Demo Report

Team: 4_for_4

Member: Xijie Guo(xg2291), Haoran Geng, Yun Wang(yw3180), Jerry Lin

Date: 11/8/2018

Time: 17:00 pm EST

Description:

The demo was held by Kiran in CESPR for the first iteration for Advanced Software Engineering. We showed the function of our travel assistant, the post-commit test, and the Junit test coverage report to the TA. Our first iteration prototype allows user to select spots on the map as a list and get the travel recommendation from the system. We finished the UI prototype, google Maps API establishment, travel recommendation and, all the unit tests for these function.

Demo Challenges:

The TA and us found several bugs we need to fix and things to add for the second iteration:

1. When adding the unknown spots after adding some validate spots to the system, the system can not handle the error spots and add the old spots instead.
2. The system cannot handle long distance travel plan recommendation. (we do not need to handle long distance travel plan recommendation, but we need to handle if some user what to do such things)
3. The delete button can only delete the last element of the list. We need to change it to delete the element we selected.
4. The first spot that user choose is consider as a starting point. We need to change it that the system allow user to choose the start point they want.
5. Database used to persist data.

User Stories, Conditions of Satisfactions Demonstrated, and Explanation of Changes Since Revised Proposal:

The user stories we demonstrated today are quite different from those on the revised proposal as we found some of the previous user stories were not very reasonable while implementing functionalities of our application.

In revised proposal, the application can generate travel schedules for users by the questionnaires of preferences filled out by them in advance and then users can adjust the generated travel schedule. Later we found it is hard to map users' preferences to spots as their preferences are subjective and not standard. Therefore we decided to let users choose the spots they want to visit themselves and the application is responsible for helping them manage those spots and giving them suggestions on visiting order. Generating travel schedules for users will be based on some analytics data submitted by other users and is left for the 2nd iteration.

Today what we demonstrated specifically are:

1. The users can create a list containing all spots they want to visit in a day and manage the list themselves.
    a. A Google map is displayed on user interface so that users can search and find spots on Google Maps.
    b. Users are able to add those spots to the spot list.
    c. Users are able to delete spots from the spot list by deleting the last spot each time.
2. The users can know the correct order for visiting those spots in order to save traffic time.
    a. After adding all spots users want to visit, they click on a button to display spots in a correct order on user interface in terms of the shortest paths starting from the first spot and returning to the first spot finally.

CI:

Our application is developed under good industrial practices. Each of our team members has set up pre-commit in our repository to check three things: 1. Whether code compiles; 2. Whether all unit tests are passed; 3. Run the code through PMD, a static analysis tool. After pre-commit has finished, the code pushed to the repo triggers the CI server. We use Travis CI for continuous integration. Our Java project is built with Ant and have all the unit tests and compilation test set up by configuring build.xml. When the CI server starts, it first run "ant test" to run through all the unit tests and the compilation test. Then it runs PMD to check code smells and code style.

Technologies used are: git, bash, pmd, ant, Travis-CI

Project Github Link:

https://github.com/MousseKwok/Travel-Assistant