment Bid
    - Product Name - Bok Choy

- Desired Amount - 50
  - Offer to the Product Request Fulfillment Bid (Non-Zero)
    - Price Offer - \$4.00 (POUND OR PIECE)
  - Offer to the Product Request Fulfillment Bid (Zero)
    - Price Offer - \$0
    -
- Expected Outputs:
  - A posted reply to the product request fulfillment bid containing their inputted price offer.
  - Offer to the Product Request Fulfillment Bid (Non-Zero)
    - Product Name - Bok Choy
    - Desired Amount - 50
    - Price Offer - \$4.00
- Process:
  - User views and selects the product request fulfillment bid
  - User inputs the price offer for the selected bid
  - The user selects to post their reply
- Testing Step
  - Replying to a fulfillment bid
    - 1) Reply to a fulfillment bid
    - 2) Check if the system sends a notification to the buyer
    - 3) The system should also allow the buyer accept or reject

### 7.3.3 Accept Product Request Fulfillment Bid

#### *Verification*

This functionality is verified when logged in users who have posted the request fulfillment bid are given the ability to:

- Accept the offer that has been replied to on their posted request fulfillment bid and it is removed and no longer viewable to other users

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Request Fulfillment view
- User must have posted a product request fulfillment bid

#### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Users are able to accept the offer that has been replied to on their product fulfillment bid, the post is removed, and no longer viewable to other users
  - The user selects to accept the received bid offer. A message is displayed saying "The bid has been accepted." The user selects to deny the received bid offer. A message is displayed saying "The bid has been denied."

- Invalid Cases:
  - Users are unable to accept the offer that has been replied to on their product fulfillment bid
  - The post is not removed after the offer has been accepted
  - The post is still viewable to other users even after the offer has been accepted

### *Type of Test to be Conducted*

System Testing, Performance Testing

### *Priority: (High) 1 - 10 (Low)*

3 - This requirement has a “3” priority because it depends on other requirements to be tested first.

### *Effort Estimate:*

3 Hours

### *Test Execution Steps:*

- Inputs:
  - Product request fulfillment bid with all of the needed information:
    - Product name, desired stock
  - Reply to the product request fulfillment bid with the price offer
  - Offer to the Product Request Fulfillment Bid (Non-Zero)
    - Product Name - Bok Choy
    - Desired Amount - 50
    - Price Offer - \$4.00
    - Accept - Accept
- Expected Outputs:
  - An accepted posted request fulfillment bid
  - The request is no longer viewable to other users besides the buyer and the seller
  -
- Process:
  - The buyer selects and views their posted product request fulfillment bid
  - They select the price offer that they want
  - They accept the price offer
- Testing Step
  - 1) When an offer is sent, the buyer gets to pick whether or not to accept or reject it
  - 2) Once an option is picked, the system will send a notification to the seller
    - a) If accept:
      - i) The system will remove the request fulfillment bid
      - ii) System inform the seller that they buyer accepts their offer
    - b) If deny:
      - i) System informs the seller that the buyer denied their offer

### 7.3.4 View Current State of Transaction

#### *Verification*

This functionality is verified when logged in users who have posted the request fulfillment bid and had their proposed price offer accepted are given the ability to:

- View the current state of the transaction

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the Request Fulfillment view
- User must have an accepted product request fulfillment bid

#### *Validation*

This functionality is validated when testing for the following valid and invalid test cases:

- Valid Cases:
  - Both parties are able to see the current state of the transaction that is taking place
  - The user selects a product request fulfillment bid that they have. The user is directed to a view containing all information regarding the selected product request fulfillment bid.
- Invalid Cases:
  - Both parties are unable to see the current state of the transaction that is taking place
  - The current state of the transaction is not accurate
    - The current state is not being updated or has been updated to the wrong state
  - The user selects a product request fulfillment bid and they are not directed to the view containing all information regarding it. Or they are directed to the view but it contains incorrect information.
  - The user has an accepted product request fulfillment bid but they are unable to view them.

#### *Type of Test to be Conducted*

Interface Testing

#### *Priority: (High) 1 - 10 (Low)*

4 - This requirement has the lowest priority because it is a requirement that does not need to be immediately tested since it is at the lowest level of dependencies.

#### *Effort Estimate:*

1 Hours



*Test Execution Steps:*

- Inputs:
  - A product request fulfillment bid that has been accepted by the buyer
  - Offer to the Product Request Fulfillment Bid (Non-Zero)
    - Product Name - Bok Choy
    - Desired Amount - 50
    - Price Offer - \$4.00
    - Accept - Accept
- Expected Outputs:
  - A view that shows all of the information regarding the request and the current state of the transaction.
  -
- Process:
  - The user selects the product request fulfillment bid
  - The user views the product reservation request and sees all of the information regarding the request
- Testing Steps
  - 1) Go to the desired transaction
  - 2) Click on the transaction
  - 3) The transaction Status is shown

## 7.4 Miles Radius Search

### Timeframe

3/15/2022 - 3/19/2022

### Total Effort For All Test Cases

4 Hours

#### 7.4.1 Input Search Location

##### *Verification*

This functionality is verified when logged in users are able to insert their location into the search bar

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be on the Miles Radius Search view

##### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:

- Users are able to input a location of their choosing into the search bar
- Users are able to select a current location if they want to use their current location
- If the user inputs an invalid location an error should show up
- The user inputs a valid location or they allow the application to use their current location. The system directs the user to a map view containing their search results.
- Invalid Cases:
  - The system doesn't correct the invalid address and just shows an empty search result
  - The user attempts to input an invalid location and a message will be displayed stating "Invalid location has been inputted." The user is directed back to the Miles Radius Search view.
  - The user inputs a valid location and the system considers it to be invalid. The user inputs an invalid location and the system considers it to be valid.

### *Type of Test to be Conducted*

Unit Testing, System testing

### *Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since the other requirements for this feature depend on it to be fully functional before they can be tested.

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - The location that the user wants to use as the search location:
    - Desired inputted location to be used as the search location
    - The current location of the user
  - Desired Inputted Location
    - 7881 Edinger Ave Ste 150, Huntington Beach, CA 92647
  - User's Current Location
    - 1250 Bellflower Blvd, Long Beach, CA 90840
  - Invalid Inputted Location
    - Abc 123
- Expected Outputs:
  - A map that shows all of the ingredient listings around their search location.
  -
- Process:
  - The user inputs the location that they want to use as their search location
  - The user selects to search using their inputted location

- The map is updated to show all of the ingredient listings around their search location
- Testing Steps
  - Inputting a Valid Location
    - 1) Go to the search bar and input a valid location
    - 2) The system will find all the products for sale near the given location
  - Inputting an invalid location
    - 1) Go to the search bar and input an invalid or incomplete location
    - 2) The system will send a error message stating that the address is not valid
  - Using current location (when the user has given permission)
    - 1) Ensure that the system is given permission to get the user's location
    - 2) go to the search bar and select current location
    - 3) The system should be able to pinpoint the location the user is at and find all the products near the user
  - Using current location (when user hasn't given permission)
    - 1) Ensure that the system has not been given permission to get the user's location
    - 2) go to the search bar and select current location
    - 3) The system should give an error stating that it is unable to get the user's location because it is not allowed to
  - Using current location with location service turned off
    - 1) Ensure that the computer's location service is turned off
    - 2) Go to the search bar and select current location
    - 3) The system should give an error stating that it is unable to access the user's current location

### 7.4.2 Select Miles Radius Search Constraint

#### *Verification*

This functionality is verified when logged in users are able to filter their search radius based on the chosen location and the selected mile

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be on the Request Fulfillment view

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users are able to filter their results down to the select miles based on their chosen location
  - The user will be unable to choose their current location as their starting point if they do not give permission to the system

- The user selects the miles radius constraint that they want to use for their search. The system will accurately display all results that are within the selected search constraint.
- Invalid Cases:
  - The user is unable to filter the results down to the selected miles
  - The user selects the miles radius constraint that they want to use for their search. The system does not accurately display the results with the selected search constraint.

### *Type of Test to be Conducted*

Unit Testing, Interface Testing

### *Priority: (High) 1 - 10 (Low)*

2 - This requirement has the lowest priority since it depends on the previously stated requirement to be completed first.

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - The location that the user wants to use as the search location:
    - Desired inputted location to be used as the search location
    - The current location of the user
  - The user selects the constraint they want to apply to their search
    - 5 miles, 10 miles, 15 miles
  - Inputs for Selecting the 5 Miles Radius Constraint
    - Desired Inputted Location
      - 7881 Edinger Ave Ste 150, Huntington Beach, CA 92647
    - User's Current Location
      - 1250 Bellflower Blvd, Long Beach, CA 90840
  - Inputs for Selecting the 10 Miles Radius Constraint
    - Desired Inputted Location
      - 7881 Edinger Ave Ste 150, Huntington Beach, CA 92647
    - User's Current Location
      - 1250 Bellflower Blvd, Long Beach, CA 90840
  - Inputs for Selecting the 15 Miles Radius Constraint
    - Desired Inputted Location
      - 7881 Edinger Ave Ste 150, Huntington Beach, CA 92647
    - User's Current Location
      - 1250 Bellflower Blvd, Long Beach, CA 90840
- Expected Outputs:
  - A map that shows all of the ingredient listings around their search location.

