System and Unit Tests Report

Product Name: SlugLife
Team Name: Athlatex
Date: December 3, 2015

System Testing:

<u>**Sprint 1:**</u>

<u>User Story 4 from Sprint 1:</u> As a user, I want to see upcoming events so that I can be informed of what's happening on campus.

Scenario:
1. Start SlugLife app;
2. Select Events tab
3. User should now see

<u>User story 5 from sprint 1 backlogged became user story 1 from sprint 2:</u> As an admin, I want to post events so that students are informed of my event.

Scenario:
1. Start SlugLife app; Login screen should appear; type
   - name = <jujwong@ucsc.edu>
   - password = <password>
   - Press Login
   - User will now see the MyEvents view
2. Press "Add" button; New event addition page should appear; fill in lines
   - nameStr = <Event name>
   - collegeStr = <College>
   - dateStr = <##.##.####>
   - descrip = <Description>
   - Press submit
3. User will now see new event in MyEvents view

<u>User story 2 from sprint 2:</u> As a student, I want to sort posted events by college so that I can find events specific to a particular college

<u>User story 3 from sprint 2:</u> As a student, I want to sort posted events by date so that I know what is happening each day

<u>User story 1 from sprint 3:</u> As an admin, I only want UCSC affiliated users so that only students arrive at my event.

<u>User story 2 from sprint 3:</u> As a user, I want my password to be hidden when I sign-up and login so now one can see it over my shoulder.

Scenario:

1. Start SlugLife app; Login Screen should appear; select sign up; type
   - name = <jujwong@ucsc.edu> must be @ucsc.edu
   - password = <password>
   - password should be converted to black dots
   - password confirmation = <password>
   - password confirmation should be converted to black dots
   - press Sign-Up
   - Check email for verification; click link
   - User should see verification message that account is now verified
   - Return to app; login screen; type
   - name = <jujwong@ucsc.edu>
   - password = <pass>
   - password should be converted to black dots
   - Press Login
2. See events view
3. Click on tab-bar Colleges
4. User should now see the colleges listed alphabetically
5. Click on tab-bar Upcoming
6. User should now see the events listed with most recent first
7. Click on tab-bar New
8. User should now see the events listed with most recent post first

User story 4 from sprint 2: As an admin, I want to be able to edit and change my events, as I need the ability to update important information to keep students informed.

Scenario:

1. Open SlugLife app; Login screen should appear; type
   - username = < jujwong@ucsc.edu>
   - password = <password>
   - Press login
   - User should now see MyEvents page
2. Select a previously posted event
   - The events textbar field's page should appear
   - Update the nameStr, collegeStr, DateStr, or descrip
   - Press Done
3. User will now see the updated event in MyEvents page

User story 3 from sprint 3: As a student, I only want to go to events posted by verified event administrators so I know I am attending a legitimate campus event.

Scenario:

1. Open SlugLife app; Login screen appears; type
   - name = <jujwong@ucsc.edu> (with student privilege)
   - password = <password>
   - press Login
2. Events page appears
   - "Add" button is greyed out
   - Unable to add an event as a student user

User story 5 from sprint 3: As a user, I want to RSVP to events so I can receive notifications about events that interest me.

Scenario:

1.Open SlugLife app; Login screen appears; type:

- name = <jujwong@ucsc.edu>
- pass = <pass>
- press Login

2. Events page should display all events

- On a specific event press add to calendar

3. Open calendar app

- Event should be created on date of event with name and description

Unit Tests:

Unit tests (25 points): Include a file/directory named 'Testing' in your Git Repository. There should be details (can be in a separate file in the directory) provided by each team member about the module and the functional testing they have done. Each team member picks a module or module and lists the equivalence classes and the test cases selected to cover all equivalence classes.

