

Team JN08

Elaboration Document

Carpooling System

Team Members:

|  |
| --- |
| 214104974, Logan Fleur  214139549, Richard Smart  213329913, Agnes Kijewski |

Version Number: 1.0

Version Date: 9 May 2016

Table of Contents

[1 FUNCTIONAL REQUIREMENTS 3](#_Toc456857081)

[1.1 Analysis Use Case Model 3](#_Toc456857082)

[1.2 Updated Use Case Glossary 4](#_Toc456857083)

[1.3 Detailed Use Case Narratives 7](#_Toc456857084)

[1.3.1 Package A: Communication 7](#_Toc456857085)

[1.3.2 Package B: Ridesharing Group 18](#_Toc456857086)

[1.3.3 Package C: Administration 39](#_Toc456857087)

[2 DATA REQUIREMENTS 60](#_Toc456857088)

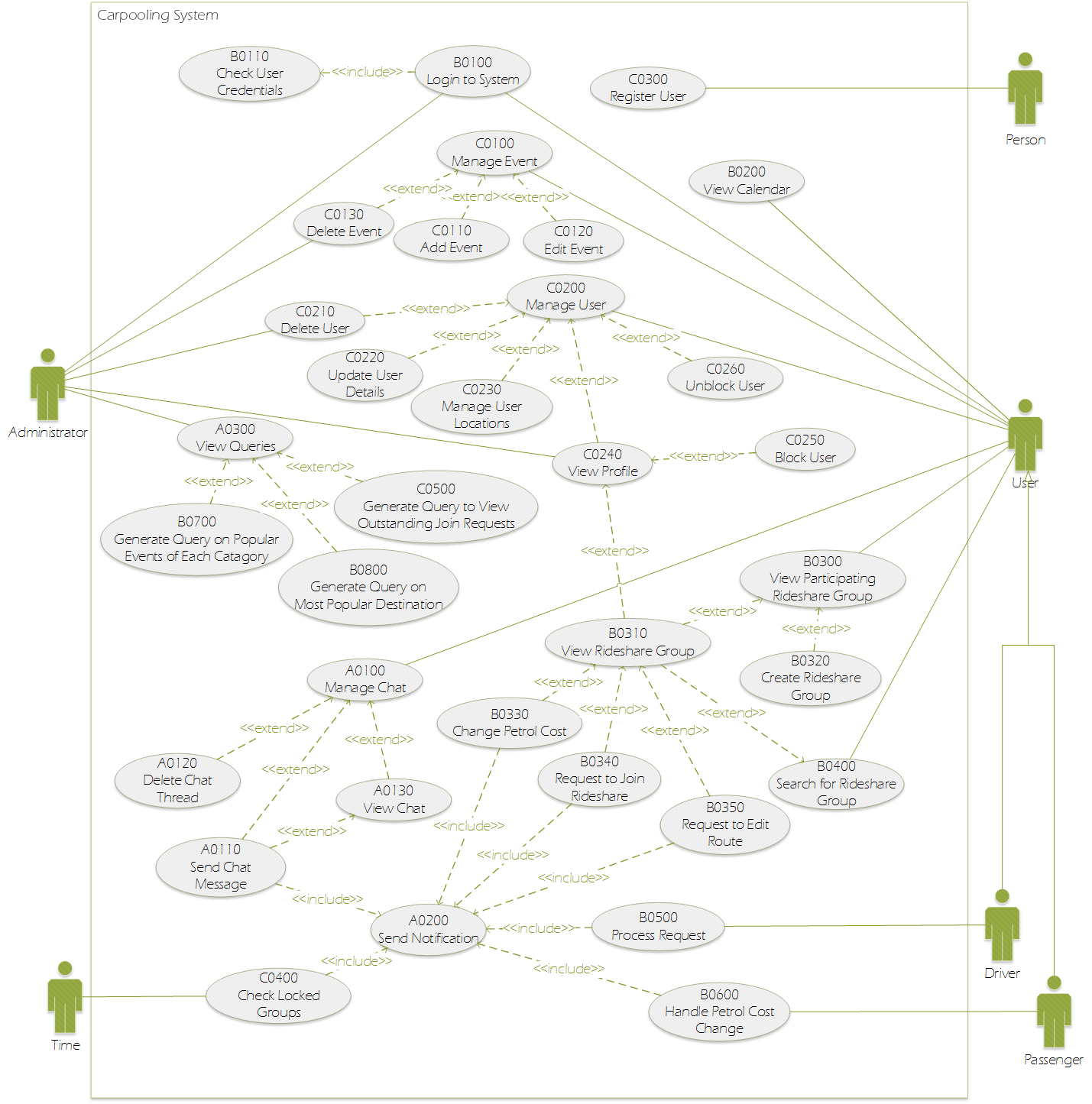
[2.1 Class Diagram 60](#_Toc456857089)

[2.2 Database Design 61](#_Toc456857090)

[3 DOCUMENT APPROVAL 62](#_Toc456857091)

