**Event-Navigator**

Bandy Wang, Calvin Tang, Matthew Raday

**1.1 Functional Requirements**

ID: Input

IN-01 : The system will take in user’s username and password as login credentials.

IN-02 : New users can input their desired username and password when creating their account.

IN-03 : Users can add comments to a particular event.

IN-04 : Users can look up a particular organization by searching for a student org keyword.

IN-05 : Event Organizers can add/promote an upcoming event by providing the event name, date, location, photos and description. The location will be in the form of the building’s name and room number (if applicable).

IN-06 : Event Organizers can edit any preexisting event hosted by them.

IN-07: Event Organizers can delete any preexisting event hosted by them.

IN-09 : Users can submit a particular date that they would like to see any events on that day.

IN-10 : Users can click on the interactive map to view the event detail.

IN-11 : There will an option for submitting applications for becoming an Event Organizer by providing the organization they represent.

ID: Output

OUT-01 : Display the location of the upcoming events on a map.

OUT-02 : Display events descriptions , location, time, and photos.

OUT-03 : Display comments on an event left by users, which includes the name of the user, as well as the time the comment was added.

OUT-04 : Displays all the events of a particular student organization

OUT-05 : Displays information of any particular student org. This will include the club’s meeting times, contact number, location, and description.

OUT-06 : Display search results list of a user-entered query for when they want to look for a particular event or organization.

OUT-07 : Display can show all the events in a given date.

OUT-08 : Display user account creation confirmations and errors. The error will appear if the desired username already exists.

OUT-09 : Display a list of all student organizations

OUT-10 : There will be a list of all events for a given date displayed for the user. Users can scroll through this list to see those events.

**1.2 Non-Functional Requirements**

ID: Process

PRO-01 : All accounts will be stored in such a way that it will be still persisted after application is closed.

PRO-02 : All events and their respective descriptions will be stored in such a way that it will be still persisted after application is closed.

PRO-03 : Events that are added by any event organizers can be viewed and accessed in the same session without any downtime upon submission

PRO-04 : Comments that are added by a user can be immediately viewed by other users.

PRO-05 : The system will verify the location of an event in order to conform with GR-01

PRO-06 : The system will remove events once they have passed.

PRO-07 : Authentication of a user during the login process. The program will be able to verify that the user has an account based on their inputted username and password.

PRO-08 : The program will be a web application.

ID: Game Rules

GR-01 : Events supported by this application can only be located in the New Brunswick/Piscataway area.

GR-02: System will only be active during fall and spring school semesters.

GR-03: Only event organizers will be able to add/edit/delete events for their specific organizations

**2. Test Design**

|  |  |
| --- | --- |
| Test-Case ID | T01 (IN-01, OUT-11) |
| Purpose | Determine if a user can register an account |
| Pre-conditions | 1. The System should have the register (/register) page pulled up |
| Inputs | 1. Email, username, and password |
| Expected Outputs | 1. Account Registration Success Notification |
| Post-Conditions | 1. The System should bring up the Login (/login) page 2. The Registration Success Notification should appear 3. The Account information should appear in the database |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T02 (IN-02, OUT-12) |
| Purpose | Determine if a user can login into a pre-existing account |
| Pre-conditions | 1. An account should already exist in the database with valid email and password (“tempEmail”, ”tempPass”) (See T01) 2. The System should have the login (/login) page pulled up |
| Inputs | A query to get all accounts with email “tempEmail” and “tempPass”. |
| Expected Outputs | 1. Login successful notification |
| Post-Conditions | 1. The System should bring up the home (/home) page 2. The Log in Success Notification should appear |
| Design Technique | Boundary analysis |

|  |  |
| --- | --- |
| Test-Case ID | T03 (PRO-01, IN-02, OUT-12) |
| Purpose | Determine if an account persists in the database after the closing of the application (PRO-01) |
| Pre-conditions | 1. An account should have just been registered (See T01) 2. The System should have been turned off and then back on 3. The System should have the login (/login) page pulled up |
| Inputs | 1. Email, and password |
| Expected Outputs | 1. Login successful notification |
| Post-Conditions | 1. The System should bring up the home (/home) page 2. The Log in Success Notification should appear |
| Design Technique | Boundary analysis |

