GUI Programming

Module Title:	GUI Programming.
Assignment Type:	Individual or pair Practical Assignment.
Project Title:	Barber/Hairdresser Appointment System.
Project Date:	October 2020.
Assignment Compiler:	Amilcar Aponte.
Weighting:	50%
Due Date:	First submission: 7th November 2020 @ 23:59.
	Second submission: 12th December 2020 @ 23:59.
	Late submissions will be accepted up to 5 days after the deadline. All late submissions are subject to a penalty of 10% of the mark awarded.
Method of Submission:	Moodle Submission.
	No email submission will be accepted.

Module Learning Outcomes Assessed

- MLO1. Write computer programs with Graphical User Interfaces.
- MLO2. Utilise tools such as layout managers, GUI IDEs and containers to write robust GUI programs.
- MLO3. Develop event-driven, reactive GUIs.
- MLO4. Incorporate best practices in user design and usability.
- MLO5. Integrate non-GUI Object Oriented code with a user interface.

Assignment Introduction

A lot of the times when visiting a barber or hairdresser you have to wait in a queue not knowing when your turn is. To avoid this waiting problem, we are going to develop an automated booking platform for client and barber/hairdresser as there is currently no system to solve this issue.

Design and implement an application that allows barbers to register their business details, their location, and their available slots, so customers can login to search and book appointments.

General System Requirements

The system will have a backend database (DB) that will engine a desktop application with a GUI implementation. You are required to design and implement the system, including the database and the platform that will manipulate the data in it.

As the system will be used by a number of different people, different user account types must be created. Depending on the account type, different options in the system will be available to them. The two types of users will be:

Customer
Customers should be able to register themselves with their full name, mobile number, email address, and password. After successful registration, a customer should be able to log into the system and search for a barber shop, this could be filtering by name and/or location of the barber. Also, customers would book themselves for an appointment, once they have selected a suitable barber.

Service Provider (Barber/Hairdresser)
Service providers should register themselves by giving their full name, location, mobile number, and password. After successful registration, the service provider is able to log into the system and they can enter their free available slots, which will be updated and available for the customer to see. Also, there should be an option to confirm the booking that the customer has made.

System Requirements

- There should be a unique login section for customers and barbers/hairdressers to access the system. When a user opens the desktop application, they should be greeted with two main options:
    - Register on the system for a new account
    - Login to the system with an existing account
- All users should be validated against data stored in the database. Customers and service providers should be then directed to their appropriate home page if the login is successful.
- Storing passwords in plain text is a security issue that should be avoided. When customers and barbers register for a new account, the password that they enter should be hashed and salted.
- All input fields should be validated using client-side validation, checking for length and blank values, and any extra condition as needed.
- Your contractor has a cloud database server that you will be given access to. Your database deployment must be done to this server and your program must connect to it in order to perform all its activities. Details of the database server will be provided in due time.

Customer Requirements

- Customers who are going to be registering themselves must enter their:
    - Full name.
    - Phone number.
    - Email address.
    - Password.
- Customers who are already registered would use their email and password to login.
- When a Customer logs into the system, they should be presented with these main options on their homepage.
    - Search Barber/Hairdresser: by name or location to book an appointment. Once they have found and selected a particular barber/hairdresser, they can check available free slots.
    - View their own bookings: they should be able to cancel a booking if they wanted.
    - Place a complaint about a service provider.
    - Logout from the system: this should be done properly and not just closing the window/browser.

Service Provider Requirements

- Barbers/Hairdressers who are going to be registering themselves must enter their:
    - Full name.
    - Phone number.
    - Email address.
    - Location.
    - Password.
- Barbers/Hairdressers who are already registered would use their email and password to login.
- When a barber/hairdresser logs into the system, they should be presented with these main options on their homepage.
    - View a list of upcoming appointments: they should have an option to let the customer know that the booking has been received and confirmed.
    - Set availability: set specific slots as available or not available.
    - Set status of the appointment: if it has been completed or if the customer did not arrived.
    - Logout from the system: this should be done properly and not just closing the window/browser.

Notes

- **The use of code generators and GUI builders is NOT allowed.**
- You are allowed to use other tools apart from the ones seen in class, as long as it is your own code. So feel free to explore other types of Layout Managers, Listeners and/or Components for your programme.
- Try to keep it simple. Planning your program before start coding is part of your assessment.
- Comment your code!!
- Plagiarism will not be tolerated. All work must be your own. If you used some snippet of code from an external source, make sure that you reference it correctly inside your code.
- In any situation, the lecturer is entitled to call you in for further explanation of your code.
- **Your code must run as no debugging will be done.**

Deliverables and Deadlines

First submission (15%) – 7th November 2020 @ 23:59.
One PDF document in report format that includes:

- Wireframes of your full application.
- Logic and graphic design justification of your application.
- If you do the assignment as part of a pair, you need to include in your report a page explaining the contribution of each member of the team and what parts of the development they'll be in charge.
- Link to public GIT repository.

Second submission (35%) – 12th December 2020 @ 23:59.
You must submit through Moodle a zip file named after your student number that contains:

- Your full Eclipse/NetBeans project with all source code.
- If external library are used, the JAR file for them must be included as part of your project.
- TXT file with link the GIT repository.

Marking Scheme Summary

Description	Weighting
First Submission (15%)	
Wireframes. Layouts and design are clear and user friendly. All functions have been considered in the design.	12
Logic and graphic design have reasonable justifications.	15
Git repository has been set up correctly and both members are added as contributors in case of team projects.	3
Second Submission (35%)	
Login section works correctly and redirects users to their corresponding dashboards. Code is well structured and properly commented.	5
All functions for each customer users are present on the dashboard and work correctly. Code is well structured and properly commented.	20
All functions for each service provider users are present on the dashboard and work correctly. Code is well structured and properly commented.	30
The programme interacts with the database correctly. Code is well structured and properly commented.	10
Logout is available and redirects users to the login page. Code is well structured and properly commented.	5
TOTAL	**100**

Assessment of Individual Contribution (where assignment is completed in pairs)

Each group member inherits the overall group project mark. This mark is then adjusted by their individual contribution.

Individual contribution will be assessed by reference to your individual contribution report (first submission) and GIT repository (second submission).

For example:

- Final group result: 76%
- Student with perfect contribution receives: 100% * 76% = 76%
- Student with just sufficient contribution receives: 50% * 76% = 38%

It is important therefore that (if you are completing this assignment as a pair) both of you need to ensure that you are contributing to the coding work!