# FUNCTIONAL REQUIREMENTS

## Analysis Use Case Model



## Updated Use Case Glossary

|  |  |  |
| --- | --- | --- |
| **Package Name: Communication** | | |
| **Use Case Id** | **Use Case Name** | **Actors** |
| A0100 | Manage Chat | User |
| A0110 | Send Chat Message | User |
| A0120 | Delete Chat Thread | User |
| A0130 | View Chat | User |
| A0200 | Send Notification | - |
| A0300 | View Queries | Administrator |

|  |  |  |
| --- | --- | --- |
| **Package Name: Rideshare Group** | | |
| **Use Case Id** | **Use Case Name** | **Actors** |
| B0100 | Login to System | Administrator, User |
| B0110 | Check User Credentials | Administrator, User |
| B0200 | View Calendar | User |
| B0300 | View Participating Rideshare Groups | User |
| B0310 | View Rideshare Group | User |
| B0320 | Create Rideshare Group | Driver |
| B0330 | Change Petrol Cost | Driver |
| B0340 | Request to Join Rideshare | User |
| B0350 | Request to Edit Route | Passenger |
| B0400 | Search for Rideshare Group | User |
| B0500 | Process Request | Driver |
| B0600 | Handle Petrol Cost Change | Passenger |
| **Queries/Reports** | | |
| B0700 | Generate Query on Popular Events of Each Category | Administrator |
| B0800 | Generate Query on Most Popular Destination | Administrator |

|  |  |  |
| --- | --- | --- |
| **Package Name:** | | |
| **Use Case Id** | **Use Case Name** | **Actors** |
| C0100 | Manage Event | User |
| C0110 | Add Event | User |
| C0120 | Edit Event | User |
| C0130 | Delete Event | Administrator, User |
| C0200 | Manage User | User |
| C0210 | Delete User | User, Administrator |
| C0220 | Update User Details | User |
| C0230 | Manage User Locations | User |
| C0240 | View Profile | User |
| C0250 | Block User | User |
| C0260 | Unblock User | User |
| C0300 | Register User | Person |
| C0400 | Check Locked Groups | - |
| **Queries/Reports** | | |
| C0500 | Generate Query to View Outstanding Join Requests | Administrator |

## Detailed Use Case Narratives

General Notes

* General Cancellation - At any point where a User/Administrator can input/select data, they can cancel the use case by selecting the cancel option, or selecting outside the dialog space.
* The event categories will be defined in the code. It will be a generic list which is hard coded within the database.

