GROUP PROJECT REPORT

TEB1033/TFB1043: OBJECT ORIENTED PROGRAMMING

MAY 2024

TITLE: KPA FITNESS

LECTURER NAME: DR NORDIN ZAKARIA

DATE SUBMITTED: 26 JULY 2024

NAME	STUDENT ID	COURSE
MUHAMAD ADAM HARIS BIN HAIRULRIZAL	24000263	IT
WAN AKID MAN IKMAL BIN SAIFUL EZUAN	24000336	IT
MOHAMMAD AMIR HAZMAN BIN NAWANDI	24000387	IT
KAGENDRAN A/L SIVAM	24000582	IT
AMRIL NAJWAN BIN AHMAD BASRI	24000254	COE

Table of Contents

Video Demonstration Link: https://www.youtube.com/watch?v=uJ3CsX56T-0

1.0 Introduction

KPA Fitness is a complete software application specifically developed to meet the requirements of both fitness professionals and enthusiasts. If you are a personal trainer, gym owner, or someone dedicated to maintaining a healthy lifestyle, KPA Fitness provides a streamlined and effective solution for monitoring fitness objectives, organizing client information, and designing customized exercise routines.

Other than that, KPA Fitness also offers a user-friendly experience with its simple Java Swing interface, enabling users to easily browse and concentrate on attaining and exceeding their fitness goals. The program is supported by a strong MySQL database, ensuring dependable data management and preservation. This creates a stable platform for users to establish and sustain their fitness journey.

Lastly, KPA Fitness distinguishes itself as a crucial tool for those dedicated to enhancing their exercise routines and client relations by including robust functionalities and a user-friendly interface.

2.0 Objective

The KPA Fitness Desktop Application's primary objective is to offer a comprehensive platform that improves and streamlines the administration of fitness activities. Our goals are to simplify client management, improve fitness tracking, and ensure secure use for users.

KPA Fitness offers fitness users a user-friendly system that efficiently manages client databases, allowing for easy storage, organization, and retrieval of client information. This guarantees a smooth and effective management of clients, enabling trainers to prioritize offering customized fitness programs instead of being overwhelmed by administrative duties. Trainers may customize their services to better suit individual customer needs by utilizing comprehensive client profiles that provide simple access to extensive information such as fitness history, progress monitoring, and personal preferences.

Other than that, KPA Fitness also offers extensive tools to track and visualize users' progress in fitness, promoting motivation and providing users with updates on their achievements. Users may conveniently record their exercises, monitor performance indicators, and visualize their progress using comprehensive progress tracking options that include charts and graphs. This feature not only assists users in maintaining their motivation by displaying their progress, but also allows them to see patterns, establish attainable objectives, and modify their exercise routines for best outcomes. The presence of concrete proof of their diligent efforts serves as a motivator for individuals to persist and remain dedicated to their pursuit of physical health.

Lastly, KPA Fitness uses robust user authentication to protect sensitive data and restrict program access. To safeguard customer data and personal information, the application uses password protection, encrypted data storage, and multi-factor authentication. This strong user authentication technique reduces the danger of data breaches and gives users confidence that their data is being handled carefully and securely. Role-based access controls can also restrict and manage user authorizations to ensure that only authorized users can perform software operations.

3.0 Problem Statement

Ensuring our well-being and physical condition is crucial in our busy lifestyle, although managing our exercise routines and client data can pose challenges. Despite being outdated and susceptible to errors, pen and paper as well as basic spreadsheets continue to be often utilized. An improved system is evidently required, one that facilitates the storage of client data, creation of training plans, and monitoring of progress, all within a single spot. KPA Fitness addresses these concerns by utilizing a digital platform that ensures precision, accessibility, and organization.

4.0 Overview

KPA Fitness is segmented into many fundamental modules, each designed to target a certain aspect of fitness administration:

