

SALON RESERVATION SYSTEM

GROUP NAME: ReserveIT

TEB1043 – Object-Oriented Programming

May. 2024

Submitted by

24000182

NURUL HANIIZATI BINTI HAZLI

24000574

SARA EUDORA BINTI SAID

22006373

DANIA ADRIANA BINTI MOHD FAIZAL

24000918

ZULAIKHA BINTI MOHD AZHAR

22005979

RAILYNN NISHALIA BINTI RHYMEI

Submitted to

Dr M Nordin B Zakaria

CIS Department

Universiti Teknologi PETRONAS, Malaysia

CONTENTS

No.		Page
1.	PROJECT TITLE	3
2.	PROJECT OBJECTIVES	3
3.	MARKET POTENTIAL	4
4.	PROJECT DETAILS	5
5.	PROJECT CODE	6
6.	USER MANUAL	7
7.	PROJECT UML DIAGRAM	14

PROJECT TITLE

Salon Reservation System

PROJECT OBJECTIVES

1. Develop a User-Friendly and Efficient Reservation System

Design a user-friendly client application for customers to easily find and select an available stylist based on their preferences, choose a convenient time slot that fits their schedule, browse and select from a variety of services including haircuts, hair coloring, hair styling and facials, enabling customers to conveniently manage their reservations. This application should provide a seamless and intuitive user experience to enhance customer satisfaction and streamline the reservation process.

2. Leverage Advanced Java Programming Techniques and Best Practices

Implement a modular and maintainable codebase by applying fundamental object-oriented programming principles. Moreover, it is to enhance user interaction and experience by creating a visually appealing and intuitive graphical user interface (GUI).

3. Provide Valuable Business Insights

Through comprehensive reporting and analytics, the system offers salon owners and managers insights into client preferences and business performance. These data-driven insights help in making informed decisions that can drive business growth and success.

MARKET POTENTIAL

Market Demand

An efficient and convenient system that will help salons improve appointment management and reduce manual customer entries. Furthermore, it will increase gadgets penetration with online booking trends to minimize customer's time spent in a salon while enhancing customer amenities.

Target Market

Our system focuses on hair and beauty salons where it is a business that heavily relies on appointments and client schedule management. The need for a reservation system is to predict customers traffic throughout the day, allowing for better staff scheduling, inventory management, and increased revenue.

The Salon Reservation System is designed to cater to a diverse range of beauty and wellness businesses, including independent salons, salon chains, and franchises. It serves small to medium-sized salons seeking to streamline their booking process, as well as large multi-location chains requiring centralized management. The system is ideal for mobile and home-based stylists who need a flexible scheduling solution, as well as full-service spas and day spas looking to manage a wide array of services efficiently. Additionally, it supports beauty schools and training centers in managing student appointments, hotel and corporate office salons in coordinating bookings with existing systems, and event and wedding planners in organizing beauty services for special occasions. By addressing the specific needs of these various segments, the Salon Reservation System enhances operational efficiency, client satisfaction, and overall business performance.

PROJECT DETAILS

Salon Reservation System will operate in the following procedure:

Input: Customers can input their reservation details such as the date, time and preferred stylist. Customer details can also be inserted such as their name, contact, email and type of services. Additionally, employees have the ability to update their availability.

Process: The program has a user-friendly and minimalistic design to ensure straightforward navigation and usability for users for all technical skill levels. Customer details and submitted information will be meticulously processed and stored in the database upon submission. The system supports adding, removing, searching, and printing customer details. Staff can manage and supervise reservations, to meet customers' needs and regulate stylist availability as required.

Output: A list of client information, reservations and stylist information will be displayed. A confirmation message will also be displayed on the window saying that the data has either been successfully added or removed. Additionally, an option to cancel the reservation is provided.

Database: Stores persistent data related to clients, stylists, and reservations. The MySQL database is accessible through **phpMyAdmin** for easy management and administration.

Running The System: To run the Salon Management System, the BlueJ IDE is used.

Follow these steps to set up and run the project:

1. Open BlueJ: Launch the **BlueJ IDE** on your computer.
2. Open the Project: Go to **Project -> Open Project** and select the directory containing the Salon Management System source code.
3. Compile the Project: Click on **Compile** to compile all classes in the project.
4. Run the FrontEndClient: Right-click on the **FrontEndClient** class and select **Run JavaFX Application** to run the application. This will launch the user interface, allowing you to interact with the system.