### Package A: Communication

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0100 | Manage Chat | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | | - |
| **Description** | Allows users to communicate with each other. The User will be able to view, send and delete chats between themselves and other Users. | |
| **Pre-Conditions** | Users must be logged into the system. | |
| **Triggers** | This is triggered when the User selects the chat option. | |
| **Post-Conditions** | Depending on what the User would like to do on the messaging system, Send Chat Message (A0110), Delete Chat Thread (A0120) and, View Chat (A0130) would be triggered respectively. | |
| **Basic Flow of Events** | | |
| 1. System displays search bar and a list of current chats in the User’s inbox 2. User enters another user’s name in the search bar 3. System returns a list of users that match the search input 4. User selects the correct User name 5. User chooses option to start a chat, (Send Chat Message - A0110) is invoked | | |
| **Alternate Flow of Events** | | |
| 1. User selects a User name from the list of already open chats 2. Return to Step 5 3. If the User chooses option to delete a chat, (Delete Chat Thread – A0120) is invoked, given that a chat has already been started with the user 4. If the User chooses option to view a chat, (View Chat – A0130) is invoked, given that a chat has already been started with the user | | |
| **Initial Screen Design** | | |
| User begins typing the name of another user he/she wants to search for. System provides a list of user’s names that match what has been typed so far    The User selects the user he/she is looking for from the list of names. Options for the selected user are displayed    User selects an appropriate option | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0110 | Send Chat Message | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | User |
| **Description** | Allows a User to start a chat with other Users. Any messages sent would appear in the recipient(s)'s chat inbox(es) under the sender’s name. Recipients can be either individuals or ridesharing groups which the User are a part of. | |
| **Pre-Conditions** | User recipients must be registered Users or active ridesharing groups. (Once off ridesharing groups will be deleted after the event has occurred) | |
| **Triggers** | Triggered by Manage Chat (A0100)  OR  Triggered from View Chat (A0130) | |
| **Post-Conditions** | Message is sent to recipient/s which the user specified and Send Notification (A0200) is invoked.  OR  User opts not to send a chat message even though they have started the process (catered for in General Notes) | |
| **Basic Flow of Events** | | |
| 1. User inputs text for message 2. User selects to send the message 3. System sends the message 4. Send Notification (A0200) is triggered | | |
| **Alternate Flow of Events** | | |
| 1. System informs the user that no text has been entered in the body of the message and the message must contain text before it can be sent 2. System informs the user that the message cannot be sent to a non-existent recipient, and aborts sending the message 3. Return to step 2 | | |
| **Initial Screen Design** | | |
| User selects the user/rideshare group he/she would like to message and types the text to be sent in the text box shown    The User sends the message and it is then displayed in both users chat logs | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0120 | Delete Chat Thread | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | User |
| **Description** | Allows a User to delete chat threads they have in their inbox | |
| **Pre-Conditions** | User must have (a) chat/s to be deleted. | |
| **Triggers** | Triggered by Manage Chat (A0100) | |
| **Post-Conditions** | Chat is deleted and is no longer available.  OR  User opts not to delete the chat and no chats are deleted (catered for in General Notes) | |
| **Basic Flow of Events** | | |
| 1. User selects a chat 2. System displays option for a drop down menu icon on selected chat thread, and displays the chat thread 3. User selects to open the options menu on the selected chat thread 4. System expands the options menu, displaying additional options that can be selected by the user 5. User selects the delete option from options which are now displayed in the dropdown menu (before the delete option was not visible) 6. System warns the User that deleting the chat(s) is irreversible with a dialog 7. User confirms that they wish to delete the chat(s) 8. System deletes the selected chat(s) from the user’s list of active chats 9. System displays the list of chats | | |
| **Alternate Flow of Events** | | |
| 1. User selects the checkbox on the message. 2. System displays checkboxes on all the other chats 3. Return to Step 3 4. User selects the checkbox on the message 5. System displays the checkboxes on all the other chats 6. User selects multiple checkboxes on the chats they wish to delete 7. Return to Step 3 | | |
| **Initial Screen Design** | | |
| The User selects the chat thread they would like to delete and selects the blue option arrow to display additional options.    The User select the delete option. (other options are displayed just to show this is a drop down menu with the potential for more options) | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0130 | View Chat | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | User |
| **Description** | The User selects the chat which they would like to view. The System displays the message selected for viewing to the User. | |
| **Pre-Conditions** | The User must be logged in and must have chosen to go into the messaging system. | |
| **Triggers** | Triggered by Manage Chat (A0100) | |
| **Post-Conditions** | The system displays the message selected for viewing to the User. From there the User opts to either exit the Manage Chat (A0100) system or Send Chat Message (A0110). | |
| **Basic Flow of Events** | | |
| 1. User selects the chat which they would like to view 2. System displays the chat thread selected for viewing to the User 3. User exits Manage Chat (A0100) screen. | | |
| **Alternate Flow of Events** | | |
| 1. User selects to send a message from the current screen. 2. Go to Step 5 of Send Chat Message (A0110). | | |
| **Initial Screen Design** | | |
| User selects one of the available chat threads to be displayed, Step 2 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0200 | Send Notification | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | - |
| **Description** | Notifies Users of any messages they’ve received, or updates regarding the events or rides they are subscribed to. | |
| **Pre-Conditions** | Users must already part of a group going to an event if the notification regards a specific event or rideshare group.  The User must be registered on the system | |
| **Triggers** | Send Chat Message (A0110); Change Petrol Cost (B0330); Request to Join Rideshare (B0340); Request to Edit Route (B0350); Process Request (B0500); Check Locked Groups (C0400); Handle Petrol Cost Change (B0600) | |
| **Post-Conditions** | Notification is visible to all relevant logged in users. System state is returned to previous Trigger | |
| **Basic Flow of Events** | | |
| 1. System sends notification to relevant Users based on message from triggering use case. 2. Return to previous Trigger | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
|  | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| A0300 | View Queries | |
| **Primary Business Actors** | | **Other participating Actors** |
| Administrator | | - |
| **Description** | Allows an Administrator to view the queries available, and to generate them in a subsequent use case | |
| **Pre-Conditions** | Administrator must be logged into the system. | |
| **Triggers** | This is triggered when the Administrator selects the option to view the available queries | |
| **Post-Conditions** | The queries what can be generated is displayed to the User | |
| **Basic Flow of Events** | | |
| 1. System invokes Generate Query on Popular Events of Each Category (B0700) 2. Administrator exits option to view queries | | |
| **Alternate Flow of Events** | | |
| 1. Administrator selects the option to view a query of the most popular destination 2. System invokes Generate Query on Most Popular Destination (B0800) 3. Administrator selects the option to view a query of outstanding join requests 4. System invokes Generate Query to View Outstanding Join Requests (C0500) | | |
| **Initial Screen Design** | | |
| Initial screen, before Administrator is redirected to the first use case. | | |
| **Final Screen Design** | | |
|  | | |