- 
- Process:
  - The user inputs the location that they want to use as their search location
  - The user inputs the constraint that they want to apply on their search
  - The user selects to search using their inputted location
  - The map is updated to show all of the ingredient listings around their search location
- Test Steps
  - Miles radius search with a valid location
    - 1) Ensure the address/current location is valid
    - 2) Select a miles radius search
    - 3) Check that all the results that show are within that radius search and no search within the radius is left out
  - Miles radius search with an invalid location
    - 1) Ensure the address/current location is invalid
    - 2) Select a miles radius search
    - 3) The system will send out an error

## 7.5 Cuisine Filter

### Timeframe

3/15/2022 - 3/19/2022

### Total Effort For All Test Cases

5 Hours

#### 7.5.1 Select and Apply Predefined Cuisine to Filter Results

##### *Verification*

This functionality is verified when logged in users are able to filter their search based on the chosen cuisine

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be on the Miles Radius Search view

##### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users are able to filter their results based on the chosen cuisines
  - Users are only able to pick the default cuisines ie. they can not input the cuisine of their choosing

- User selects the predefined cuisine that they want to filter their search results with. The system will display their search results with the filters that they have selected.
- Invalid cases:
  - Results do not get filtered down based on cuisines
  - User selects a predefined cuisine that they want to filter their results with, but the system does not accurately display their results.

### *Type of Test to be Conducted*

Unit Testing, Interface Testing

### *Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since it does not have any other requirement for this particular feature.

### *Effort Estimate:*

5 Hours

### *Test Execution Steps:*

- Inputs:
  - The generated map after executing the miles radius search feature
  - The selected predefined filter:
    - Asian: Chinese, Japanese, Indian, Korean, Thai, Vietnamese
    - European: French, Greek, Italian, Spanish
    - Latin American: Mexican
  - Search Location
    - 1250 Bellflower Blvd, Long Beach, CA 90840
  - Filter Selected
    - Japanese
- Expected Outputs:
  - A map that shows all of the ingredient listings around their search location filtered by the cuisine filter the user has selected.
  -
- Process:
  - The user inputs the location that they want to use as their search location
  - The user selects to search using their inputted location
  - The map is updated to show all of the ingredient listings around their search location
  - The user selects the cuisine filter that they want to apply to their search results
  - The map is updated to show all of the ingredient listings around their search location filtered by the selected cuisine filter
- Testing Step
  - Miles radius search with a valid location

- 1) Ensure the address/current location is valid
  - 2) Select a miles radius search
  - 3) Select the cuisine choice
  - 4) Check that all the results that show are within that radius search and that all the results are the correct cuisine
- Miles radius search with an invalid location
    - 1) Ensure the address/current location is invalid
    - 2) Select a miles radius search and cuisine
    - 4) The system will send out an error

## 7.6 Real-Time Notifications

### Timeframe

3/22/2022 - 3/26/2022

### Total Effort For All Test Cases

12 Hours

#### 7.6.1 Receive Notifications

##### *Verification*

This functionality is verified when logged in users are able to receive notifications as long as they opt-in for it

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in

##### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The user has the ability to opt-in to get notifications
  - The user is able to receive notifications
  - The user selects to receive notifications. The system sends a notification to the user stating "You will now receive notifications."
- Invalid cases:
  - Even if the user opts into getting notifications, the notifications are appearing
  - The user selects to receive notifications and the system does not send a notification to the user.
  - The user does not select to receive notifications but the system sends notifications to the user.

*Type of Test to be Conducted*

System Testing, Interface Testing

*Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since the other requirements for this feature depend on it to be fully functional before they can be tested.

*Effort Estimate:*

4 Hours

*Test Execution Steps:*

- Inputs:
  - The user that receives default notifications i.e notifications that users are already able to receive by default
  - User:
    - John Doe
  - Default Notifications Received:
    - Current Status of Product Reservation
      - Your Bok Choy reservation request has transitioned the the "Ready" phase
- Expected Outputs:
  - A notification that states what the notification is about:
    - Following a seller:
      - Shows newly posted products, updates to products
    - Their current reservation or request:
      - Shows if it has been accepted or denied
      - Shows updates to the current state of their transaction
    - Direct messages
    - Ratings and reviews
  -
- Process:
  - The user gets a notification regarding product updates, direct messages, ratings and reviews made about them or their products, and update notifications regarding their current reservations
  - The user views the notification
- Testing Step
  - 1) The system will ask the user if they want to opt into notifications when they first log into the web application
    - a) Opt into the notifications
      - i) Notifications should be sent
    - b) Opt out
      - i) No notifications should be viewable
  -



## 7.6.2 View Notifications

### *Verification*

This functionality is verified when logged in users are able to view a list of their notifications in the last 60 days

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in

### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users are able to see a list of their notifications within the last 60 days
  - The user selects to view all of the notifications that they have received within the last 60 days. The system redirects the user to the view containing all of the notifications that they have received.
- Invalid cases:
  - Notifications aren't showing up in the list
  - The user selects to view all of the notifications that they have received within the last 60 days. The system does not direct the user to the view containing all of the notifications that they have received.
  - The user is directed to the view but it does not contain accurate information regarding their notifications

### *Type of Test to be Conducted*

Interface Testing, System Testing

### *Priority: (High) 1 - 10 (Low)*

2 - This requirement has a "2" priority because it is the requirement that depends on the previously stated requirement. This requirement also has dependencies (7.6.1).

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - The user that receives notifications because they follow a seller, and have a current reservation or request
  - User:
    - John Doe
  - Notifications
    - Product update

- Lychee has been restocked
- Expected Outputs:
  - A view that shows what the notification is about:
    - Following a seller:
      - Shows newly posted products, updates to products
    - Have a current reservation or request:
      - Shows if it has been accepted or denied
      - Shows updates to the current state of their transaction
- Process:
  - The user selects the notification that they have received
  - The user views the notification
- Testing step
  - 1) Send a test notification
  - 2) See if the notification get saved into t

### 7.6.3 Opt-In to Receive Notifications

#### *Verification*

This functionality is verified when logged in users are able to opt into getting notifications into receiving notifications anytime on the settings page

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the settings view

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The users have the ability to opt in to receiving notifications
  - The user selects to opt-in to receive notifications other than default notifications. The system sends a notification to the user stating "You will now receive notifications."
- Invalid cases:
  - The system won't let the user opt in to receiving notifications
  - The user selects to receive notifications and the system does not send a notification to the user.
  - The user does not select to receive notifications but the system sends notifications to the user.

#### *Type of Test to be Conducted*

System Testing, Security Testing

*Priority: (High) 1 - 10 (Low)*

3 - This requirement has a “3” priority because it depends on other requirements to be tested first.

*Effort Estimate:*

2 Hours

*Test Execution Steps:*

- Inputs:
  - Notifications that users are only able to get for following a seller
    - Product update, newly posted products
  - User:
    - John Doe
  - Notification to opt-in to
    - Product update
      - Lychee has been restocked
- Expected Outputs:
  - The user has now opted-in to receive “followed seller” notifications and the default notifications.
- Process:
  - The user opts-in to receive “followed seller” notifications
  - There is a newly posted product from a seller they follow or an update to a posted product
  - The user gets a notification regarding newly posted products or an update to a product
  - The user views the notification
- Test Step
  - 1) Go into the settings page
  - 2) Find the notifications and turn it on
  - 3) Notifications should be sent

#### 7.6.4 Opt to Not Receive Notifications

*Verification*

This functionality is verified when logged in users are able opt-out of receiving notifications

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the settings view

*Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - At any time, users are able to opt-out in to receive product update notifications, direct message notifications, ratings and reviews made about them or their products, and notifications from sellers that they follow
- Invalid cases:
  - Even after the user opts out of notifications, the notifications still show up
  - The user is unable to opt-out in the settings

#### *Type of Test to be Conducted*

System Testing, Security Testing

#### *Priority: (High) 1 - 10 (Low)*

4 - This requirement has a "4" priority because it depends on other requirements to be tested first.

#### *Effort Estimate:*

2 Hours

#### *Test Execution Steps:*

- Inputs:
  - Default notifications that users are able to receive and notifications that they have opted-in to receive
  - User:
    - John Doe
  - Notifications to opt-out of:
    - Notification regarding product updates, direct messages, ratings and reviews made about them or their products, and update notifications regarding their current reservations
- Expected Outputs:
  - The user has now opted out of receiving default notifications and notifications that they have opted-in to receive.
- Process:
  - The user opts out to receive default notifications and "followed seller" notifications
  - There is a notification regarding product updates, direct messages, ratings and reviews made about them or their products, and update notifications regarding their current reservations
- Testing Steps
  - User rejects receiving notifications at the beginning
    - 1) The system should ask the user if the user wants to turn on notification or not
    - 2) The user will deny
    - 3) Check if the system sends notifications
  - User goes into setting and turns off notification

- 1) Go to the settings page and turn off the notifications
- 2) check if the system stops sending out notifications

### 7.6.5 Deleting Notifications

#### *Verification*

This functionality is verified when logged in users are able delete notifications from their list

Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must be on the setting view

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The user has the ability to delete notifications
  - Any notifications older than 60 days will be deleted by the system
  - The user selects a notification(s) for deletion and they confirm it. The system permanently deletes the notifications that the user has selected. A message is displayed stating "Deletion is successful."
- Invalid cases:
  - There is no option for the user to delete notifications
  - The system doesn't delete the notifications after 60 days
  - The user selects the notification(s) that they want for deletion. The system does not permanently delete the notifications.
  - A system message is not shown upon deletion or an incorrect message is displayed.
  - The user is unable to determine whether the notification has successfully been deleted.

