Ingegneria del Software e Progettazione Web

Progetto A.A. 2022/2023

SPORTIFY

Chiara D'Ambrogio

Matteo La Gioia

Indice

1. Software Requirement Specification

1.1. Introduction

1.1.1. The aim of Document

The aim of the project is to encourage people to practice a sport, helping them to choose the best sport and the gym where to practice it.

1.1.2. Overview of the defined system

Sportify is used to search for courses and gyms and to leave reviews on their service. Meanwhile, it can be used by gyms to add courses and into schedule them to their Sportify profile. As follows, you can find the guaranteed features:

- User:
  - Search for sports
  - Fill out the test to understand the suitable sport.
  - Find your nearest gym
  - Writing reviews of popular gyms
- Gyms:
  - Add courses and timetables
  - Read reviews of your gym
  - Delete possible unconstructive reviews.

1.1.3. Operational Settings

There are many operating settings for Sportify; the program will be used as a mobile application or computer program by users looking for courses or gyms, who want to understand the most suitable sport, and by gyms that want to advertise themselves.

For the development of the software, we used the following:

- Figma.com to get the HTML code of the storyboards that we modified later with WebStorm.
- StarUML for the realization of diagrams related to the design of the system.
- Scene Builder to create FXML files to create the graphical interface (MVC views).
- JavaFX to manage and extend FXML files (MVC graphics controller).

- IntelliJ and especially Java as an IDE and programming environment. To use the software, you must extract the Sportify folder from the zip file and open it as a Maven project in IntelliJ. In the folder, there is already the '*pom.xml*' file to configure how to run the application. If this is not possible, the class responsible for starting the application is '*src/main/java/sportify/MainAppLauncher.java*'.

1.1.4. Related Systems

`(at least 2), Pros and Cons.`

Performing a careful analysis of the competitors, although there are few similar services, we identified the two in particular: *Fit Finder – Gym Locator, Sportclubby*.

• *Fit Finder – Gym Locator*: This smartphone-only app can also be accessed from gyms and by Google and Facebook as well. Once you accessed to your account, you can search for the closest gym to your location, looking to the gyms relative profiles, allowing one to add a given gym to your favorites. However, Sportify, unlike Fit Finder, also offers a desktop application (for PC) and the possibility to fill out a test that recommends the suitable sport, based on your preferences.

• *Sportclubby*: This smartphone-only app can be accessed by Google and Facebook. Once you have accessed your account, you can search for the closest gym to your location by looking at the relative profiles of the gyms, allowing you to add a given gym to your favorites, considering the choice of favorite activities made at registration. You can also book and pay for courses through the same app, as well as invite friends to courses booked by the app. Sportify, however, unlike Sportclubby also offers a desktop application (for PC) and the possibility to fill out a test that recommends the suitable sport, based on your preferences.

1.2. User Stories

(3 per member)

- **As a new user, I want to discover the best suitable sport for me considering my needs so that I can start practice it.**
- **As a user, I want to review a gym based on my experience so that it could be useful to other users.**
- **As a gym owner, I want to know the ten most popular sports practiced in Italy, so I can add them to my gym offer.**
- **As a new user, I want to know which gyms have a selected sport to not waste time.**
- **As a user, I want to know the main information about a selected sport so that I can decide whether to practice it.**
- **As a gym, I want to sponsor myself to gain more visibility.**
- As a gym owner, I want to read the reviews written about my gym, so that I can modify and improve my gym offer.
- As a user, I want to keep track of what level I reached in the sports I play, so that I can continuously improve.

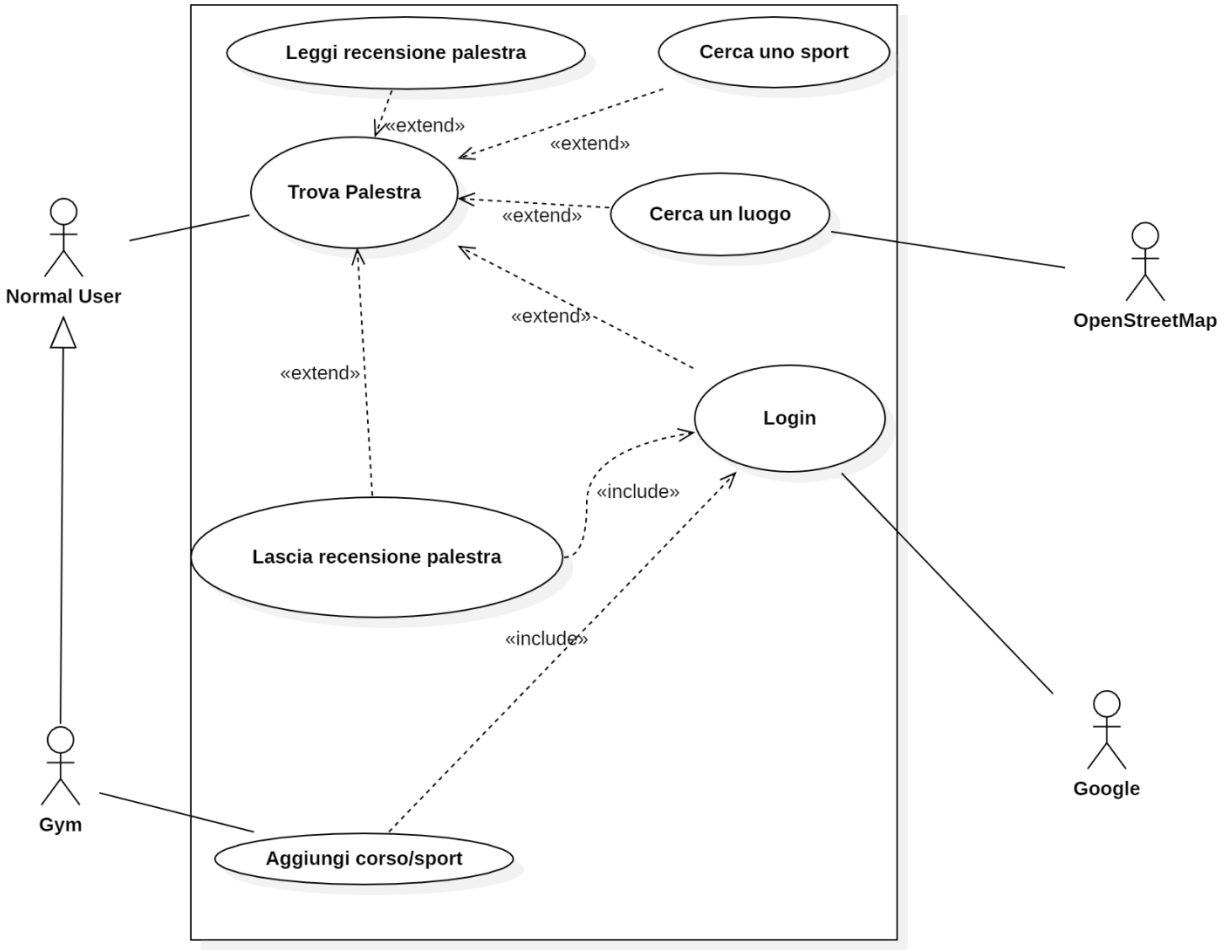
1.3. Functional requirements

(3 per member)