### Package B: Ridesharing Group

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0100 | Login to System | |
| **Primary Business Actors** | | **Other participating Actors** |
| Administrator, User | | - |
| **Description** | Administrators and Users need to login in order to use the carpooling system and its features. They will login with their unique username and their password. | |
| **Pre-Conditions** | The Administrator/User must not be logged in yet. | |
| **Triggers** | Selecting to login to the system. | |
| **Post-Conditions** | The Administrator/User will be logged into the system.  OR  A general cancellation occurs at some point (catered for in General Notes) | |
| **Basic Flow of Events** | | |
| 1. Administrator/User selects the login option. 2. System prompts the Administrator/User for their username and password. 3. Administrator/User enters their credentials and submits them. 4. Check User Credentials (B0110) is invoked. 5. System logs the Administrator/User in. | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| Login interface after Step 3 where details have been entered | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0110 | Check User Credentials | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | Administrator, User |
| **Description** | The system checks the username and password which the Administrator/User have entered. | |
| **Pre-Conditions** | The Administrator/User must be registered. | |
| **Triggers** | Triggered by Step 4 in Login to System (B0100) | |
| **Post-Conditions** | System state is returned to Login to System (B0100) | |
| **Basic Flow of Events** | | |
| 1. System checks the credentials entered by the Administrator/User against the database. 2. System finds that the credentials are correct. 3. Return to Step 5 of Login to System (B0100). | | |
| **Alternate Flow of Events** | | |
| * 1. The credentials entered are incorrect   2. System displays an incorrect username and password message.   3. Returns to Step 2 of Login to System (B0100). | | |
| **Initial Screen Design** | | |
| Textboxes illuminated red to indicate a failed login, alternate flow 2.1.2 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0200 | View Calendar | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | Users can view their calendar with upcoming rides. | |
| **Pre-Conditions** | User must be logged in | |
| **Triggers** | Triggered when the View Calendar option is clicked. | |
| **Post-Conditions** | A calendar is displayed with the upcoming rides put into the calendar. | |
| **Basic Flow of Events** | | |
| 1. User selects the option to view their calendar. 2. System then displays the calendar to the User with the user’s events indicated in the calendar. | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| State of interface after system displays user agenda, Step 2 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0300 | View Participating Rideshare Groups | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | | - |
| **Description** | Users can view the Rideshare Groups they are currently in. | |
| **Pre-Conditions** | User must be logged in. | |
| **Triggers** | User selects to view the Rideshare Groups they are a part of | |
| **Post-Conditions** | The User’s list of Rideshare Groups which they are a part of will be displayed | |
| **Basic Flow of Events** | | |
| 1. System displays list of Rideshare Groups the User is a part of 2. User selects a Rideshare group to view 3. View Rideshare Group (B0310) is invoked | | |
| **Alternate Flow of Events** | | |
| 1. User types part of the name of the Rideshare Group 2. System displays an updated list of Rideshare Groups which match the entered text 3. Go to Step 2 | | |
| **Initial Screen Design** | | |
| State of interface after Step 1, displaying the Rideshare Groups the User is a part of | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0310 | View Rideshare Group | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | User |
| **Description** | Users can view the current Rideshare Group’s attributes | |
| **Pre-Conditions** | User must be logged in. | |
| **Triggers** | Rideshare Group is selected in View Participating Rideshare Groups (B0300) | |
| **Post-Conditions** | Depending on the User, the group may simply be viewed, | |
| **Basic Flow of Events** | | |
| 1. System displays the Rideshare Group 2. User returns to the home screen | | |
| **Alternate Flow of Events** | | |
| 1. User selects an option specific to the type of user they are, either Driver or Passenger, in relation to the Rideshare Group | | |
| **Initial Screen Design** | | |
| General interface when a User selects to view a Rideshare Group | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0320 | Create Rideshare Group | |
| **Primary Business Actors** | | **Other participating Actors** |
| Driver | | - |
| **Description** | Users can create new ridesharing groups. This ridesharing group is used to allow users travel to the same event/destination in one vehicle. By a User creating this ridesharing group, the User becomes the Driver of the group. | |
| **Pre-Conditions** | A User must be registered as a Driver User type in the System. User must be logged in. | |
| **Triggers** | The User selects the option to create a ridesharing group. | |
| **Post-Conditions** | A group is created with user set as “Driver” and the initial route from drivers start location to event is determined by system. | |
| **Basic Flow of Events** | | |
| 1. User selects the option to create a rideshare group 2. System displays a dialog requesting entry of data 3. User enters the data, and selects an event required to form a rideshare 4. User submits the entered details to create a rideshare group 5. System creates a rideshare group with User as the Driver 6. User is returned to the Main Screen | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| State of interface during Step 3, as data is being entered | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0330 | Change Petrol Cost | |
| **Primary Business Actors** | | **Other participating Actors** |
| Driver | | - |
| **Description** | Driver changes price per km in a ridesharing group to an event. | |
| **Pre-Conditions** | Driver need to be logged into the system. | |
| **Triggers** | The Driver chooses to change the petrol cost of a ridesharing group which he is a part of. | |
| **Post-Conditions** | Price change is sent through Send Notification (A0200) being triggered | |
| **Basic Flow of Events** | | |
| 1. Driver selects to change price per km. 2. System displays dialog to change price 3. Driver enters new price per km 4. Driver selects to change the price per km 5. Send Notification (A0200) is triggered. | | |
| **Alternate Flow of Events** | | |
| Initial state of interface before changing the petrol cost    Dialog shown when changing the petrol price, Step 2 | | |
| **Initial Screen Design** | | |
|  | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0340 | Request to Join Rideshare | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | Passenger requests to join a rideshare group but only when the Passenger is not already part of another group going to the same destination at the same time. | |
| **Pre-Conditions** | User must be logged in and cannot already be part of a rideshare group going to the same event. | |
