Incremental Testing and Regression Testing

CS408
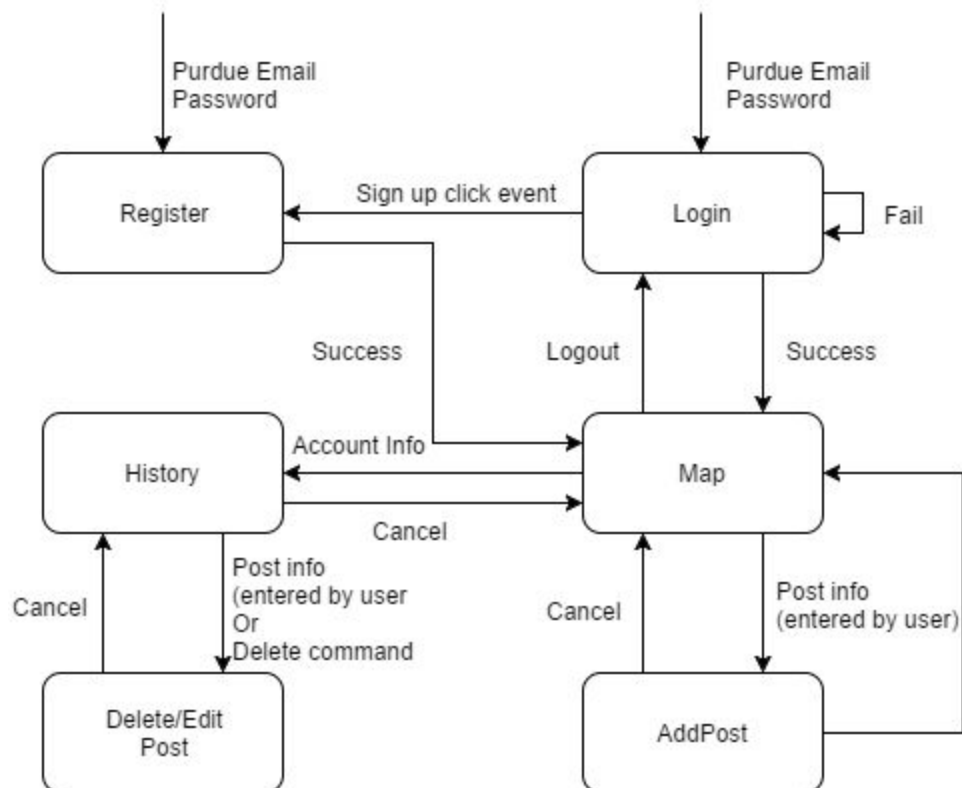
Team 20: Heng Li, Yiyan Zhou, Yibin Wu, Ketao Chang, Weichu Hu

The project is currently deployed at https://lofo-purdue.herokuapp.com
Right click to create a new post

1 Classification of Components

1.1 Define all components

1.2 Which form of incremental testing did you follow

• Top-down or Bottom-up? Explain.

We choose to follow Top-Down incremental testing. Because this test pattern basically follows the control flow. It's pretty straightforward and time-saving because it's easier to write stub modules than driver module.

2 Incremental and Regression Testing

2.1 Automation
In root directory, after install the packages required, input "npm test" in command line, the automated tests will be operated, there are total 13 test cases to check every functions used in LOFO. (This is in Dev branch by now.)

2.2 Defect log

Incremental Testing

Module: Login & Register			
Defect #	Description	Severity	How Corrected
1	After we launch our website with domain 'https://lofo-purdue.herokuapp.com' user can go to the main map page by using url even though the login or register process is not finished.	1	If not detect cookie, the website will navigate the guests to the login page when they try to go to map or history page.
2	After user click 'SIGNUP' button in login page, system would redirect to register page but the logo image won't load correctly.	3	Double check the logo image setting and correct the image file path.

