***Gourmet Pitt Tour***

***User Requirements***

***Document***

*05 Jul 2015*

*Version 0.1*

**1. Introduction**

The objective of this User Requirement Document (URD) is intended to demonstrate clearly about SuperFly’s expectation of the Gourmet Pitt Tour Android App. Specific software requirement is given in the tables so that the design features of the App is well defined.

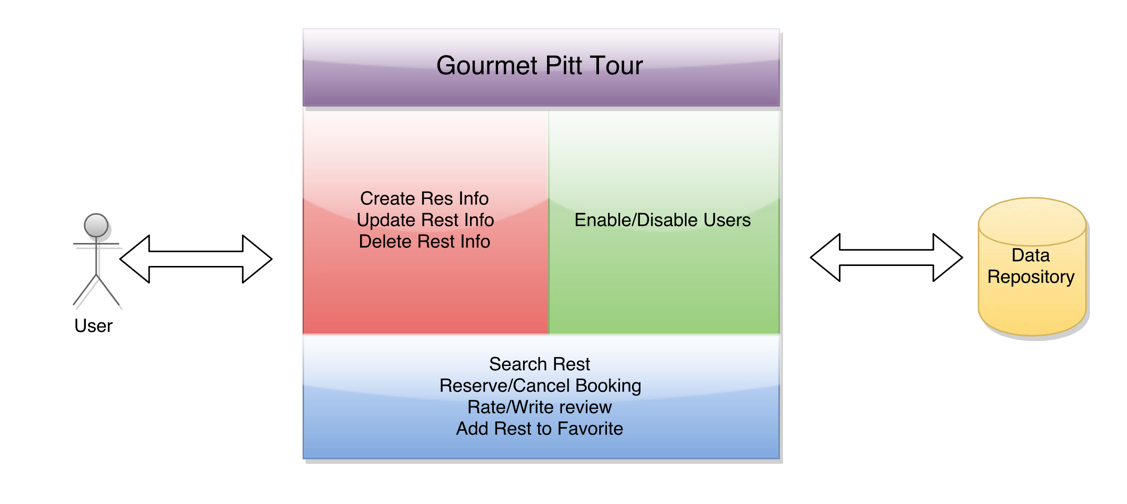
**2. Hardware Used**

Three hardwares will be used in our project - GPS, camera, music player. GPS will be used to locate use and find nearby restaurant on the map. When user go to a restaurant and want to write a review with photo, he can use camera to take photo. Music player is used to play background music.

**3. Document Scope**

This URD provides an overall description of the Gourmet Pitt Tour (GPT) Android App, which will be developed by team SuperFly. This URD provides detailed descriptions of all of the GPT’s intended functionality. This URD outlines the events experience of different kind of target users and specific software functional requirement of GPT.

**4. System Overview**



**5. General Requirements**

Team SuperFly is intended to develop the GPT App, which helps locals, and tourists to Pittsburgh find desired restaurants catering to their taste. Also, this App helps the restaurant owners to advertise their restaurants and delicious food.

**5.1 System in Context**

The GPT App would realize the following functions necessary for sharing and rating food and restaurant information between consumers and restaurant owners:

· Sharing function for Business to share food and restaurant information on this App.

· Searching and rating functions for Consumers to find their desired food and restaurant and rate their experience using this App.

· Security function for authentication and encryption.

**5.2 Constraints**

A. Different Main UI by different kinds of users.

B. Secure login screens, reverting to screen saving mode.

C. Ease of use of functions of the App.

**5.3 System-Wide Requirements**

**5.3.1 Actors**

The GPT App recognizes 3 types of users. The first is restaurant owners (Business), the entities that will first create their customized website on our App and upload necessary information. The second type of users is common customer (Customer) who can be either locals or tourists travelling to Pittsburgh. They can search nearby restaurants catering to their tastes or choose those recommended restaurants nearby. The last type is Administrator who will manage the account, approve/deny the application to create business websites and get feedback for other two types of users.