#### *Type of Test to be Conducted*

System testing, Interface testing

#### *Priority: (High) 1 - 10 (Low)*

5 - This requirement has the lowest priority since it depends on all the other requirements to be tested first.

#### *Effort Estimate:*

2 Hours

#### *Test Execution Steps:*

- Inputs:

- Notifications that the user has received
- Product Reservation Notification
  - “Your product reservation request has been accepted by the seller.”
  - Product Name - Lychee
  - Price - \$3.00 per pound
  - Current Stock - 25
  - Desired Inputted Amount - 5
- Expected Outputs:
  - The notifications that have been selected for deletion have been deleted and are no longer viewable
- Process:
  - The user selects the received notifications to delete
  - The user saves the edit
- Testing Step
  - 1) Go to the notification list and find a notification
  - 2) select the notification and select delete
  - 3) Check if the notification is deleted

## 7.7 Transactions View

### Timeframe

3/22/2022 - 3/26/2022

### Total Effort For All Test Cases

8 Hours

#### 7.7.1 View All Product Reservations and Request Fulfillment Bids

##### *Verification*

This functionality is verified when logged in users are able to view relevant information regarding their product reservation and requests

##### Pre-conditions:

- User must have an active authenticated session on the device,
- User must be logged in
- User must have a product reservation or request fulfillment bid
- User must be on the Transactions view

##### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:

- Users are able to make product reservations given that their reservation amount is less than or equal to the amount the seller has
- Users are able to post their own request for a specific product along with the amount they need
- All the reservation and request a user makes will be stored in a list
- Users will be able to see the status of their reservation and request
- The user selects to view all of the product reservations and request fulfillment bids that they have. The system redirects the user to the view containing all of the reservations and request fulfillment bids.
- Invalid cases:
  - Users are unable to post a request
  - Users are unable to reserve a product
  - Users are able to reserve an amount greater than the stock the seller has
  - The user selects to view all of the product reservations and request fulfillment bids that they have. The system does not direct the user to the view containing all of the reservations and request fulfillment bids.
  - The user is directed to the view but it does not contain accurate information regarding their product reservations and fulfillment bids

#### *Type of Test to be Conducted*

System Testing, Interface Testing

#### *Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since the other requirements for this feature depend on it to be fully functional before they can be tested.

#### *Effort Estimate:*

3 Hours

#### *Test Execution Steps:*

- Inputs:
  - An accepted product reservation request or product request fulfillment bid
  - Product Request Fulfillment Bid (Normal)
    - Product Name - Bok Choy
    - Desired Amount - 50
  - Product Reservation Request
    - Product Name - Lychee
    - Price - \$3.00 per pound
    - Current Stock - 25
    - Images - Two 40 MB images showing the product
    - Description - Freshly grown, non-GMO lychee
- Expected Outputs:
  - A view that shows all of the information regarding the request.

- Product Request Fulfillment Bid (Normal)
  - Product Name - Bok Choy
  - Desired Amount - 50
- Product Reservation Request
  - Product Name - Lychee
  - Price - \$3.00 per pound
  - Current Stock - 25
  - Images - Two 40 MB images showing the product
  - Description - Freshly grown, non-GMO lychee
- Process:
  - The user selects the product request
  - The user views the product reservation request and sees all of the information regarding the request
- Testing Steps
  - Go to the page where the all the reservations and requests are saved
  - Check that the user is able to click on the reservation/request and view the relevant information

### 7.7.2 Edit Product Reservations and Request Fulfillment Bids

#### *Verification*

This functionality is verified when logged in users are able to edit or cancel the reservation and requests

Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must have a product reservation or request fulfillment bid
- User must be on the Transactions view

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users are able to edit the reservation and request
  - The user selects the product reservation or request fulfillment bid that they want to modify. The user inputs a valid amount of stock that they want to reserve. The system changes the initial reserved amount with the new amount. A message is displayed stating "The reserved amount of stock has successfully been changed." The user is redirected to the Transactions view.
- Invalid cases:
  - Users are unable to edit the reservation and request



- The user inputs an invalid amount of stock. A message is displayed stating “Invalid data has been inputted.” The user is redirected back to the Transactions view.
- The user inputs a valid amount of stock and the system considers it invalid. Or the user inputs an invalid amount of stock and the system considers it valid.
- Upon successful modification, the system does not change the initial reserved amount to the new amount.

### *Type of Test to be Conducted*

System Testing, Interface Testing

### *Priority: (High) 1 - 10 (Low)*

2 - This requirement has a “2” priority because it is the requirement that depends on the previously stated requirement. This requirement also has dependencies (7.7.1).

### *Effort Estimate:*

3 Hours

### *Test Execution Steps:*

- Inputs:
  - A product reservation request or request fulfillment bid that has been accepted
  - Product Request Fulfillment Bid (Normal)
    - Product Name - Bok Choy
    - Desired Amount - 50
  - Desired new Amount - 20
- Expected Outputs:
  - An updated product reservation request or request fulfillment that nows shows the newly inputted information
  - Product Request Fulfillment Bid (Normal)
    - Product Name - Bok Choy
    - Desired Amount - 20
- Process:
  - The user selects and views their product request
  - The user updates the information fields that they want to change
  - The user saves their edit
- Testing steps
  - 1) Select a reservation or request
  - 2) Click on the edit button
  - 3) Edit the post
    - a) if the post is within limits (ie word count/ image size)
      - i) The system will allow the user to save
    - b) if the post is not within the limit
      - i) The system will return an error

- ii) The system will not allow the user to save
- 4) Once saved, the post will be update
- 5) Ensure that the changes in the post is viewable

### 7.7.3 Open Direct Chat to Other Party

#### *Verification*

This functionality is verified when logged in users are able to chat to with other users

#### *Pre-conditions:*

- User must have an active authenticated session on the device
- User must be logged in
- User must have a product reservation or request fulfillment bid
- User must be on the Transactions view

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - There is a button that will allow users to open a chat with each other
  - The user selects the seller that they would like to open a direct chat with. A message is displayed stating "Successfully sent a chat request to the user." The user is redirected to the Transactions view.
- Invalid cases:
  - Users are unable to chat with each other
  - The user selects the seller that they would like to open a direct chat with. A message is displayed stating "Unsuccessfully sent a chat request to the user." The user is redirected to the Transactions view.
  - The system sends a direct chat request to a different user
  - The user is not redirected back to the Transactions view

#### *Type of Test to be Conducted*

Unit Testing, System Testing

#### *Priority: (High) 1 - 10 (Low)*

3 - This requirement has the lowest priority because it depends on requirement 7.7.1 but does not need to be tested earlier than requirement 7.7.2.

#### *Effort Estimate:*

2 Hours

#### *Test Execution Steps:*

- Inputs:

- A product reservation request or request fulfillment bid that has been accepted
- Product Request Fulfillment Bid (Normal)
  - Product Name - Bok Choy
  - Desired Amount - 50
- Expected Outputs:
  - Opens a chat between the buyer and seller for a product request
- Process:
  - The user selects and views their product request
  - The user selects to open the chat with the other party
- Testing steps
  - 1) Users have the ability to open a chat with another user
  - 2) When the user first tries to send a message to another user, the system will ask the sendee wants to chat with the sender
    - a) If accepts:
      - i) The system will make a new chat for the two users
    - b) If deny:
      - i) The system will inform the other user that the message can't get sent through
  - 3) Once the sendee accepts the invitation, they both should be able to send messages back and forth

## 7.8 Meeting Point Directions

### Timeframe

3/29/2022 - 4/2/2022

### Total Effort For All Test Cases

5 Hours

#### 7.8.1 View Street Directions to the Meeting Point Location

##### *Verification*

This functionality is verified when logged in users are able to view the meeting location once the transaction has been confirmed

##### Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must have a product reservation or request fulfillment bid
- User must be on the Transactions view

### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users will be able to see the meeting location address and be able to receive directions to the meeting point once the transaction has been confirmed
  - The user selects the production reservation or request fulfillment bid that is ready to be picked up. The system directs the user to a view that contains map directions to the meet-up location that has been set.
- Invalid cases:
  - The meeting location address is not showing up even after the transaction has been confirmed
  - Meeting location is visible before a transaction has been confirmed
  - The user selects the production reservation or request fulfillment bid that is ready to be picked up. The system does not direct the user to a view that contains map directions to the meet-up location that has been set.
  - The system shows incorrect map directions to the meet-up location

### *Type of Test to be Conducted*

Interface Testing, Security Testing

### *Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since the other requirements for this feature depend on it to be fully functional before they can be tested.

