New Yorker: a Trip Planning Desktop Application for NYC

Team: 4for4
Haoran Geng (hg2496)
Jerry Lin (sl4299)
Yun Wang (yw3180)
Xijie Guo (xg2291)
Instructor: Prof. Gail Kaiser

Language & Platform
Java & Mac OSX

Summary
Travelling is fantastic. However, people have to spend a lot of time planning for a wonderful trip, including the time spent on looking for attractive spots, searching for the best route, planning for visiting time, and considering others' experiences. Therefore, we decided to design a trip planning application, New Yorker, to help people in NYC make reliable travel itineraries without spending too much time or energy figuring them out. In New Yorker, user only needs to provide their personal travel information and a travel plan with destinations, routes, time, cost and tips will be generated based on user's preference.

Detailed Project Description
Users will sign in with their username and password. Before starting a schedule, users can fill out a form about the general travel information including current location, date of arrival and departure, preferred transportation, budget, etc. After they predefine the overall travel information, users can proceed to select the local attractions where they want to visit and rank the sites by their priorities. Then the app offers multiple travel plans that are calculated from other users' travel plan and feedback. If a user is not satisfied with a particular travel plan,he or she can always make adjustment to the plan or select an alternative. Google Maps API will be used to fetch geolocation data, and NoSQL database will be used to store travel plans and user preferences.

Technologies
Development Framework/APIs: Swing/JavaFX, Morphia, Google Maps API
Static Analysis tool: PMD, Checkstyle
Logging/Testing tools: JUnit Test, Log4j2
Data Store: NoSQL (MongoDB)

User Stories

1. As a traveler who has limited experience and time, I need a personal travel assistant to customize a travel agenda based on my travel requirements so that I don't need to spend too much time researching myself. My conditions of satisfaction are:
    - I can fill out a form about the basic travel information and requirements provided by the application and obtain a feasible travel agenda based on the questionnaire after the application runs an algorithm.
    - I can select a travel schedule from a list of travel plans generated by the system, which are computed from existing schedules and user data.

2. As a long-time user of New Yorker who has enjoyed the auto-generated travel plans of certain cities/countries, I want to submit my feedback to the system so that other travellers can have a more comprehensive travel plan. My conditions of satisfaction are:
    - I can submit my feedback and changes of the plans back to the system for analytics, so that the system can better facilitate the trip planning for other users on the platform.

3. As a traveller who has high expectations on the incoming travel, I want an intelligent travel assistant to improve my present travel agenda for a better travel. My conditions of satisfaction are:
    - I can make some adjustments including changing the order of sites and removing or adding stops on the generated travel schedule based on the app's recommendation and my preferences.

Acceptance Testing

Case 1: user types into search bar to find a location.

- Expected behavior: search bar returns locations that are related to the user input, and recognizes invalid address inputs.
- Unexpected behavior: search bar returns unrelated location results.

Case 2: user clicks "my trips"

- Expected behavior: app returns trips previously generated for the user.
- Unexpected behavior: app returns other users' travel plans.

Case 3: when user tries to add a location to the list of locations he likes to visit, he clicks the location returned from the search bar

- Expected behavior: app adds trips to the list.
- Unexpected behavior: app doesn't add the location or adds another location.

Case 4: user clicks "generate trips" button after inputting a list of locations and preferences

- Expected behavior: app generates a list of feasible plans.
- Unexpected behavior: app generates a list of plans that are impossible to fulfill.

Case 5: user wants to edit an ongoing trip

- Expected behavior: app allows user to add/delete stops and save the updated version.
- Unexpected behavior: app doesn't allow user to commit these actions or doesn't save.

Case 6: user submits a New York travel schedule

- Expected behavior: app allows user to submit if the user did not submit before and notifies user whether it's submitted successfully.
- Unexpected behavior: app allows user to submit the same schedule unlimited times.