- **The system, based on specific questions, must decide the best sport for the user's needs.**
- **The system must provide the ability to read user reviews.**
- **The system shall provide all gyms in a range of choice from a chosen location in which they can practice the selected sport.**
- **The system creates a list of the ten most practiced sports in Italy based on the number of times a sport has been chosen.**
- **The system must provide the main information about a selected sport.**
- **The system sends an alert to the user who is searching for a particular sport from gyms that offer that sport and that are in a range of choice from a selected location.**
- The system must keep track of the level the user reaches in each sport.
- The system must provide a review of the user experience in the gyms by adding a comment in the appropriate box on the gym profile pages.

1.4. Use cases: Overview Diagram

There are two primary actors (*Normal User, Gym*), one is a generalization of the other (*Gym*), and two secondary actors (*OpenStreetMap, Google*).



2. Storyboards

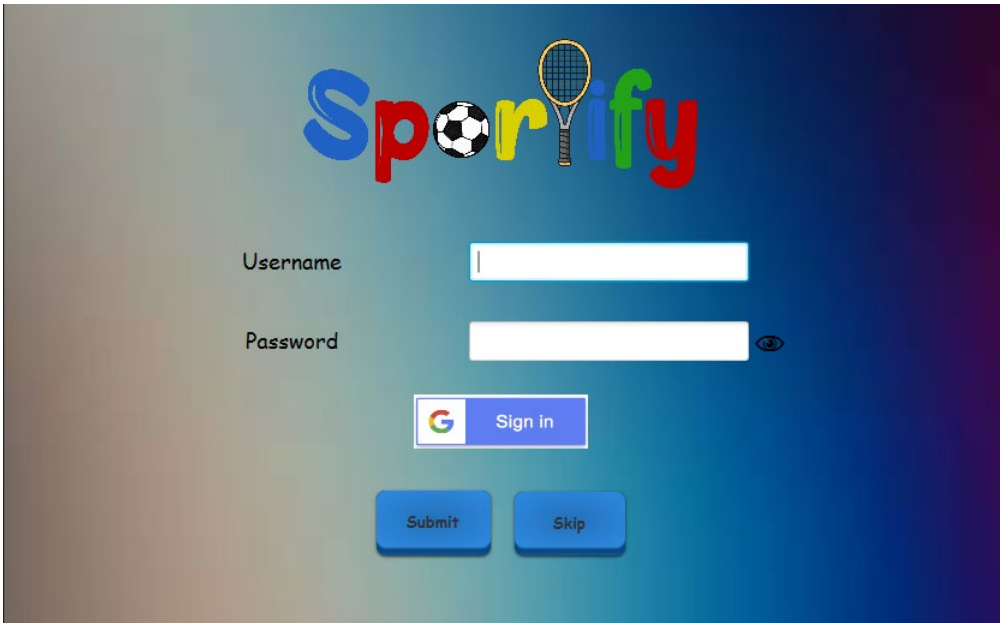
2.1. Format: HTML

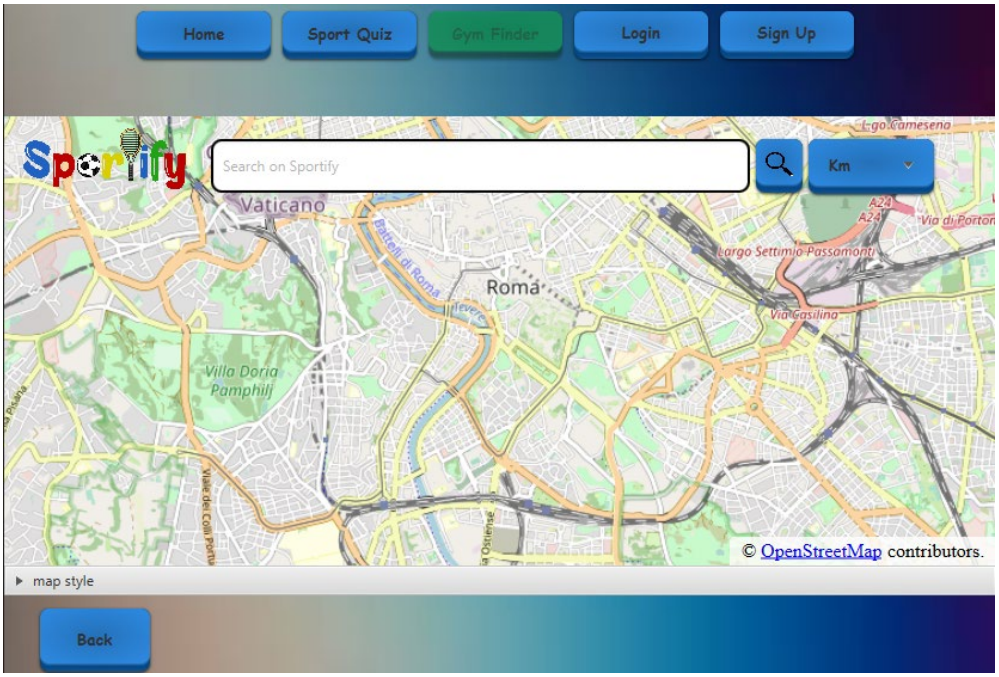
Storyboards in HTML5 format are present in the folder '*../Storyboard*' of the project with its style '*.css*' files in '*../Storyboard/css*' and the assets they need in the '../Storyboard/img'.

2.2. Screens

```
Quantity: two screens per member, covering all the functionalities described in SRS,
developed using Draw.io or similar
```

The screens present the gym login screen and the search screen through the map.