### *Effort Estimate:*

3 Hours

### *Test Execution Steps:*

- Inputs:
  - A product reservation request or request fulfillment bid that is ready to be picked up.
  - Product Request Fulfillment Bid (Normal)
    - Product Name - Bok Choy
    - Desired Amount - 50
    - Meet-up Location: 7881 Edinger Ave Ste 150, Huntington Beach, CA 92647
- Expected Outputs:
  - Access to a map showing street directions from a location of their choosing to the seller's meetup location.
- Process:
  - The user selects and views their product request that is ready to be picked up
  - The user inputs the location that they want to use as the origin point

- Inputted the desired location
  - Current location
  - The map is updated to show the street locations to the meet-up location from their desired origin point
- Testing Steps:
  - A transaction has been confirmed between two parties
    - 1) The system should automatically show the meeting location address once the transaction has been confirmed
    - 2) The user should be able to click on the location and the system will show the meeting location directions:
      - a) From the address the user provides
      - b) From the current address
        - i) User must have the location on and have given permission to the system to use it
  - A transaction has not been confirmed between two parties
    - 1) The system should not show the meeting location

### 7.8.2 Input Origin Location

#### *Verification*

This functionality is verified when logged in users have a designated spot to either choose their current location or an address

#### Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must have a product reservation or request fulfillment bid
- User must be on the Meeting Point Directions view

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - There is a designated spot for the user's current location or an address to be inputted
  - If there is an error in the inputted address then the system will show an error
  - The user inputs a valid location or they allow the application to use their current location. The system directs the user to a map view containing the map directions to the meet-up location.
- Invalid cases:
  - There is not a designated spot for the user to add a location
  - The user attempts to input an invalid location and a message will be displayed stating "Invalid location has been inputted." The user is directed back to the Miles Radius Search view.

- The user inputs a valid location and the system considers it to be invalid. The user inputs an invalid location and the system considers it to be valid.

### *Type of Test to be Conducted*

System Testing, Interface Testing

### *Priority: (High) 1 - 10 (Low)*

2 - This requirement has the lowest priority since it does not depend on the previous requirement (7.8.1).

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - The location that the user wants to use as the origin location:
    - Desired inputted location to be used as the origin location
    - The current location of the user
  - Desired Inputted Location
    - 7881 Edinger Ave Ste 150, Huntington Beach, CA 92647
  - User's Current Location
    - 1250 Bellflower Blvd, Long Beach, CA 90840
  - Invalid Inputted Location
    - Abc 123
- Expected Outputs:
  - The origin location is updated to be the desired inputted location or the user's current location.
  -
- Process:
  - The user inputs the location that they want to use as their origin location
- Testing Steps
  - User input their own location
    - Address is correct
      - 1) Input an correct location into the search bar
      - 2) The system should give directions to the meeting location
    - Address is invalid
      - 1) The system should output and error message stating that the address is invalid
  - User uses their current location
    - Given the system permission
      - 1) The system will retrieve the user's location
      - 2) The system will return directions to the meeting location
    - Haven't given the system permission

- 1) The system will inform the user that it doesn't have permission to retrieve the location
- Turned off location service
- 1) The system will inform the user that it is unable to retrieve the user's location

## 7.9 Following Sellers

### Timeframe

4/4/2022 - 4/9/2022

### Total Effort For All Test Cases

6 Hours

#### 7.9.1 Follow Sellers

##### *Verification*

This functionality is verified when logged in users are able able to follow sellers

Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must not be following more than 300 sellers
- User must be on the view of a Seller's Profile

##### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The user is able to follow a seller
  - The user selects the other user that they want to follow and is directed to the view of their profile. The user then confirms that they want to follow the other user. A message is displayed stating "You have successfully followed the user." The user is directed back to the followed user's profile.
- Invalid cases:
  - The user is unable to follow the seller
  - The user selects the other user that they want to follow and is directed to the view of their profile. The user then confirms that they want to follow the other user. A message is displayed stating "You have unsuccessfully followed the user." The user is directed back to the followed user's profile.
  - The user successfully followed the user, but there were directed to an incorrect view

- The user is currently following 300 users and the system allows them to follow more than the limit
- The user is currently following less than 300 users and the system is preventing them from following a user

### *Type of Test to be Conducted*

#### System Testing

### *Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since the other requirements for this feature depend on it to be fully functional before they can be tested.

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - The user selects to follow a seller
  - User:
    - John Doe
  - Following List:
    - Sam Smith, Robert Lee
  - Follow:
    - Christopher Johnson
- Expected Outputs:
  - The seller is added to the user's following list.
  - User:
    - John Doe
  - Following List:
    - Sam Smith, Robert Lee, Christopher Johnson
- Process:
  - The user selects to follow a seller
- Testing step
  - 1) Follower a seller
  - 2) Check that notifications will show up whenever the seller updates their profile
  - 3) Check that the seller's name is in the following page

## 7.9.2 Unfollow Sellers

### *Verification*

This functionality is verified when logged in users are able to unfollow sellers



**Pre-conditions:**

- User must have an active authenticated session on the device
- User must be logged in
- User must at least be following 1 seller
- User must be on the view of their Following List

**Validation**

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The user is able to successfully unfollow the seller
  - The user selects the other user that they want to unfollow and confirms that they want to unfollow the other user. A message is displayed stating “You have successfully unfollowed the user.” The user is directed back to their following list view.
- Invalid cases:
  - The system prevents the user from unfollowing sellers
  - The user selects the other user that they want to unfollow and confirms that they want to unfollow the other user. A message is displayed stating “You have unsuccessfully unfollowed the user.” The user is directed back to their following list view.
  - The user successfully unfollowed the user, but there were directed to an incorrect view

**Type of Test to be Conducted**

System Testing, Interface Testing

**Priority: (High) 1 - 10 (Low)**

2 - This requirement has a priority of “2” since it depends on requirement 7.9.2 to be completed first and the next requirement does not depend on this one.

**Effort Estimate:**

2 Hours

**Test Execution Steps:**

- Inputs:
  - The user selects to unfollow a seller
  - User:
    - John Doe
  - Following List:
    - Sam Smith, Robert Lee
  - Unfollow:
    - Christopher Johnson
- Expected Outputs:

- The seller is removed from the user's following list.
  - User:
    - John Doe
  - Following List:
    - Sam Smith, Robert Lee
- Process:
  - The user selects the seller that they want to unfollow
  - The user saves the edit
- Testing Step
  - 1) Unfollow a user
  - 2) Check that the notification stops for that specific user
  - 3) Check that the user is removed for the following list

### 7.9.3 View Following List

#### *Verification*

This functionality is verified when logged in users are able to view a list of people they are following

Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must be on the view of their profile

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The user is able to see all of the people they are currently following
  - When the user selects the seller in the list, it brings them to the seller's profile
  - The user selects to view all of the users that they are currently following. The system redirects the user to the view containing all of the users that they are following.
- Invalid cases:
  - The user can't see any or only a few of the people they are following
  - When the user selects the seller in the list, it doesn't bring the user to the seller's profile
  - The user selects to view all of the users that they are following. The system does not direct the user to the view containing all of the users that they are following.
  - The user is directed to the view but it does not contain accurate information regarding their following list

#### *Type of Test to be Conducted*

Interface Testing

*Priority: (High) 1 - 10 (Low)*

3 - This requirement has the lowest priority since it only depends on requirement 7.9.1 to be completed first and it does not have to be tested earlier than requirement 7.9.2.

*Effort Estimate:*

2 Hours

*Test Execution Steps:*

- Inputs:
  - A user's following list
  - User:
    - John Doe
  - Following List:
    - Sam Smith, Robert Lee, Christopher Johnson
- Expected Outputs:
  - A view that shows all of the sellers the user is following.
  - User:
    - John Doe
  - Following List:
    - Sam Smith, Robert Lee, Christopher Johnson
- Process:
  - The user selects to view their following list
- Testing Step
  - 1) Select the followings page
  - 2) The page should show all the sellers that the user is following

## 7.10 Rating and Review

Timeframe

4/12/2022 - 4/16/2022

Total Effort For All Test Cases

7 Hours

### 7.10.1 Review and Rating From a 1-5 Star Scale

*Verification*

This functionality is verified when logged in users are able to rate each other as long as there is a least one transaction between them

Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must have completed a product reservation or request fulfillment bid
- User must be on the view of the other user's profile or the product that was received

### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The users are able to rate and review each other once at least one transaction has been completed
  - The user is able to edit their review and rating
  - The user selects the product or user that they want to make a rating and review on. The user selects their rating from a 1-5 star scale. The user inputs valid information into the required fields for the review. The user confirms to post their rating and review. A message is displayed stating "Successfully posted the rating and review." The user is directed to the view containing the rating and review that they have made.
- Invalid cases:
  - The user is able to rate and review a user without completing any transaction with each other
  - The user is unable to rate and review once a transaction has been done
  - The user is unable to edit their review and rating
  - The user selects the product or user that they want to make a rating and review on. The user selects their rating from a 1-5 star scale. The user inputs valid information into the required fields for the review. The user confirms to post their rating and review. A message is displayed stating "Unsuccessfully posted the rating and review." The user is directed to the view containing the rating and review that they have made.
  - The user successfully posts their rating and review but they are not directed to the rating and review they made

### *Type of Test to be Conducted*

Interface Testing, System Testing

### *Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since the other requirements for this feature depend on it to be fully functional before they can be tested.

