DPIT121 - Object Oriented Design and Programming

Assignment III (due end of week 12)

This assignment is the last Iteration or Sprint of your work in DPIT121 and your third phase and final official delivery of the project in this session. However software is a live entity and you may keep working on this again and again as maintenance phase and to keep updating the software based on your customer's new requirements. For instance making it a web application, or adding a mobile application as a client. However, by end of phase three you should have a small Desktop GUI Application with Three Tier Architecture and some UML documentations as well as test cases and UI sketching. You can use this as a demo of your work if you want to apply for an internship programming job in particular if you have done standard or advanced for your assignments.

This assignment has three levels: advanced (max mark is 12 out of 12), standard (max mark is 9 out of 12), or core (maximum 6 out of 12).

Core Level: maximum 6% if completed in week 12

Standard Level: maximum 9% if completed in week 12

Advanced Level: maximum 12% if completed in week 12

Remember that you need to submit the project folder including all java files and forms as a zip file (*.java and *.form). In addition submit JAR file (The executable file) separately, and a zip folder including screenshots from all forms to show that UI is working as specified. Screenshots should demonstrate data entry, reporting, etc. Submitting only the project without separate JAR file and screenshots results in 0 to be allocated as your mark.

Core Level

You can start this from Assignment I solution provided. You also can use Swing Tool box. Your program will provide a swing based GUI.

- The program starts from a login form (Check the lab 7 jar demo). In lab 7 the user login was done. Now the login checks the login details if it is the admin it shows the admin form, if not and it is a user, it shows the user form if user validation is successful. In the case of unsuccessful validation, it shows the error message (the same as lab 7) ✓

User Form: Including several panels, menus, and forms for all the tasks and reports that a user is able to control. Minimum requirement is the same as lab 7. If you missed any menu, UI elements, or tab items, do it now for your submission. You may add new tabs/ menus to make the user form more user friendly and to do extra tasks compared with lab 7. **(3 marks)** ✓
List of use cases that has been done in Lab 7 is as below:

1. Edit user information. Current values are populated. ID and username are not editable. ✓
2. Issue a new policy (ThirdParty or Comprehensive) ✓
3. Find a policy by ID (you may have an autofill combobox of policy IDs instead of current textbox) ✓
4. Display a list of policies in a Table ✓
5. Let user to edit/update or delete a policy ✓
6. The option to sort policies ✓
7. Filter a list of polices based on car model ✓
8. Filter a list of policies based on expiry date ✓

Admin Form: You need to design a proper GUI for the admin page. You may use some ideas from Lab 7 but be creative. Admin menu includes several panels, menus, or forms for all the tasks and reports that the insurance company admin can control (Hint: all methods with admin login details but not limited to them) such as create/delete user, add/delete policies, reports, etc.). In particular do the following tasks but don't limit your GUI to this: **(3 marks)** ✓

Note that Admin can do all the requirements specified in user menu (1-8) but for a given user. You only need to extend all the panels from UserUI (the exact classes) and only add a combobox to the panel (or an autofill combobox which is even more user friendly) filled with all user IDs. Admin can ✓

choose the user from the list and the same panel and information that you had in UserUI will be displayed for the selected user. Watch the assignment explanation video for hints on this. ✓

1. Populate a list of all users in the company in a combo box for admin to choose a user from ✓
2. Issue a new policy for a given user. (You can use your panel from user form). Admin chooses a user from the combobox and then adds a policy. ✓
3. Find a policy for a given user (again show the list of users in a ComboBox) ✓
4. Display a list of polices for a given user ✓
5. Admin to edit or delete a policy for a user. ✓
6. Admin to be able to sort the list of polices for a user. ✓
7. Filter a list of policies based on car model for a user or all the policies in the system ✓
8. Filter a list of policies based on expiry date for a user or all the policies in the system ✓
9. Show a list of users in a table and the total premium payment for each user. Admin can sort the list of users based on name or total premium payment (by using Comparable or Comparator) ✓
10. Admin to generate the report **City Name – Total Premium Payment** ✓
11. Admin to generate the report **Car Model – Total Premium – Average Premium** ✓

In addition to project, submit screenshots from all of your forms and panels with data to show they work properly.

Standard Level

- Modify all the methods specified in lab 8 to Lambda and functional programming. In addition modify other data aggregation reports (Car Model – total premium payment and average premium payment for User and Company). **(1.5 marks)** ✓ Model array
City Hush

- Modify or write 3 methods from Assignment II Advanced range methods by using Lambda. You can select any three arbitrary methods from the list. **(1.5 marks)** ✓

PolicyCount – user
Policy car model – user
user count – Company

Advanced Level

In addition to using binary and text file for your data storage use a proper database such as Access or MySql to store the data (insurance company, users, and policies) as a proper Persistence Layer. You need to design your relational database and create several tables. In addition you need to develop a Data Access Layer in Java to read the data from the database and create objects and fill the ArrayLists and HashMaps. Use the sample codes provided to understand how to do it by using JDBC (Java Database Connectivity). **(3 marks)** **Refer to Week 12 lecture codes on JDBC, Database and Data Access Layer design** ✓

MySQL - Create Users - Save
Create Policies - Save

Submit your database files in addition to jar file and screenshots