**Test Plan - EventHub**

1. **Purpose**

This test plan is to test the EventHub website functionality.

1. **Unit Testing**

The following unit tests will be performed on the modules:

**2.1. Follow user module:** This module allows users to follow other users.

**2.1.1. Equivalence classes for Follow user module:**

**2.1.1.1. The equivalence classes for item “follow”:**

**1.** click ‘follow’ - acceptable

**2.** anything else - no response

**2.1.1.2. The equivalence classes for item “unfollow”:**  
 **1.** click ‘unfollow’ - acceptable

**2.** anything else - no response

**2.1.2. Checklist:** The following black-box tests will be performed.

( ) Follow a user

( ) Unfollow a user

( ) Follow many users then exit

( ) Reopen and check the follow counts

**2.2. RSVP module:** This module allows users to RSVP to events.

**2.2.1. Equivalence classes for RSVP module:**

**2.2.1.1. The equivalence classes for item “RSVP”:**

**1.** click ‘attending’ - acceptable

**2.** click ‘attending’ again - nothing happens

**3.** click ‘not attending’ - acceptable

**4.** anything else - no response

**2.2.2. Checklist:** The following black-box tests will be performed.

( ) Click ‘attending’ on an event page

( ) Click ‘not attending’ on the same event page

( ) Click ‘attending’ on many event pages then exit

( ) Reopen and check the RSVP count on said event pages

( ) Check profile for attending list

( ) Click ‘not attending’ on the same event pages

( ) Check that everything is the same as it was in the beginning

**2.3. Invite user module:** This module allows users to invite other users to events.

**2.3.1. Equivalence classes for Invite user module:**

**2.3.1.1. The equivalence classes for item “invite”:**

**1.** click ‘invite’ on your event page - acceptable

**2.** choose which followers to invite - acceptable

**3.** anything else - no response

**2.3.1.2. The equivalence classes for item “response”:**  
 **1.** click ‘attending’ - acceptable

**2.** click ‘dismiss’ - acceptable

**3.** anything else - no response

**2.3.2. Checklist:** The following black-box tests will be performed.

( ) Invite a follower to an event

( ) Check if user got invitation

( ) Respond as user with attending

( ) Respond as user with dismiss

( ) Check if attending that the RSVP count is increased

**2.4. Create event module:** This module allows users to create events.

**2.4.1. Equivalence classes for Create event module:**

**2.4.1.1. The equivalence classes for item “Event Name”:**

**1.** 1-45 alphanumeric characters - acceptable

**2.** anything else - error message

**2.4.1.2. The equivalence classes for item “Event Date”:**  
 **1.** Mm/dd/yyyy - acceptable

**2.** anything else - error message

**2.4.1.3. The equivalence classes for item “Description”:**  
 **1.** 0-1000 alphanumeric characters - acceptable

**2.** anything else - error message

**2.4.1.4 The equivalence classes for item “Tags”:**  
 **1.** 0-3 tags 0-15 alphabet characters - acceptable

**2.** anything else - error message

**2.4.1.5 The equivalence classes for item “Private”:**  
 **1.** set private - acceptable

**2.** set public - acceptable

**3.** anything else - no response

**2.4.1.6 The equivalence classes for item “Location”:**  
  **1.** 6-50 alphanumeric characters - acceptable

**2.** anything else - error message

**2.4.2. Checklist:** The following white-box tests will be performed.

( ) Enter alphanumerical permutations for event\_name

( ) Enter alphanumerical permutations for event\_location

( ) Enter alphanumerical permutations for event\_description

( ) Enter values for event\_date in proper and improper notation

( ) Set event to private and test security

**2.5. Event Deletion module:** This module deletes an event.

**2.5.1. Equivalence classes for event deletion module:**

**2.5.1.1. The equivalence classes for item “delete button”:**

**1.** delete button - acceptable

**2.** outside of button – no action

**3.** different button – error

**2.5.1.2. The equivalence classes for item “cancel button”:**

**1.** cancel button – acceptable

**2.** outside of button – no action

**3.** different button – error

**2.5.2. Checklist:** The following black-box tests will be performed:

( ) Click on delete event button.

( ) Cancel a delete event action.

( ) Click outside of the delete button.

( ) Click outside of cancel button.

( ) Check if event was deleted.

( ) Accept the deletion on confirmation popup.

( ) Cancel the deletion on confirmation popup.

( ) Drag confirmation window.

( ) Check to ensure users are no longer “following” deleted event.

**2.6. Event Search module:** This module allows users to search for events.

**2.6.1. Equivalence classes for event search module:**

**2.6.1.1. The equivalence classes for item “search query”:**

**1.** city, state, particular building, or full address – acceptable

**2.** other input – error

**2.6.1.2. The equivalence classes for item “sub categories”:**

**1.** item from dropdown list – acceptable

**2.** other input – should not occur, error

**2.6.2. Checklist:** The following black-box tests will be performed:

( ) Search for an event at a location that exists.

( ) Search for an event at a location at an invalid location.

( ) Search for an event at a location with no events.

( ) Search for an event at a location with many events.

( ) Select all the subcategories one-by-one.

( ) Click on the submission button to submit a search.

( ) Check organization of search results.

( ) Do multiple searches in succession.