### *Effort Estimate:*

5 Hours

### *Test Execution Steps:*

- Inputs:

- There is at least one transaction between both parties
- The user is able to post a review containing UTF-8 characters and 40 MB images
- Review Containing UTF-8 Characters
  - Freshly grown daikon radish.
- Review Not Containing UTF-8 Characters
  -
- Uploading 40 MB images
  -
- Uploading Non-40 MB Images
  -
- Expected Outputs:
  - The rating and review have been posted on the product posting or user's profile.
- Process:
  - The user selects to rate and review their experience
  - The user selects the rating that they want to give from the 1-5 star scale
  - The user writes their review containing UTF-8 characters and up to 4 40 MB images
  - The user posts their rating and review
- Testing Step
  - After a transaction has been completed
    - 1) User should be able to write a review and rate another user
    - 2) Post the review
    - 3) Check if the review is viewable to the public
  - Before a transaction has been completed
    - 1) The system should not allow this to happen

### 7.10.2 View Posted Ratings and Reviews

#### *Verification*

This functionality is verified when logged in users are able view ratings and review on the users profile

#### Pre-conditions:

- User must have an active authenticated session on the device
- User must be on the view of the other user's profile or the product that was received

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The reviews and rating is viewable to the public
  - The user selects to view all of the ratings and reviews for a product or user. The system redirects the user to the view containing all of the ratings and reviews for the selected product or user.

- Invalid cases:
  - One or more review and rating is hidden
  - The user selects to view all of the ratings and reviews for a product or user. The system does not direct the user to the view containing all of the ratings and reviews for the selected product or user.
  - The user is directed to the view but it does not contain accurate information regarding the ratings and reviews for the selected user or product

### *Type of Test to be Conducted*

#### Interface Testing

### *Priority: (High) 1 - 10 (Low)*

2 - This requirement has the lowest priority since it depends on requirement 7.10.1 to be completed first before this requirement can be fully tested.

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - The ratings and reviews of a user and their product
  - Ratings and Review for a User:
    - John Doe
      - 1-5 Stars: 4
      - Review: He gave top quality products that were freshly picked the day I picked it up.
  - Ratings and Review for a Product:
    - Lychee
      - 1-5 Stars: 5
      - Review: Juicy lychee that you will not be able to find anywhere else!
- Expected Outputs:
  - A view that shows all of the ratings and reviews of a user and their posted product.
- Process:
  - The user selects to view the user or product
- Testing Step
  - 1) Once a test review has been written, check if the review is viewable from another account

## 7.11 User Block Feature

### Timeframe

4/11/2022 - 4/16/2022

### Total Effort For All Test Cases

6 Hours

#### 7.11.1 Block Users

##### *Verification*

This functionality is verified when logged in users are able to block each other

Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must have no more than 1000 currently blocked users
- User must be on the view of the other user's profile or on their profile

##### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The user is able to block any other user and it will be added to the list
  - Once blocked they are unable to interact with each other until the user the initiated the block unblocks the other user
  - The user selects the other user that they want to block and confirms that they want to block the other user. A message is displayed stating "You have successfully blocked the user." The user is directed back to their profile page.
- Invalid cases:
  - The users are still able to interact with each other once blocked
  - One user is unable to block a user
  - The user selects the other user that they want to block and confirms that they want to block the other user. A message is displayed stating "You have unsuccessfully blocked the user." The user is directed back to their profile.
  - The user successfully blocked the user, but there were directed to an incorrect view
  - The user currently has 1000 blocked users in their blocked list and the system allows them to block more users
  - The user currently has less than 1000 blocked users in their blocks list and the system is preventing them from blocking a user

*Type of Test to be Conducted*

System Testing, Interface Testing

*Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since another requirement (7.11.3) for this feature depends on it to be fully functional before it can be tested.

*Effort Estimate:*

2 Hours

*Test Execution Steps:*

- Inputs:
  - The user selects a user that they want to block
  - User:
    - John Doe
  - Blocked List:
    - Sam Smith, Robert Lee
  - Block:
    - Christopher Johnson
- Expected Outputs:
  - The seller is added to the user's blocked list.
  - User:
    - John Doe
  - Blocked List:
    - Sam Smith, Robert Lee, Christopher Johnson
- Process:
  - The user selects to a user to block
- Testing Step
  - 1) Block a user
  - 2) Make sure that the blocked user is in the blocked list
  - 3) Ensure that both parties are unable to see each other's profile
  - 4) Ensure that both parties no longer receive notifications about the other party
  - 5) Ensure that both parties are not longer able to chat with each other

## 7.11.2 Unblock Users

*Verification*

This functionality is verified when logged in users are able to unblock a user

Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in



- User must have at least 1 blocked user
- User must be on the Blocked Users view

### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The user is able to go and find a user from their blocked list and unblock them
  - The user selects the other user that they want to unblock and confirms that they want to unblock the other user. A message is displayed stating “You have successfully unblocked the user.” The user is directed back to their profile page.
- Invalid cases:
  - The user is unable to unblock one or more user
  - The user selects the other user that they want to unblock and confirms that they want to unblock the other user. A message is displayed stating “You have unsuccessfully unblocked the user.” The user is directed back to their profile.
  - The user successfully unblocked the user, but there were directed to an incorrect view

### *Type of Test to be Conducted*

System Testing, Interface Testing

### *Priority: (High) 1 - 10 (Low)*

2 - This requirement has a priority of “2” because it does not depend on another requirement but has a dependency (7.11.3).

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - The user selects a user that they want to unblock from their blocked list
  - User:
    - John Doe
  - Blocked List:
    - Sam Smith, Robert Lee, Christopher Johnson
  - Unblock:
    - Christopher Johnson
- Expected Outputs:
  - The seller is removed from the user’s blocked list.
  - User:
    - John Doe
  - Blocked List:
    - Sam Smith, Robert Lee

- Process:
  - The user selects the seller that they want to unblock
  - The user saves the edit
- Testing Step
  - 1) Unblock a user from the blocked list
  - 2) Ensure that both party's profile are viewable to each other
  - 3) Ensure that both party are able to chat with each other

### 7.11.3 View Block List

#### *Verification*

This functionality is verified when logged in users are able view and access a list of users they have blocked

Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must be on the view of their profile

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The users are able to view a list of users the have blocked
  - The user selects to view all of the users that they are currently blocking. The system redirects the user to the view containing all of the users that they are blocking.
- Invalid cases:
  - The users are unable to view one or more users that they have blocked
  - The user selects to view all of the users that they are blocking. The system does not direct the user to the view containing all of the users that they are blocking.
  - The user is directed to the view but it does not contain accurate information regarding their block list

#### *Type of Test to be Conducted*

Interface Testing

#### *Priority: (High) 1 - 10 (Low)*

3 - This requirement has the lowest priority out of all the current function's requirements because it depends on the previous two requirements (7.11.1 and 7.11.2) to be functional.

#### *Effort Estimate:*

2 Hours

*Test Execution Steps:*

- Inputs:
  - A user's blocked list
  - User:
    - John Doe
  - Blocked List:
    - Sam Smith, Robert Lee, Christopher Johnson
- Expected Outputs:
  - A user is able to view all of the users that they have blocked.
- Process:
  - The user selects to view their blocked list
- Testing Step
  - 1) Go to the blocked list and ensure all the user that has been blocked shows up on the list

## 7.12 Chat Feature

### Timeframe

4/18/2022 - 4/23/2022

### Total Effort For All Test Cases

9 Hours

#### 7.12.1 Send a Chat Request

##### *Verification*

This functionality is verified when logged in users are able to send chat requests to any user

##### Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must be on the profile for the user they want to chat with

##### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users have the ability to send a chat request to another user
  - The user selects the user that they would like to open a direct chat with. A message is displayed stating "Successfully sent a chat request to the user." The user is redirected to the view of the selected user's profile.
- Invalid cases:

- Users are unable to send a chat request to another user
- The user selects the user that they would like to open a direct chat with. A message is displayed stating “Unsuccessfully sent a chat request to the user.” The user is redirected to the view of the selected user’s profile.
- The system sends a direct chat request to a different user
- The user is not redirected back to the view of the selected user’s profile

### *Type of Test to be Conducted*

#### System Testing

### *Priority: (High) 1 - 10 (Low)*

1 - This requirement has the highest priority since the other requirements for this feature depend on it to be fully functional before they can be tested.

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - The user that they want to open a chat with
  - User to receive the chat request
    - John Doe
- Expected Outputs:
  - A request has been sent to the user that they want to open a chat with
- Process:
  - The user selects another user that they want to chat with
  - The user selects to send a request to open a chat
- Testing Step
  - 1) Go to a user and send a chat request

## 7.12.2 Accept or Decline a Chat Request

### *Verification*

This functionality is verified when logged in users receive a chat request and is able to accept and decline it.

### Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must have received a chat request
- User must be on the view for the chat request notification

### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The user that is receiving the chat request has the ability to accept or decline it
  - Before users can chat with each other the user that is receiving the request must first accept the request
  - The user selects to accept the received chat request. A message is displayed saying “The chat request has been accepted.” The user is directed to the now opened chat with the other user.
  - The user selects to deny the received chat request. A message is displayed saying “The chat request has been denied.” The user is directed back to their chat request notification
- Invalid cases:
  - The user is unable to accept or decline a requisition
  - Users are able to chat with each other before the user receiving the request accepts it
  - The user selects to accept or deny the received chat request, and an incorrect message is displayed or no message has been displayed.
  - The user selects to accept the received chat request. The system incorrectly marks an accepted request denied and a denied request accepted.
  - The user successfully opens the chat with the other user, but they are not directed to the view of the opened chat

### *Type of Test to be Conducted*

System Testing

*Priority: (High) 1 - 10 (Low)*

2 - This requirement has a priority of “2” since it depends on requirement 7.12.1 to function.

*Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - A chat request that has been received
  - User to receive the chat request
    - John Doe
- Expected Outputs:
  - Notification sent to the other user stating whether their request has been accepted or denied
    - If it was accepted then the chat will be open
    - If it was declined then the chat will remain closed

- Process:
  - The user receives the chat request and views it
  - The user selects to accept or deny the request
- Testing Steps
  - 1) Once a chat request has been sent, the other user will get a notification and have the ability to accept or deny the request
    - a) If accept
      - i) The system will open a new chat with the two user
    - b) If deny
      - i) The system will send notification to the user that requested the chat that the other user denied the request

### 7.12.3 Communicate With Users Using the Chat

#### *Verification*

This functionality is verified when logged in users are able talk to each other using the chat feature

Pre-conditions:

- User must have an active authenticated session on the device
- User must be logged in
- User must have accepted the chat request of the other user
- User must be on the view of the Chat view

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users can send instant messages to each other
    - UTC-8 characters
    - 40 MB images
  - The user uploads a 40 MB image or less to be sent to the chat. The system saves the image and sends it to the other user through the chat.
  - The user types out a message that they want to send the user. The system sends the message to the other user through the chat.
- Invalid cases:
  - Users are unable to send messages to each other
    - UTC-8 characters
    - 40 MB images
  - The user uploads a 40 MB image or less to be sent to the chat. The system does not save the image and attempts to send it to the other user through the chat.
  - The user types out a message that they want to send the user. The system does not send the message to the other user through the chat.

*Type of Test to be Conducted*

Unit Testing, System Testing

*Priority: (High) 1 - 10 (Low)*

3 - This requirement has the lowest priority out of the other requirements for this feature because it depends on them (7.12.1 and 7.12.2) to function.

*Effort Estimate:*

5 Hours

*Test Execution Steps:*

- Inputs:
  - A message that contains or does not contain UTF-8 characters and 40 MB images
  - Testing for description
    - UTF-8 Message
      - Hello! My name is John!
    - Non UTF-8 Message
      -
  - Testing for uploading images
    - 40 MB image
    - Non 40 MB image
- Expected Outputs:
  - The typed out message or image is sent to the other user
- Process:
  - The user types out the message or uploads the image
  - The user selects to send the message or image
- Testing step
  - Send a message
    - 1) Both users should be able to send a message
    - 2) Check that the message gets received in a timely manner
  - Send a message that goes out of the limits
    - 1) Try to send a message that goes out of limits
    - 2) The system should not allow this and send an error message

## 8. Testing For Core Components Requirements

### 8.1 Security

#### 8.1.1 Authentication

##### Timeframe

2/1/2022 - 2/5/2022

##### *Verification*

This functionality is verified when users are required to sign up/sign in before being able to access any other feature of the web application. Using the user's id, the system will only bring up information that pertains to their account

##### Requirements:

- The user must provide a valid username and password when trying to authenticate with the system
- After 5 attempt within a 24 hour span for the same account, the account will be disabled
  - After the 24 hours the fail count resets to 0
  - The account is locked until a valid account recovery is performed by valid users or the admin. Upon successful account recover the fail count resets to zero
  - For each failed attempt, the system will record the the account that is trying to be access and the IP address of the computer trying to login
- System failure of the feature should not cause the system to go offline.

##### Pre-conditions:

- User must not be already logged on the current device
- User must be trying to log in

##### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users have the ability to sign up
  - Users that have an account has the ability to log in
  - Logged in users can access only their account
  - Everything that was save prior should show up the next time a user access their account
  - System locks up user account after 5 tries
- Invalid cases:
  - Users are unable to sign up for an account
  - Users are unable to log into their account



- Logged in users are not able to access their account
- Everything that was saved prior does not show up

### *Type of Test to be Conducted*

Unit Testing, System Testing

### *Priority: (High) 1 - 10 (Low)*

1 - This requirement is the highest priority because all the features depend on this one function.

### *Effort Estimate:*

2 Hours

### *Test Execution Steps:*

- Inputs:
  - All the required information to sign up
    - ie. Username, password, full name, email, etc
  - During log in, the user must have a username and password
- Expected Outputs:
  - When the user sign up, the information should be saved in the system
  - When the user sign up, the system should route them back to the login page
  - When the user logs in, the system should grab their username and have the relevant user information before starting the web application
- Process:
  - The user input the relevant information in the sign up page
  - The user inputs the relevant information in the login page
- Testing step
  - Signing up
    - 1) The user should be able to click on a button in the login page that reroutes them to the sign up page
    - 2) In the sign up page, there are required information the user must fill out
      - a) If the user leaves required input boxes black
        - i) The system will not add save their information and will return an error saying that all required information must be filled
      - b) If the user uses the same username
        - i) The system will not save the user's information and ask the user to choose a different username
      - c) If the user inputs an invalide email
        - i) The system will not save the user's information and ask the user to input a valid email
      - d) If the user inputs all their information correctly
        - i) The system will reroute them to the login page
  - Logging in

- 1) In the log in page, the user will be required to fill out the username and password
  - a) If the username and password is correct
    - i) The system will generate the user's account page and reroute the user to their home page
  - b) if the username and password is incorrect
    - i) The system will not allow the user to continue and inform the user that their username or password is incorrect
  - c) If the username and password has been inputted incorrectly 3 times
    - i) The account will be locked for a certain amount of time
- Username or password not correct during login
  - 1) Input incorrect login
  - 2) System should generate a message stating that the login is incorrect
  - 3) System should increase the failed attempt number by one
- Tries to login after 5 tries
  - 1) On the fifth try, if the user fails to log in, the system will generate a message stating that the account will be locked for 24 hours
  - 2) The system will lock the user account for 24 hours
    - a) Check that no one is able to login even with valid credentials
  - 3) The system will unlock the account after 24 hours and set the fail attempt number to 0

### 8.1.2 Authorization

#### Timeframe

2/9/2022 - 2/12/2022

#### *Verification*

This functionality is verified when authenticated users are allowed access only to resources that their accounts have permission to access.

#### Pre-conditions:

- The user must be authenticated and logged in to an active account.

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases.

- Valid Cases:
  - User attempts to access a protected functionality within authorization scope. Access is granted to perform functionality.
  - User attempts to access protected data within authorization scope. Access is granted to perform read operations.

- User attempts to modify protected data within authorization scope. Access is granted to perform write operations.
- User attempts to access protected views within authorization scope. Access is granted to the view. User is automatically navigated to view.
- User attempts to access a protected functionality outside of authorization scope. Access is denied and a system message displays “Unauthorized access”.
- User attempts to access protected data outside of authorization scope. Access is denied and a system message displays “Unauthorized access to data”.
- User attempts to modify protected data outside of authorization scope. Access is denied and a system message displays “Unauthorized access to data”.
- User attempts to access protected views outside of authorization scope. Access is denied and a system message displays “Unauthorized access to view”.
- User attempts to access protected views within authorization scope, but contains protected data that is not within read scope. Access is granted to the view. Upon completion of automatic navigation to view, a system message displays “Unauthorized access to data” with protected data not visible within the view.
- User attempts to access protected views within authorization scope, but contains protected data that is not within write scope. Access is granted to the view. Upon completion of automatic navigation to view, protected data is visible within the view. Attempts to modify the data will result in a system message that displays “Unauthorized access to data”.
- Invalid Cases:
  - Unauthorized access is not recorded by system when authorization fails. A system log of failure is attempted.
  - Authorized access is attempted, and the user is denied access to the protected resource.
  - Authorized access is attempted, and a system message declaring unauthorized access is provided.
  - Unauthorized access is attempted, and the user is given access to the protected resource.
  - User accesses protected view within authorization scope, but contains protected data not within read scope. Access is granted to the view, and the protected data outside of read scope is visible.
  - User accesses protected view within authorization scope, but contains protected data not within write scope. Access is granted to the view, and the protected data outside of write scope can be modified by the user.

### *Type of Test to Be Conducted*

Unit Testing, System Testing

### *Priority*

2 - This requirement is of high priority because it is required for use with several other system functions, but it requires the Authentication functionality to be complete for proper function.