1. Login Module: Offers users a safe method of logging in and using the program.

2. Main Module: Serves as the app's focal point and makes navigating between its many sections simple.

3. Client Module: Oversees client data, including fitness measurements and personal information.

4. Fitness Progress Module: Allows users to monitor their progress by tracking their fitness activities over time.

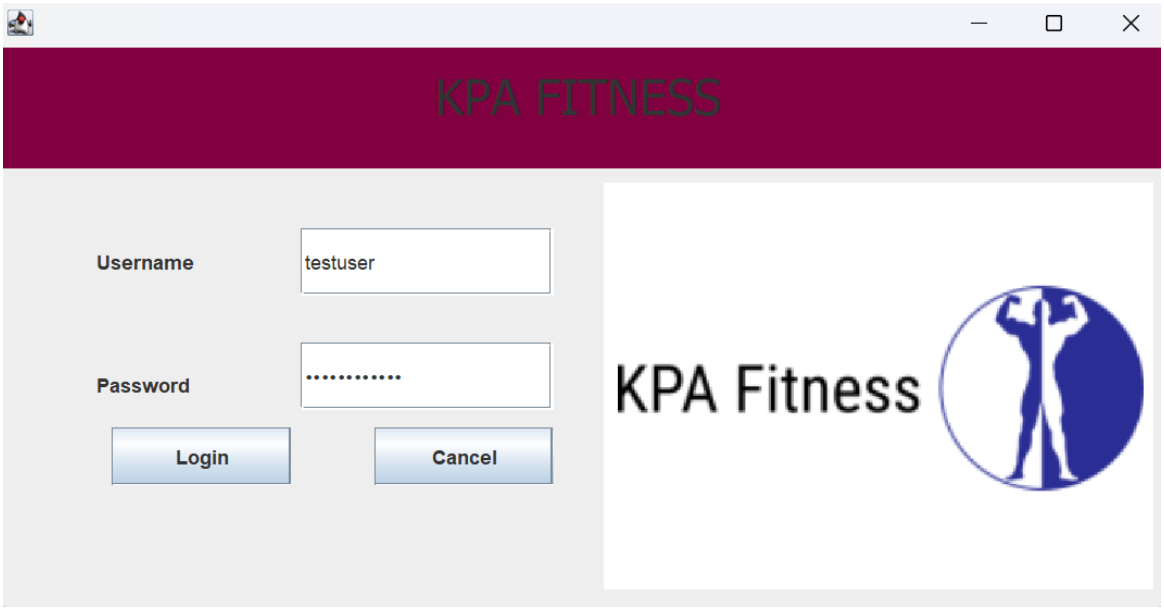
5. Exercise Plan Module: Assists in developing and overseeing customized exercise regimens for every customer

5.0 Application

The KPA Fitness app features an interface that is both user-friendly and clear. The main module offers an organized and well-structured interface for utilizing the application, while the login module ensures secure access. Users may proficiently manage client data, track fitness progress, and develop customized fitness regimens. Among the important features of KPA Fitness are secure authentication, client management, progress tracking, and customizable plans.

The secure authentication mechanism of KPA Fitness app ensures that only authorized users may have access to the software, therefore protecting user data and maintaining strict secrecy. The process of managing clients is optimized, enabling fitness professionals to organize and update customer information for convenient retrieval and modifications, and will enhances the efficiency of client handling. Progress monitoring systems enhance user engagement and motivation by presenting graphical depictions of their fitness progress, hence fostering dedication towards their objectives. Moreover, the availability of customized programs allows for the creation of individualized training routines that are specifically designed to meet the distinct requirements and preferences of every user. This ensures that every workout is fine-tuned to maximize the desired outcomes. When these elements combined, it will builds a strong and easy-to-use platform that helps both fitness experts and users in accomplishing their fitness goals.

6.0 Implementation



(Figure 1 shown interface for KPA Fitness login face)

```
CREATE TABLE users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(50) NOT NULL,
   password VARCHAR(100) NOT NULL
);
```

(Figure 2 shown Database for login face)

Purpose

The class login is responsible for handling the user login process. It creates a graphical user interface (GUI) where users can enter their username and password to authenticate themselves.

Key Components

- JFrame frame: The main window of the application.
- JTextField textField_1: Text field for entering the username.
- JPasswordField passwordField: Password field for entering the password.
- JPanel panel: Panel for organizing components.
- JButton btnLogin: Button for submitting login credentials.
- JButton btnNewButton (Cancel): Button for canceling the login process.
- JLabel lblUsername, lblPassword: Labels for username and password fields.
- ImageIcon lblimg: Image label for displaying an image.

Methods

- main(String[] args): Entry point of the application. It invokes the Login window.
- Login(): Constructor that initializes the frame.
- initialize(): Initializes the contents of the frame.
- authenticateUser(): Authenticates the user by checking the entered username and password against the database.

