First demo report
Hotel Management System

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

1. The date and time at which you already completed this demo, and briefly describe any challenges that arose during the demo.

Date: Dec. 7th,2018 1:30

Challenges: The demo is pretty smooth, we did not encounter any challenges.

2. The specific user stories and conditions of satisfaction that were demonstrated, with an explanation of any changes since your revised proposal.

For the first iteration, we have demonstrated the following user stories and conditions of satisfaction:

<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.

Changes: For this part, we have added that if the information we input is incorrect, then check-in or check-out will fail and will pop out a window indicating the failure.

<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.

Changes: For this part, we have added the following function: first we input the start and end date, then we present all the rooms that are available during that period of time.

<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.

Changes: For this part, we actually have two part, one is in the check-in part where we can add requirements in the form of comments, like “having pets”or “no smoking area”. And besides that, we have additional button for more requirements like room services.

3. A brief discussion of your CI mechanisms, including which technology you used.

For the CI part, we used Travis CI for the post commit test. We first set up the travis.yml file.Then we connect it to github repository.

Mechanism :Travis CI is configured by adding a file named .travis.yml, which is a YAML format text file, to the root directory of the repository.This file specifies the programming language used, the desired building and testing environment (including dependencies which must be installed before the software can be built and tested), and various other parameters.

And for the pre-commit, we applied the style-check for that with the Google_checks rule sets. We set up the pre-commit locally in the .git files.

For the pre-commit part, if there exists grammar faults like missing a “;”, the commit will fail, then we would have to modify the code and clean up all the error in order to commit and put the code to github.

After we have pushed the code with tests to github, since we have connected github with travis CI, then travis ci will run the test automatically.

4. A link to the github repository where your entire codebase resides. Tag the revisions that were shown in the demo.

The repo we used for demo is:

https://github.com/ChengShen1996/ASE/tree/iteration2-chaiquan

The repo we submitted with report is:

https://github.com/ChengShen1996/ASE/tree/iteration1-with-report