Home
Sport Quiz
Gym Finder
Login
Sign Up
Sportify
How old are you?
0 - 18
19 - 30
31 - 50
over 50
Back
Next


Home
Sport Quiz
Gym Finder
Login
Sign Up
Sportify
*Username
First Name
Last Name
*Password
*Email
Gym
User
Submit
Skip

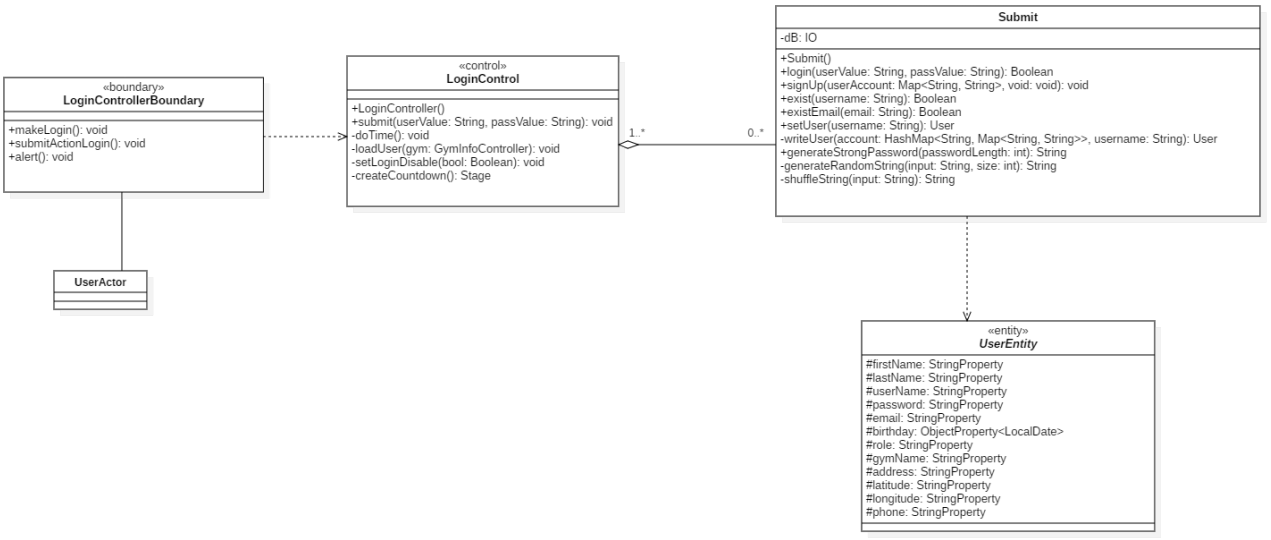
3. Design

3.1. Class diagram

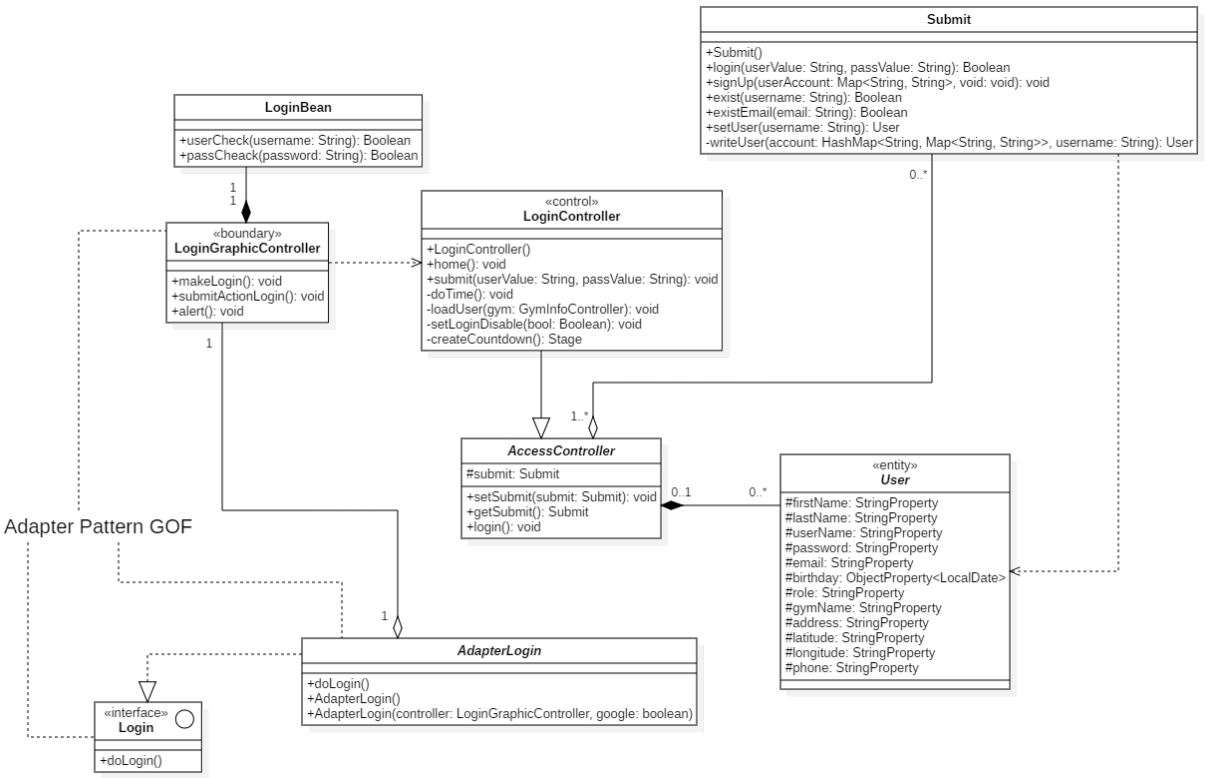
- *1 VOPC per member. (analysis)*
- *One design-level diagram per member (e.g., that includes patterns or specific solutions that improve the engineering level of the system)*

The class diagrams files are also available in pdf in the folder "*..\Document\Documentation*"