| **Triggers** | Triggered by View Rideshare Group (B0310) | |
| **Post-Conditions** | Request is sent to Driver for confirmation. Process Request (B0500) is invoked. | |
| **Basic Flow of Events** | | |
| 1. User selects appropriate rideshare group to join. 2. System sends an application to join the above respective group to the Driver. 3. Process Request (B0500) is invoked. | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| State of system after User opens rideshare group, before User performs any further steps | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0350 | Request to Edit Route | |
| **Primary Business Actors** | | **Other participating Actors** |
| Passenger | | - |
| **Description** | Passenger can choose to edit their pickup location on the route. | |
| **Pre-Conditions** | Passenger must be logged in. Passengers must be notified and agree to changes. | |
| **Triggers** | Passenger selects edit route option. | |
| **Post-Conditions** | Route is updated and Passengers will be notified that route has successfully been updated.  OR  A general cancellation occurs at some point (catered for in General Notes) | |
| **Basic Flow of Events** | | |
| 1. Passenger opts to change route on the system. 2. The system displays the screen for changing the route. 3. Passenger selects the new pickup location to be proposed. 4. The passenger submits the new changes on the system. 5. Send Notification (A0200) is invoked | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
|  | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0400 | Search for Rideshare Group | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | | - |
| **Description** | User searches for rideshare groups for a specific event (using event name). | |
| **Pre-Conditions** | User must be logged in. | |
| **Triggers** | User initiates rideshare group search by using the search option. | |
| **Post-Conditions** | User presented with list of possible groups.  OR  A general cancellation occurs at some point. (catered for in General Notes) | |
| **Basic Flow of Events** | | |
| 1. System displays list of events in order of proximity to the User 2. User starts typing event name into the search bar 3. System displays a short list of suggested events (if any exist), based on their current search terms and proximity to event 4. User selects event from the suggestions made by the system 5. System presents User with a list of most appropriate groups. 6. User selects most appropriate group (or group of his/her choice). 7. Request to Request to Join Rideshare (B0340) is invoked 8. Return User to home screen | | |
| **Alternate Flow of Events** | | |
| 1. User continues typing the name of a recipient or ridesharing group 2. Return to Step 3 3. After 1 second of inactivity, System verifies that the event exists. If it does not exist then a warning will be displayed. 4. User has no groups to select from. 5. Return to Step 2 | | |
| **Initial Screen Design** | | |
| Selected group displayed, Step 6 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0500 | Process Request | |
| **Primary Business Actors** | | **Other participating Actors** |
| Driver | | - |
| **Description** | Driver decides whether to accept Users into a rideshare group where they all are all travelling together to an event or destination. | |
| **Pre-Conditions** | User must apply to join Driver’s rideshare group | |
| **Triggers** | Request to Join Rideshare (B0340) or Request to Edit Route (B0350). | |
| **Post-Conditions** | Passenger is accepted into the rideshare group. Send Notification (A0200) is triggered. | |
| **Basic Flow of Events** | | |
| 1. System displays received request or route modification 2. Driver accepts request to join rideshare group or edit route. 3. Passenger is added to the rideshare group or route is modified by System. 4. Send Notification (A0200) is invoked 5. System displays list of User’s notifications | | |
| **Alternate Flow of Events** | | |
| 1. Driver rejects request or route modification 2. Go to Step 4 | | |
| **Initial Screen Design** | | |
| Systems displays list of pending rideshare join requests    Driver opts to either accept or deny a join request. A confirmation message is displayed should a driver opt to accept a new passenger into the group | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0600 | Handle Petrol Cost Change | |
| **Primary Business Actors** | | **Other participating Actors** |
| Passenger | | - |
| **Description** | Passengers can accept or deny the request, by leaving the rideshare group, to change the petrol cost by the Driver. | |
| **Pre-Conditions** | A change to the petrol cost must exist within the rideshare group | |
| **Triggers** | Passenger acts on the notification received regarding a change in petrol cost | |
| **Post-Conditions** | The Passenger remains within the group or leaves the group if they do not accept the change in the petrol cost | |
| **Basic Flow of Events** | | |
| 1. Passenger accepts the proposed change in petrol cost. 2. Send Notification (A0200) is triggered. | | |
| **Alternate Flow of Events** | | |
| 1. Passenger rejects the change in petrol cost. 2. System warns the Passenger that they will be removed from the group if they reject the change 3. Passenger accepts being removed from the rideshare group 4. System removes the Passenger from the rideshare group 5. Go to Step 2 | | |
| **Initial Screen Design** | | |
| Dialog warning the User of the change in the Price/km, before any action is taken by the User | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0700 | Generate Query on Popular Events of Each Category | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | Administrator |
| **Description** | Shows events where the number of attendees is above the 10th percentile of total users attending per event in any time period | |
| **Pre-Conditions** | More than one Event must exist in the Category.  Administrator must be logged in | |
| **Triggers** | Request for Query on selection by the Administrator from View Queries (A0300) | |
| **Post-Conditions** | Query results are displayed to the Administrator | |
| **Basic Flow of Events** | | |
| 1. Administrator selects a time period during which to search for events. 2. System calculates results and returns them for viewing by the Administrator. | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| State of interface after query is executed, Step 2 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| B0800 | Generate Query on Most Popular Destination | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | Administrator |
| **Description** | Shows which event was the most popular in any time period | |
| **Pre-Conditions** | More than one Event must exist in the System.  Administrator must be logged in | |
| **Triggers** | Request for Query on selection by the Administrator from View Queries (A0300) | |
| **Post-Conditions** | Query results are displayed to the Administrator | |
| **Basic Flow of Events** | | |
| 1. Administrator selects a time period during which to search for events. 2. System calculates results and returns them for viewing by the Administrator. | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| State of interface after query is generated, Step 2 | | |
| **Final Screen Design** | | |
|  | | |