|  |  |
| --- | --- |
| Test-Case ID | T04 (PRO-03, IN-05, OUT-01, OUT-02, OUT-04) |
| Purpose | Determine if a user can add events |
| Pre-conditions | 1. A user is already logged into their account 2. Their account is registered as a Student Organizer 3. The user is at the Events page |
| Inputs | 1. A user selects the Add Event option on the Events page 2. event name, date, location, photos and description. |
| Expected Outputs | 1. Display the location of the upcoming events on the map. 2. Display events descriptions , location, time, and photos. 3. The event is displayed on the appropriate Organizations event page |
| Post-Conditions | 1. The event can be searched for using the event search function 2. All events are displayed on the event display page 3. The System should have experienced no down time during this process |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T05 (IN-03, OUT-03) |
| Purpose | Determine if users can add comments to a particular event |
| Pre-conditions | 1. User is at an event’s specific description page |
| Inputs | 1. User adds a comment, including its contents 2. The System automatically adds the users username/the timestamp |
| Expected Outputs | 1. Display comments on an event left by users, which includes the name of the user, as well as the time the comment was added. |
| Post-Conditions | 1. Comments can also be commented on by other users, but the comments of comments can not be commented on. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T06 (IN-10) |
| Purpose | To assert that every marker on the map will have their own respective description that will show the name of the event, location, time, and description |
| Pre-conditions | Must be in the home page, can be either logged in or logged out |
| Inputs | A click on any markers that are presented in the map displayed on the home page. |
| Expected Outputs | A pop-up with the marker’s name, location,time and description. |
| Post-Conditions | user will remain on the home page, login status will be the same |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T07 for IN-11 |
| Purpose | To make sure that registering users has the option to become an event organizer through registration. This ensures that the user will become on through VALID registration. |
| Pre-conditions | The user is not logged in, on the registration page, and has the checkbox to become an event organizer checked out. He should also have the registration form filled out with input email and input username not existing in the database. |
| Inputs | The submission of the registration form through a submit button. |
| Expected Outputs | The page will lead to the login page, as well as a registration successful notification. In the database, the account should exist with register credentials and the account is labeled as an event organizer. |
| Post-Conditions | The user will be on the login page. |
| Design Technique | Boundary analysis |

|  |  |
| --- | --- |
| Test-Case ID | T08 for IN-11 |
| Purpose | To make sure that registering users has the option to become an event organizer through registration. This ensures that the user will become on through INVALID registration. |
| Pre-conditions | The user is not logged in, on the registration page, and has a checkbox to become an event organizer checked out. He should also have the registration form filled out with either input email and input username existing in the database. |
| Inputs | The submission of the registration form through a submit button. There will be a submit request with an attempt to write in the database with the inputted fields. |
| Expected Outputs | The page will stay in register page, as well as a registration error notification. In the database, the account should NOT exist with register credentials and the account is labeled as an event organizer. |
| Post-Conditions | The user will be on the register page.  The database will remain the same. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T09 (PRO-05, IN-05, OUT-13, GR-01) |
| Purpose | Determine if the System will allow Events that aren’t in the New Brunswick/Piscataway area. |
| Pre-conditions | 1. User is at the Add Events page |
| Inputs | 1. event name, date, a location that is not within the New Brunswick/Piscataway area, photos and description. |
| Expected Outputs | 1. A Notification explaining that an inappropriate location was given |
| Post-Conditions | 1. The event is not added 2. The user is prompted for another location 3. The location field is left populated but marked as incorrect 4. All other fields on the add events page are left populated and unmarked |
| Design Technique | Equivalence Class |