PROJECT CODE

Key Classes:

Client: Represents individual clients of the salon. Stores client-specific information such as name, contact details, and appointment history.

Methods: getName(), getPhoneNumber(), getEmail()

Clients: Manages a collection of Client objects. Provides methods to add, remove, print, and find clients.

Methods: addClient(), removeClient(), findClient()

Reservation: Represents an appointment or booking between a client and a stylist. Stores details such as appointment date, time, service type, and status.

Methods: getDate(), getTime(), getServiceType(), getClientName(), getPhoneNum(), getEmail()

Reservations: Manages a collection of Reservation objects. Provides methods to add, remove, print, and find reservations.

Methods: addReservation(), removeReservationByName(), findReservationByName(), printReservations()

Stylist: Represents individual stylists working at the salon. Stores stylist-specific information such as name, specialty, and availability schedule.

Methods: getName(), getRate(), getReservation(),

Stylists: Manages a collection of Stylist objects. Provides methods to add, remove, print, and find stylists.

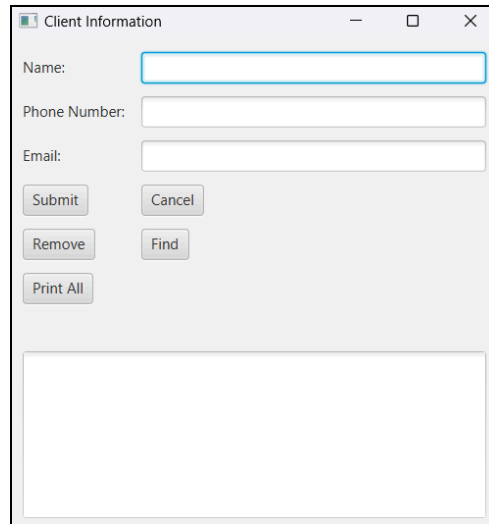
Methods: addStylist(), removeStylist(), findStylist(), printStylist()

USER MANUAL

Add

Client Information:

1. The customer provides their name, phone number, and email address, then confirms the information by clicking **Submit**.

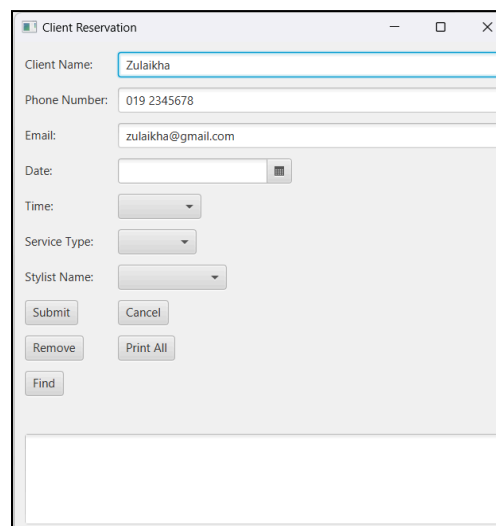


A screenshot of a software window titled "Client Information". It contains three text input fields labeled "Name:", "Phone Number:", and "Email:". Below these fields are five buttons: "Submit", "Cancel", "Remove", "Find", and "Print All". At the bottom of the window is a large, empty rectangular area, likely for displaying a list of clients.

Client Information Interface

Client Reservation:

1. The customer enters the desired date, time, service type, and stylist for their appointment and then confirms their selection by clicking **Submit**.



A screenshot of a software window titled "Client Reservation". It contains several input fields: "Client Name:" (with the value "Zulaikha"), "Phone Number:" (with the value "019 2345678"), and "Email:" (with the value "zulaikha@gmail.com"). Below these are fields for "Date:" (with a calendar icon), "Time:" (with a dropdown arrow), "Service Type:" (with a dropdown arrow), and "Stylist Name:" (with a dropdown arrow). At the bottom left are four buttons: "Submit", "Cancel", "Remove", and "Find". At the bottom right is a button labeled "Print All". At the very bottom of the window is a large, empty rectangular area for displaying reservation details.

