



# GYM APP

OOP Project 2022

## Abstract

Java app meant for displaying useful gym-related information to users

Prof: Bindea Bogdan

Cristian Adrian Dora

doracristian@yahoo.com

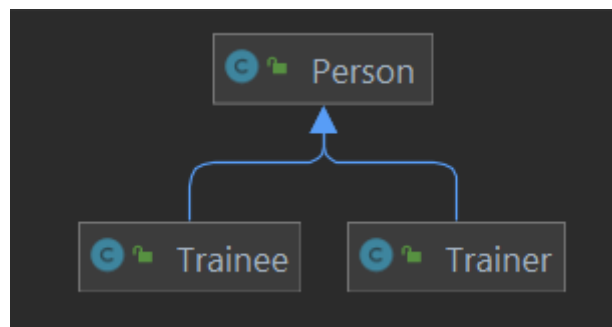
## Introduction

The app helps people who want to start going to the gym or simply just exercising. After the user creates an account, he will see a new window where he can buy supplements and request a training regime from one of the available trainers. Also, the user has a bonus feature. After all the effort put into losing weight, the app can reflect the user's progress. The user has the option to change the weight displayed on the screen with his new weight.

For this application, besides Java programming language, I also used a personal Postgres database and Swing for GUI.

## Implementation details

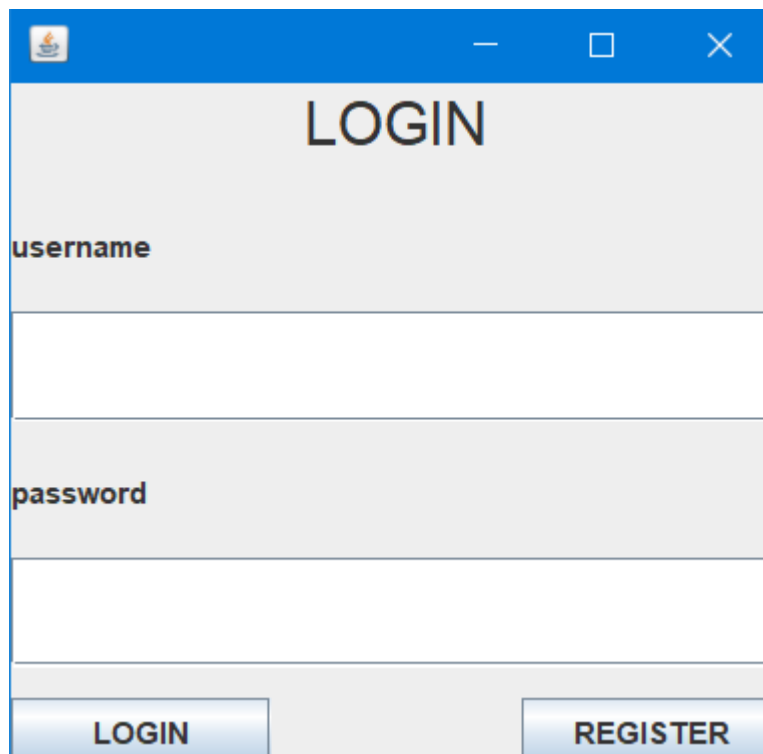
The application follows MVC structure. Each view has its control class where the view is instantiated. There are also classes for Trainers, Trainees (those 2 extend the class Person), for Supplements and Subscriptions. Those are part of the model package.



Each frame(view) is instantiated as an object and all 3 views are “glued” by the control classes which implement the ActionListener Interface. The control classes override the original actionPerformed method.

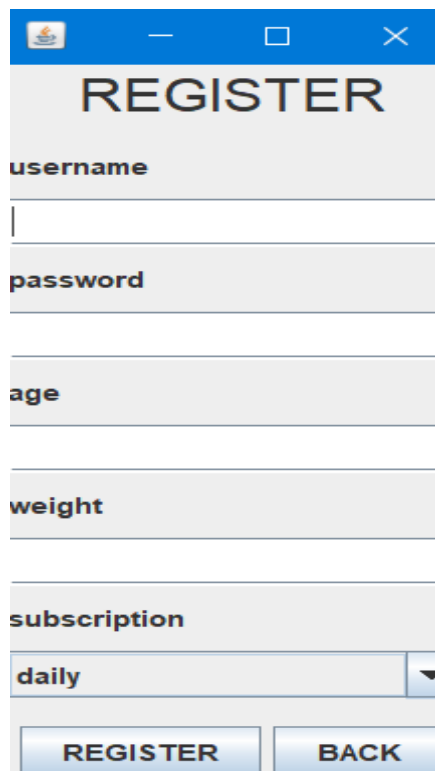
```
@Override
public void actionPerformed(ActionEvent e) {
```

