|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC-01 | | |
| Use Case Name: | Register new account | | |
| Created By: |  | Last Updated By: | Yong Hao |
| Date Created: |  | Date Last Updated: | 10.23.2018 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allows Helia Users who do not have an account to create an account |
| Preconditions: | 1. The User must not have an account  2. The User must not be logged in |
| Postconditions: | The User has created an account |
| Priority: | Nil |
| Frequency of Use: | Nil |
| Flow of Events: | 1.1  The User enters in his username, password, and email.  1.2  The User press the Submit button  1.3  The System checks if the Username and Email has been taken    AF-1: The Username has been taken    AF-2: The Email has been taken    1.4  The System saves the User information into the database  The System set the User avatar to the default |
| Alternative Flows: | AF-1: The Username has been taken  1.     The System prompts the User that the Username has been taken.  2.     The User acknowledges and is taken back to 1.3  AF-2: The Email has been taken  1.     The System prompts the User that the Email has been used.  The User acknowledge and is taken back to 1.3 |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | Nil |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC-02 | | |
| Use Case Name: | Login | | |
| Created By: |  | Last Updated By: | Yong Hao |
| Date Created: |  | Date Last Updated: | 10.23.2018 |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allows Helia Users login to their account |
| Preconditions: | The user must not be logged in |
| Postconditions: | The user is logged in |
| Priority: | Nil |
| Frequency of Use: | This use case is applicable to all users who are to use the app. |
| Flow of Events: | 1.1   The User enters Their Username and Password  1.2   The User presses the Login button  1.3   The System checks if the Username and Password match in the database    AF-1: The Username does not exist    AF-2: The Password is incorrect    The System logs the user in |
| Alternative Flows: | AF-1: The Username does not exist  1.     The System prompts the User that the Username does not exist.  2.     The User acknowledge and is taken back to 1.1  AF-2: The Password is incorrect  1.     The System prompts the User that the Password is incorrect.  The User acknowledge and is taken back to 1.1 |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | Nil |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC-03 | | |
| Use Case Name: | Modify Personal Particulars | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allows Helia Users to modify their personal particulars and avatar |
| Preconditions: | The User is logged in |
| Postconditions: | The User information is saved |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1.1   The user input updated password or email, or both.  AF-1: The user chooses to change avatar  1.2   The user presses the submit button  1.3  The system save the changes made |
| Alternative Flows: | AF-1: The user chooses to change avatar  1.1 The user chooses from the four avatar images provided  1.2 The user presses confirm button |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | Nil |

|  |  |
| --- | --- |
| Use Case ID | UC – 04 |
| Use Case Name | Discover Activities |
| Use Case  Description | Allows Helia Users to discover various activities to promote a healthier lifestyle |
| Participating  Actor(s) | User, External data sets |
| Pre-Conditions | 1. The User is logged in |
| Post-Conditions | 1. A list of activities will be displayed |
| Normal Flow | * 1. The User presses on the Discover button   2. The System checks the UV Index, Pollutant Standard Index and generates a list of recommended activities.   3. The System displays the activities that would be appropriate for the user   AF-1: The User presses on Recommend Venues  AF-2: The User presses on Healthy Recommend Eateries |
| Alternative Flows | AF-1: The User clicks on Venue Recommendation   1. The flow continues in UC-05 Venue Recommendation   AF-2: The User clicks on Healthy Eateries Recommendation   1. The flow continues in UC-06 Healthy Eateries Recommendation |
| Exceptions | - |
| Includes | - |
| Alternative Flows: | Nil |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | Nil |

|  |  |
| --- | --- |
| Use Case ID | UC – 05 |
| Use Case Name | Venue Recommendation |
| Use Case  Description | Allows Helia Users to view recommended venue for the sports activity recommended to them. |
| Participating  Actor(s) | User, External data sets |
| Pre-Conditions | 1. This use case extends the UC-04 Discover Activities. It is initiated when the User wants to view recommended venue nearby. 2. The User is logged in 3. An activity has been recommended to the User in previous page |
| Post-Conditions | 1. Information about the recommended venue is displayed |
| Normal Flow | 1. User presses the “Recommend Venue” button on the page that displays the recommended activity 2. The system detects User’s current location 3. The system gets information about nearby available facilities/venues from external data sets from data.gov.sg. |
| Alternative Flows |  |
| Exceptions | - |
| Includes | Nil |