Main.java

Purpose

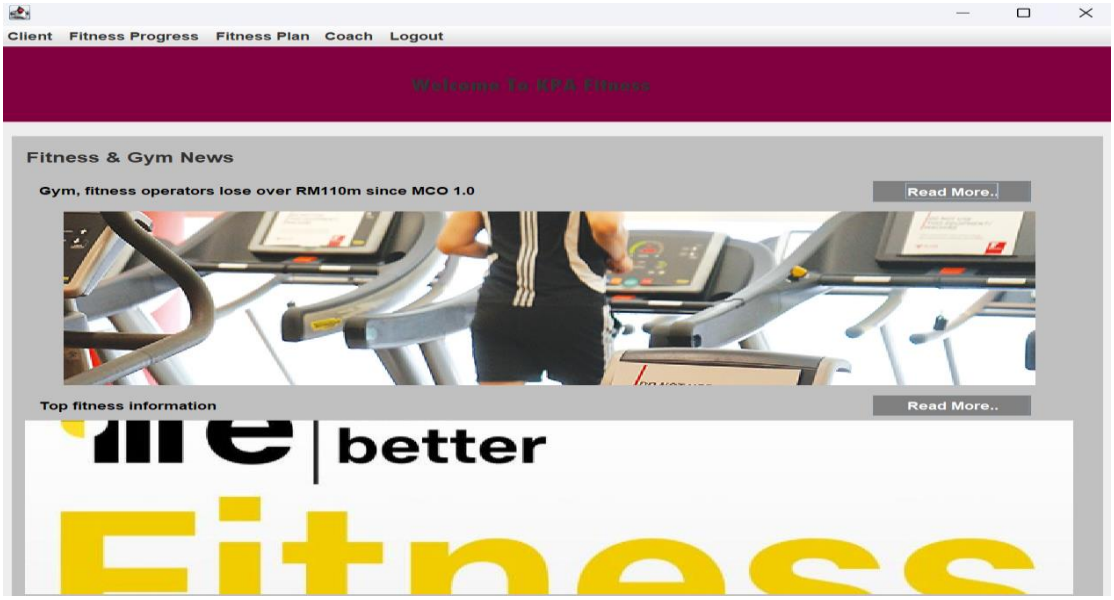
The Main class is the main window that users see after successfully logging in. It provides options to navigate to different sections of the application such as Client information, Fitness Progress, Fitness Plan, and Coach information.

Key Components

- JFrame frame: The main window of the application.
- JMenuBar menuBar: Menu bar containing different menu options.
- JMenu menuClient, menuFitnessProgress, menuFitnessPlan, menuCoach, menuLogout: Different menus in the menu bar.
- JMenuItem menuItemClient, menuItemFitnessProgress, menuItemFitnessPlan, menuItemCoach, menuItemLogout: Menu items for different sections.
- JPanel panel: Panel for organizing components.
- JPanel newsPanel: Panel for displaying news.
- JLabel lblNewLabel, lblNewLabel_1, lblNewLabel_2, lblNewLabel_2_1: Labels for different text and images.
- JButton btnNewButton, btnNewButton_1: Buttons for reading more news.