Sean: Sign up
Justin: Login
Christina: Add Event
Daniel: Edit Event
Tim: Sorting Events Tab
Jake: Tab Navigation Controller

Sign up
This module allows a new user to sign up for an account, provided that they are affiliated with UCSC. The user must enter their UCSC email address in the appropriate text field, and the module checks that they entered an @ucsc.edu email address. The user also enters their new password in the following two text fields. The module checks that these strings are the same before registering the new password into the database. A successful sign up navigates the user back to the login screen. At this time Parse sends a verification email to the new user's UCSC email address. After clicking the link sent in the email, the new user can then go back to the application and login.

Equivalence Classes:

1.1 valid @ucsc.edu email address

1.2 not a valid @ucsc.edu email address

2.1 passwords match

2.2 passwords do not match

Test Case	Input	Expected Output	Pass/Fail
1.1x2.1	Valid @ucsc.edu email address and passwords match	Signup Success	**P**
1.1x2.2	Valid @ucsc.edu email address and passwords do not match	Signup Failure	**P**
1.2x2.1	not a valid @ucsc.edu email address and passwords match	Signup Failure	**P**
1.2x2.2	not a valid @ucsc.edu email address and passwords do not match	Signup Failure	**P**

Login

This module is logging into our application which functionally will connect our app to the database that stores all our information. The first thing our application does is open to the login screen. Thus in order to access the rest of the app, a successful login is needed. There are only a few cases which are possible when entering into the username text field and password text field. For the username, having already been registered it will be in our database's users, therefore it can only be in the system or not. For the password when a user registers they must enter password to associate with that particular username, so either the password entered will be associated with the user or it will not. There is only one case that allows a successful login, when a registered username is paired with the associated password.

Equivalence Classes:

1.1 Invalid username

1.2 Valid username

2.1 Invalid password

2.2 Valid password

Test Case	Input	Expected Output	Pass/Fail

1.1x2.1	Invalid username and Invalid password	Login Fail	**P**
1.1x2.2	Invalid username and Valid password	Login Fail	**P**
1.2x2.1	Valid username and Invalid password	Login Fail	**P**
1.2x2.2	Valid username and Valid password	Login Success	**P**

Add event

This module allows campus administrators to add their college events to Parse by taking them to a new view where they will be allowed to enter relevant data, while students will be excluded from adding events. When the administrator has clicked the "Add" button the view in the app changes, but if a student considers adding something their "Add" button is slightly lowlighted and when they do click on it nothing happens at all.

Equivalence Classes:

1.1 Has Admin status

1.2 Has Student status

2.1 Add button availability

Test Case	Input	Expected Output	Pass/Fail
1.1x2.1	Valid status and Add button pressed	Go to Add Event View	**P**
1.2x2.1	Invalid status and Add button disabled	Stay on current MyEvents view	**P**

Edit event

This module allows users with administrator privileges to edit the events they have already posted. It is accessed for each individual event by clicking on the event from the My Events page.

Equivalence Classes:
1.1 Edited Name
1.2 Original Name
2.1 Edited College
2.2 Original College
3.1 Edited Date
3.2 Original Date
4.1 Edited Description
4.2 Original Description

Test Case	Input	Expected Output	Pass/Fail
1.1x2.1x 3.1 x 4.1	All fields with updated information	Event updated with all new fields	**P**
1.1x2.1x 3.1 x 4.2	Updated name, college, and date with original description.	Name, college, and date update but description does not change	**P**
1.1x2.1x 3.2 x 4.1	Updated name, college, and description with original date.	Name, college, and description update but date does not change	**P**
1.1x2.1x 3.2 x 4.2	Updated name and college with original date and description	Name and College update but date and description do not change.	**P**
1.1x2.2x 3.1 x 3.1	Updated name, date, and description with original college	Name, date and description update but college does not change.	**P**