### Package C: Administration

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0100 | Manage Event | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | | - |
| **Description** | Maintain Event allows the User to create events, edit events, and delete events. | |
| **Pre-Conditions** | User must be logged into the system | |
| **Triggers** | User selects to the maintain event option | |
| **Post-Conditions** | Depending on the User, Add Event (C0110), Edit Event (C0120) or Delete Event (C0130) is invoked | |
| **Basic Flow of Events** | | |
| 1. User searches for a specific event 2. System provides a list of events relative to the search criteria 3. User selects an event from the list provided, triggering Edit Event (C0120) | | |
| **Alternate Flow of Events** | | |
| 1. User selects to Add Event (C0110) 2. User selects to Delete Event (C0130) | | |
| **Initial Screen Design** | | |
| Initial landing page, with blank search space. | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0110 | Add Event | |
| **Primary Business Actors** | | **Other participating Actors** |
| - | | User |
| **Description** | Allows the User to add a new Event to the system | |
| **Pre-Conditions** | User must be logged in | |
| **Triggers** | Manage Event (C0100) | |
| **Post-Conditions** | Event may be added to the system; User is returned to Event list screen | |
| **Basic Flow of Events** | | |
| 1. System displays New Event dialog 2. User enters event particulars 3. User selects to save the event 4. System validates entered data 5. System stores the event in the database 6. System displays Manage Event (C0100) screen | | |
| **Alternate Flow of Events** | | |
| 1. System warns user that invalid data has been entered 2. Go back to Step 2 | | |
| **Initial Screen Design** | | |
| State of interface after Step 2, where details have been entered    Warning after invalid details were added, in alternate flow 5.1.1 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0120 | Edit Event | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | Allows Users to edit the Event’s | |
| **Pre-Conditions** | User must be logged in, User must be the creator of the Event | |
| **Triggers** | Manage Event (C0100) | |
| **Post-Conditions** | Changes to the Event are discarded or saved; User is returned to Event list screen | |
| **Basic Flow of Events** | | |
| 1. System displays dialog of current event details 2. User modifies the detail events 3. User submits the modified event 4. System validates the entered details 5. System stores the event in the database 6. System displays Manage Event (C0100) screen | | |
| **Alternate Flow of Events** | | |
| 1. User selects to delete event, triggering Delete Event (C0130) 2. System warns user that invalid data has been entered 3. Return to Step 2 | | |
| **Initial Screen Design** | | |
| State of interface after Step 1, showing the current details of the event    Warning of invalid information being entered, displayed in alternate flow 5.1.1 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0130 | Delete Event | |
| **Primary Business Actors** | | **Other participating Actors** |
| Administrator, User | |  |
| **Description** | Allows the deletion of an Event from the system | |
| **Pre-Conditions** | User/Administrator must be logged in  User must be the creator of the Event | |
| **Triggers** | Triggered by delete option in Manage Event (C0100) | |
| **Post-Conditions** | Event may be deleted from the system | |
| **Basic Flow of Events** | | |
| 1. System warns the User/Administrator that the deletion is irreversible. 2. User/Administrator confirms the deletion 3. System removes the Event from the system 4. User/Administrator is returned to Manage Event (C0100) | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| Dialog to warn User that deletion of an event is irreversible, Step 1 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0200 | Manage User | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | Allows a User to delete, register or update a User and their locations, view profiles and unblock a user | |
| **Pre-Conditions** | User must be logged in | |
| **Triggers** | On selecting the Manage User option by an User | |
| **Post-Conditions** | Depending on the User, Delete User (C0210), Update User Details (C0220) or Manage User Locations (C0230), View Profile (C0240), Unblock User (C0260) is invoked | |
| **Basic Flow of Events** | | |
| 1. System invokes Update User Details (C0220) 2. User exits the option to manage their details | | |
| **Alternate Flow of Events** | | |
| 1. User selects the option to delete another user 2. System invokes Delete User (C0210) 3. User selects the option to update their locations 4. System invokes Manage User Locations (C0230) 5. User selects the option to view another User’s profile 6. System invokes View Profile (C0240) 7. User selects the option to unblock a User 8. System invokes Unblock User (C0260) | | |
| **Initial Screen Design** | | |
| Initial screen, before Step 1 is executed, selecting an option to display | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0210 | Delete User | |
| **Primary Business Actors** | | **Other participating Actors** |
| User, Administrator | | - |
| **Description** | A User deletes themselves off the system, or Administrators delete a User from the system. | |
| **Pre-Conditions** | The User/Administrator must be logged in. | |
| **Triggers** | User/Administrator selects the delete account option. | |
| **Post-Conditions** | A User’s details are deleted from the system. | |
| **Basic Flow of Events** | | |
| 1. System displays a warning informing the User/Administrator that the deletion is irreversible 2. User/Administrator confirms by selecting to delete the account. 3. System deletes the User from the system | | |
| **Alternate Flow of Events** | | |
| 1. User selects to cancel the deletion of their account. 2. User is returned to Manage User (C0200) | | |
| **Initial Screen Design** | | |
| Initial screen, before User is has selected an option in Step 1 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0220 | Update User Details | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | Allows a User to update their profile details | |
| **Pre-Conditions** | User must be logged in | |
| **Triggers** | Manage User (C0200) | |
| **Post-Conditions** | User’s details are updated or unaffected | |
| **Basic Flow of Events** | | |
| 1. System displays dialog populated with current details 2. User modifies the displayed details 3. User selects to submit the modifications 4. System saves the changes to the User’s profile 5. User is returned to Manage User (C0200) | | |
| **Alternate Flow of Events** | | |
| 1. User declines to save the modifications 2. Go to Step 5 | | |
| **Initial Screen Design** | | |
| State of interface after Step 1 has displayed the current User’s details | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0230 | Manage User Locations | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | |  |
| **Description** | Allows a User to Add or Delete saved locations | |
| **Pre-Conditions** | User must be logged in | |
| **Triggers** | Manage User (C0200) | |
| **Post-Conditions** | User’s stored locations may be modified or unaffected | |
| **Basic Flow of Events** | | |
| 1. System displays list of stored locations 2. User selects to add a location 3. System displays dialog prompting for details 4. User completes entry form 5. User selects to submit the entry as a new location 6. System saves the location to the database 7. System returns User to Manage User (C0200) | | |
| **Alternate Flow of Events** | | |