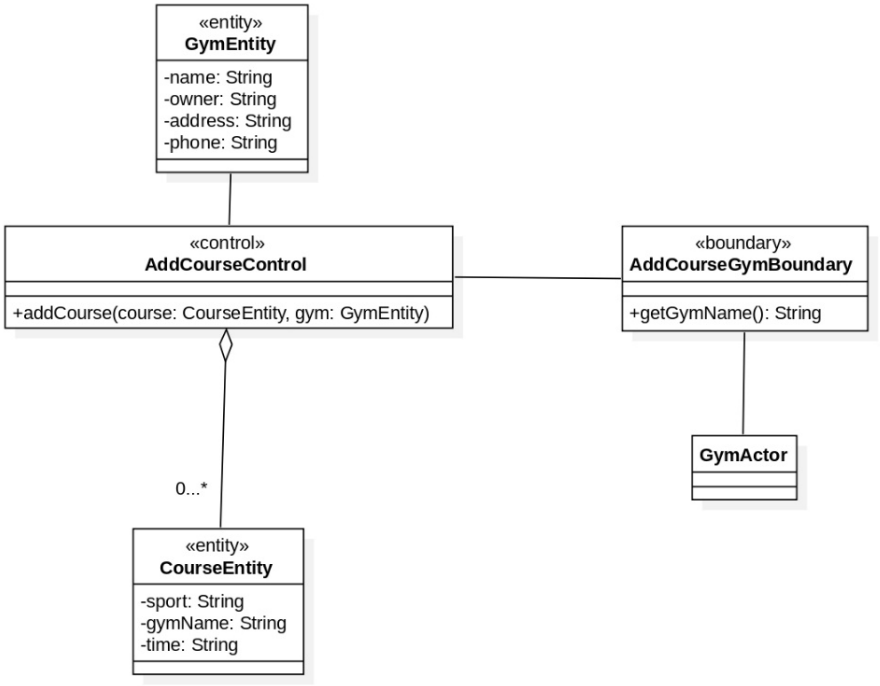
Login VOPC



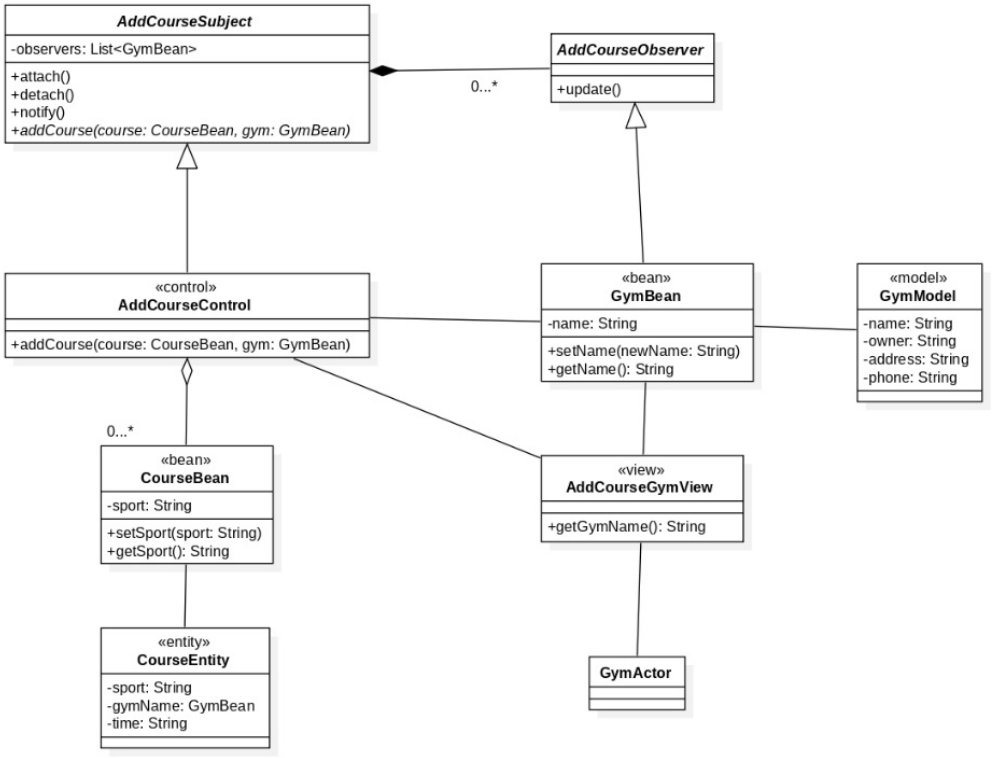
Login Design Level



AddCourse VOPC



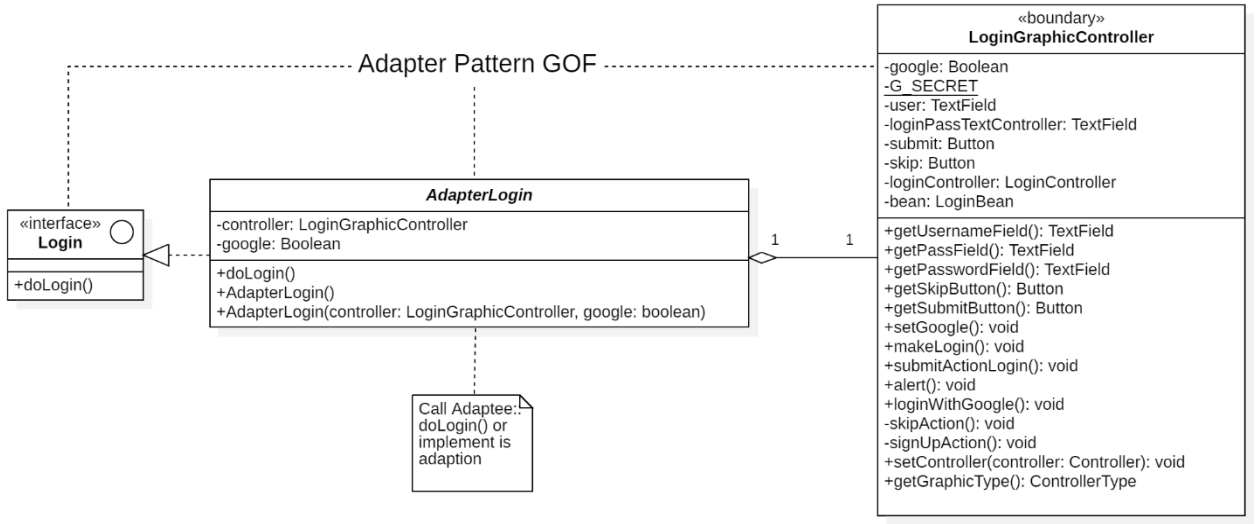
