Hotel Management System

Team name: jigsaw
Shenghao Jiang (sj2914)
Wen Sun (ws2544)
Lingsong Gao (lg3018)
Cheng Shen (cs3750)

Part 0:

Language:Java
Platform: Mac

Part 1 Summary:

Hotel management is a critical part of the hotel operation chain. Our goal is to create an efficient, secure, and user-friendly hotel management system to provide excellent service to the customers while maximizing the profit for the hotel. Our hotel management system is an all-in-one application, which is designed to accommodate the needs of different staff members in the hotel. The front desk staffs are able to make the most suitable room arrangement for the customers. In addition, the front desk staffs may recommend the restaurant and suggest directions to places of interests to the customers. The hotel managers are able to view and analyze the profit and loss over certain period of time. We believe that our hotel management is capable of facilitating the service and helping the hotel to develop at a fast pace.

Interface: This is a desktop application and we will have a GUI which is demoable.
Data Storage: SQL (MySQL)
Framework : JavaFX
External API / Framework :
Google map direction api:
https://developers.google.com/maps/documentation/directions/start
Google timeline api :
https://developers.google.com/chart/interactive/docs/gallery/timeline
Google calendar api:
https://developers.google.com/chart/interactive/docs/gallery/calendar
Tableau JavaScript api :
https://onlinehelp.tableau.com/current/api/js_api/en-us/JavaScriptAPI/js_api.htm
Yelp api:
https://www.yelp.com/developers/documentation/v3
Tool: Github

Part 2 User Story:

Manager:

<1> : As a manager, I want to see the revenue so that I can evaluate the operation status of the hotel. My condition of satisfaction is that I can record daily profits and outputs and visualize our data for analysis and comparison.

Front desk:

<1> : As front desk, I want to be able to help customers check in and check out so that we have the right information. My condition of satisfaction is that I can check in and check out our customers on our software interface and the availability status of the room will be changed accordingly.

<2> : As front desk, I want to be able to tell the customers our available rooms and the prices, so that I arrange the most suitable rooms for them. My condition of satisfaction is that the information about our available rooms is correct and can be shown on our software interface.

<3> : As front desk, I want to be able to record requirements of customers and add other room service fees for our customers, so that customers can enjoy room service and hotels can make profits. My condition of satisfaction is that the information can be recorded on the interface and additional fees will be added to the total charge.

<4> : As front desk, I want to be able to recommend restaurants around our hotel, so that customers can go to desirable restaurants. My condition of satisfaction is that the information should include lists of restaurants according to some criterion.

<5> : As front desk, I want to be able to give information with respect to map directions and corresponding traffic routes, so that customers can easily transport to their desired places. My condition of satisfaction is instantaneous and the information should correctly answer customer’s questions.

Part 3 Acceptance testing:

Manager:

<1>: The sample input can be a record of rooms booked at a given date. If the inputs are recorded correctly, the expected result is the sum of all the prices of the booked rooms. If the result does not equal to the sum of all the prices of the booked rooms, the test fails.

Also, the input can be a file of daily profits in the database. If the inputs are recorded correctly, the correct output should be sum of profits of a chosen period of time. Also, the correct output can be represented as a diagram showing the profit change of that period of time. If the number or the diagram does not represent the profits of that period of time, then the test fails.

Front desk:

<1>: For checking in, the input is customer information including names, phone-number, room number and etc. If the input information has been correctly entered, then in the database we can see the information in the database, and the selected room will not appear in available room anymore. If any of the mentioned scenario appear differently, the test fails.

<2>: For providing room and price, we can test by requesting this information from the gui, the expected output is list of information which includes room type, room prices. If available room is not listed or unavailable room is listed, then the test would fail.

<3>: For adding requirement, first, we test by inputting the room number into the system and the correct name of customer would appear, otherwise it would fail. Then we add the requirement of the customer and the corresponding price to the system. If this step succeed, we can see the requirement in the database. Otherwise, it would fail.

<4> For recommending restaurant, when we make a request to the Yelp API, the correct response would be a list of popular restaurant coming from yelp, if that is not the case, then the test fails. Our input could be the current location like the name of the city we are in.

<5>:The sample inputs could be our current hotel location (and the desired destination). If the test is successful, we should get the desired direction and routes. If the test fails, either wrong or null information is retrieved.