**5.3.2 Events**

GPT is a restaurant and food App with functions of searching, recommending, and rating the food in area of Pittsburgh. The most critical events are: 1) Create restaurant information; 2) Update restaurant information; 3) Delete restaurant information; 4) Search nearby restaurants; 5) Reserve/Cancel a restaurant booking; 6) Rate/Review a restaurant; 7) Add restaurants to favorites; 8) Enable/Disable users. Among the above events, the main actors of event 1-3 are restaurant owners (Business), main actors of events 4-7 are Customers and Administrator is the actor of the last event.

**6. Detailed Requirements**

**6.1** **Functional Requirements**

After login, a restaurant owner (Business) can choose to create a website about his restaurant. Basic information such as location, phone number and cuisine style as well as some pictures of recommended food can be uploaded for creating their customized website. When successfully creating the website, Business can choose to add new features to their website by updating it or remove some contents using the deleting function.

After login, a consumer (Customer) can choose to search the desired restaurants using our filter function, which will help find the best restaurants catering to their tastes. If Customer cannot decide which restaurant to pick, he can use our function, which will recommend a restaurant based on their previous dining experiences. After choosing a restaurant, Customer can reserve a seat at that restaurant and can cancel it if he changes his idea. After the dining, Customer can rate and write review about this restaurant or some specific dishes supplied by the restaurant. These rating and review will be a precious resource for future Customers.

After login, an Administrator can see all the reminding messages including the registration message, the application from Business to create the website or feedbacks from other users.

**6.2** **Detailed Requirements**

|  |  |  |
| --- | --- | --- |
| Use Case ID | Use Case Name | Priority |
| AUTH-USR | Authenticate User | High |
| CRE-RES | Create Restaurant Information | High |
| APPR-RES | Approve Restaurant Information | High |
| UPD-RES | Update Restaurant Information | High |
| DEL-RES | Delete Restaurant Information | High |
| NEAR-RES | Find Nearby restaurant | High |
| FILTER | Filter Restaurant | High |
| DETAIL | Restaurant Detail | High |
| RESERVE | Reserve a Restaurant | Medium |
| CANCEL-RES | Cancel a Reservation | Medium |
| ADD-FAV | Add to Favorite | Medium |
| REMV-FAV | Remove Favorite | Medium |
| SEND-RES | Send Reservation message | Medium |
| SEND-CAN | Send a Cancel Message | Medium |
| RATE-RES | Rate a Restaurant | High |
| ENABLE | Enable User | High |
| FB | Feedback | Medium |
| CHECK-MSG | Check Message | High |

**Table 1. AUTH-USR: Authenticate User**

|  |  |
| --- | --- |
| Use Case ID: AUTH-USR | Use Case Name: Authenticate User |
| Primary Actor(s): | All authorized users of Gourmet Pit Tour(GPT) |
| Secondary Actor(s): | N/A |
| Description: | Requests username and password and authenticates user into PGT system. |
| Preconditions: | User login display dialog is available and waiting for user input. |
| Normal Flow of Events: | 1. User enters user name and then tabs or clicks into password field and enters password 2. User clicks ok 3. New user, then go to User Register UI. |
| Postconditions: | After step 2, the main PGT UI displays. |
| Frequency of Use: | High |
| Alternative Flows: | User can click cancel within the login dialog display to cancel or clear a misspelled ID or password |
| Exceptions: | 1. User does not enter anything in the username field or password field, then a message “username/password field cannot be empty” will be prompted 2. User enters  wrong username or password, then a message “invalid username or password” will be prompted |
| Assumptions: | PGT is started, user has obtained a valid ID and password from the administrator. |
| Issues: | User never has no id or password. |
| Source: | Use case document |
| Includes: | TBD |
| Associated Requirements: | TBD |