AddCourse Design Level



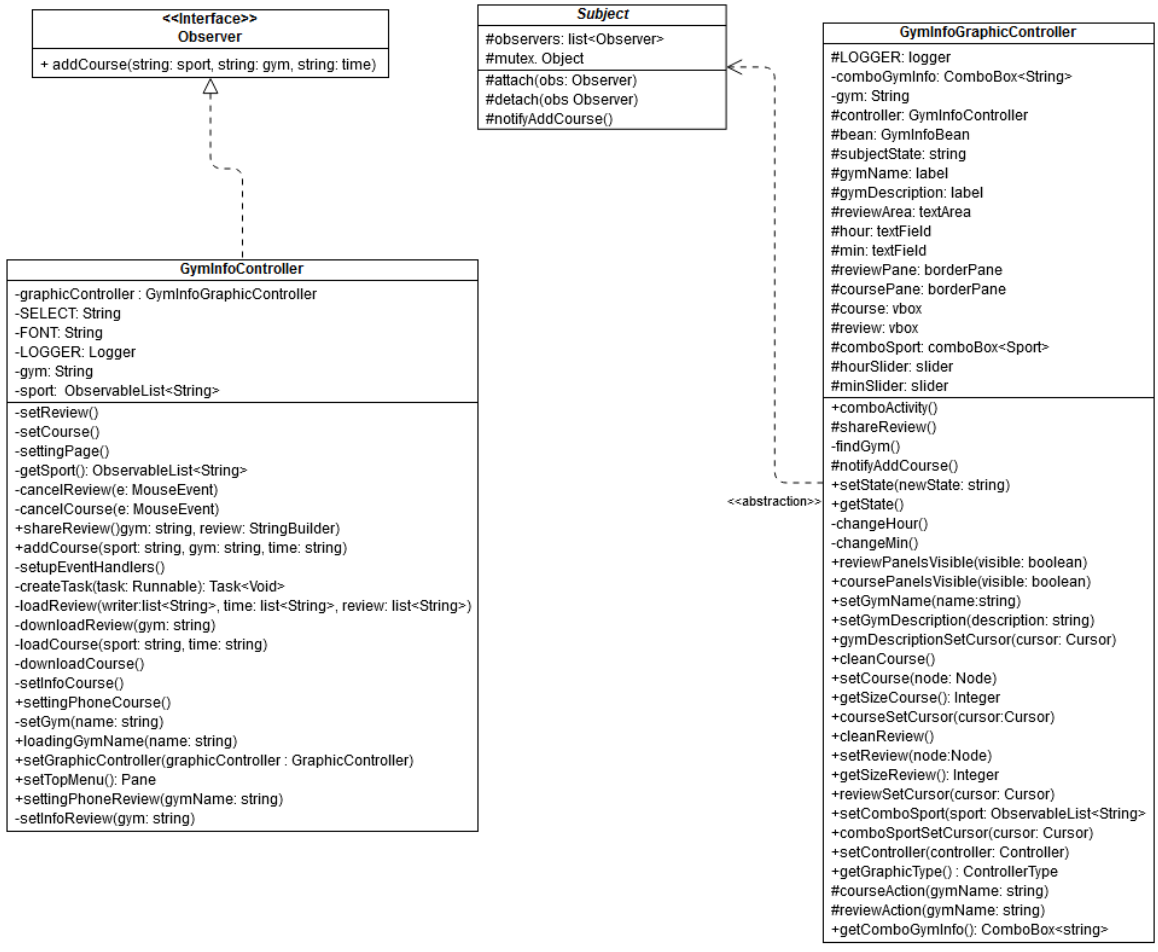
3.2. Design Patterns

One different pattern per member. Possibly, try applying the pattern within the context of the project.

