**Dawson Hotel Reservation System**

**Project Specifications**

**Phase V**

**Due: Friday, December 9th**

In this last phase, you will build a text user interface and a GUI for the hotel reservation system. We will use the Model-View-Controller design pattern, so that the Model (which is our Hotel class) can notify various views (text-based and GUI) whenever its state changes.

**NOTE: You do not need to create separate test applications! You’re welcome ☺**

**Part 1 – Make the Hotel an Observable class**

Your Hotel class should already extend the Observable class: Hotel is our model and as such it will notify views when there are changes.

Add the setChanged and notifyObservers method invocations in the methods listed below; these are the only methods which will require the view to be made aware of a change in state. Note that the Object argument passed in the notifyObservers invocation will vary depending on the method:

createReservation-> invoke setChanged and notifyObservers(optionalReservation)

findCustomer-> invoke setChanged and notifyObservers(customer)

findReservations-> invoke setChanged and notifyObservers(List<Reservation>)

registerCustomer-> invoke setChanged and notifyObservers(customer)

updateCreditCard-> invoke setChanged and notifyObservers(customer)

**Part 2 – Simple Text User Interface (TUI) application**

We will now code a basic menu driven TUI application that will allow a user to connect and interact with the hotel database. This application will provide commands that reflect some of the DawsonHotel system. Refer to the given sample output run to see which commands must be implemented (page 4).

The additional text user interface types required:

1. The groupX.hotel.ui.TextView class: This class is your responsibility

* It must implement Observer.
* The constructor gets a handle to an Observable model and adds itself to the model’s observer list
* The update method public void update(Observable o, Object arg) must look at the type of the Object arg, and based on the type, display the appropriate information. Use private methods for each of the different displays. Note that when you are checking if the type of a generic type, you will need to use the wildcard syntax (i.e., if (arg instanceof Optional<?>), and you will get an unchecked warning when you cast e.g., (Optional<Reservation>) arg)
* Notice that the view to display a Customer will be the same, even though two different actions can cause it (i.e., a Customer is displayed both when you register and when you update credit card information)

1. The groupX.hotel.ui.TextController class (partial code provided in S:\CompSci\317\ReservationSys\code). Please review the provided code so that you understand it, and the additions that you must make (TODO). You will notice that some helper methods have been provided, you can use them or code your own. Notice that:

* it has the following private member:

private HotelManager model;

* The Command enum is used to determine the user’s request:

private enum Command

{

CUSTOMER\_INFO, REGISTER\_CUSTOMER, CREATE\_RESERVATION, RESERVATION\_INFO, CHANGE\_CREDITCARD, STOP

}

* it has the following constructor which sets the private variable:

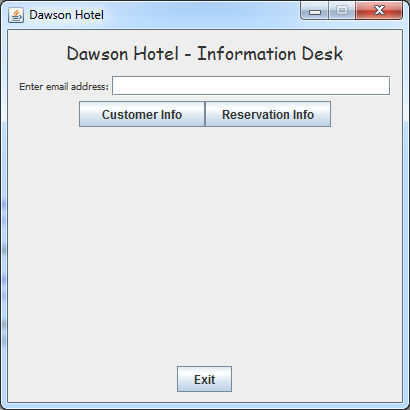
public TextController(HotelManager model)

* Notice that the controller will interact with a HotelManager -> this is the Model interface (Hotel implements HotelManager), but it is not Observable. Since the Controller is not observing the model, we can program to the interface.
* The method public void run() which is the main loop of the application. This method will:
* create and display the text menu
* get the user input and invoke the appropriate methods
* You are responsible for each method. You can use the private methods that I supplied to get specific types of user input. Each method must invoke the appropriate method on the model, and catch exceptions and display error messages. You don’t have to do anything with the returned objects from the model, since the view is observing.
* when the user chooses Stop, the hotel is closed, and all changes are persisted.

1. The groupX.hotel.ui.TextApp class is provided in S:\CompSci\317\ReservationSys\code. It instantiates the required objects and invokes the controller’s run method.

Execute the application to ensure that it functions as expected. Refer to the given sample output run (page 4).

**Part 3 – Graphical User Interface**



1. Due to the time constraints, the GUI application is simplified – we are taking care of the information screen, not the registration and reservation screens. I have provided you partial code of the groupX.hotel.ui.GUIViewController class. This takes care of the look&feel but has NO event handling or notification handling.