*Effort Estimate*

6 Hours

## Test Execution Steps

- Inputs
  - Two sets of protected data, a protected view containing the protected data, and a protected functionality.
  - A user account "User 1", with all of the listed protected resources within authorization scope.
  - A user account "User 2", with none of the protected resources within authorization scope.
  - A user account "User 3", with all of the listed protected resources except write permissions for one set of data within authorization scope.
  - A user account "User 4", with all of the listed protected resources except read permissions for one set of data within authorization scope.
- Expected outputs
  - User 1 has access to all protected resources.
  - User 2 has access to none of the protected resources.
  - User 3 has permissions to all of the protected resources except for write permissions for the data they do not have write permissions for.
  - User 4 has permissions to all of the protected resources except for read permissions for the data they do not have read permissions for.
- Process
  - User 1 attempts to access the view, read and modify data from both sets, and access the functionality.
    - All attempts are successful.
  - User 2 attempts to access the view, read and modify data from both sets, and access the functionality.
    - None of the attempts are successful. The correct system message is sent after each failed attempt.
  - User 3 attempts to access the view, read and modify data from both sets, and access the functionality.
    - All of the attempts except modifying the data they do not have permission to are successful. The correct system message is sent after the failed modification attempt.
  - User 4 attempts to access the view, read and modify data from both sets, and access the functionality.
    - All of the attempts except viewing the data they do not have permission to are successful. The correct system message is sent after the failed viewing attempt.
- Testing step

- User 1
  - 1) User 1 attempts to access the view.
    - The user should be automatically navigated to the view and all data from both sets should be viewable.
  - 2) User 1 attempts to modify data from one set.
    - The data should be successfully modified.
  - 3) User 1 attempts to modify data from the other set.
    - The data should be successfully modified.
  - 4) User 1 attempts to access the functionality.
    - The functionality should be successfully invoked.
- User 2
  - 1) User 2 attempts to access the view.
    - The user should not be automatically navigated to the view.
    - A system message "Unauthorized access to view" should be displayed.
  - 2) User 2 attempts to read data from one set.
    - A system message "Unauthorized access to data" should be displayed.
  - 3) User 2 attempts to read data from the other set.
    - A system message "Unauthorized access to data" should be displayed.
  - 4) User 2 attempts to modify data from one set.
    - A system message "Unauthorized access to data" should be displayed.
  - 5) User 2 attempts to modify data from the other set.
    - A system message "Unauthorized access to data" should be displayed.
  - 6) User 2 attempts access to the functionality.
    - A system message "Unauthorized access" should be displayed.
- User 3
  - 1) User 3 attempts to access the view.
    - The user should be automatically navigated to the view and all data from both sets should be viewable.
  - 2) User 3 attempts to modify data from the set they have permission to.
    - The data should be successfully modified.
  - 3) User 3 attempts to modify data from the set they do not have permission to.
    - A system message "Unauthorized access to data" should be displayed.
  - 4) User 4 attempts to access the functionality.
    - The functionality should be successfully invoked.
- User 4
  - 1) User 4 attempts to access the view.

- The user should be automatically navigated to the view and all data from the set within read authorization scope should be viewable.
- 2) User 4 attempts to modify data from one set.
  - The data should be successfully modified.
- 3) User 4 attempts to modify data from the other set.
  - The data should be successfully modified.
- 4) User 4 attempts to access the functionality.
  - The functionality should be successfully invoked.

### 8.1.3 Logout

#### Timeframe

2/9/2022 - 2/12/2022

#### *Verification*

This functionality is verified when users are attempting to log out of their account

Pre-conditions:

- The user must be currently logged in before being able to access the logout feature
- User must be able to access the logout button

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Logged in users are able to log out of their account successfully within 5 seconds of the action. The system should reroute the user to the login page A message should pop up informing the user has successfully logged out of their account.
- Invalid cases:
  - Logged in users are not able to log out of their account successfully
  - When users log out of their account their data isn't saved properly
  - Users are not rerouted back to the login page after logging out
  - The process takes longer than 5 seconds

#### *Type of Test to be Conducted*

Unit Testing, System Testing

#### *Priority: (High) 1 - 10 (Low)*

1 - This requirement is the highest priority because this is what allows individual users to access only their information in a given time.

*Effort Estimate:*

2 Hours

*Test Execution Steps:*

- Inputs:
  - The user must be logged into an account
- Expected Outputs:
  - The user will be rerouted to the login page.
- Process:
  - The login user tries to logout of their account
- Testing step
  - Logging out successfully
    - 1) The user presses the logout button
    - 2) The system will save all the information pertaining to their account
    - 3) The system brings the user back to the login page
  - Logging out unsuccessfully (doesn't reroute to the login page)
    - 1) The user presses the logout button
    - 2) The system save all the relevant information
    - 3) The system does not reroute the user back to the login page
  - Logging out unsuccessfully (doesn't save the users changes)
    - 1) The user presses the logout button
    - 2) The system is unable to save the changes
    - 3) The system should send out an error stating that the changes are not being stored

## 8.2 User Administration

### 8.2.1 Account Creation (Registration)

Timeframe

1/24/2022 - 1/28/2022

*Verification*

This functionality is verified when users who do not currently have an active authenticated session successfully register for an account.

Pre-conditions:

- User must not have an active authenticated session
- User must be on account creation view

### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - User registers with a valid email and valid passphrase. A system message displays “Email confirmation pending”. The user receives a confirmation email within 15 seconds upon invocation of the system message. The user completes email confirmation within 24 hours. The user is notified of their username. A system message displays “Account created successfully”
- Invalid Cases:
  - User registers with an invalid email or invalid passphrase. A system message displays “Account creation error. Retry again or contact system administrator” or no system message.
  - User registers with a valid email and valid passphrase. A system message displays “Email confirmation pending”. The user does not receive a confirmation email at all or an email within 15 seconds upon invocation of the system message.
  - User registers with a valid email and valid passphrase. A system message displays “Email confirmation pending”. The user receives an email confirmation within 15 seconds upon invocation of the system message. The user does not complete email confirmation within 24 hours.
  - User registers with a valid email and valid passphrase. A system message displays “Email confirmation pending”. The user receives an email confirmation within 15 seconds upon invocation of the system message. The user completes email confirmation within 24 hours. User is not notified of username
  - User registers with a valid email and valid passphrase. A system message displays “Email confirmation pending”. The user receives an email confirmation within 15 seconds upon invocation of the system message. The user completes email confirmation within 24 hours. The user is notified of their username. A system message does not display
  - User registers with a valid email and valid passphrase. A system message displays “Email confirmation pending”. The user receives an email confirmation within 15 seconds upon invocation of the system message. The user completes email confirmation within 24 hours. The user is notified of their username. A system message displays “Account created successfully”, but account is not created and stored in the data store

### *Type of Test to be Conducted*

Unit Testing, System Testing, Interface Testing

### *Priority: (High) 1 - 10 (Low)*

1 - This requirement will have the highest priority out of the other requirements for this feature because the other features depend on account creation to work and accounts to exist within our system.



*Effort Estimate:*

6 Hours

*Test Execution Steps:*

- Inputs:
  - Valid Email Address:
    - [abc@gmail.com](mailto:abc@gmail.com)
  - Invalid Email Address:
    - test@notreal.com
  - Secret Passphrase
    - AaBbCc123!
  - Invalid Secret Passphrase
    - <Hello>
- Expected Outputs:
  - A user will be created using the email address of [abc@gmail.com](mailto:abc@gmail.com) and a secret passphrase of AaBbCc123!
  - A user will not be created using the email address of [test@notreal.com](mailto:test@notreal.com) and a secret passphrase of <Hello>
- Process:
  - User attempts to create an account using the valid email address and valid secret passphrase
    - Account creation will be successful
  - User attempts to create an account using the invalid email address and invalid secret passphrase
    - Account creation will be unsuccessful
- Testing step
  - Success Case
    - User inputs a valid email address
    - User inputs a valid secret passphrase
    - User attempts to create the account
  - Failure Case
    - User inputs an invalid email address
    - User inputs an invalid secret passphrase
    - User attempts to create the account

### 8.2.2 Account Deletion

#### Timeframe

1/26/2022 - 1/29/2022

#### *Verification*

This functionality is verified when users are able to delete their account from the system

**Pre-conditions:**

- The user must currently have an account
- User must be logged into their account at the time of attempting to delete the account
- User must have permission to delete the account

**Validation**

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Users are able to delete the account. Upon completing the action, the system should send a message stating that the account has been successfully deleted. Once the user has deleted the account the account, the system should bring the user back to the login page
- Invalid cases:
  - User is unable to delete the account
  - The user is able to delete the account and receives a success message but the account information stays in the system
  - The user data isn't deleted in the system
  - System does not show complete message
  - User is not able to acknowledge the system message
  - The user does not get redirected to the login page

**Type of Test to be Conducted**

Unit Testing, System Testing

**Priority: (High) 1 - 10 (Low)**

1 - This requirement is the highest priority because this is what allows individual users to access only their information in a given time.

**Effort Estimate:**

4 Hours

**Test Execution Steps:**

- Inputs:
  - The user must be logged into an account
- Expected Outputs:
  - The user will be rerouted to the login page. The account and the account data should be deleted from the system
- Process:
  - The login user tries to delete their account
- Testing step
  - Deleting the account

- 1) Click the delete account button
- 2) The system should ask the user if they are sure they want to delete the account
  - a) If the user chooses no
    - i) the system gets rid of the message and doesn't do anything
  - b) If the user chooses yes
    - i) the system should start deleting the system account and the data pertaining to the account
    - ii) Once the system successfully deletes the account, the system should inform that the user that the account has been successfully deleted
    - iii) The system should reroute the user to the login page

### 8.2.3 User Management

#### Timeframe

2/2/2022 - 2/5/2022

#### *Verification*

This functionality is verified when the user is able to perform any single or Bulk UM operation within a specific amount of time with message display.