Adapter pattern GOF

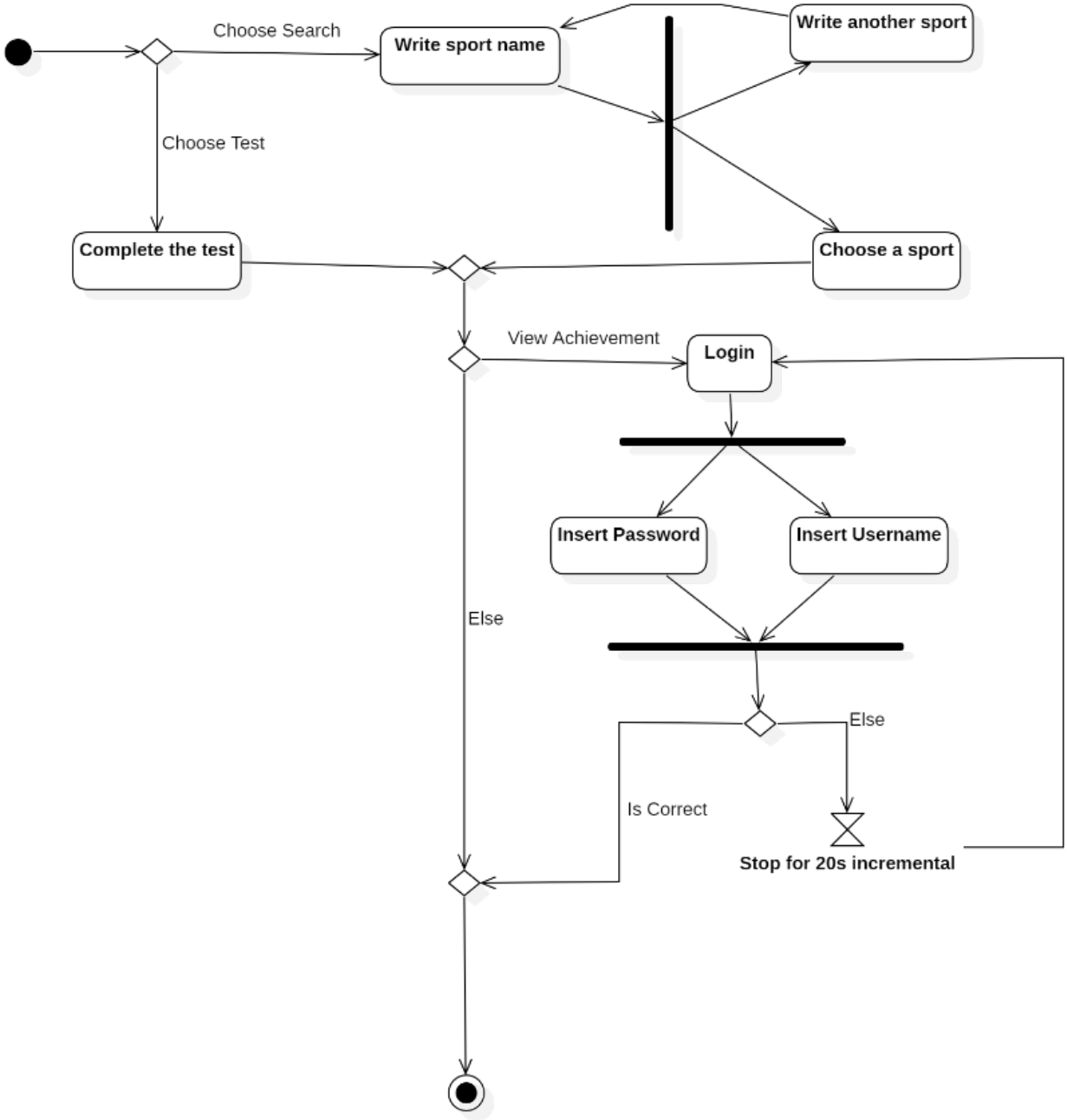


Observer pattern GOF

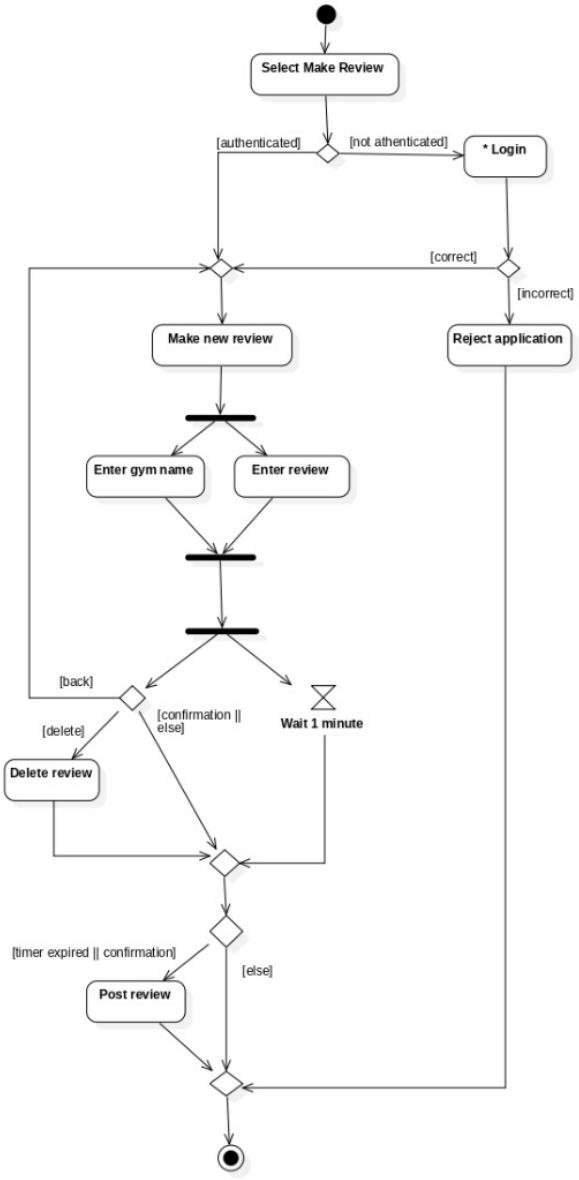


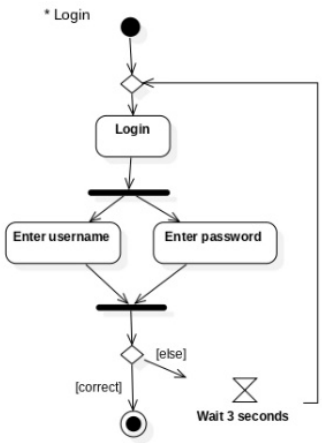
3.3. Activity diagram

One per member.
Login Activity Diagram (La Gioia)



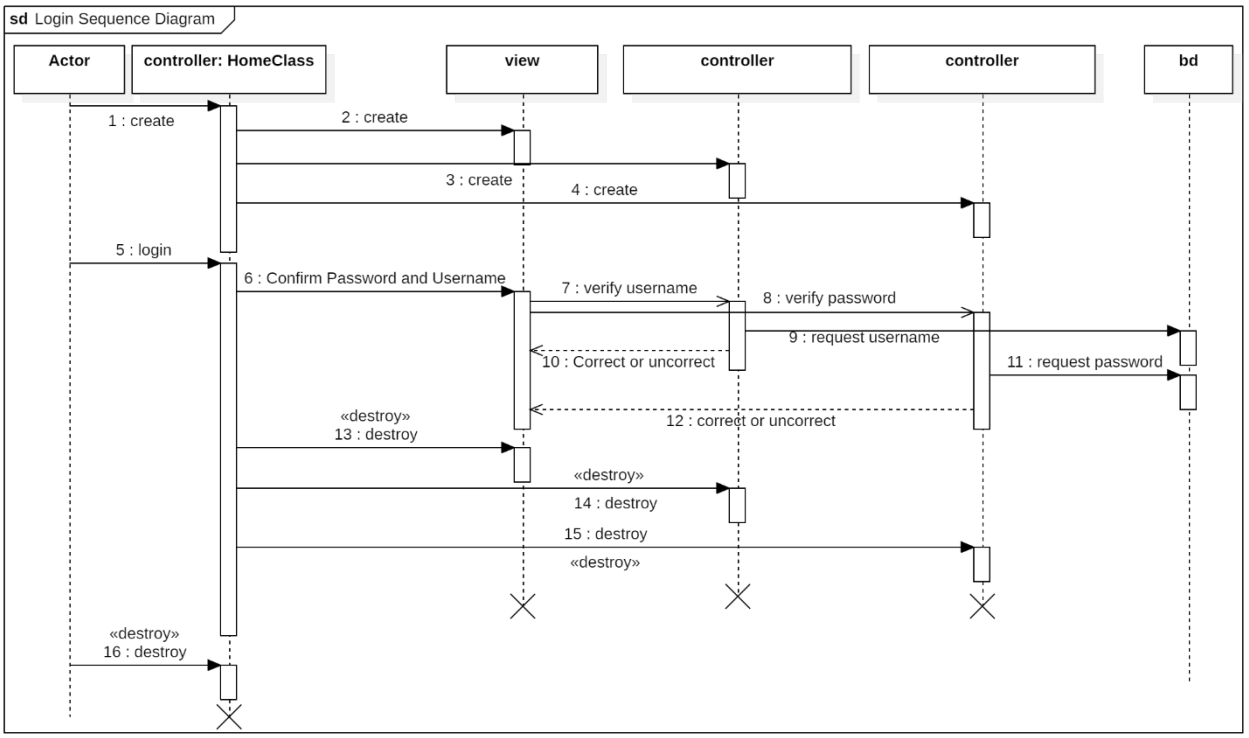
`Make Review Activity Diagram(D'Ambrogio)`



