

# Sprint#4 Progress report

Team6

## 1. What we have done

- We implemented UI for 'main page', 'search page', and 'user info page'.
- We implemented 'signup' page with tests and improved UI using Material-UI
- We implemented 'login' to verify the user, store the userinfo and token. This token is used for each call of http request, so that only authenticated user can access, and modify.
- We made a frame of the list that shows travel plan overviews.
- We added edit/settings/delete option buttons to each overview blocks, when a user is in the user info page of himself/herself.
- Travel detail items such as transportation and restaurant are implemented to create travel plan.
- Assign some travel blocks to 'create travel' and modify their width and height.
- Adjust droppable and draggable wrappers to components for moving blocks smooth
- Set up backend server with django and MySQL.
- We studied about recommendation methods, and select what method to apply, design the architecture to apply.

## 2. What we are planning to do for next sprint

- We will implement UI for travel 'settings page' and 'edit user info page'.
- We will connect User backend to 'user info page' UI so that the page shows different buttons for owner and non-owner of the user page, assuming multiple users.
- We will add a link for each button to make our pages connected.
- The edit travel page which is similar with create travel page will be covered in next sprint.
- In this sprint, the create travel page was implemented but functionality detail is not added. To create travel with real data, connecting the google maps API should be added in next sprint
- We will construct backend model for 'travel' and connect related frontend with 'travel' model, adding proper tests.
- Triplannet service covers recommendation and searching travels. In next sprint we will add the tags to each travel to search the travel with keywords.
- Content-based recommendation will be implemented, and tested to check whether it gives appropriate result.

## 3. Not implemented features

- We made UI for 'main page', 'search page', and 'user info page', but we did not implement functions for operating these pages. We found that the backend model for travel should be designed with more detail, so we will implement functions for those pages after designing backend models.
- There are still some minor bugs in create travel page which can be used for drag and drop functionality. Before deploy the service we should add more tests to catch these bugs.

- We tried to implement backend model for 'travel', but it needs more consideration for architecture to provide 'commit' and 'merge' features. Thus, we chose to implement with tree structure and implementation is still ongoing.

#### **4. Organization issues**

- We plan to implement a fork, commit, merge functional for the sharing and editing the plan. It was hard to think appropriate backend design to deal with this. Also, it was hard to think simple, but well organized design for the user who is not familiar with this kind of functional.
- So we discussed it for a long time, and made a draft design. This design manage tree structure of commits(edit), and each commit have travel blocks IDs. We can deal with merge conflict by this commit history of travel block. Also, user can easily resolve it by block unit so it can be convenient.

#### **5. What tests did we prepare and not testing yet**

- As we have to show the demo of our project in the mid term presentation, we wrote the codes for tests that satisfies minimum test coverage to shorten development time. We did quite well in separating 'components' and 'containers', so we did write tests for rendering components. However, we didn't write firm test codes for features that are not completed and for container functions/branches. In the following sprint, we will write firm tests for all features we have implemented, as well as new test codes for newly-implemented features. For instance, we will write a test code for 'TravelOverviewBlock.js' considering different conditions such as whether a user is the author of this block, a travel is private or public, a travel is forked or not.
- Some custom components(e.g. Button.js, Responsive.js) should be replaced with material-UI components, so the test codes for these components are not necessary. We are planning to fix the overall UI of our project in the last sprint, and replacing and deleting custom components is not done yet, which lowers our test coverage. We will not write test codes for these components in the following sprint.
- Also, draggable and droppable wrapper were hard to test scenarios. Each wrappers has its style, action function before drag/drop, action function after drag/drop. It was hard to drag some components when test case and test functionality was also hard.

#### **6. Coverall Report**

Overall 93%

<https://coveralls.io/github/swsnu/swpp2019-team6?branch=master>

The tests for frontend containers such as 'createTravel' and 'HeaderContainer' will be added to lift up the coverage.