**Table 2. CRE-USR: Create Restaurant Information**

|  |  |
| --- | --- |
| Use Case ID: CRE-RES | Use Case Name: Create Restaurant Information |
| Primary Actor(s): | Business(Restaurant owner) |
| Secondary Actor(s): | N/A |
| Description: | Restaurant owner can upload materials for creating their simple restaurant website on PGT |
| Preconditions: | Restaurant owner has successfully logged in PGT and selects “create my restaurant” after clicking on the “My Restaurant” in the Main UI. |
| Normal Flow of Events: | 1. The restaurant owner can choose to upload basic information about the restaurant and pictures about the restaurants/foord. 2. After uploading, click “OK” to upload.  3. Wait for the message from Administrator that the Developed has already built the website or the creating website action is denied by the Administrator. |
| Postconditions: | After step 2, return to the Main UI of Business. |
| Frequency of Use: | High |
| Alternative Flows: | User can click “cancel” to stop uploading materials. |
| Exceptions: | User upload invalid file format other than .txt and .jpg, then a message “incompatible file type” will be prompted |
| Assumptions: | PGT is started, Business has obtained a valid ID and password from the administrator. |
| Issues: | TBD |
| Source: | Use case document |
| Includes: | TBD |
| Associated Requirements: | TBD |

**Table 3. APPR-RES: Approve Restaurant Information**

|  |  |
| --- | --- |
| Use Case ID: APPR-RES | Use Case Name: Approve Restaurant Information |
| Primary Actor(s): | Administrator |
| Secondary Actor(s): | N/A |
| Description: | When the Business choose to create their restaurant website and upload required material, Administrator will get these information. After reviewing these information, Administrator can choose whether to approve the application for building the restaurant website and informs the Developed or deny the application |
| Preconditions: | 1.PGT is started, Administrator logins.  2.Bussiness has uploaded the material and Administrator gets the message. |
| Normal Flow of Events: | 1. Administrator review the application  2. If it is approved, send out a message to inform the Developer  3. If denied, send out a message to inform the Business |
| Postconditions: | After step 2 and 3, the Main Administrator UI  displays. |
| Frequency of Use: | High |
| Alternative Flows: | TBD |
| Exceptions: | TBD |
| Assumptions: | PGT is started, Administrator logins. |
| Issues: | TBD |
| Source: | Use case document |
| Includes: | TBD |
| Associated Requirements: | TBD |

**Table 4. UPD-RES: Update Restaurant Information**

|  |  |
| --- | --- |
| Use Case ID: UPD-RES | Use Case Name: Update Restaurant Information |
| Primary Actor(s): | Business(Restaurant owner) |
| Secondary Actor(s): | N/A |
| Description: | Restaurant owner can update information of their restaurant website on PGT |
| Preconditions: | 1. Restaurant owner has successfully logged in PGT and selects “edit my restaurant” after clicking on the “My Restaurant” in the Main UI.  2. Restaurant owner has successfully created that restaurant website before. |
| Normal Flow of Events: | 1. The restaurant owner can choose where they want to update and upload new materials for updating  2. After uploading, click “OK” to upload.  3. Wait for the message from Administrator that the Developed has already updated the website or the updating website action is denied by the Administrator. |
| Postconditions: | After step 2, return to the Main UI of Business. |
| Frequency of Use: | High |
| Alternative Flows: | User can click “cancel” to stop updating the website and the website will be the same as before |
| Exceptions: | User upload invalid file format other than .txt and .jpg, then a message “incompatible file type” will be prompted. |
| Assumptions: | PGT is started, Business has obtained a valid ID and password from the administrator, Business has created the website. |
| Issues: | TBD |
| Source: | Use case document |
| Includes: | TBD |
| Associated Requirements: | TBD |

**Table 5. DEL-RES: Delete Restaurant Information**

|  |  |
| --- | --- |
| Use Case ID: DEL-RES | Use Case Name: Delete  Restaurant Information |
| Primary Actor(s): | Business(Restaurant owner) |
| Secondary Actor(s): | N/A |
| Description: | Restaurant owner can delete information of their restaurant website on PGT |
| Preconditions: | 1. Restaurant owner has successfully logged in PGT and selects “delete my restaurant” after clicking on the “My Restaurant” in the Main UI.  2. Restaurant owner has successfully created that restaurant website before. |
| Normal Flow of Events: | 1. The restaurant owner can choose where they want to delete  2.click “OK” to confirm deletion.  3. Wait for the message from Administrator that the Developed has already deleted the website. |
| Postconditions: | After step 2, return to the Main UI of Business. |
| Frequency of Use: | High |
| Alternative Flows: | User can click “cancel” to stop deleting the website and the website will be the same as before |
| Exceptions: | The user wants to delete something that has not been created before and it will fail. |
| Assumptions: | PGT is started, Business has obtained a valid ID and password from the administrator, Business has created the website. |
| Issues: | TBD |
| Source: | Use case document |
| Includes: | TBD |
| Associated Requirements: | TBD |