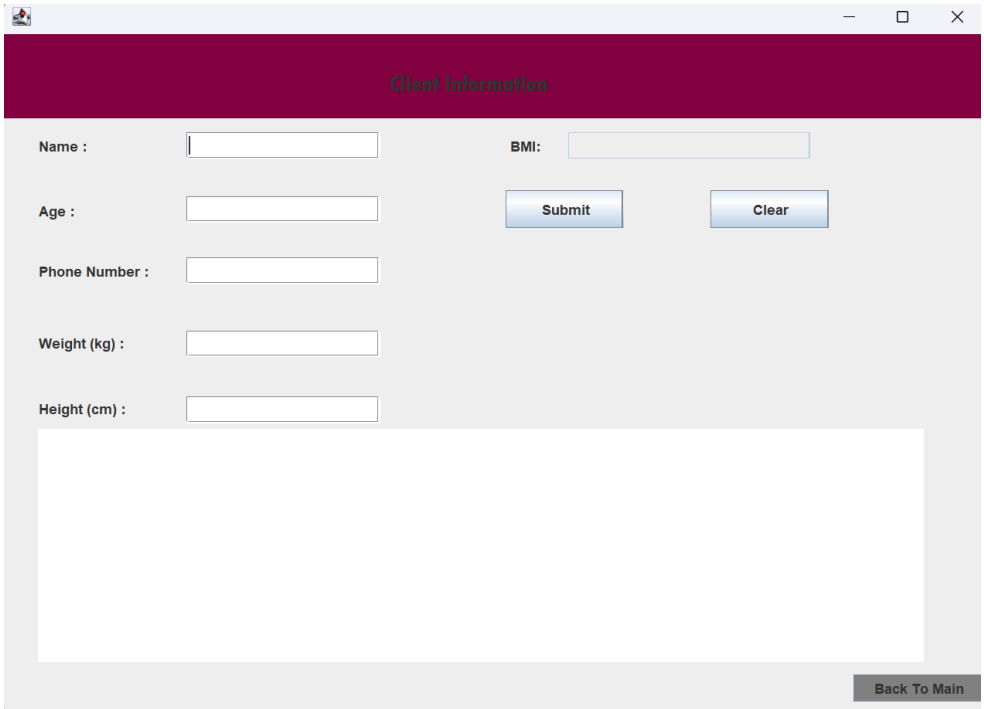
Methods

- main(String[] args): Entry point of the application. It invokes the Main window.
- Main(): Constructor that initializes the frame.
- initialize(): Initializes the contents of the frame.
- setVisible(boolean b): Sets the visibility of the frame



(Figure 3: Homepage for Kpa Fitness)

Client.java



(Figure 4: Client Infromation Interface)

Purpose

The Client class is responsible for managing client information. It allows users to enter client details such as name, phone number, age, weight, and height. It also calculates the Body Mass Index (BMI) based on the entered weight and height.

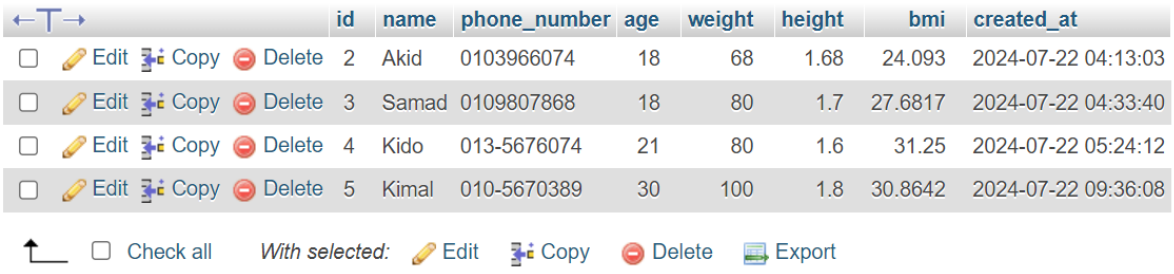
Key Components

- JFrame frame: The main window of the application.
- JTextField textName, textPN, textAge, textWeight, textHeight, textField: Text fields for entering client details.
- JTextPane textPane: Text pane for displaying additional information.
- JButton btnSubmit, btnClear, btnback: Buttons for submitting, clearing, and navigating back to the main window.
- JPanel panel: Panel for organizing components.
- JLabel lblName, lblPhoneNum, lblFitPlan, lblAge, lblWeight, lblHeight, lblClientInformation: Labels for different fields.

Methods

- Client(): Constructor that initializes the frame.
- getFrame(): Returns the frame.
- initialize(): Initializes the contents of the frame.
- calculateAndSaveData(): Calculates the BMI and saves the client data to the database.

```
CREATE TABLE IF NOT EXISTS clients (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    phone_number VARCHAR(255) NOT NULL,
    age INT NOT NULL,
    weight DOUBLE NOT NULL,
    height_cm DOUBLE NOT NULL  -- Height in centimeters
);
```



	id	name	phone_number	age	weight	height	bmi	created_at
Edit Copy Delete	2	Akid	0103966074	18	68	1.68	24.093	2024-07-22 04:13:03
Edit Copy Delete	3	Samad	0109807868	18	80	1.7	27.6817	2024-07-22 04:33:40
Edit Copy Delete	4	Kido	013-5676074	21	80	1.6	31.25	2024-07-22 05:24:12
Edit Copy Delete	5	Kimal	010-5670389	30	100	1.8	30.8642	2024-07-22 09:36:08