**2.7. Event Feed module:** This module displays a feed of events happening nearby.

**2.7.1. Equivalence classes for event feed module:**

**2.7.1.1. The equivalence classes for item “feed”:**

**1.** display events - acceptable

**2.** other - error

**2.7.1.2. The equivalence classes for item “scroll”:**

**1.** up and down scroll – acceptable

**2.** left and right scroll – unacceptable, error

**3.** scroll during update - acceptable

**2.7.2. Checklist:** The following black-box tests will be performed:

( ) Click on events listed in feed.

( ) Follow events from the feed.

( ) Unfollow events from the feed.

( ) Scroll up and down the event feed.

( ) Scroll during an update to the feed.

( ) Attempt to scroll from left to right.

( ) Highlight event data with cursor.

( ) Attempt to drag an event out of the feed.

**2.8. Map module:** This module displays the location of an event on a map.

**2.8.1. Equivalence classes for map module:**

**2.8.1.1. The equivalence classes for item “map”:**

**1.** shows small surrounding radius of event – acceptable

**2.** any other display - error

**2.8.1.2. The equivalence classes for item “event marker”:**

**1.** placed on the site of the event – acceptable

**2.** placed anywhere else - error

**2.8.2. Checklist:** The following black-box tests will be performed:

( ) Click on map.

( ) Attempt to drag map around screen.

( ) Click on event marker.

( ) Highlight map address information.

( ) Attempt to zoom in/out of map.

( ) Attempt to zoom in on the event.

( ) Click and drag cursor in center map.

( ) Click and drag event marker.

( ) Click and drag on outside edge of map.

**2.9. User Creation module:** This module creates a new EventHub user.

**2.9.1. Equivalence classes for user creation module:**

**2.9.1.1. The equivalence classes for item “create button”:**

**1.** accept button - acceptable

**2.** outside of button – no action

**2.9.1.2. The equivalence classes for item “cancel button”:**

**1.** cancel button – acceptable

**2.** outside of button – no action

**2.9.2. Checklist:** The following white-box tests will be performed:

( ) Enter sanitary “username.”

( ) Enter unsanitary “username.”

( ) Enter empty “username.”

( ) Enter “username” with less than 6 characters.

( ) Enter “username” with more than 20 characters.

( ) Enter “username” 6 to 20 characters long.

( ) Enter empty “password.”

( ) Enter “password” with less than 6 characters.

( ) Enter “password” with more than 20 characters.

( ) Enter “password” 6 to 20 characters long.

( ) Enter “password\_confirm” that matches “password.”

( ) Enter “password\_confirm” that doesn’t match “password.”

( ) Enter sanitary “email.”

( ) Enter unsanitary “email.”

( ) Enter empty “email.”

( ) Enter valid “email.”

( ) Enter invalid “email.”

**2.10. Who will perform the tests:** Jay Gallagher and Josef Grzenda

**3.0 Integration Testing**

**3.1. Purpose:** The purpose of integration testing is to ensure that all modules of the project can effectively work together. This is important so that modules don’t unexpectedly cause complications with each other.

**3.2. Integration Testing Checklist:** The following tests black-box tests will be performed:

( ) RSVP to event then accept an invitation

( ) Search for existing and deleted events

( ) Delete a profile while hosting an event

( ) Create an event then change location

( ) RSVP then delete profile

( ) Change location of events and delete them

( ) Follow and unfollow users with invitations

**3.3. Who will perform the tests:** Jay Gallagher and Josef Grzenda

**4.0 System Testing  
 4.1. Purpose:** System testing is done to ensure that the EventHub system works as a whole.

**4.2. System Testing Checklist:** The following black-box tests will be performed:

( ) When user enters URL, the webpage appears.

( ) User can successfully log in.

( ) User can successfully sign up.

( ) Search functionality performs properly.

( ) User can create a new event.

( ) User can delete an event.

( ) User can view the event feed.

( ) The view feed loads properly.

( ) The view feed updates properly.

( ) Map functionality is integrated properly.

( ) Map functionality correctly shows location of an event.

( ) User can follow another user.

( ) User can RSVP to an event.

( ) User can invite another user to an event.

( ) All buttons/menus perform some action upon clicking.

( ) User’s changes to profile are successfully updated.

( ) No strange artifacts present upon loading pages.

( )Website does not freeze under expected acceptable load.

( ) Website is secure.

( ) Invitation system is implemented properly.

( ) Modules load properly upon refresh of page.

**4.3. Who will perform the tests:** Jay Gallagher and Josef Grzenda

1. **Acceptance Testing**

**5.1. Purpose:** Acceptance testing is done to ensure that the client’s requested functionality is present in the final product.

**5.2. Acceptance Testing Checklist:** The following black-box tests will be performed:

( ) Search functionality is present.

( ) Search functionality is easy to use.

( ) User profiles are present.

( ) Events are robust enough.

( ) User profiles are satisfactory.

( ) General display/presentation of website is acceptable.

( ) Website is easy to navigate.

( ) Users can easily find help if needed.

( ) Event feed looks satisfactory.

( ) Event feed performs acceptably.

( ) Website easily updatable for future changes.

( ) Enough events present at launch of website.

( ) Event creation and deletion is easy to use.

( ) Presence of map is not awkward.

( ) Event feed does not cause “lag” on webpage.

**5.3 Who will perform the tests:** Jay Gallagher and Josef Grzenda