1.1x2.2x 3.2 x 3.1	Updated name and description with original college and date	Name and description update but college and date do not change	**P**
1.1x2.2x 3.1 x 4.2	Updated name and date with original college and description	Name and date update but college and description do not	**P**
1.1x2.2x 3.2x 4.2	Updated name with original college, date, and description.	Name is updated but college, date, and description do not.	**P**
1.2x2.1x 3.1x 4.1	Updated college, date, and description with original name.	College, date, and description are updated but name does not.	**P**
1.2x2.2x 3.1 x 4.1	Updated date and description with original name and college.	Date and description are updated but name and college do not change.	**P**
1.2x2.1x 3.2 x 4.1	Updated college and description with original name and date.	College and description are updated but name and date do not change.	**P**
1.2 x 2.1x 3.1 x 4.2	Updated college and date with orignal name and description.	College and date are updated but name and description do not change.	**P**
1.2 x 2.2x 3.1 x 4.2	Updated date with original name, college, and description	Date is updated but name, college, and	**P**

			description do not change.	
1.2 x 2.2x 3.2 x 4.1	Updated description with original name, college, and date.	Description is updated but name, college, and date do not change.	**P**	
1.2 x 2.1x 3.2 x 4.2	Updated college with original name, date and description	College is updated but name, date and description do not change.	**P**	
1.2 x 2.2x 3.2 x 4.2	Original name, college, date, and description.	Nothing changes	**P**	

Sorting Events Tab

This module allows a user to sort the all events page according to newest post to oldest, soonest event to most distant, and college in alphabetical order. It is the bar across the top of the all events page. The order the events are currently in is highlighted by yellow.

Equivalence Classes:
1.1 From Newest sort
1.2 From Upcoming sort
1.3 From College sort
2.1 To Newest sort
2.2 To Upcoming sort
2.3 to College sort

Test Case	Input	Expected Output	Pass/Fail
1.1x2.1	From newest sort, pressing newest sort tab.	Events stay sorted from newest to	**P**

		oldest.	
1.1x2.2	From newest sort, pressing upcoming sort tab.	Events are displayed in order of soonest to most distant	**P**
1.1x2.3	From newest sort, pressing college sort tab.	Events are displayed in alphabetical college order.	**P**
1.2x2.1	From upcoming sort, pressing newest sort tab.	Events are displayed in order from newest to oldest.	**P**
1.2x2.2	From upcoming sort, pressing upcoming sort tab.	Events stay sorted from soonest to most distant.	**P**
1.2x2.3	From upcoming sort, pressing college sort tab.	Events are displayed in alphabetical college order.	**P**
1.3x2.1	From college sort, pressing newest sort tab.	Events are displayed in order of newest post to oldest.	**P**
1.3x2.2	From college sort, pressing upcoming sort tab.	Events are displayed in order or soonest to most distant.	**P**
1.3x2.3	From college sort, pressing college sort tab.	Events stay sorted by college alphabetically	**P**

Tab Navigation Controller
This module allows student and administrator users to navigate between various view controllers. It is a bar that is constantly along the bottom of the app unless the user is signing-up, logging in, adding or editing a post.

Equivalence Classes:
1.1 From My Events
1.2 From All Events
1.3 From Map
2.1 To My Events
2.2 To All Events
2.3 To Map

Test Case	Input	Expected Output	Pass/Fail
1.1x2.1	From My Events page, pressing My Events tab.	Staying in My Events page.	**P**
1.1x2.2	From My Events page, pressing All Events tab.	Going to All Events page.	**P**
1.1x2.3	From My Events page, pressing Map tab.	Going to Map page.	**P**
1.2x2.1	From All Events page, pressing My Events tab.	Going to My Events page.	**P**
1.2x2.2	From All Events page, pressing All Events tab.	Staying in All Events page.	**P**
1.2x2.3	From All Events page, pressing Map tab.	Going to Map page.	**P**
1.3x2.1	From Map page, pressing My Events tab.	Going to My Events page.	**P**
1.3x2.2	From Map page, pressing All Events tab.	Going to All Events Page.	**P**

| 1.3x2.3 | From Map page, pressing Map tab. | Staying in Map page. | **P** |