Assignment 3

Modelling, Click Dummies, Software Processes

Assignment Due: Friday 11th October, 2020 @ 5pm

This assignment is worth 7.5% of your overall mark.
Please upload a **single ZIP file** containing all deliverables to the **assignment dropbox (adb.auckland.ac.nz)** before the due time. Failure to follow these instructions may result in your assignment not being marked!

Task 1: Analysis Data Model (35%)

A new travel agency "**Aroha Explore**" have approached you to create a *travel booking & sale management system software*. They have given you the following description:

The "Aroha Explore" travel agency needs a program to manage their customers, travel service providers, hotels, airlines, products, product specials, sales and employees. Customers have a name, email, optional address and contact number, and a credit card number. Different types of products are offered by the travel agency: hotel deals, flights, and package deals. All those products have a price.

Hotel deals are for a particular hotel and have a certain duration in days. Flights have a departure and destination location, and are offered by a particular airline. A package deal has a name and is organized by a travel service provider. It includes a hotel deal and a flight, which are also available as separate products. Travel service providers have a name, and offer at least one package deal. Hotels have a name and an address, and may or may not offer hotel deals at the moment. Airlines just have a name, and offer at least one flight each.

For example, the travel provider "Taehua Tours" offers the "Fiji Adventure" package, which costs only $999 and includes a cheap AirNZ flight from Auckland to Suva and a 6 days stay at the "Taehua Hut Resort" in Suva. Product specials are also products, but they refer to another product and offer it under a special price (usually less than the original price). A product special has a start and an end date, and is only valid during that period. A sale records the following data: the product that was sold, the customer who bought the product, the employee who sold the product, and the date on which the sale was done. That is, for each product a customer buys the system should record a separate sale. For each employee the system should store a name, email, contact number, IRD number and an address.

Draw an analysis data model (a class diagram with classes, typed attributes, associations with multiplicities and inheritance) with either UMLet tool, Papyrus, draw.io or any other tool you like. However, you must submit the diagram as a PDF file and optional .uxf file or working copy.

Task 2: User Interface Model (20%)

Aroha Explore have also given you the following requirements for the system's user interface:

The program for the travel agency should have an easy-to-use graphical user interface, which should work similar to this: the main screen should let the user choose between general activities: managing sales, managing customers, managing employees and managing products. Choosing managing sales opens a new screen, which shows a list of all the recent sales and lets a user add and delete sales. The user can also go back to the main screen.

The function for adding a sale opens a screen where the user can set all relevant information for the new sale (i.e. customer, product). The user can confirm the new sale or cancel a sale, and both lead back to the sales management screen. Sales can be deleted by choosing them in the list and invoking the delete function, which asks for confirmation before deleting the sale. If the product sold is a flight, and if the user tries to delete the sale, then the system should show a warning saying that cancelling the flight may incur a cancellation fee. Analogous to the screen for managing sales, there is a screen for managing customers. This screen shows a list of customers and offers the functions 'add' and 'modify'. The user can also go back to the main screen. Customer records are never deleted.

The add and modify functions open a new window offering input fields for the customer data. When adding a record, the fields are initially blank. When modifying a record, the old values are shown. The system refuses to accept customer records where the customer name is left blank. The system also rejects a record if a name is given, but the credit card number is invalid. In both cases, the system shows a warning and then the screen for modifying the record is shown again. The screen for managing products is similar to the screen for managing customers: it shows a list of all products and lets the user add new products. The user can also go back to the main screen. At this time, deletion or modification of the products does not need to be supported, and there need only be functions for adding hotel deals and flights. Depending on the product type, a new screen should be shown with input controls for the data that is necessary when adding a new product of the respective type. If the user confirms the addition but has not given all the necessary data, a warning is shown and the user returns to the screen where the data can be entered. Otherwise the user returns to the product management screen.