|  |  |
| --- | --- |
| Test-Case ID | T10 (GR-03) |
| Purpose | To determine if only event organizers are able to add/edit/delete events for their specific organizations |
| Pre-conditions | 1. User is already logged into their account 2. User’s account is registered as a regular Student 3. There is no Add/Edit/Delete link options viewable |
| Inputs | 1. The User manually inputs the URLs that the Add/Edit/Delete links use |
| Expected Outputs | 1. An error message explaining that these URLs are not viewable |
| Post-Conditions | 1. The non-Student Organizer accounts should not be able to reach the Add/Edit/Delete URLs |
| Design Technique | Boundary Analysis |

|  |  |
| --- | --- |
| Test-Case ID | T11 (PRO-02, IN-05, OUT-01) |
| Purpose | Determine if an Event persists in the database after the closing of the application (PRO-02) |
| Pre-conditions | 1. An event should have just been created (See T01) 2. The System should have been turned off and then back on 3. The System should have the login (/login) page pulled up |
| Inputs | 1. Scroll to the Event’s location on the map |
| Expected Outputs | 1. Display the location of the specific Event on the map |
| Post-Conditions | 1. The event can be searched for using the event search function 2. The event is displayed on the event display page |
| Design Technique | Boundary analysis |

|  |  |
| --- | --- |
| Test-Case ID | T12 (PRO-04, IN-03, OUT-03) |
| Purpose | Determine if recently added comments can be seen immediately by users |
| Pre-conditions | 1. User A and B are at an event’s specific description page |
| Inputs | 1. User A adds a comment, including its contents 2. The System automatically adds the users username/the timestamp |
| Expected Outputs | 1. Display comments on an event left by users, which includes the name of the user, as well as the time the comment was added. |
| Post-Conditions | 1. User B, after User A has sent his comment, should be able to see User A’s comment without refreshing the page. |
| Design Technique | Boundary analysis |

|  |  |
| --- | --- |
| Test-Case ID | T-13 |
| Purpose | Ensure that the list of registered organisations can be queried and retrieved from the database. |
| Pre-conditions | Database will contain a number of preexisting organizations. |
| Inputs | A query will be made to the database that will retrieve all of the student orgs once there is a GET request in the organization page. |
| Expected Outputs | The result should be a table with all the student orgs as rows. |
| Post-Conditions | The results will be used to populate the table in /organization. |
| Design Technique | Boundary analysis |