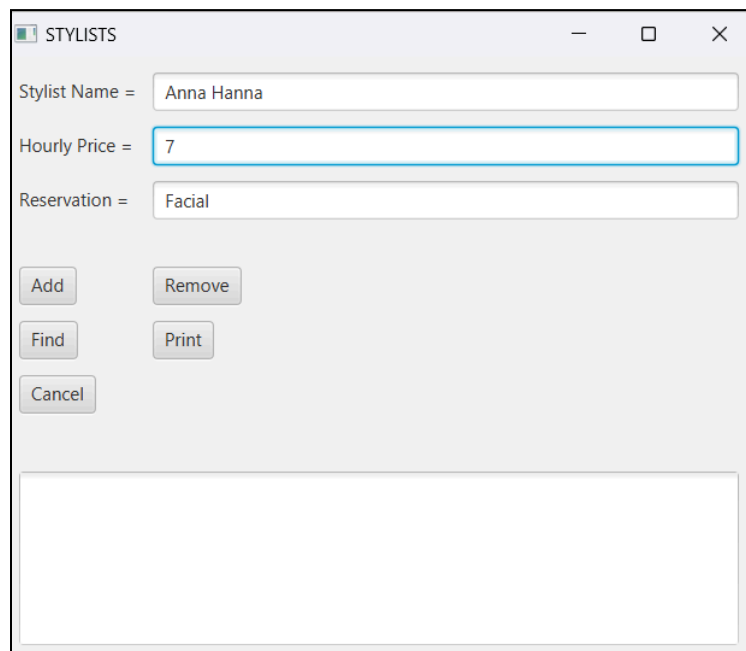
Client Reservation Interface

2. The customer's reservation is finalized and stored in the system.



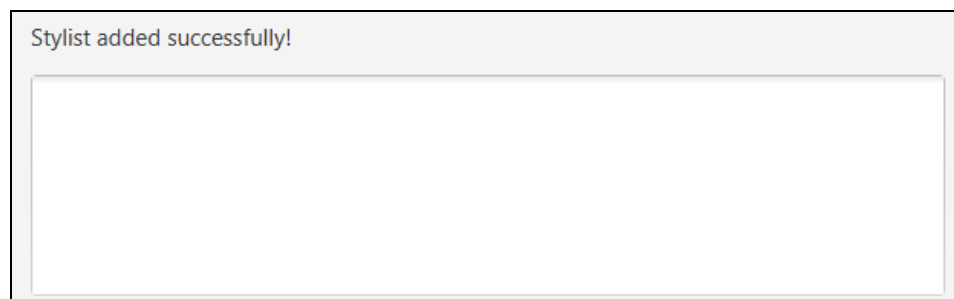
Stylist:

1. Stylists input their hourly charge and confirm by clicking **Add**.



Stylists Interface

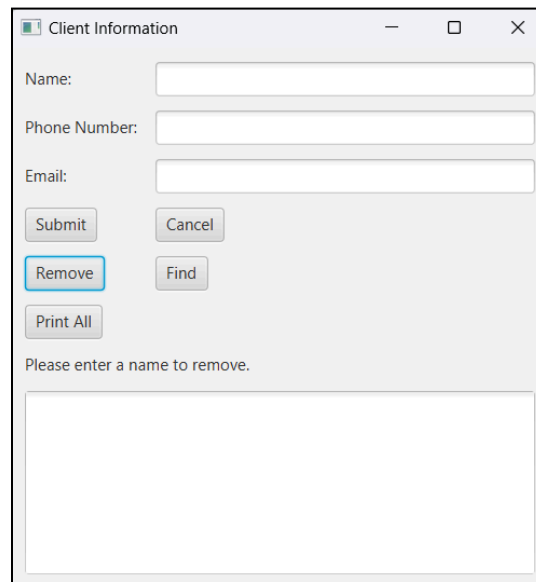
3. The stylist's information has been successfully recorded in the system.