## GUI



A screenshot of a Windows-style application window titled "LOGIN". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is light gray and contains two input fields. The first field is labeled "username" and the second is labeled "password". Below these fields are two buttons: "LOGIN" on the left and "REGISTER" on the right. Both buttons have a blue gradient and a slight shadow effect.

This is the main(first) window. From here the user can log into his/her account with a valid username/password. The process is completed when LOGIN button is pressed. If the user doesn't have an account, a new window will appear, with a registration form.



A screenshot of a Windows-style application window titled "REGISTER". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is light gray and contains five input fields. The first field is labeled "username", the second is labeled "password", the third is labeled "age", and the fourth is labeled "weight". Below these fields is a dropdown menu labeled "subscription" with "daily" selected. At the bottom of the window are two buttons: "REGISTER" on the left and "BACK" on the right. Both buttons have a blue gradient and a slight shadow effect.

There are a few error checks like the following:

The image displays three separate error message dialog boxes. The first, titled 'BAD INPUT', shows a form with fields for 'age' (containing 'a') and 'weight' (containing '66'). The message states: 'Bad input: you entered a character in a field which accepts only numbers'. The second, titled 'REGISTRATION', shows a form with 'username' and 'password' fields. The message states: 'Invalid username or password!'. The third, titled 'BAD USERNAME', shows a message: 'This username is already taken'. Each dialog box includes a red 'X' icon and an 'OK' button.

The username & password must not be blank, and weight & age must be valid numbers. In case of bad input, an appropriate error message is shown.

After a successful login, the user is now in his account where he/she can see useful information:

The image shows a user account dashboard for a user named 'mihai'. The dashboard is divided into several sections. On the left, there is a 'Current weight:' section with a text input field containing '61', a checkbox labeled 'Enable weight edit', and a 'SUBMIT' button. Below this is an 'Available trainers:' section with a dropdown menu showing 'Jucan Rares' and a 'REQUEST TRAINING' button. In the center, there is a large empty box. On the right, there is an 'Available supplements:' section with a dropdown menu showing 'Beta-Alanine' and an 'ADD SUPPLEMENT TO CART' button. At the bottom, there is a 'Total cost:' label and a 'PAY NOW' button.

If the user presses the enable weight checkbox, he will be able to edit his/her weight but the value will be updated only after pressing submit.

If the user presses REQUEST TRAINING, the routine of the chosen trainer is shown in the middle.

The screenshot shows a web interface for a user named 'mihai'. At the top right, it says 'Welcome, mihai'. On the left, there is a section for 'Current weight:' with a text input field containing '61'. Below this is a checkbox labeled 'Enable weight edit'. A blue 'SUBMIT' button is positioned below the checkbox. Further down, under 'Available trainers:', there is a dropdown menu showing 'Jucan Rares' and a blue 'REQUEST TRAINING' button. On the right side, a white box displays a list of exercises: '1 Bench press; 2 Triceps dip; 3 Incline dumbbell press; 4 Incline dumbbell flye; 5 Triceps extension'.

On the right, the user can select a supplement, and add it to the cart. The final price will be displayed on the bottom. The user has the option to pay for them.

The screenshot shows a supplement selection interface. On the right, there is a dropdown menu with 'Carbohydrates' selected. Below it is a blue button labeled 'ADD SUPPLEMENT TO CART'. At the bottom left, the text 'Total cost: 80' is displayed next to a blue 'PAY NOW' button.

## Conclusions

This project was a fun challenge, giving me the opportunity to develop my java knowledge. For future updates, the app may include:

- Automatic due date completion so the user knows when his/her current subscription expires.
- Better GUI aspect
- Option to remove items from the cart
- Ability to log in as trainer and add specific routines based on weight/age