|  |  |
| --- | --- |
| Test-Case ID | T-14 |
| Purpose | Ensure event organizations can delete their own hosted events |
| Pre-conditions | Database will contain a single event with a name that they would want to delete. |
| Inputs | A query will be made to the database to delete that event with the same name |
| Expected Outputs | The result should be a table with no rows as we would have deleted the only event with the queried name from ou database. |
| Post-Conditions | The database will have no events containing the same name of the event that we just deleted. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T-15 |
| Purpose | Ensure event organizations can edit their own hosted events. |
| Pre-conditions | Database will contain a single event named “testEvent” that they would want to edit. |
| Inputs | A query will be made to the database to edit that event named “testEvent. In this case, we will edit the event’s description.  We will then query to get all events name “testEvent” |
| Expected Outputs | The result should be a table a single row. That row will have the description of our updated description. |
| Post-Conditions | The database will have an event named “testEvent” that will contain the updated description. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T-16 |
| Purpose | Ensure event organizations CANNOT add events has the same name one that is already registered. |
| Pre-conditions | Database will contain a single event named “testEvent”. |
| Inputs | A query will be made to the database to add an event also name “testEvent”.  There will then be a query that should retrieve all events name “testEvent”. |
| Expected Outputs | The result should be a table a single row. That row will contain the information of the already pre existing event name “testEvent” and should not be overwritten. |
| Post-Conditions | The database will only contain one copy “testEvent” that was there first. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T-17 |
| Purpose | Ensure events that are past the current time will be removed from the database |
| Pre-conditions | Database contains an event e that is before the current time, This event will be named “pastEvent” |
| Inputs | A query will be made to get all events named “pastEvent” |
| Expected Outputs | The result should be a table with no rows. |
| Post-Conditions | The database will not contain an event name “pastEvent” |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T-18 |
| Purpose | Ensure that we can filter the events based on a certain date |
| Pre-conditions | Database contains 2 events that has event time set to different days. |
| Inputs | A query will be made to get all events that has an event time that is of the two already in the database. |
| Expected Outputs | The result should be a table with a row containing the event information of the event that has the same date as the one queried. |
| Post-Conditions | The database will remain the same. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T-19 |
| Purpose | Ensure an account can not be registered if its desired USERNAME is already in the database. |
| Pre-conditions | Database contains an account with username “testName” |
| Inputs | A query will be made to insert an account also named “testName”.  There will then be a query to get all accounts named “testName” |
| Expected Outputs | The result should be a table with a single row with the first account named “testName”. The row should not be overwritten after the insert attempt. |
| Post-Conditions | The database will remain the same. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T-20 |
| Purpose | Ensure an account can not be registered if its desired EMAIL is already in the database. |
| Pre-conditions | Database contains an account with email “testEmail” |
| Inputs | A query will be made to insert an account also containing the email “testEmail”.  There will then be a query to get all accounts with email “testEmail” |
| Expected Outputs | The result should be a table with a single row with the first account with email “testEmail”. The row should not be overwritten after the insert attempt. |
| Post-Conditions | The database will remain the same. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T21 (IN-04, OUT-06) |
| Purpose | Determine if a user can search for Events using the provided Keyword Search bar |
| Pre-conditions | 1. A user is already logged into their account 2. An event has been created, with a name “tempEvent” 3. The user is at the Events page |
| Inputs | 1. The user inputs the entire name of the previously created event into the search bar and selects “Search” 2. The user then inputs the “tempEvent” into the search bar, and selects “Search”. 3. This will run a query that will look for all events named “tempEvent” |
| Expected Outputs | 1. Return a table with a row containing the event name “tempEvent” |
| Post-Conditions | 1. Database should return the same |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T22 (OUT-06) |
| Purpose | Determine if a user can search for Organizations using the provided Keyword Search bar |
| Pre-conditions | 1. A user is already logged into their account 2. An organization has been created, with its name “tempOrg” 3. The user is at the Organizations page |
| Inputs | 1. The user inputs the “tempOrg” into the search bar and selects “Search” 2. This will run a query that will get all student orgs named “tempOrg”. |
| Expected Outputs | 1. Return a table with a row containing the event name “temporg” |
| Post-Conditions | 1. Database should remain the same, the organization table in the org page should display the results |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T23 (IN-09) |
| Purpose | Make sure Users can submit a particular date that they would like to see any events on that day |
| Pre-conditions | 1. A user is already logged into their account 2. 2 events have been created, each on different days 3. The user is at the Events page |
| Inputs | 1. The user swaps the filtered settings to “date” and selects the date of one of the Events |
| Expected Outputs | 1. The user should be able to see the single event on that date, but not any other events. |
| Post-Conditions | 1. The user should be able to repeat the process for the other event and find it, or repeat the process with a third date and be unable to see either date. |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T24 (PRO-07) |
| Purpose | Determine if the System verifies login credentials correctly. In this case, we will test using a VALID email but INVALID password |
| Pre-conditions | Within the database, there will be an account with email “tempEmail” and password “tempPass1” |
| Inputs | A query to search for all accounts with email “tempEmail” and password “tempPass2” |
| Expected Outputs | 1. Incorrect login information notification 2. The query should return a table with no rows |
| Post-Conditions | 1. The System should bring up the login (/login) page 2. The Incorrect login information notification should appear on the login page 3. The database will be the same |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T25 (IN-12) |
| Purpose | Make sure Users can submit a particular date that they would like to see any events on that day within the map function |
| Pre-conditions | 1. A user is already logged into their account 2. 2 events have been created, each on different days 3. The user is at the map function |
| Inputs | 1. The user swaps the filtered settings to “date” on the map and selects the date of one of the Events |
| Expected Outputs | 1. The user should be able to see the single event on that date on the map, but not any other events. |
| Post-Conditions | 1. The user should be able to repeat the process for the other event and find it, or repeat the process with a third date and be unable to see either date on the map. |
| Design Technique | Boundary analysis |