Remove

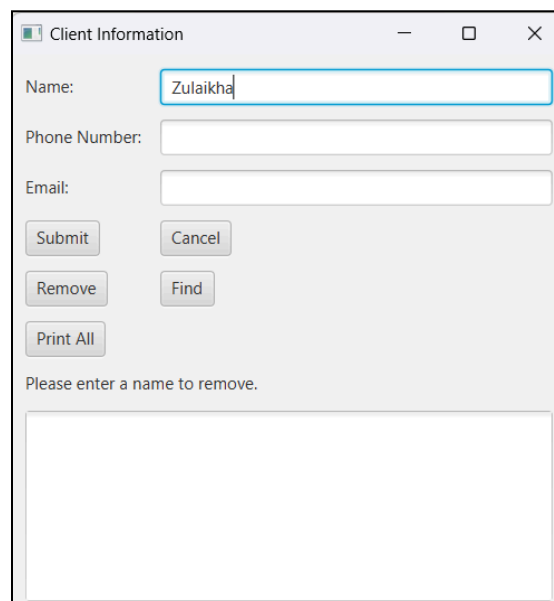
Customer Information:

1. To remove a customer's information, click the **Remove** button.



The screenshot shows a window titled "Client Information" with a standard Windows title bar (minimize, maximize, close buttons). Inside the window, there are three text input fields labeled "Name:", "Phone Number:", and "Email:". Below these fields are five buttons: "Submit", "Cancel", "Remove", "Find", and "Print All". The "Remove" button is highlighted with a blue border. At the bottom of the window, there is a label "Please enter a name to remove." followed by a large, empty text area.

2. Enter the customer's name and click the **Remove** button again.



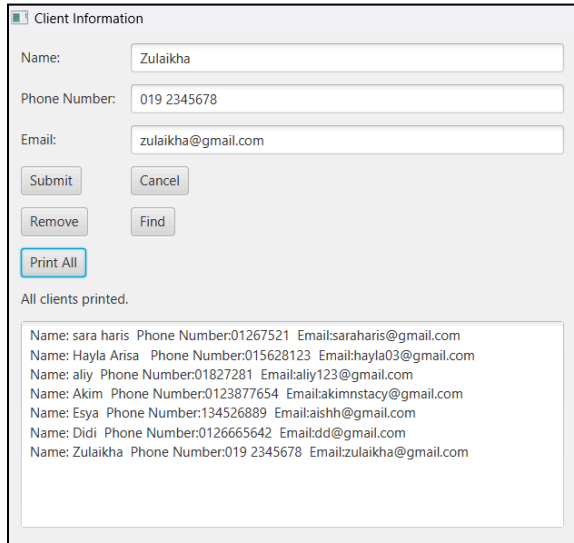
This screenshot shows the same "Client Information" window. The "Name:" input field now contains the text "Zulaikha". The "Remove" button is no longer highlighted. All other elements, including the "Phone Number:", "Email:" fields, the "Submit", "Cancel", "Find", and "Print All" buttons, and the bottom section with the "Please enter a name to remove." label and text area, remain the same as in the previous screenshot.

3. The customer's information is successfully removed from the database.
4. To confirm the cancellation, click **Print All** to view the updated list of reservations.
5. Steps 1-4 are repeated for **Client Reservation** and **Stylist** to remove their information.

Print

Client Information:

1. Click **Print All** to generate the list of all customer information.



Client Information

Name:

Phone Number:

Email:

Submit Cancel

Remove Find

Print All

All clients printed.

```

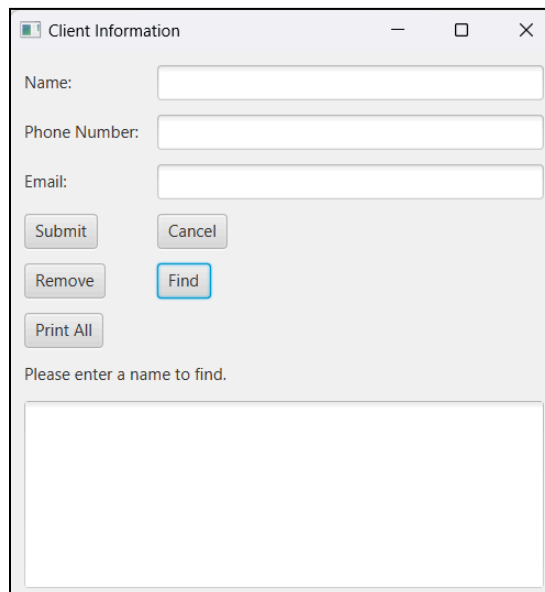
Name: sara haris Phone Number:01267521 Email:saraharis@gmail.com
Name: Hayla Arisa Phone Number:015628123 Email:hayla03@gmail.com
Name: aliy Phone Number:01827281 Email:aliy123@gmail.com
Name: Akim Phone Number:0123877654 Email:akimnstacy@gmail.com
Name: Esya Phone Number:134526889 Email:aishh@gmail.com
Name: Didi Phone Number:0126665642 Email:dd@gmail.com
Name: Zulaikha Phone Number:019 2345678 Email:zulaikha@gmail.com
  