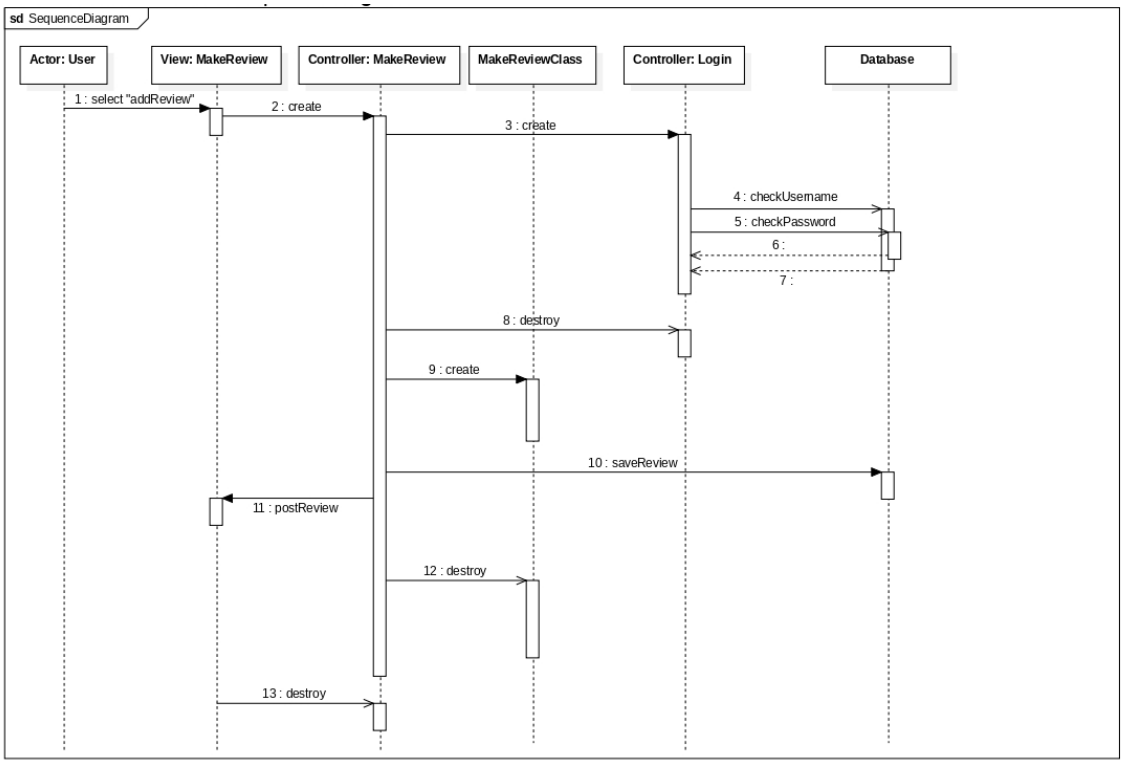


3.4. Sequence diagram

One per member.



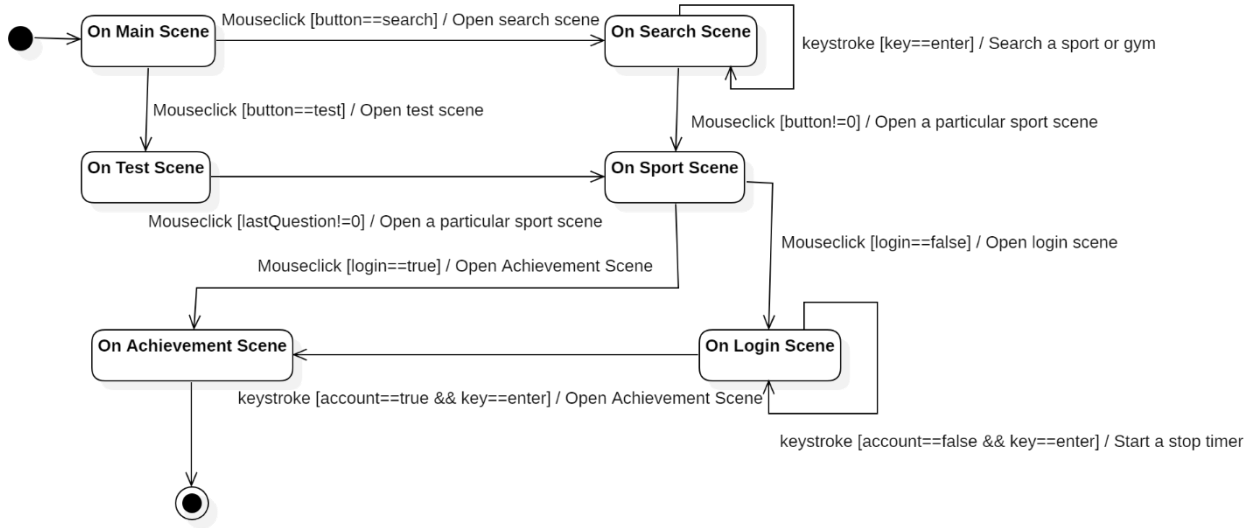
Make Review Sequence Diagram(D'Ambrogio)



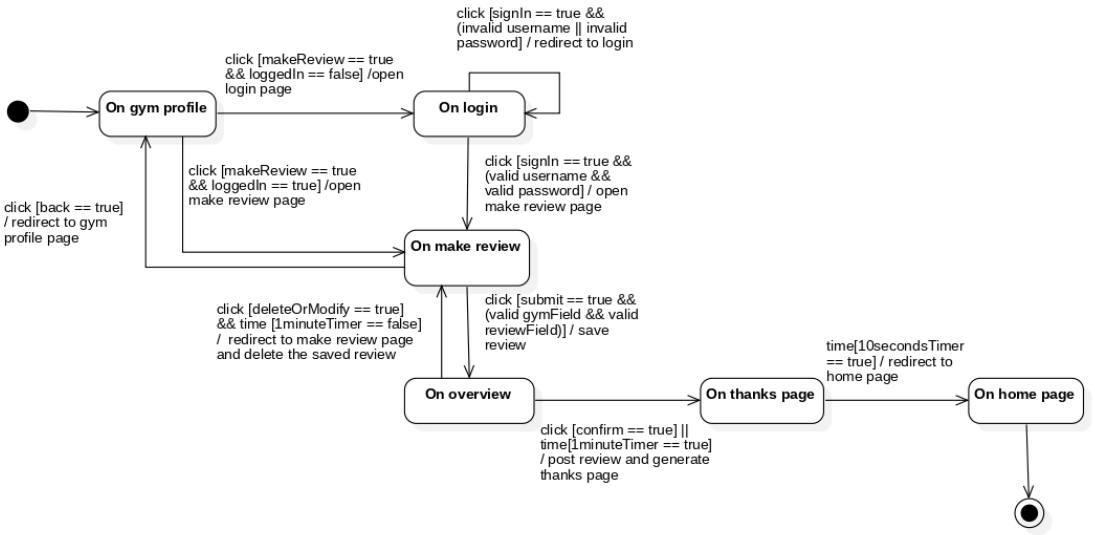
3.5. State diagram

One per member.