Draw a screen diagram for the user interface using pencil/pen and paper, then scan or photograph it. It is possible to separate the diagram into several sub-diagrams, as long as it is clear how the sub-diagrams fit together as a whole. Hand in the graphics files for the diagram. Only PNG, JPG or PDF files are acceptable.

Task 3: Click Dummy (35%)

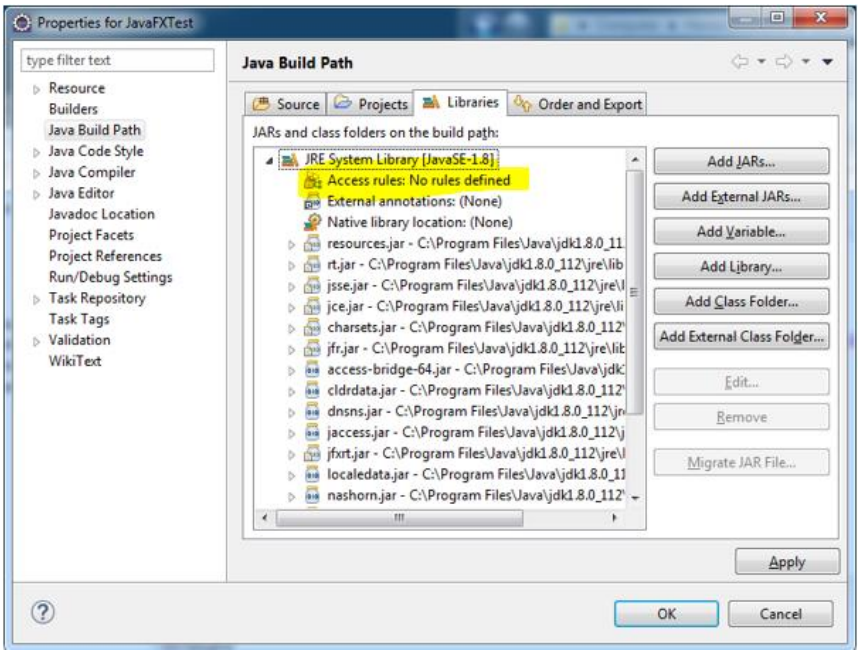
Create a click dummy UI prototype for the leasing company with Scenebuilder (JavaFX) or WindowBuilder (Java Swing) for Eclipse. The click dummy should be based on your screen diagram from Task 2. If there are case distinctions in the screen diagram (*e.g.* when sometimes a system warning is displayed and sometimes not), you only need to implement the common case. That is, when the user activates a control the click dummy will always react in the same way. Make sure that your click dummy compiles and runs without problems. **Then create runnable JAR file of your click dummy.**

Hand in the complete Java code or Eclipse project directory *(not the entire workspace, just the single Eclipse project)* of the click dummy program with runnable JAR file.

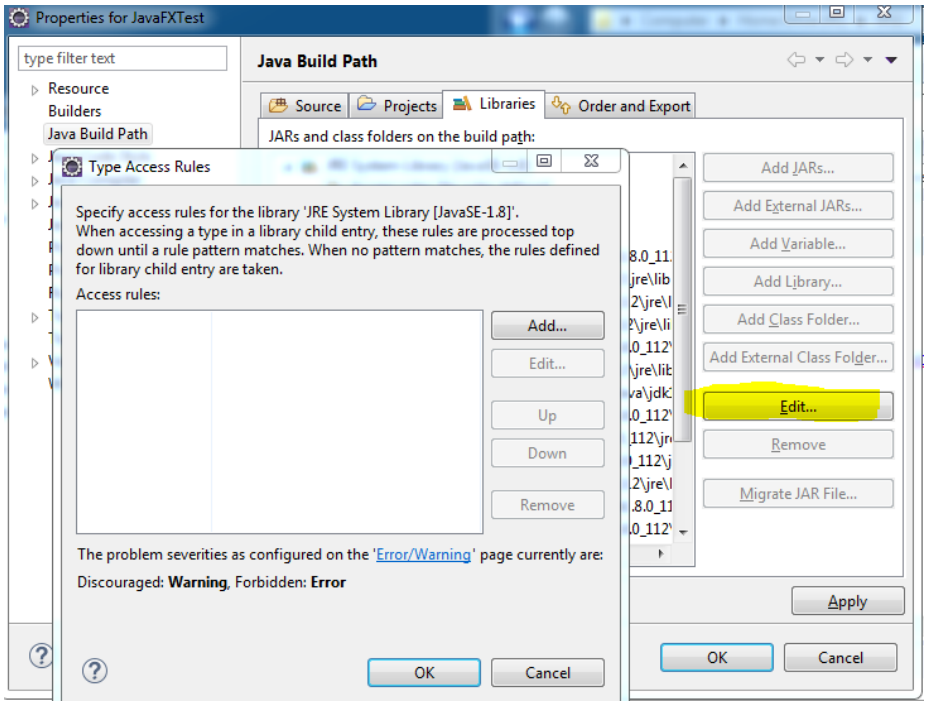
Scene-Builder:
Download Scenebuilder installer or executable JAR file (Java 1.8 version) from following URL http://gluonhq.com/products/scene-builder/