```

2. The same step is done for **Client Reservation** and **Stylist**.

Find

Client Information:

1. Click the **Find** button to find a specific customer's information.



Client Information

Name:

Phone Number:

Email:

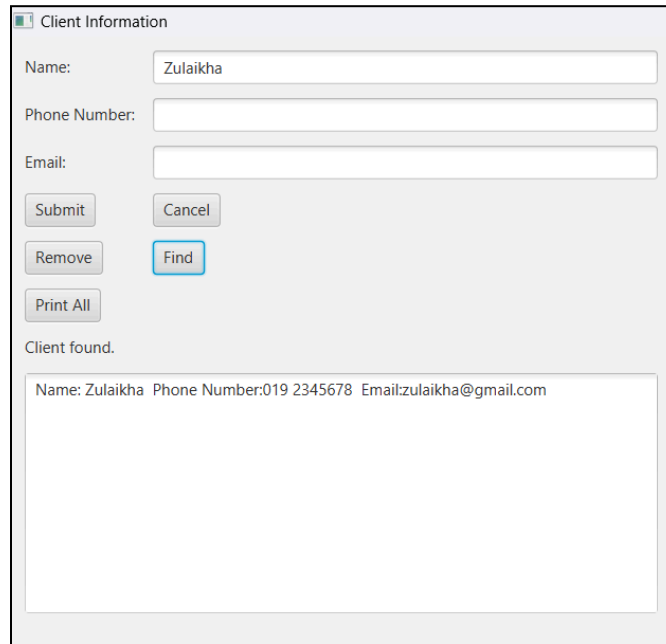
Submit Cancel

Remove **Find**

Print All

Please enter a name to find.

2. Enter the customer's name and click the **Find** button again.

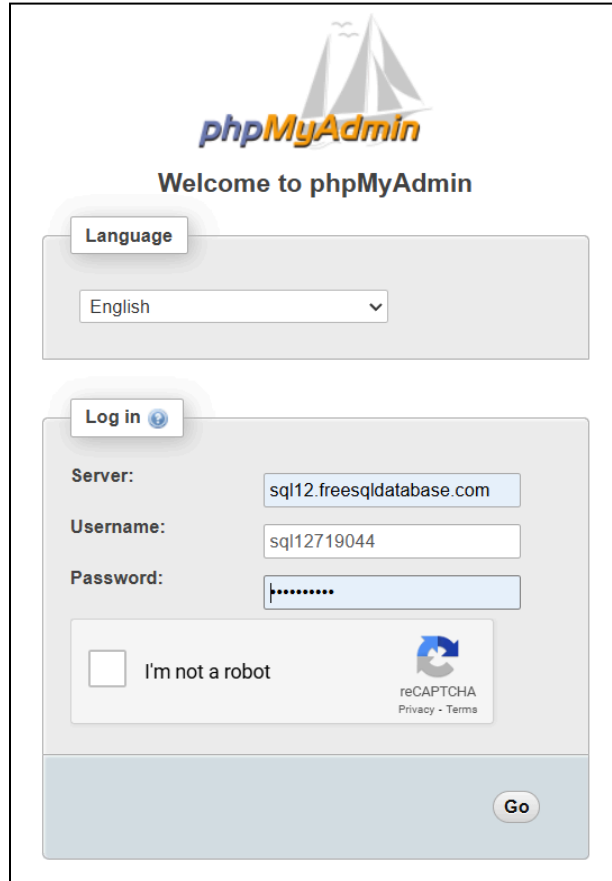


The screenshot shows a web form titled "Client Information". It contains three input fields: "Name:" with the value "Zulaikha", "Phone Number:", and "Email:". Below the fields are five buttons: "Submit", "Cancel", "Remove", "Find" (highlighted with a blue border), and "Print All". Below the buttons, the text "Client found." is displayed, followed by a text box containing the details: "Name: Zulaikha Phone Number:019 2345678 Email:zulaikha@gmail.com".