| 1. User selects a location to delete 2. System displays selected location 3. User selects to delete the selected location 4. System warns User that deleting a location is irreversible 5. User selects to delete the location 6. System deletes the location from the database 7. Return to Step 7 8. User selects a location to edit 9. System displays selected location 10. User modifies selected location 11. User selects to save the edited location 12. System updates the location in the database 13. Return to Step 7 | | |
| **Initial Screen Design** | | |
| List of stored locations is shown, from Step 1    Selected location data is displayed, alternate flow 2.2.2    Warning dialog shown after user selects the delete option, alternate flow 2.1.3 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0240 | View Profile | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | | - |
| **Description** | A User can view their own profile or another User’s profile on the system. | |
| **Pre-Conditions** | The User must be registered on the system | |
| **Triggers** | User chooses to view their own profile or someone else’s profile from the options given in Manage User (C0200) | |
| **Post-Conditions** | The profile displayed to the User | |
| **Basic Flow of Events** | | |
| 1. User starts typing User’s name into the search bar 2. System presents the User with a list of people, having the common characters in their name as what has be typed 3. User selects the User, whose profile they would like to view, from the list given by the system 4. System displays the User’s profile 5. User exits back to the home screen | | |
| **Alternate Flow of Events** | | |
| 1. User continues typing the name of the User they are searching for 2. Return to Step 3 3. User has no options displayed to them which they can select from[[1]](#footnote-1) 4. Return to Step 2 5. User selects to block the displayed User. Invoke Block User (C0250) | | |
| **Initial Screen Design** | | |
| State of interface after Step 4, selected user is displayed | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0250 | Block User | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | | - |
| **Description** | A User can block another User so that there is no communication between the blocker and the blocked user. A User can only be blocked by the option given on their profile. | |
| **Pre-Conditions** | The User must be registered on the system | |
| **Triggers** | User selects Block User option from the screen in View Profile (C0240) | |
| **Post-Conditions** | The User is blocked  OR  The User remains unblocked | |
| **Basic Flow of Events** | | |
| 1. System asks the User if they want to block the User 2. User selects to block the User 3. System blocks the User, barring communication between the two. | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| Dialog shown to user during Step 1 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0260 | Unblock User | |
| **Primary Business Actors** | | **Other participating Actors** |
| User | | - |
| **Description** | A User can select to unblock another User to restore communication abilities between Users. | |
| **Pre-Conditions** | The User must be registered on the system | |
| **Triggers** | User selects Block User option | |
| **Post-Conditions** | The User is unblocked  OR  The User remains blocked | |
| **Basic Flow of Events** | | |
| 1. System displays list of Users which are currently blocked 2. User selects a blocked User 3. System displays the blocked User’s profile 4. User selects to unblock the User 5. System unblocks the User, removing them from the blocked users list 6. System displays updated list of blocked Users | | |
| **Alternate Flow of Events** | | |
| 1. User types part of another User’s name in the search box 2. System updates the displayed list to only contain matching user names 3. Go to Step 2 | | |
| **Initial Screen Design** | | |
| State of interface after user selects a user to view, Step 3 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0300 | Register User | |
| **Primary Business Actors** | | **Other participating Actors** |
| Person | | - |
| **Description** | Allows an unregistered user of the System to become a User on the System | |
| **Pre-Conditions** | No User may be logged in on the device accessing the system and the unregistered user cannot have an existing account. | |
| **Triggers** | An unregistered user selects the registration option | |
| **Post-Conditions** | A unregistered user is Registered on the system as a User | |
| **Basic Flow of Events** | | |
| 1. System displays a dialog for registration 2. Unregistered user enters personal details required for registration 3. Unregistered user selects to register with a vehicle 4. Unregistered user enters vehicle registration details 5. Unregistered user selects to submit registration application 6. System creates a User for the now registered user 7. User is taken to the Home Screen | | |
| **Alternate Flow of Events** | | |
| 1. Unregistered user selects to register without a vehicle 2. Go to Step 5 3. System does not register a new User since the entered particulars already exist in the database. 4. System warns the unregistered user that the registration details already exist in the system 5. Return to Step 2 6. System displays to the unregistered user that there are compulsory fields which are empty. 7. Return to Step 2 8. System warns the unregistered user that details are not entered correctly. 9. Return to Step 2 | | |
| **Initial Screen Design** | | |
| State of interface after Step 2, user entered details for registration    Warning shown to user due to duplication, alternate flow 6.1.2    Warning shown to user that some fields are incomplete which cannot be, alternate flow 6.2.1    Warning shown to user that some data is invalid, alternate flow 6.3.1 | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0400 | Check Locked Groups | |
| **Primary Business Actors** | | **Other participating Actors** |
| Time | | User |
| **Description** | The system finds ridesharing groups where the event will take place exactly in a week's time and if there are such ridesharing groups, then the system locks those groups at 12 AM each day. | |
| **Pre-Conditions** | N/A | |
| **Triggers** | The current time must be 00:00 | |
| **Post-Conditions** | The ridesharing groups will become locked groups. Send Notification (A0200) is triggered to send out a notification to all the Users within the ridesharing group to show that the group is now locked. | |
| **Basic Flow of Events** | | |
| 1. The system finds ridesharing groups where the event will take place exactly in a week's’ time. 2. The system locks those ridesharing groups. 3. Send Notification (A0200) is triggered. | | |
| **Alternate Flow of Events** | | |
| 1. System does not find any groups to lock | | |
| **Initial Screen Design** | | |
|  | | |
| **Final Screen Design** | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Use Case Name** | |
| C0500 | Generate Query to View Outstanding Join Requests | |
| **Primary Business Actors** | | **Other participating Actors** |
| Administrator | |  |
| **Description** | Shows which rideshare groups are due to be locked at 12 AM on the day of the request | |
| **Pre-Conditions** | Administrator must be logged in | |
| **Triggers** | Request for Query on selection by the Administrator | |
| **Post-Conditions** | Query results are displayed to the Administrator | |
| **Basic Flow of Events** | | |
| 1. System generates a list of outstanding join requests and displays it to the user | | |
| **Alternate Flow of Events** | | |
|  | | |
| **Initial Screen Design** | | |
| Example query shown after selecting the specific query option, Step 1 | | |
| **Final Screen Design** | | |
|  | | |