**Table 6. NEAR-RES: Find Nearby Restaurant**

|  |  |
| --- | --- |
| Use Case ID: NEAR-RES | User Case Name: Find nearby restaurant |
| Primary Actor(s): | Customers |
| Secondary Actor(s): | N/A |
| Description: | Locate the user and show a list of restaurant within distance user specified |
| Preconditions: | Main UI is available and it's for customer to select “Find Nearby” |
| Normal Flow of Events: | 1. User select “Find Nearby” on main UI; 2. Screen inquires whether the user allows app to access his/her location 3. User selects “Yes” |
| Postconditions: | After step 3, a list of restaurant is displayed sorted by the distance |
| Frequency of Use: | High |
| Alternative Flows: | At step 3, user can click “No” and then return to the main UI |
| Exceptions: | TBD |
| Assumptions: | User has successfully logged in with a valid account;  User has an available GPS in the tablet |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | TBD |
| Associate Requirements | TBD |

**Table 7. FILTER: Filter Restaurant**

|  |  |
| --- | --- |
| Use Case ID: FILTER | User Case Name: Filter Restaurant |
| Primary Actor(s): | Customers |
| Secondary Actor(s): | N/A |
| Description: | Refine the search after a restaurant list is displayed |
| Preconditions: | A restaurant list is displayed , the filter list view menu is ready for customer to select |
| Normal Flow of Events: | 1. Customer enters the specification 2. The menu is filtered according to the specification |
| Postconditions: | After step 2, a list of filtered restaurants is displayed in a random order |
| Frequency of Use: | High |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | TBD |
| Associate Requirements | TBD |

**Table 8. DETAIL: Restaurant Detail**

|  |  |
| --- | --- |
| Use Case ID: DETAIL | User Case Name: Restaurant Detail |
| Primary Actor(s): | Customers |
| Secondary Actor(s): | N/A |
| Description: | Show information of a restaurant |
| Preconditions: | A restaurant list is displayed , a detail button waits for user to click |
| Normal Flow of Events: | 1. Customer clicks the “detail” button； 2. A page of restaurant detail is displayed(from top to bottom): The restaurant name, its average rating, and its foods uploaded by customers/business owners, customers’ review; 3. A “rating” button waits for customers to give rating and review for this restaurant; 4. A “reserve” button waits for customer to make a reservation for this restaurant. After clicking on this button, the user is led to another interface for him/her to specify the reservation time/date 5. An “add to favorite” button waits for customer to click |
| Postconditions: | After step 1, all the above details are shown |
| Frequency of Use: | High |
| Alternative Flows: | N/A |
| Exceptions: | N/A |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | TBD |
| Associate Requirements | TBD |

**Table 9. RESERVE: Reserve a restaurant**

|  |  |
| --- | --- |
| Use Case ID: RESERVE | User Case Name: Reserve a restaurant |
| Primary Actor(s): | Customers |
| Secondary Actor(s): | N/A |
| Description: | Reserve a seat |
| Preconditions: | A restaurant detail page is displayed, the “reserve” button waits for user to click |
| Normal Flow of Events: | 1. Customer select “reserve”; 2. A reservation page is displayed, then the user can enter his/her reservation time and date, the “confirm” button waits for the user to click |
| Postconditions: | After step 1, a reserve page waits for user to specify his reservation;  After step 2, the user clicking “confirm”, this page changes into another page with information of the reservation and a “cancel” button; a reserve message will be sent to the business owner’s UI |
| Frequency of Use: | Medium |
| Alternative Flows: | A customer/business owner clicks on the “delete”, next time after a user clicks on “reserve” on detail page, he/she will go to a normal reservation page |
| Exceptions: | In reservation page, if a customer does not make any specifications and clicks “confirm”, then the specifications will be set to default values |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | The reservation deal |
| Associate Requirements | TBD |

