Queens College, CUNY, Department of Computer Science

Software Engineering

CSCI 370

Spring 2020

Instructor: Dr. Sateesh Mane

due date Thursday May 21, 2020 at 11:59 pm

2 Project 2b

- **Students work in teams for Project 2.**
- This is "Project 2b" and is the coding portion of the full Project 2.
- **Every student must be a member of a team.**
- Solo submissions will not be accepted.
- **Once a team is set, its membership may not be changed.**
- You may not switch teams (e.g. halfway through the project, etc.).
- *If any students drop out of the course, I will deal with the situation on a case-by-case basis.*
- **Please upload your submission (project report) via Blackboard, in a pdf or docx file.**
  1. Other document formats will not be accepted.
  2. Your file name must contain your name and student id.
  3. See Sec. 2.4 for details about your project report.
  4. You may submit update submissions up to the due date.
  5. **Only your final submission will be graded.**

2.1 Project description

- This project is to create a hotel reservation system and contains the following entities:
  1. Map
  2. Customers
  3. Hotels
  4. Search engine
  5. Current date (see next section)

Map

- The map is a web page which displays the names and locations of all the hotels in the project.
- The map is **read only.**
- All the hotels are located on an integer grid (i, j) where $0 \leq i, j \leq 100$.
- Therefore the "address" of a hotel is an (i, j) integer pair.
- **Two different hotels are not permitted to have the same (i, j) coordinate location.**
- The map is **publicly accessible.**
- A password or account is not required to view the map.
- There is no "map administrator" who controls the contents of the map.

Customer

- A customer must have a **password protected account.**
  1. A customer can pull up a read-only screen to display all his/her reservations.
  2. A customer **is not allowed to see the data for other customers.**
- A customer can book/cancel reservations for hotels.
  1. A customer can go to a search engine (= web page) to look up available hotels and room rates.
  2. A customer can also go directly to the web page of a hotel and book a reservation there.
  3. A customer can also cancel a reservation.
  4. A customer must go to a hotel website or the search engine to book or cancel reservations.
- Creation of customer account:
  1. When a customer attempts to book a hotel reservation, he/she will be prompted to input their account login information.
  2. This can happen either when using the search engine else at a hotel website.

3. If the customer does not have an account, instructions must be displayed to register for a new customer account.
4. If a customer account with the same name already exists, **a message mist appear that the account creation failed and the person must input a different name.**

Hotel

- A hotel must have a web page which displays its name and address, also instructions to view rooms and rates and book/cancel a reservation.
- The above web page is NOT password protected: anyone can view it.
- Web page for rooms and rates.
  1. This web page is NOT password protected: anyone can view it.
  2. A viewer can **read the room and rates information but cannot edit it.**
  3. *All room rates must be quoted in US dollars only.*
  4. Every hotel must have a list of rooms/suites with different rates.
  5. Every hotel must have a limited number of "cheap rate" rooms.
  6. Every hotel must have a limited number of expensive "deluxe" suites.
  7. **Availability:** If rooms/suites in a certain price range are sold out, the web page must say those rooms/suites are not available.
- Hotel reservations page.
  1. Customers can book (and cancel) reservations using a hotel's reservations page.
  2. This web page is NOT password protected: anyone can access it.
  3. If rooms in a certain price range are full, **customers cannot book reservations for those rooms.**
- Every hotel is a member of a chain and **has an administrator (for the hotel chain).**
  1. A "hotel chain" means there are multiple franchises of the hotel (different locations).
  2. The web page of a hotel must have a mechanism to **login as the administrator.**
  3. The administrator account is **password protected.**
  4. The administrator can view **all the customer reservations in the hotel chain.**
  5. The administrator can also view **all the customers who are currently checked into hotels in the chain.**
  6. The administrator can **add a new franchise to the hotel chain (= new hotel location.**
  7. The administrator must specify the location of the new hotel franchise.
  8. The administrator must specify number and types of rooms/suites and rates of the new hotel franchise.