Install Scenebuilder on your computer. In Eclipse, make sure you have the option to create JavaFX projects. Refer to **Part 2 Lecture 4b** and **Lab 3** for more on setting up WindowBuilder or SceneBuilder.

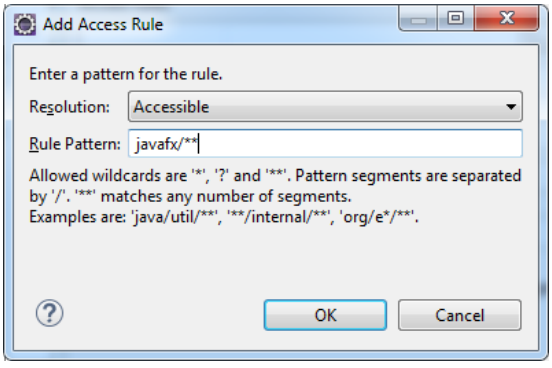
Note:
If you are using old version of Eclipse then JavaFX code might not work. One way to make it work is to add access rule to Eclipse project build path. Following are main steps for adding access rule:
Right click on *project >Build Path> Configure Build path> Libraries tab > select "Access rules"* as shown



Then click on *Edit*



Add new Access Rule *" javafx/**"* as shown below and click *OK*.



After adding this access rule your JavaFX code will work.

WindowBuilder:

Go to this link to find the Eclipse update site link for your Eclipse version:
https://eclipse.org/windowbuilder/download.php
Once you've got the link, open Eclipse and go to *Help* → *Install New Software* → *Add…* and paste the update site link in the Location text field. Install (at least) all of "Swing Designer" and "WindowBuilder Engine" except "WindowBuilder XML Core".

Task 4: Processes (10%)

Imagine that this (Aroha Explore) is a real project. How can a software development process make sure that the application will be of high quality? Briefly describe five concrete ways in which a process can help (no more than 3 sentences for each of the five ways).

Submit your answer as a Word or PDF file.

Submission Instructions

Please submit all necessary files to the assignment dropbox before the due date and time. The files should be submitted as a **single zip file** (containing four subfolders) with the following file structure:

- **<yourUPI_SE254_A3>.zip**
  - T1 *(folder)*
    - A single *PDF* file, and optional *.uxf* file optional .uxf file or working copy, representing your UML diagram.
  - T2 *(folder)*
    - One or more PNG, JPG, or PDF files representing your screen diagram.
  - T3 *(folder)*
    - Runnable Jar file with Java source code or Eclipse project folder for your click dummy. You'll know you've included the correct folder as it will have a *.classpath* and *.project* file, and a *src* folder in it (among other things).
  - T4 *(folder)*
    - A single Word or PDF document containing your answer to question asked in Task 4.

You must adhere to this submission structure. **Failure to do so may result in your assignment not being marked.**