**Table 10. CANCEL-RES: Cancel a reservation**

|  |  |
| --- | --- |
| Use Case ID: CANCEL-RES | User Case Name: Cancel a reservation |
| Primary Actor(s): | Customers/Business owner |
| Secondary Actor(s): | N/A |
| Description: | Cancel a reservation |
| Preconditions: | A confirmation list view on the user’s personal information page showing reservation detail is displayed. At the bottom there is a “delete” button waiting for user to click in order to delete his/her reservation |
| Normal Flow of Events: | User clicks the “delete” button |
| Postconditions: | Next time after a user clicks on “reserve” on detail page, he/she will go to a normal reservation page. The deleted reservation will disappear correspondingly from both the customer and the business owner’s reservation list view which is placed in the personal information page |
| Frequency of Use: | Medium |
| Alternative Flows: | TBD |
| Exceptions: | The user will have to choose delete mode at first then choose the reservation to be deleted. If not, an exception message will be toasted |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | The reservation deal |
| Associate Requirements | TBD |

**Table 11.ADD-FAV: Add to favorite**

|  |  |
| --- | --- |
| Use Case ID: ADD-FAV | User Case Name: Add to favorite |
| Primary Actor(s): | Customers |
| Secondary Actor(s): | N/A |
| Description: | Add a restaurant to a customer’s favorite |
| Preconditions: | A restaurant detail page is shown, at the bottom there is a “AddFavorite” button waits for customer to click |
| Normal Flow of Events: | User clicks on “AddFavorite” button |
| Postconditions: | In the MyFavorite page of the customer, the link of this restaurant’s detail page is added, with a “remove” button on the right |
| Frequency of Use: | Medium |
| Alternative Flows: | TBD |
| Exceptions: | If the restaurant has already been added to the user’s favorite, next time if a customer clicks on “AddFavorite”, nothing will be done |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | The restaurant’s detail page link |
| Associate Requirements | TBD |

**Table 12.REMV-FAV: Remove favorite**

|  |  |
| --- | --- |
| Use Case ID: REMV-FAV | User Case Name: Remove favorite |
| Primary Actor(s): | Customers |
| Secondary Actor(s): | N/A |
| Description: | Remove from favorite |
| Preconditions: | On the “MyFavorite” page of the customer, links of favorite restaurants are listed, each with a “remove” button waits for user to click |
| Normal Flow of Events: | User clicks on “remove” button |
| Postconditions: | The link of the removed restaurant will be cleared |
| Frequency of Use: | Medium |
| Alternative Flows: | TBD |
| Exceptions: | The user has to click on “delete mode” to enter the delete favorite mode, the choose the favorite restaurant name to delete this favorite. If doing this in an wrong order, an exception message will be toasted to remind the user of the correct order |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | The restaurant’s detail page link |
| Associate Requirements | TBD |

**Table 13.SEND-RES: Sending reservation message**

|  |  |
| --- | --- |
| Use Case ID: SEND-RES | User Case Name: Sending reservation message |
| Primary Actor(s): | Customers |
| Secondary Actor(s): | N/A |
| Description: | Send a reservation message |
| Preconditions: | On reserve page, a “confirm” button waits for user to click |
| Normal Flow of Events: | User specifies the reservation information and clicks on the “confirm” button |
| Postconditions: | The reservation information will be posted on the other involver’s personal information interface |
| Frequency of Use: | Medium |
| Alternative Flows: | TBD |
| Exceptions: | TBD |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | The reservation deal |
| Associate Requirements | TBD |

