

System and Unit Test Report

Team Name : *Calendar Application, CAP*

Date : November 30th, 2019.

System Test Scenarios

• Sprint 1

- **User Story 1** : As a client, I would like to see a calendar template on the website so I can select an event to sign up too.

Scenario :

- * User (client or admin) reaches default homepage and sees a calendar that allows for interaction, but has no functionality.

• Sprint 2

- **User Story 1** : As an user, I would like to create a user account so that I may sign up for events.

Scenario:

1. Visit web application site, select *Need a New Account?* hyperlink.
 2.
 - * firstname = <John>
 - * lastname = <Doe>
 - * username = <j.doe>
 - * password = <supercalifragilistic>
 - * email = <j.doe@wahoo.com>
 3. Press Enter or Create button.
 4. Either a new user successfully created prompted is displayed or the page is reloaded.
- **User Story 2** : As an admin/user, I would like to be able to login so that I can sign up and view events.

Scenario :

1. Visit home login website.
2. Enter the following :
 - * username = <j.doe>
 - * password = <j.doe's password>
3. Click login.
4. Either the page is reloaded because of some error or the calendar web application shows up, and contingent on whether you are an admin or not, will show certain features.

• Sprint 3

- **User Story 1** : As an administrator, I would like to add events to a calendar so that clients can choose to join.

Scenario :

1. Visit website, admin enters login credentials as such :
 - * username : <admin username>
 - * password : <admin password>
2. Submits form. Either the login is successful or the page is reloaded and all fields are empty.
3. A calendar with plus buttons will appear. Click on button corresponding to day of desired event creation.
4. An event modal/form will appear allowing for event information to be submitted, enter the following :

- * Title : <title name>
- * Location : <location of event>
- * Description : <description of event>
- * Begin Date : <Dec dd yyyy>
- * Start Time : <hh:mm am/pm>
- * End Date : <Dec dd yyyy>
- * End Time : <hh:mm am/pm>

5. Click Create Event button.

6. Will create the event and display it on the calendar application.

– **User Story 2 :** As an administrator, I would like to delete events from the calendar.

Scenario :

1. Visit website, admin enters login credentials as such :

- * username : <admin username>
- * password : <admin password>

2. Submits form. Either the login is successful or the page is reloaded and all fields are empty.

3. A calendar with plus buttons will appear with events if there exist any. Click on event title corresponding to the desired event for editing.

4. An modal/form will appear allowing for additional buttons to be selected.

5. Click Delete button.

6. It will update the calendar and remove with the given event.

– **User Story 3 :** As an administrator, I would like to edit events on the calendar.

Scenario :

1. Visit website, admin enters login credentials as such :

- * username : <admin username>
- * password : <admin password>

2. Submits form. Either the login is successful or the page is reloaded and all fields are empty.

3. A calendar with plus buttons will appear with events if there exist any. Click on event title corresponding to the desired event for editing.

4. An modal/form will appear allowing for additional buttons to be selected.

5. Click Edit button, a modal/form will appear with the following fields, similar to create event, enter desired updates.

- * Title : <title name>
- * Location : <location of event>
- * Description : <description of event>
- * Begin Date : <Dec dd yyyy>
- * Start Time : <hh:mm am/pm>
- * End Date : <Dec dd yyyy>
- * End Time : <hh:mm am/pm>

6. Click Edit Event button.

7. It will update the event with submitted information and adjust accordingly.

– **User Story 4 :** As an user, I would like to sign up for events that interest me.

Scenario :

1. Visit website, the client will enter login credentials as such :
 - * username : < username >
 - * password : < password >
2. Submits form. Either the login is successful or the page is reloaded and all fields are empty.
3. A calendar will appear with events if there exist any. Click on event title corresponding to the desired event to sign up for.
4. An modal/form will appear allowing for additional buttons to be selected along with displaying event information.
5. Click Signup button. User signs up to event.
6. Click Close.

Sprint 4

- **User Story 1 :** As an admin, I would like to retrieve all users signed up for an event for viewing purposes.

Scenario :

1. Visit website, admin enters login credentials as such :
 - * username : <admin username>
 - * password : <admin password>
 2. Submits form. Either the login is successful or the page is reloaded and all fields are empty.
 3. Admin clicks on some given event, and there is a spinner action occurring and retrieves either all users for the event or no users are signed up.
- **User Story 2 :** As an admin, I would like to be able to edit an event given there is a rescheduling of some sort.
This was a story from the previous sprint that had some low hanging fruit that needed to be finished.
 - **User Story 3 :** As a developer, I would like to add a field in the sign up page for admin, so the database can recognize it.
 - **User Story 4 :** As a developer, I would like to add a flag on the event, so user can tell if signed up to the event or not.
 - **User Story 5 :** As an user/admin, I would like to list the events of a current month in a list.

Scenario :

1. Visit website, admin/client enters login credentials as such :
 - * username : <admin username>
 - * password : <admin password>
2. Submits form. Either the login is successful or the page is reloaded and all fields are empty.
3. Admin/Client clicks on Agenda in top right corner, and lists all events of that given month.

Unit Tests

Frontend

There were Unit Tests done on the Frontend. Primary methods of testing were unit tests on the different reducers to make sure they were working properly. Any time the logic would change on the code the unit tests would notify that the reducers would no longer work, and had to be fixed. There was an attempt to create unit tests for our action creators but the problem with that is we are using asynchronous action creators which are much more difficult to get working. After spending about 2 days, 16ish hours of work on them, pivoted and gave up as that time could be put towards other unit tests. For what could not be unit test, implemented very basic console.log tests to make sure that information and data was properly being routed around correctly. By doing this, the frontend was able to fix a few variable naming bugs and it really

cleared up what was going wrong at first. After that, we made sure to click on everything possible, such as sign up for an event, edit an event, delete an event, create an event, etc. After all of this, we found a few bugs that unfortunately we will not be able to handle before turning in the project.

Backend

We attempted to create unit tests for the backend. For the more trivial functions not involving the main class where CORS was applied, we found we were able to write some unit tests, such as create event and user id's. Towards the end of sprint four, Alfredo was able to further isolate classes which allowed for more testability but ran out of time to create further unit tests for it. Our other main form of testing was exploratory. Code, test, review the output. We created shell scripts that allowed for a tester to execute the shell script with the needed parameters, such as proper/failing dictionary keys, associated values, etc. The return values would either be error responses or the retrieved values from the backend that interacted with the database. Overall, the team did a pretty great job on this project and on the testing and attempting to incorporate unit testing.