2nd Iteration Demo Report

Team: 4_for_4

Team members: Xijie Guo, Yun Wang, Jerry Lin, Haoran Geng

Date: 10/4/2018 Time: 3:00-4:00 PM

Description:

The demo was held by Kiran in CESPR for the second iteration for Advanced Software Engineering. We showed the function of our travel assistant to the TA. We now have our user stories fully implemented. We plugged in a database to persist data about user info and added login system for the application. Besides, we added new features on UI and backend algorithm that allow users to choose the different transportation and time or distance based travel plan. Users can save all the travel plans into the database and they are displayed on UI. The system can recommend three spots mostly visited by other users. We finished the UI, google Maps API, Mongodb database, authentication, 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. After input the invalid address, the system will keep reporting invalid address when inputting the correct address
2. The system uses location name as its location identifier, but since certain places have same names, the system failed to identify the desired location. Need to use the coordinates instead.

User Stories Demo-ed (changes from 1st iteration underlined):

1. As a traveler who doesn't want to spend so much time on route planning, I need a personal travel assistant to customize a travel agenda so that I can have a travel plan that can get me to selected places with least time spent on the road. My conditions of satisfaction are:
   - I can select spots I want to visit and the system will generate a travel plan for me.

2. As a traveler who prefers flexibility in transportation methods, I need a personal travel assistant to tailor the itineraries based on different modes of transport, so that I can have the quickest and the most flexible way to travel in the New York City. My conditions of satisfaction are:
    - I can obtain travel plans based on different modes of transport
    - I can prioritize the distance or travel time for my travel plan. Thus I can select the most suitable plan for me.
3. As a traveler who want to visit a lot of places, I need a system that can save all my potential travel plan, so that I can select the plan I want and do not need to fill in the travel information every time. My conditions of satisfaction are:
    - I can login the system with my username and password.
    - I can see my saved travel plan each time I login.

CI Mechanisms

Our CI mechanisms contain both pre-commit and post-commit, which are standard industrial practices. Each of our team members has set up pre-commit in our repository to do four things: 1. Check whether code compiles; 2. Check whether all unit tests are passed, prevent the commit if any unit test fails, and output unit test reports; 3. Check the unit test coverage via JaCoCo and outputs a report to report folder; 4. Analyze the code via PMD, a static analysis tool and generate reports.

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, coverage test, and compilation test set up in build.xml. When the CI server starts, it first run “ant test” to run through all the unit tests, coverage test, and the compilation test. Then it runs PMD to check code smells and code style. If any of the unit test fails on the CI server, all the team members will be immediately alerted.

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

Github Repo Link:

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