Module: Post			
Defect #	Description	Severity	How Corrected
1	When submit a new post, if the type is not selected. After click 'submit' button, post should fail.	3	If not select a type, a pop-up will show up, telling users that they need select a type then submit.

Module: Map			
Defect #	Description	Severity	How Corrected
1	Angular 2 doesn't have official library support for marker clustering. So the group/ungroup function is not likely to implement.	1	Use google map wrapper interface, and implement our own script.

Module: History			
Defect #	Description	Severity	How Corrected
1	After updating an edited post, the history page should be reloaded and refreshed.	3	Added refresh function. Now the refresh event could be triggered when a post is updated.
2	After refreshing the page, the browser should be able to direct the website to the same page, instead of getting a 404 error.	2	Angular by default uses HTML5 pushstate. It request "http:localhost:3000/[route URL]" that does not exist. We map the corresponding file to the index.html by using "**HashLocationStrategy**"
3	A guest(user who does not log in) should not be able to access to the history page by entering url into the navigation bar directly.	1	If not detect cookie, the website will navigate the guests to the login page when they try to go to map or history page.

Module: Edit

Defect #	Description	Severity	How Corrected
1	After clicking the update button, a "success" pop-up should show up.	3	Add a function that pop an alert when click event is detected.
2	On the post pop-up the"item type" dropdown should be initialized with a selected type.	2	Add function to allow the communication between component. Now the type property of a post could be passed to this component.
3	After changing the lost item type in the dropdown and clicking the update button, the lost item type should be updated.	2	Fixed a bug that blocks the communication between frontend and backend.
4	If deletion or edition fail, a "fail" pop-up should show up, not the "success" pop-up.	2	Add a function to fetch the message sent by backend, notifying the user whether deletion and edition failed or passed..

Regression Testing

Module Map			
Defect #	Description	Severity	How Corrected
1	With the clustering script, the single (ungrouped) markers are not shown as the custom icons representing tags.	2	Instead of setting features of clustering objects generally in script, assign iconUrl attribute to markers when calling the wrapper interface.
2	With the clustering script, when click on the single (ungrouped) markers, the infoWindow will not appear.	3	Instead of using the official infoWindow, construct and assign infoWindow using the wrapper interface.
3	When click on the clusterer, a new post marker will appear which should not appear.	3	Set right click action to create a new post instead of single click

3 Update Product Backlog

Backlog Id	Functional Requirement	Hours	Status
1	As a user, I want to log in through Purdue email.	10	Complete in sprint 1
2	As a user, I want to create a post.	5	Complete in sprint 1
3	As a user, I want to upload the photo in the post.	10	In progress: move to sprint 2
4	As a user, I want to have description of lost item and the location found/lost in a post.	6	Complete in sprint 1
5	As a user, I want to have my contact information section in a post.	3	Complete in sprint 1
6	As a user, I want to add tags to a post.	3	Complete in sprint 1
7	As a user, I want to edit my post.	5	Complete in sprint 1
8	As a user , I want to delete an ongoing post and my personal contact information.	5	Complete in sprint 1
9	As a user, I want to see the lost and found items pin on a map.	15	Planned for sprint 2
10	As a user, I want to view either the lost or found post on the map.	4	Planned for sprint 2
11	As a user, I want to view items by tags on the map.	3	Planned for sprint 2
12	As a user, I want to view posts by specific created time interval.	3	Planned for sprint 2
13	As a user, I want to see a list of items when I click on a certain building.	3	Planned for sprint 2
14	As a user, I want to see details of a post when I click on a post in the list.	5	Planned for sprint 2
15	As a user, I want to see my own post history.	6	Complete in

			sprint 1
16	As a user, I want to verify the identity of each other when a lost and found matched.	8	Planned for sprint 2
17	As a user, I want to see all the solved and expired posts.	5	Planned for sprint 2
18	As a user, I want to report a falsely claim.	10	Planned for sprint 2