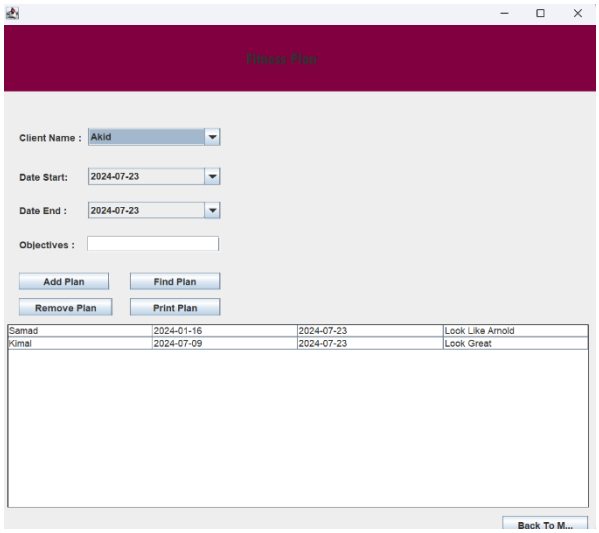
Check all *With selected:* Edit Copy Delete Export

(Figure 5: Database for clients)

Explanation of the Code Flow

1. **Login Process:**
    - The application starts with the class login, displaying a login window.
    - The user enters their username and password.
    - Upon clicking the "Login" button, the method authenthicateUser() is called, which checks the credentials against the database.
    - If the credentials are valid, the login window is closed, and the Main window is displayed.
2. **Main Window:**
    - The Main class displays a window with different menu options.
    - Users can navigate to different sections like Client, Fitness Progress, Fitness Plan, and Coach information by clicking on the respective menu items.
    - The "Logout" option allows users to log out and return to the login window.
3. **Client Information:**
    - The Client class displays a window for entering client details.
    - Users can enter the client's name, phone number, age, weight, and height.
    - The "Submit" button calculates the BMI and saves the client data to the database.
    - The "Clear" button clears the form fields.
    - The "Back To Main" button navigates back to the main window.

FitnessPlan.java



(Figure 6: Fitness Plan Interface)

Purpose

The Fitness Plan interface is designed to help users manage and track fitness plans for clients in a gym or fitness setting. It provides a graphical user interface (GUI) that allows users to perform various tasks related to fitness plans.

Key Components:

- JFrame frame: The main window for the fitness plan application.
- JTextField textField: Input field for the objectives.
- JTable table: Table for displaying fitness plans.
- JComboBox<String> comboBoxClient: Dropdown for selecting a client.
- JComboBox<String> comboBoxWeekStart: Dropdown for selecting the start date of the fitness plan.
- JComboBox<String> comboBoxWeekEnd: Dropdown for selecting the end date of the fitness plan.
- DefaultTableModel tableModel: Model for managing the data in the table

Constructor:

- Initializes the application by setting up the frame, loading client names, week dates, and existing fitness plans.

getFrame:

- Returns the main frame of the application.

initialize:

- Sets up the main frame and its components, including buttons, labels, combo boxes, and the table.

loadClientNames:

- Loads client names from the database into the client dropdown list.

loadWeekDates:

- Loads the last 52 weeks' dates into the start and end date dropdown lists.

loadTableData:

- Loads existing fitness plans from the database into the table model.

addPlan:

- Adds a new fitness plan to the database and updates the table with the new entry.

removePlan:

- Removes the selected fitness plan from the database and updates the table to reflect the removal.

findPlan:

- Finds and displays fitness plans for the selected client in the table.

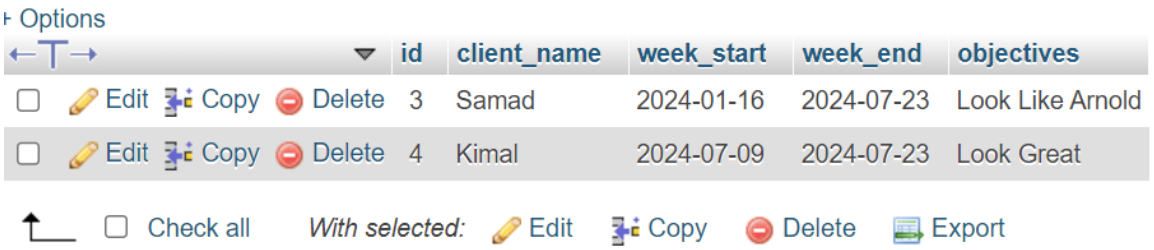
printPlan:

- Displays the details of the selected fitness plan in a dialog box.

main:

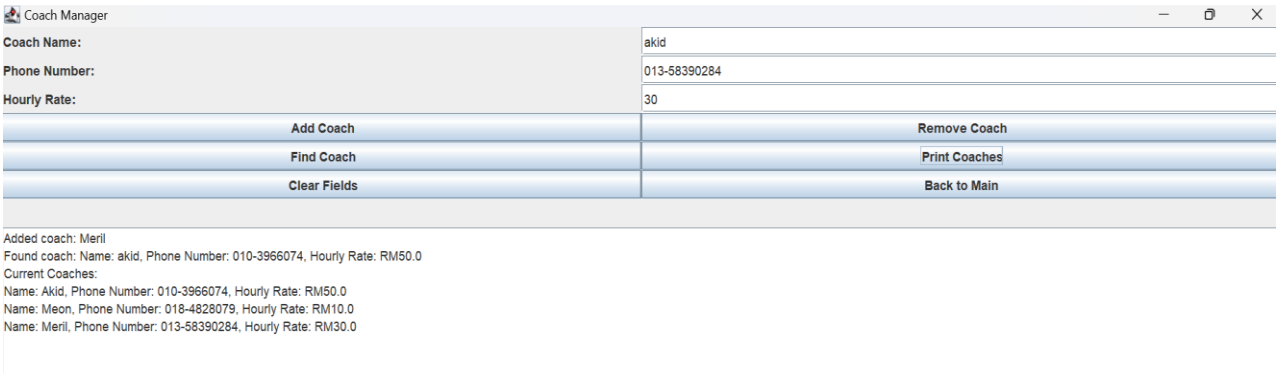
- Launches the FitnessPlan application on the Event Dispatch Thread using SwingUtilities.invokeLater.

```
CREATE TABLE fitness_plans (
    id INT AUTO_INCREMENT PRIMARY KEY,
    client_name VARCHAR(255),
    week_start DATE,
    week_end DATE,
    objectives VARCHAR(255)
);
```



(Figure 7: Database for fitnessPlan)

FitnessProgress.Java



(Figure 8: FitnessProgress Interface)

Purpose

The FitnessProgress class is designed to help users track and manage their progress in a fitness program. This class allows users to log their progress, view their progress history, and analyze their improvements over time.

Key Components

1. JFrame (frame): The main window of the application that contains all other components.
2. JComboBox (comboBoxClient, comboBoxDate): Dropdown lists for selecting a client and the date of progress entry.
3. JTextField (textFieldWeight, textFieldBodyFat, textFieldMuscleMass): Input fields for entering the client's weight, body fat percentage, and muscle mass.

4. JButton (btnAddProgress, btnRemoveProgress, btnFindProgress, btnPrintProgress, btnBackToMain): Buttons to perform various actions like adding, removing, finding, printing progress entries, and navigating back to the main interface.
5. JTable (table): Table to display the progress entries.
6. DefaultTableModel (tableModel): Model for managing the data displayed in the table.
7. JPanel (panel): Panel to organize and hold components.
8. JLabel (lblFitnessProgress, lblClientName, lblDate, lblWeight, lblBodyFat, lblMuscleMass): Labels to identify different input fields and sections.

Methods

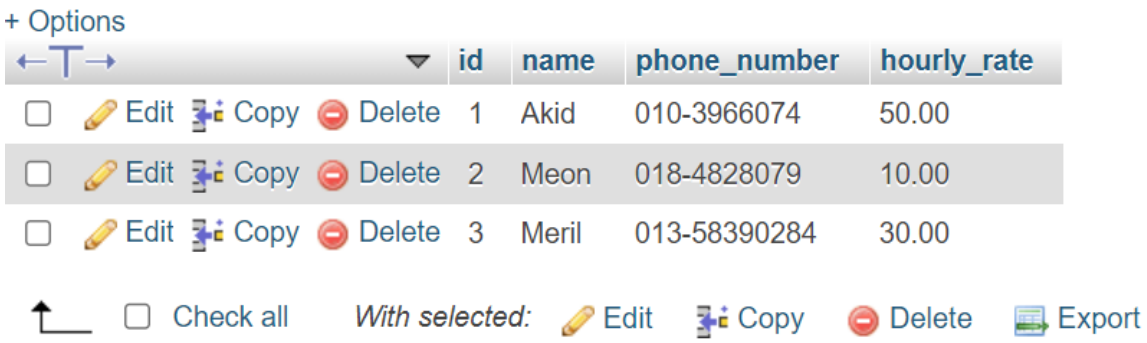
1. Constructor:
    - Initializes the application by setting up the frame, loading client names, dates, and existing progress entries.
2. getFrame:
    - Returns the main frame of the application.
3. initialize:
    - Sets up the main frame and its components, including buttons, labels, combo boxes, and the table.
4. loadClientNames:
    - Loads client names from the database into the client dropdown list.
5. loadDates:
    - Loads dates from the database into the date dropdown list.
6. loadTableData:
    - Loads existing progress entries from the database into the table model.
7. addProgress:
    - Adds a new progress entry to the database and updates the table with the new entry.
8. removeProgress:
    - Removes the selected progress entry from the database and updates the table to reflect the removal.
9. findProgress:
    - Finds and displays progress entries for the selected client and date in the table.
10. printProgress:

- Displays the details of the selected progress entry in a dialog box.

11. main:

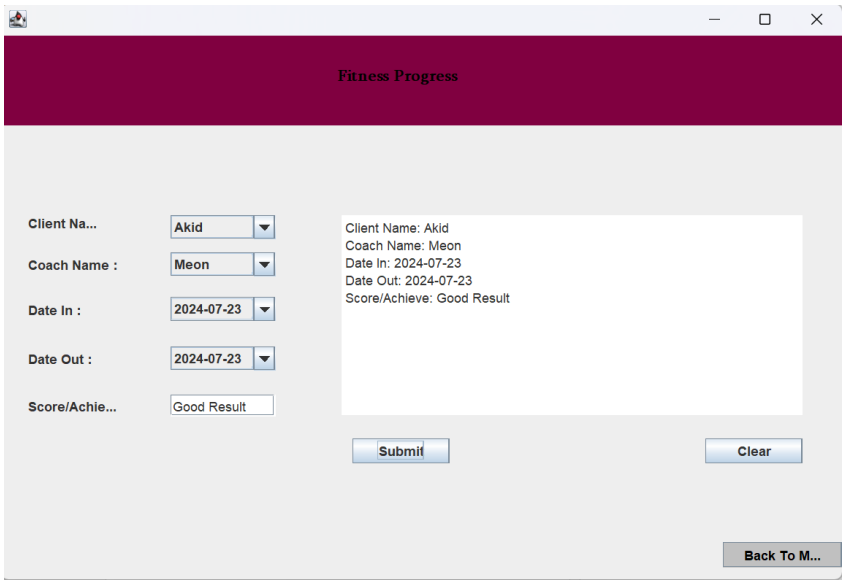
- Launches the FitnessProgress application on the Event Dispatch Thread using SwingUtilities.invokeLater.

```
CREATE TABLE IF NOT EXISTS coaches (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    phone_number VARCHAR(255) NOT NULL,
    hourly_rate DECIMAL(10, 2) NOT NULL
);
```



(Figure 9: Database for FitnessProgress)

Fitness Progress



(Figure 10: Fitness Progress Interface)

Purpose

The FitnessProgress class in the provided code is part of a Java Swing application for managing and displaying fitness progress data. It allows users to enter and submit fitness-related information, including client and coach names, dates, and scores. This data is then stored in a database and displayed in a text pane within the application.

Key Components

1. JFrame (frame): The main window of the application.
2. JTextField (textFieldScore): A text field for inputting the score or achievement.
3. JComboBox (comboBoxClient, comboBoxCoach, comboBoxDateIn, comboBoxDateOut): Drop-down lists for selecting client names, coach names, and dates.
4. JTextPane (textPane): A text area that displays the submitted data.
5. JButton (btnBack, btnSubmit, btnClear): Buttons for navigating back to the main window, submitting data, and clearing the form.
6. JLabel (lblNewLabel, lblDI, lblDO, lblSA, lblCN, lblCN2): Labels for various fields and instructions.
7. JPanel (panel): A panel used for grouping and styling components.