|  |  |
| --- | --- |
| Use Case ID | UC – 06 |
| Use Case Name | Healthier Eateries Recommendation |
| Use Case  Description | Allows Helia Users to view recommended healthier eateries. |
| Participating  Actor(s) | User, External data sets |
| Pre-Conditions | 1. This use case extends the UC-04 Discover Activities. It is initiated when the User wants healthier eateries recommended. 2. The User is logged in |
| Post-Conditions | 1. A map marked with locations of the nearby healthier eateries is displayed |
| Normal Flow | 1. User taps the “Recommended Eateries” button on the page that displays the recommended activity 2. The system detects User’s current location 3. The system gets information about nearby healthier eateries from external data sets from data.gov.sg. 4. The system displays a map marked with locations of the healthier eateries around the User’s current location |
| Alternative Flows |  |
| Exceptions | Nil |
| Includes | Nil |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC-07 | | |
| Use Case Name: | View Events | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allows Helia Users to view the upcoming events |
| Preconditions: | The user is logged in |
| Postconditions: | A list of events will be shown |
| Priority: | - |
| Frequency of Use: | - |
| Flow of Events: | 1.1  The user swipes right from the Discover Section to enter Event Section  1.2  The System generates the list of events in the database that have been approved by the admin  1.3  The User press on the Event that he is interested in viewing more details  AF-1: The User filters the list of events  1.4  The System displays the event details       AF-2 The User clicks on Interested  AF-3: The User clicks on Register |
| Alternative Flows: | AF-1: The user filters the list of events  1. The system displays a list of categories that the User can choose from  2. The user chooses which category the User is interested in  3. The system displays the list of event base on the category    AF-2: The user clicks on Interested button  1. The System adds the event to the user’s list of interested events  AF-3: The user clicks on Registered button   1. The system adds the event to the user’s list of registered events |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | Nil |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC-08 | | |
| Use Case Name: | Create Events | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allows Helia users to create events for other users to goto |
| Preconditions: | The user is logged in |
| Postconditions: | The event will be created |
| Priority: | - |
| Frequency of Use: | - |
| Flow of Events: | 1.1  The User presses on Create Events  1.2  The System redirects the User to the Create Events Page  1.3  The User enters the Information of the events and clicks on the Create button  1.4  The System validates the information  AF-1: There are invalid inputs  1.5  The System add the event to the database |
| Alternative Flows: | AF-1: There are invalid inputs  1.     The System prompts the User of the invalid input  2.     The User acknowledge and is taken back to 1.1 |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | The events will be later checked by admin and to decide whether it is to be approved or rejected. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC-09 | | |
| Use Case Name: | View Events created by the user | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allows Helia Users view the Events that they have created |
| Preconditions: | The User is logged in |
| Postconditions: | A list of events that the User have created will be shown |
| Priority: | - |
| Frequency of Use: | - |
| Flow of Events: | 1.1  The User presses on View My Events button  1.2  The system displays a list of events that the user has created  1.3  The user clicks on one of the events that he is interested in checking  1.4  The System displays the events details |
| Alternative Flows: | Nil |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | Nil |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC – 10 | | |
| Use Case Name: | View Interested and Registered | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allows Helia Users view all the Events that they are interested in or have registered for |
| Preconditions: | The User is logged in |
| Postconditions: | A list of events that the User have is interest in or have registered for will be shown |
| Priority: | - |
| Frequency of Use: | - |
| Flow of Events: | 1.1  The User clicks on Interested and Registered Events  1.2  The System redirects the User to the Interested Events page  1.3  The System displays the list of events that the User is interested in  AF-1: The User swipes right to see events he is Registered in  1.4  The User clicks on one of the events  1.5  The System displays the event details  AF-2: The user removes himself from the registered list for the event  AF-3: The user removes the event from his interested list of events |
| Alternative Flows: | AF-1: The User swipes right to see events he is registered in  1.1  The System displays the list of events that the User is interested in  1.2  The User clicks on one of the events  1.3  The System displays the events details  AF-2: The user removes himself from the registered list for the event  2.1  The User clicks on the Remove from Register icon  2.2  The System removes the event from the registered list in the database  AF-3: The user removes the event from his interested list of events  3.1  The User clicks on the Remove from Interested icon  3.2  The System removes the event from the interested list in the database |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | Nil |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | UC-11 | | |
| Use Case Name: | Events Moderation | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | Admin |
| Description: | Allows Helia Admin view and manage all the Events that has been created |
| Preconditions: | The user is logged in |
| Postconditions: | A list of events will be shown for the admin to moderate |
| Priority: | - |
| Frequency of Use: | - |
| Flow of Events: | 1.1 The Admin presses on “Events to be approved”        AF-1: The Admin presses on “Approved Event”  1.2  The System displays a list of events that has not been approved  1.3  The Admin presses on one of the events  1.4  The System displays the events details  1.5  The Admin presses on Approved        AF-2: The Admin presses on Reject  1.6  The System adds the event to the Approved list in the database |
| Alternative Flows: | AF-1: The Admin presses on “Approved Event”  1.     The System displays a list of events that has been approved  2.     The Admin presses on one of the events  3.     The System displays the events details  4.     The Admin presses on Remove  5.     The System removes the event from the database       AF-2: The Admin presses on Reject  1.     The System modifies the event to the rejected status in the database |
| Exceptions: | Nil |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | Nil |
| Notes and Issues: | Nil |