9. **Two different hotels are not permitted to have the same (i, j) coordinate location, hence an error must be reported if the new hotel is placed at the location of an existing hotel in the map.**
10. After a new hotel is successfully created, **its name and location must automatically appear on the map.**

- **Different hotel chains have different administrators.**
- **A hotel administrator cannot access the data for a different hotel chain.**

Search engine

- Customers can go to a search engine (= web page) to look up information for hotels.
- The search engine is **publicly accessible.**
- A password or account is not required to access the search engine.
- There is no "administrator" who controls the search engine.
- Customers input selection criteria that your team thinks are relevant, and the search engine displays the results of the customer's query.
- Customers can sort the display by room rates or hotel names.
- The search engine must display how many rooms are available in the hotels that meet the customer's query.
- If rooms in a hotel (in a particular price range) are sold out the search engine display must say so.
- Customers can book room reservations (and cancel reservations) using the search engine.

2.2 Current date

- When I access your team's website, the first screen I see must ask me to enter an input date.
- I shall refer to this as the **current date.**
- The current date is a global variable available to all parts of your team's application.
- The value of the current date is set in the initial web page and cannot be changed in the rest of the application during a session.
    1. The value of the current date can only be set when I logout and login again.
    2. **Every time I exit the application and re-enter, the application must request me to input a value for the current date.**
    3. *The value of the current date should be displayed in all relevant screens in your team's application.*
- A hotel administrator can view all the customers who are checked into the hotels in his/her chain on the current date.
- A hotel administrator can view all the customer reservations for dates in the future (= later than the current date).
- This project does not support "historical data" in the following sense.
    1. If a customer booking has ended, i.e. the checkout date is in the past (= earlier than the current date), it should not appear in the hotel administrator's display.
    2. If a customer booking has finished, i.e. the checkout date is in the past (= earlier than the current date), it should not appear in the customer's account of hotel reservations.
    3. Only data on or after the current date should be displayed.

2.3 General information

- There is only one map.
- There is only one "FindHotels" search engine.
- There must be at least three hotel chains.
    1. Each hotel chain must have at least two franchises when I access your application the first time.
    2. I can login as a hotel administrator to add more hotels later.
    3. Your project report must include information for me to access the accounts of all the hotel administrators.
- **You may implement the project using any programming language, also any framework, etc.**
- This project requires **persistence across invocations.**
    1. "Persistence across invocations" means that all the data must be saved and must be available when I exit and access your team's application multiple times.
    2. For example if new user accounts are created they must be available when I exit an access your team's application again.
    3. Your team is free to decide how to implement persistence across invocations: use a database or text files, etc.

2.4 Project report

- **Place your team's application on a remote host.**
  1. I will not download or install anything on my computer or iPhone, etc.
  2. Applications which require me to install or download anything **will NOT be accepted.**
  3. Your team may employ the Mars/Venus server if that works for your team.
- Every student must submit a project report.
  1. **Your report must be a pdf or docx file.**
  2. Other document formats will not be accepted.
  3. **It is acceptable if all team members submit the same report.**
  4. Write the names of all the team members (including yourself) at the top of your report.
  5. Your report must contain **complete instructions how to access and run your team's application.**
  6. Your project report **must include information for me to access the accounts of all the hotel administrators,** so I can login as a hotel administrator (to add more hotels, etc.).
  7. Your report must explain how to navigate between different screens, e.g. how to access the search engine.
  8. A higher grade will be awarded for applications where the navigation is user-friendly, i.e. clear from the web pages themselves, without having to consult the project report document many times.
- **Your report must contain screen shots/images of all the web pages in your application,** with an explanation what each screen shot means.
- In other words, you must submit proof that your team implemented all parts of the project.
- Submissions which do not implement all the parts of the project will score a lower grade.
- **Your report must contain a section with a list of the following items.**
  1. Format of team meetings (online?), e.g. videoconference? email? Facebook? etc.
  2. Summary/schematic of the project architecture.
  3. Assignment of project responsibilities to team members, i.e. what did each person do.
  4. A list of the project goals/milestones.
  5. A timeline of progess for the completion of the project milestones.
  6. A list of unit tests and a timetable when each test was successfully passed.
  7. If necessary, what corrective steps were taken if the work fell behind schedule.
- **Do not submit program code in your report.**