#### *Pre-conditions*

- User must have an active authenticated session
- User must be on user management view
- User must be a system administrator

#### *Validations*

This functionality is validated when testing for the following valid and invalid cases

#### *Valid cases:*

1. User is able to perform any single UM operation within 5 seconds upon invocation. A system message displays "UM operation was successful"
2. The user is able to perform less than 1K UM operations in bulk within 60 seconds. A system message displays "Bulk UM operation was successful"
3. The user is able to perform 1K UM operations in bulk within 60 seconds. A system message displays "Bulk UM operation was successful"

#### *Invalid cases:*

1. Single UM operation takes longer than 5 seconds
2. Bulk UM operations takes longer than 30 seconds

3. Single UM operation completes within 5 seconds, but no system message is shown or inaccurate system message is shown
4. Bulk UM operations completes within 30 seconds, but no system message is shown or inaccurate system message is shown
5. Single UM operation completes within 5 seconds, with system message "UM operation was successful" shown, but latest data is not written to data store
6. Bulk UM operations completes within 60 seconds, with system message "Bulk U operation was successful" shown, but latest data is not written to data store
7. 1K Bulk UM operations completes takes longer than 60 seconds
8. 1K Bulk UM operations completes within 60s seconds, but no system message is shown or inaccurate system message is shown
9. 1K Bulk UM operations completes within 60 seconds, with system message "Bulk UM operation was successful" shown, but latest data is not written to data store

#### *Type of Test to be Conducted*

Unit Testing, System testing

#### *Priority: (High) 1 - 10 (Low)*

3 - It's related to the main function of the application.

#### *Effort Estimate:*

4h

#### *Test Execution Steps:*

- Inputs: The users can create, update and delete their account.
  - Single:
    - Create - Sign up with information needed
      - Username, password, email etc.
    - Update - change information or preference settings
      - Username, password, email etc.
    - Delete - delete user account and data
      - Select delete account
  - Bulk:
    - Update - change information or preference settings
      - block/unblock a group of people
      - follow/unfollow a group of people
- Expected Outputs: Operation should be completed within 5 seconds.
  - Single:
    - Create
      - If all data is valid, data will be saved to the system.

- Update
    - The valid and new data will be saved to the system and replace the old data.
  - Delete
    - Account deleted
- Bulk: Operation should be completed within 15 seconds
- Update - change information or preference settings
    - block/unblock a group of people
    - follow/unfollow a group of people
  - Process:
    - Users enter required information.
    - Users change their information and preference.
    - Users delete the account.
  - Testing steps:
    - Users can create account
      - check if operation time is within 5 seconds
    - Users can make any change in information or preference settings.
      - Block one user
        - Check if operation time is within 5 seconds
      - Block 100 users
        - Check if operation time is within 15 seconds
    - Users can delete account
      - check if operation time is within 5 seconds

## 8.3 System Observability

### 8.3.1 Usage Analysis Dashboard

#### Timeframe

4/18/2022 - 4/23/2022

#### *Verification*

This functionality is verified when the user is able to navigate to the view with all KPI data's auto update.

#### Pre-conditions

- The user must have an active authenticated session on the device on Usage Analysis Dashboard view as a system administrator.

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid cases:
  - User is able to navigate to the view within 15 seconds. All KPIs automatically refreshes data within 60 seconds.
- Invalid cases:
  - User is a system administrator, but unable to navigate to the page.
  - User is able to navigate to the page, but view does not load within 15 seconds upon navigation completion.
  - The user is able to navigate to the page, view loads within 15 seconds, but no KPI data is refreshed.
  - The user is able to navigate to the page, view loads within 15 seconds, but not all KPI data is refreshed.
  - The user is able to navigate to the page, view loads within 15 seconds, but all KPI refresh takes longer than 60 seconds.

#### *Type of Test to be Conducted*

Unit Testing, System testing

#### *Priority: (High) 1 - 10 (Low)*

6 - It's a visual form to show the performance of the application or it's status of deliverables, but it will not affect the main function of the application.

#### *Effort Estimate:*

4h

#### *Test Execution Steps:*

- Inputs:
  - User's behavior within the system.
- Expected Outputs:
  - All data collected is use to create charts for Key Performance Indicators
- Processes:
  - Navigate to view of charts:
    - The top 5 most visited view of all time (bar chart)
    - The top 5 average duration per view of all time (bar chart)
    - The number of logins per day within the span of 3 months (trend chart)
    - The number of registrations per day within the span of 3 month (trend chart)
    - The number of post created per day within the span of 3 month(trend chart)
    - The number of success transaction per day within the span of 3 month(trend chart)
  - All KPIs automatically refreshes data

- Testing Steps:
  - Make sure the page can navigate to the view
    - Check the time to navigate the view is within 15 seconds or not
    - Check if all KPI data are refresh
    - Check the time to refresh KPI data is within 60 seconds or not

### 8.3.2 Logging

#### Timeframe

2/16/2022 - 2/19/2022

#### *Verification*

This functionality is verified when all attempted system actions are recorded to a persistent data store with the relevant information.

Pre-conditions:

- The persistent data store must be active.
- The persistent data store must be accessible to the system.
- The persistent data store must have storage capacity for log entry.

#### *Validation*

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - The system logs system success events
  - The system logs system failure events
  - The system logs user success events
  - The system logs user failure events
- Invalid cases:
  - The logging process took longer than 5 seconds to complete upon invocation
  - The logging process blocks a user from interacting with the system
  - The logging process completes within 5 seconds, but did not save to a persistent data store
  - The logging process completes within 5 seconds, but did not accurately save the event to the persistent data store (i.e. timestamp, log level, category, message, etc.)
  - Previously saved log entries are modifiable

#### *Type of Test to be Conducted*

Unit testing, system testing

*Priority: (High) 1 - 10 (Low)*

2 - The logging function depends on the existence of a user system to properly function, but should be completed early for integration with other core and app-specific features.

*Effort Estimate:*

4 Hours

*Test Execution Steps:*

- Inputs:
  - Two user IDs
  - An enumeration of log categories (View, Business, Server, Data, Data Store)
  - An enumeration of log levels (Info, Debug, Warning, Error)
- Expected Outputs:
  - Log entries on the persistent data store accurately reflecting the information input to the logging function.
- Process:
  - Randomly generate log entries using a random user ID, category, log level, and message.
- Testing step
  - 1) Once per second for 15 seconds, attempt to log an entry with the user ID, a random category from the enumeration, a random log level, and a randomly generated message String between 10-32 characters long. Save the inputs for each entry and the UTC timestamp at invocation.
  - 2) Request each entry from the persistent data store and compare each entry with its corresponding inputs that were saved during invocation.
    - The inputs should match with the entry that was requested from the data store.

### 8.3.3 Archiving

#### Timeframe

2/16/2022 - 2/19/2022

#### Total Effort

5 Hours

#### *Verification*

This functionality is verified when all log entries have been consolidated and compressed to preserve system resources..



**Pre-conditions:**

- Persistent data store must be active
- Persistent data store must be accessible by the system
- Archival destination location must have storage capacity

**Validation**

This functionality is validated when testing for the following valid and invalid cases

- Valid Cases:
  - Archival process executes at 00:00:00 AM (local time) on the 1st of the month. All log entries older than 30 days are consolidated, compressed and relocated to another location. All archived logs are removed from the system. The entire archival process completes within 60 seconds upon invocation.
- Invalid Cases:
  - Archival process did not start at 00:00:00 AM
  - Archival process started at 00:00:00 AM, but not local time
  - Archival process started at 00:00:00 AM (local time), but not on the 1st of the month
  - Archival process started at 00:00:00 AM (local time) on the 1st of the month, but did not archive any log entries even though there are logs older than 30 days
  - Archival process started at 00:00:00 AM (local time) on the 1st of the month, but did not archive all log entries older than 30 days
  - Archival process started at 00:00:00 AM (local time) on the 1st of the month. All log entries older than 30 days are not consolidated.
  - Archival process started at 00:00:00 AM (local time) on the 1st of the month. All log entries older than 30 days are consolidated, but are not compressed.
  - Archival process started at 00:00:00 AM (local time) on the 1st of the month. All log entries older than 30 days are consolidated and compressed, but are not relocated to another location
  - Archival process started at 00:00:00 AM (local time) on the 1st of the month. All log entries older than 30 days are consolidated, compressed and are relocated to another location, but archived logs are not removed from the system.
  - Archival process took longer than 60 seconds to complete upon invocation

**Type of Test to be Conducted**

Unit Testing, System Testing

**Priority: (High) 1 - 10 (Low)**

3 - The archiving function depends on the log entries from the logging function in order to properly function.

**Effort Estimate:**

5 Hours

*Test Execution Steps:*

- Inputs:
  - An enumeration of log entries that are located within our system
  - An enumeration of log categories (View, Business, Server, Data, Data Store)
  - An enumeration of log levels (Info, Debug, Warning, Error)
- Expected Outputs:
  - All log entries older than 30 days are consolidated, compressed and relocated to another location. All archived logs are removed from the system
- Process:
  - Randomly generate a list of log entries using a random user ID, category, log level, message, and age. Then get the enumeration of logs that are older than 30 days and consolidate, compress and relocate it to another location. All the logs that have been archived will be removed from the system. Compare the enumeration of log entries before archiving has taken place and after it has.
- Testing step
  - Once per second for 15 seconds, attempt to log an entry with the user ID, a random category from the enumeration, a random log level, a randomly generated message String between 10-32 characters long, and a randomly generated age. Save the inputs for each entry and the UTC timestamp at invocation.
  - Request each entry from the persistent datastore and get the log entries that are older than 30 days.
  - Once all log entries have been retrieved, consolidate, compress, and relocate it to another location.
  - Remove all of the log entries that have been archived from the persistent datastore.
  - Compare the log entries in the persistent datastore to see if all of the log entries that are 30 days and older have been removed. Then look for the archived logs in the location that it has been related to.
    - There should only be log entries that are less than 30 days old