The GUI looks like this (sorry-not-sorry for the Comic-sans font), feel free to improve

1. Add the required event handling (for Controller functionality) and notification handling (for View functionality) (referring to lab 11 for an example). Changes to be made include:
2. make the GUIViewController implement Observer.
3. change the GUIViewController constructor so that it gets a handle to a Hotel model, and add a private instance field to keep a reference to the model. Within the constructor, also add this to the model’s observer list. Finally, control whether the resultPanel JPanel is visible. (use the setVisible method)
4. you need to listen for the events when someone clicks on a button. You can use anonymous inner classes or create new private inner classes that implement ActionListener to take care of each button event (your choice, in lab 11 we created inner classes)
5. Within the actionPerformed methods, all you will do is invoke the appropriate model method. Catch potential exceptions, and display a JOptionPane message dialog.
6. Within the update method, you will need to write results to the resultText, and ensure the resultPanel is visible.

Also refer to the screenshots on page 7.

1. The GUIApp is provided. Use this application to run your GUI.

**Submittal:**

- softcopy of zipped ReservationSys and CommonX

Dawson Hotel Menu

Select a choice from the menu:

1 - CUSTOMER\_INFO

2 - REGISTER\_CUSTOMER

3 - CREATE\_RESERVATION

4 - RESERVATION\_INFO

5 - CHANGE\_CREDITCARD

6 - STOP

Enter your choice: 1

Please enter the email address: raj@king.ru

Customer information

Name: Raj Wong

Email: raj@king.ru

Credit card: visa 4556737586899855

Dawson Hotel Menu

Select a choice from the menu:

1 - CUSTOMER\_INFO

2 - REGISTER\_CUSTOMER

3 - CREATE\_RESERVATION

4 - RESERVATION\_INFO

5 - CHANGE\_CREDITCARD

6 - STOP

Enter your choice: 2

Please enter the first name: jaya

Please enter the last name: thebest

Please enter the email address: j@thebest.com

Customer information

Name: jaya thebest

Email: j@thebest.com

No credit card on file.

Dawson Hotel Menu

Select a choice from the menu:

1 - CUSTOMER\_INFO

2 - REGISTER\_CUSTOMER

3 - CREATE\_RESERVATION

4 - RESERVATION\_INFO

5 - CHANGE\_CREDITCARD

6 - STOP

Enter your choice: 5

Please enter the email address: j@thebest.com

Please enter the card number1111111111111111

Select the credit card type from the menu:

1 - mastercard

2 - visa

3 - amex

Enter your choice: 1

Update unsuccessful! MasterCard must start with 5: 1111111111111111

Please try again.

Dawson Hotel Menu

Select a choice from the menu:

1 - CUSTOMER\_INFO

2 - REGISTER\_CUSTOMER

3 - CREATE\_RESERVATION

4 - RESERVATION\_INFO

5 - CHANGE\_CREDITCARD

6 - STOP

Enter your choice: 3

Please enter the email address: j@thebest.com

Customer information

Name: jaya thebest

Email: j@thebest.com

No credit card on file.

Enter the check in date.

Please enter the year : 2017

Please enter the month (1-12): 3

Please enter the day (1-31): 30

Enter the check out date.

Please enter the year : 2017

Please enter the month (1-12): 3

Please enter the day (1-31): 31

Select the room type from the menu:

1 - normal

2 - suite

3 - penthouse

Enter your choice: 3

Your reservation is for:

jaya thebest

at email: j@thebest.com

Check in on 2017-03-30

Check out on 2017-03-31

Your room: 801 is a penthouse on the 8floor

Dawson Hotel Menu

Select a choice from the menu:

1 - CUSTOMER\_INFO

2 - REGISTER\_CUSTOMER

3 - CREATE\_RESERVATION

4 - RESERVATION\_INFO

5 - CHANGE\_CREDITCARD

6 - STOP

Enter your choice: 4

Please enter the email address: raj@king.com

No customer found with that email address!

Please register, or try again.

Dawson Hotel Menu

Select a choice from the menu:

1 - CUSTOMER\_INFO

2 - REGISTER\_CUSTOMER

3 - CREATE\_RESERVATION

4 - RESERVATION\_INFO

5 - CHANGE\_CREDITCARD

6 - STOP

Enter your choice: 4

Please enter the email address: raj@king.ru

Customer information

Name: Raj Wong

Email: raj@king.ru

Credit card: visa 4556737586899855

You have 0 reservations:

Dawson Hotel Menu

Select a choice from the menu:

1 - CUSTOMER\_INFO

2 - REGISTER\_CUSTOMER

3 - CREATE\_RESERVATION

4 - RESERVATION\_INFO

5 - CHANGE\_CREDITCARD

6 - STOP