# DATA REQUIREMENTS

## Project Class Diagram.pngClass Diagram

## Database Design

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Identifiers <pk> <fk>** | **Attributes** |
| Notfication | NotificationID <pk> | NotificationMessage, Trigger |
| NotifiedUser | NotificationID <pk> <fk1>, IDNumber <pk> <fk2> |  |
| BlockedUser | BlockID <pk>, BlockerID <fk1>, BlockedID <fk2> | BlockReason |
| User | IDNumber <pk> | Title, Name, Surname, Username, Password, UserType, Picture, Blocked |
| Driver | IDNumber <pk> <fk> | CarRegistrationNo, Rating, CellphoneNo |
| Passenger | IDNumber <pk> <fk> | Rating, CellphoneNo |
| Administrator | IDNumber <pk> <fk> |  |
| Message | MessageID <pk> | MessageNumber, Subject, MessagePath, Date, Time |
| MessageRecipient | MessageID <pk> <fk1>, IDNumber <pk> <fk2> | ReadReciept, Type |
| Location | IDNumber <pk> <fk>, AreaID <pk> | StreetNumber, StreetName, Suburb, Town, Province |
| RouteMarker | RideshareNo <pk> <fk1>, IDNumber <pk> <fk2>, AreaID <fk3> | Order |
| RideshareGroup | RideshareNo <pk>, EventID <fk> | PricePerkm, MaxPassengers, LockStatus, RecurringFrequency |
| PendingJoin | IDNumber <pk> <fk1>, RideshareNo <pk> <fk1> |  |
| Event | EventID <pk>, CategoryID <fk> | EventName, StreetNumber, StreetName, Town, Suburb, Date |
| EventCategory | CategoryID <pk> | CategoryDesc |

# DOCUMENT APPROVAL

The undersigned acknowledge that they have reviewed the **ELABORATION DOCUMENT** and agree with the information presented within this document. Changes to this **ELABOARTION DOCUMENT** will be coordinated with, and approved by the undersigned.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature:** |  | **Date:** |  |
| **Print Name:** |  |  |  |
| **Role:** | PROJECT SUPERVISOR  ***\* TO BE SIGNED AFTER DOCUMENT ASSESSMENT \**** | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature:** |  | **Date:** |  |
| **Print Name:** |  |  |  |
| **Role:** | Developer |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature:** |  | **Date:** |  |
| **Print Name:** |  |  |  |
| **Role:** | Developer |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature:** |  | **Date:** |  |
| **Print Name:** |  |  |  |
| **Role:** | Developer |  |  |

1. This would happen if the characters/name the User has input does not match any user on the system. [↑](#footnote-ref-1)