3. Client's information is displayed.
4. Repeat steps 1-3 to find customer's reservation details on **Client Reservation** and stylists details on **Stylist**.

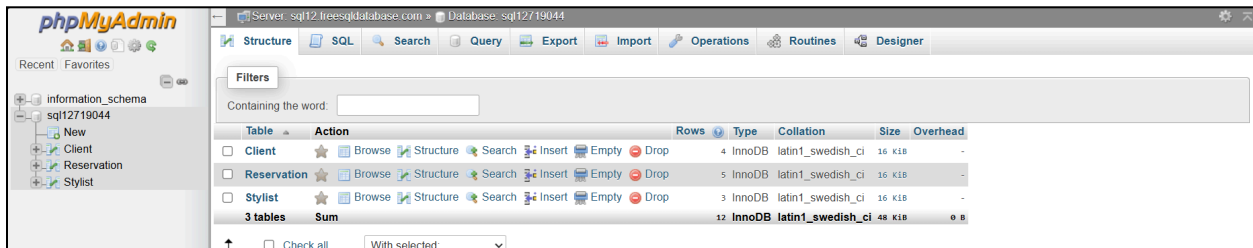
Database - phpMyAdmin

1. To access the database phpMyAdmin is used.
2. Log into phpMyAdmin with Username: sql12719044 and Password: PQUSjvXBEP



The image shows the phpMyAdmin login page. At the top is the phpMyAdmin logo and the text "Welcome to phpMyAdmin". Below this is a "Language" dropdown menu set to "English". Underneath is a "Log in" button. The login form contains fields for "Server:" (sql12.freesqldatabase.com), "Username:" (sql12719044), and "Password:" (masked with dots). There is also a checkbox for "I'm not a robot" and a reCAPTCHA logo. A "Go" button is at the bottom right of the form.

Log in to phpMyAdmin



The image shows the phpMyAdmin interface after logging in. The left sidebar shows the database structure for "sql12719044" with tables: Client, Reservation, and Stylist. The main area displays a table list for the "Reservation" table. The table has columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The table "Reservation" has 5 rows and is of type InnoDB with latin1_swedish_ci collation. The table "Stylist" has 3 rows and is of type InnoDB with latin1_swedish_ci collation. The table "Client" has 4 rows and is of type InnoDB with latin1_swedish_ci collation. The table "Reservation" is selected.

Table	Action	Rows	Type	Collation	Size	Overhead
Client	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 K	18
Reservation	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 K	18
Stylist	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 K	18
3 tables	Sum	12	InnoDB	latin1_swedish_ci	48 K	18

Tables are created for each interface

+ Options

Name	Phone Number	Email
Zulaikha	019 2345678	zulaikha@gmail.com
ADRIANA	017364839	adri@gmail.com
LALISA	0173452864	lisa777@gmail.com
ROSE	0192673472	rosehere@gmail.com

Client Table

+ Options

Client Name	Phone Number	Email	Date	Time	Service Type	Stylist
Zulaikha	019 2345678	zulaikha@gmail.com	2024-07-26	2:00 PM	Facial	Anna Hanna
ADRIANA	017364839	adri@gmail.com	2024-08-03	12:00 PM	Styling	Emma Stone
LALISA	0173452864	lisa777@gmail.com	2024-07-31	1:00 PM	Styling	Emma Stone
KUROMI	0162363854	kuro333@gmail.com	2024-07-30	2:00 PM	Haircut	Adam Rizq
ROSE	0192673472	rosehere@gmail.com	2024-08-01	3:00 PM	Facial	Adam Rizq

Reservation Table

+ Options

Stylist Name	Hourly Price	Reservations
Anna Hanna	10	Facial
Anna Hanna	9	Coloring
Adam Rizq	7	Facial

Stylist Table

PROJECT UML DIAGRAM