Methods

1. public FitnessProgress(): Constructor that initializes the frame and sets up the UI components.
2. private void initialize(): Initializes and arranges all the UI components. Sets up action listeners and loads data from the database.
3. private void loadNames(): Connects to the database, retrieves client and coach names, and populates the corresponding combo boxes.
4. private String[] getDates(): Generates an array of date strings for the past 365 days, formatted as "yyyy-MM-dd".
5. private void submitData(): Retrieves data from the form, connects to the database, inserts the data into the fitness_progress table, and updates the text pane with the submitted information.
6. private void clearForm(): Resets all form fields and the text pane to their default state.
7. public static void main(String[] args): Entry point of the application that initializes and displays the FitnessProgress frame.

```
CREATE TABLE fitness_progress (
    id INT AUTO_INCREMENT PRIMARY KEY,
    client_name VARCHAR(255),
    coach_name VARCHAR(255),
    date_in DATE,
    date_out DATE,
    score VARCHAR(255)
);
```

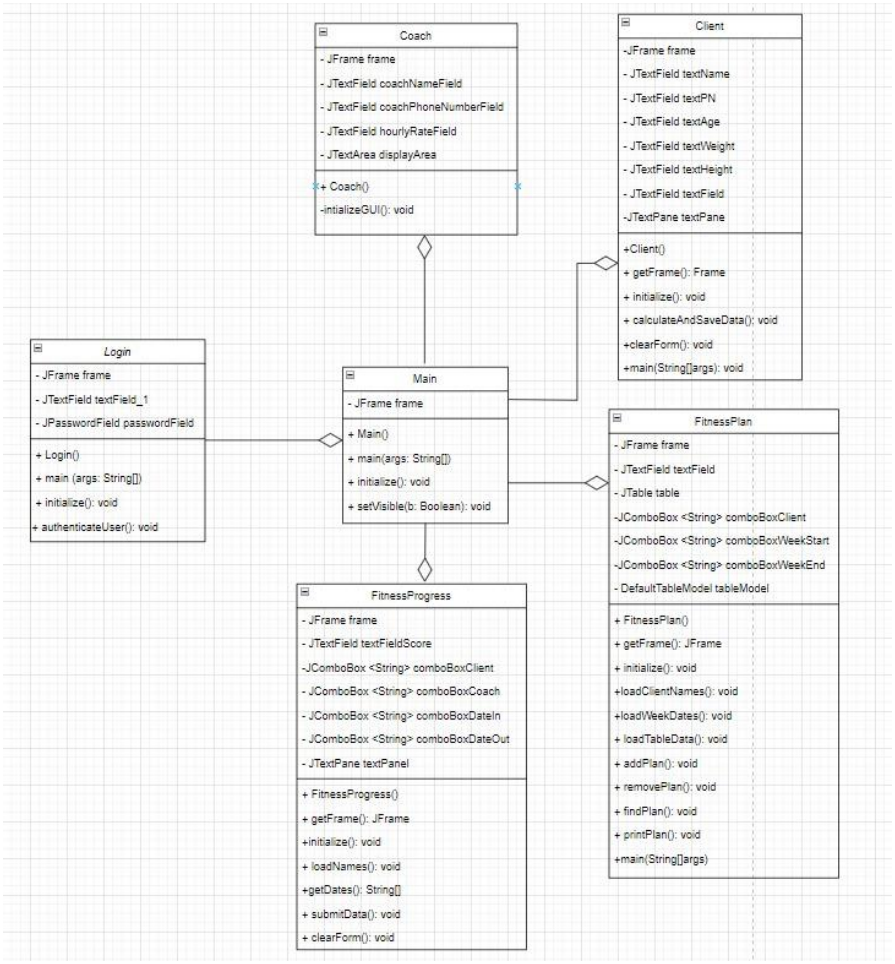
+ Options

	id	client_name	coach_name	date_in	date_out	score
Edit Copy Delete	1	Samad	Akid	2024-07-13	2024-07-22	Lose 5 kg
Edit Copy Delete	2	Samad	Akid	2023-07-24	2024-07-22	Strong Body
Edit Copy Delete	3	Kimal	Akid	2024-07-23	2024-07-23	Under Progress
Edit Copy Delete	4	Akid	Meon	2024-07-23	2024-07-23	Good Result

Check all *With selected:* Edit Copy Delete Export

(Figure 11: Database for FitnessProgress)

7.0 UML Diagram



(Figure 12: KPA Fitness UML Diagram)

8.0 Conclusion

In conclusion, KPA Fitness is a comprehensive solution designed to tackle the challenges of managing and tracking fitness activities. By integrating multiple functionalities into one application, it simplifies client management, progress tracking, and fitness planning. With its user-friendly interface and secure database, KPA Fitness provides a reliable platform for fitness professionals and enthusiasts. This allows users to focus on achieving their fitness goals while maintaining an organized and efficient system.