|  |  |
| --- | --- |
| Test-Case ID | T26 (PRO-08) |
| Purpose | 1. Make sure URLs that require you to be logged in redirect to the login page |
| Pre-conditions | 1. No one is logged in |
| Inputs | 1. The user, without logging in, attempts to access the home, events, and organization pages |
| Expected Outputs | 1. The user should only be prompted to login first |
| Post-Conditions | 1. The user should not be able to view any of these pages, instead being redirected to the Login page |
| Design Technique | Boundary analysis |

|  |  |
| --- | --- |
| Test-Case ID | T27 (OUT-05) |
| Purpose | To ensure that the event information such as time, location, and host organization can be acquired with just the event’s name |
| Pre-conditions | A database will contain an event named “eventName” and will contain a description “testDescription”, and “2020:01:01” |
| Inputs | A query that will retrieve all events name “eventName” |
| Expected Outputs | A row with an event named “eventName” and will contain a description “testDescription”, and “2020:01:01” |
| Post-Conditions | The database will be the same |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T28 |
| Purpose | To test the ability of getting all of a specified student org’s events with just the student org’s name. |
| Pre-conditions | A database will contain many events, some with student org as “tempOrg”, others with different student orgs |
| Inputs | A query that will retrieve all events with host student org as “tempOrg” |
| Expected Outputs | A table that contains all of events that are hosted by “tempOrg” |
| Post-Conditions | The database will be the same |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T29 (PRO-07, IN-01) |
| Purpose | Determine if the System verifies login credentials correctly. In this case, we will test using a INVALID email |
| Pre-conditions | Within the database, there will be an account with email “tempEmail1” and password “tempPass1” |
| Inputs | A query to search for all accounts with email “tempEmail2” and password “tempPass1” |
| Expected Outputs | 1. Incorrect login information notification 2. The query should return a table with no rows |
| Post-Conditions | 1. The System should bring up the login (/login) page 2. The Incorrect login information notification should appear on the login page 3. The database will be the same |
| Design Technique | Equivalence class |

|  |  |
| --- | --- |
| Test-Case ID | T30 |
| Purpose | To make sure that event organizers does not have the ability to edit events that are NOT hosted by them |
| Pre-conditions | A database will contain a event that are hosted by “eventHost1” and named “tempEvent”  However, the current event organizer is logged in is “eventHost2” |
| Inputs | The current event organizer will attempt to change the name of the event “tempEvent” to “tempEvent2”.  There will then be a query that will get all events in the database |
| Expected Outputs | A query should result in an event named “temptEvent”. It should not be changed from preconditions. |
| Post-Conditions | The database will be the same as pre-conditions. |
| Design Technique | Equivalence class |

**3. Tracability**

|  |  |
| --- | --- |
| Test Case Number | List of the Requirements Tested |
| T01 | IN-01, OUT-11 |
| T02 | IN-02, OUT-12 |
| T03 | PRO-01, IN-02, OUT-12 |
| T04 | PRO-03, IN-05, OUT-01, OUT-02, OUT-04 |
| T05 | IN-03, OUT-03 |
| T06 | IN-10 |
| T07 | IN-11 |
| T08 | IN-11 |
| T09 | IN-05, OUT-13, GR-01 |
| T10 | GR-03 |
| T11 | PRO-02, IN-05, OUT-01 |
| T12 | PRO-04, IN-03, OUT-03 |
| T13 | OUT-09 |
| T14 | IN-07 |
| T15 | IN-06 |
| T16 | IN-05 (attempting to add event that contains the same name) |
| T17 | PR-06 |
| T18 | OUT-07, OUT- 10 |
| T19 | OUT-08 |
| T20 | OUT-08 |
| T21 | IN-04, OUT-06 |
| T22 | OUT-06 |
| T23 | IN-09 |
| T24 | PRO-07,IN-01 (valid email + invaild pass with login) |
| T25 | IN-12 |
| T26 | PRO-08 |
| T27 | OUT-05 |
| T28 | OUT-04 |
| T29 | PRO-07,IN-01 (invalid with login) |
| T30 | IN-06 |