**Table 14.SEND-CAN: Sending a cancel message**

|  |  |
| --- | --- |
| Use Case ID: SEND-CAN | User Case Name: Send a cancel message |
| Primary Actor(s): | Customers/Business owner |
| Secondary Actor(s): | N/A |
| Description: | Sending a cancel message |
| Preconditions: | A list view showing reservation detail is displayed on the user/business owner’s personal information interface. At the bottom there is a “delete” button waits for user to click |
| Normal Flow of Events: | User clicks the “delete” button |
| Postconditions: | Next time after a user(customer) clicks on “reserve” on detail page, he/she will go to a normal reservation page. A cancelled reservation will disappear from the business owner personal information interface;  A user(business owner) clicks on the “cancel” button, the reservation will disappear from the reservation list view of the customer’s personal information |
| Frequency of Use: | Medium |
| Alternative Flows: | TBD |
| Exceptions: | TBD |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | The reservation deal |
| Associate Requirements | TBD |

**Table 15.RATE-RES: Rate a restaurant**

|  |  |
| --- | --- |
| Use Case ID: RATE-RES | User Case Name: Rate a restaurant |
| Primary Actor(s): | Customers |
| Secondary Actor(s): | N/A |
| Description: | Rating a restaurant |
| Preconditions: | On restaurant detail page, there is a “rating” button waits for user to click. |
| Normal Flow of Events: | 1. User give a rate for this restaurant, click ”rate us” button  2. New average rating is calculated and the rating on the restaurant detail page under the restaurant name is updated |
| Postconditions: | The rate us button along with the rate edit view is shown on the restaurant detail interface, which shows the rating he/she gives for this restaurant, whether the user has rated this restaurant, he/she can give a new score and click “rate us”, and average rating for the restaurant will be recalculated |
| Frequency of Use: | Medium |
| Alternative Flows: | TBD |
| Exceptions: | If the customer gives a zero or negative or more than ten or blank for this restaurant, then this rating will not be taken into account, any rating has to be greater than zero, and the exception message will be toasted |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | TBD |
| Associate Requirements | TBD |

**Table 16.ENABLE: Enable User**

|  |  |
| --- | --- |
| Use Case ID: ENABLE | User Case Name: Enable User |
| Primary Actor(s): | Administrator |
| Secondary Actor(s): | N/A |
| Description: | Approving a sign up request |
| Preconditions: | A customer/business owner tries to sign up for an account, a request message is sent to the administrator’s inbox. |
| Normal Flow of Events: | User’s information is added automatically into the server database |
| Postconditions: | The request is approved |
| Frequency of Use: | High |
| Alternative Flows: | User checks the message and select “Decline”, the request get declined |
| Exceptions: | TBD |
| Assumptions: | N/A |
| Issues: | N/A |
| Source: | Sample user cases and user requirements documents |
| Includes | TBD |
| Associate Requirements | TBD |

**Table 17. FB: Feedback**

|  |  |
| --- | --- |
| Use Case ID: FB | User Case Name: Feedback |
| Primary Actor(s): | Customers/Business owner |
| Secondary Actor(s): | N/A |
| Description: | User gives feedback for the app to the Administrator |
| Preconditions: | After logging in, at the bottom of the main UI, there is a “feedback” button waiting to be clicked |
| Normal Flow of Event | 1. User clicks on the “feedback” choice on the navigation drawer; 2. User is led to a feedback page, which has an empty box to fill in feeback; 3. After filling in the opinions, the user can click on the “submit” button at the bottom. |
| Postconditions: | Feedback message is sent via email to the developer’s email inbox |
| Frequency of Use: | Medium |
| Alternative Flows: | TBD |
| Exceptions: | TBD |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | TBD |
| Associate Requirements | TBD |

**Table 18.CHECK-MSG: Check message**

|  |  |
| --- | --- |
| Use Case ID: CHECK-MSG | User Case Name: check messages |
| Primary Actor(s): | Developer |
| Secondary Actor(s): | N/A |
| Description: | Check messages |
| Preconditions: | User has used the feedback interface to write and send a feedback email to the developer’s email inbox |
| Normal Flow of Events: | User sent a feedback email by using the feedback interface, developer logs into the developer’s email inbox |
| Postconditions: | None |
| Frequency of Use: | High |
| Alternative Flows: | TBD |
| Exceptions: | TBD |
| Assumptions: | User has successfully logged in with a valid account |
| Issues: | User does not have a valid account |
| Source: | Sample user cases and user requirements documents |
| Includes | TBD |
| Associate Requirements | TBD |