Sport State Diagram



Make Review State Diagram(D'Ambrogio)



4. Testing

4.1. Test cases

Develop at least three test cases per person. In each test (class) file, report (via Java comments) the name of the person in charge.

The tests we have implemented are for Matteo: '*src/test/java/sportify/HomePhoneTest.java*', '*src/test/java/sportify/LoginTest.java*', '*src/test/java/sportify/MainAppTest.java*', while for Chiara: '*src/test/java/sportify/MenuTest.java*', '*src/test/java/sportify/QuizPhoneTest.java*', '*src/test/java/sportify/QuizTest.java*', which test the corresponding functions of the screens.

4.2. Selenium GUI Test

1 Selenium test using the GUI per member.

The two Selenium GUI tests are present in the project test folder in the 'selenium' package.

4.3. Test Selenium API

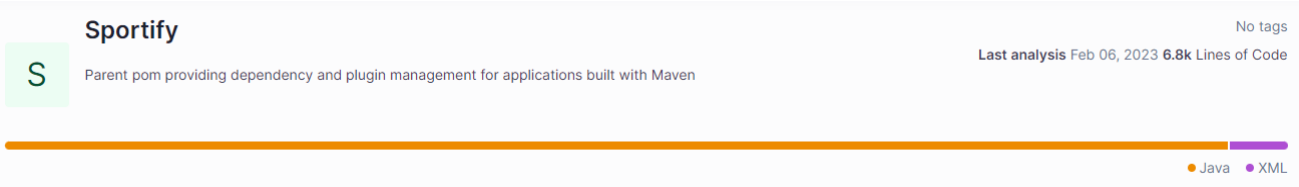
1 Selenium test through API per member.

The two Selenium API tests are present in the project test folder in the 'selenium' package.

5. Code

5.1. ~4K LOC

6.8K LOC has been programmed for the entire program.



5.2. Similar functionality implemented with GUIs.

5.3. Exceptions

at least two per member (do not just catch and back-propagate the exceptions, but effectively manage them. Possibly, define your own error logic by means of exceptions).

The completely managed are:

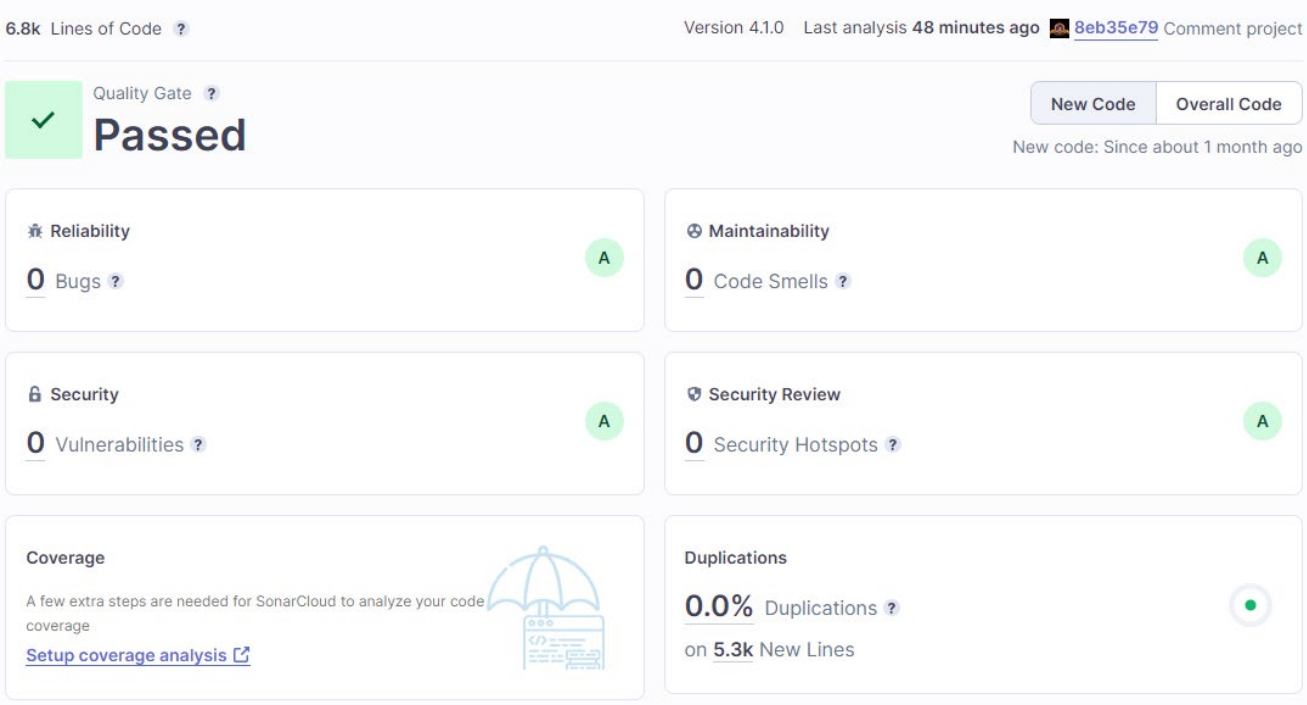
- Exception due to the inability to connect to the database: first try the connection one more time and then conclude in case of denied connection with an alert message to the user via graphic controller.
- Exception due to incorrect entry of username and/or password: Exception managed in logical controllers using the NewException class that allows the exception to be thrown.

- Exception due to incorrect age entry in the case of the test done via smartphone screen: exception managed in logical controllers through the NewException class that allows the exception to be thrown.
- Exception due to the erroneous click on the map of the position indicator and not of a gym: managed by alert message for the user sent via logical controller.

5.4. Svn (or Git) + SonarCloud

Show that Svn (or Git) + SonarCloud is correctly installed on one of your computers, and it can analyze your project for rule violations. No rule must be violated (no smells, no vulnerabilities, no bugs. This will be checked during the exam.

The project after analyzing SonarCloud does not present code smells, vulnerabilities, or bugs.



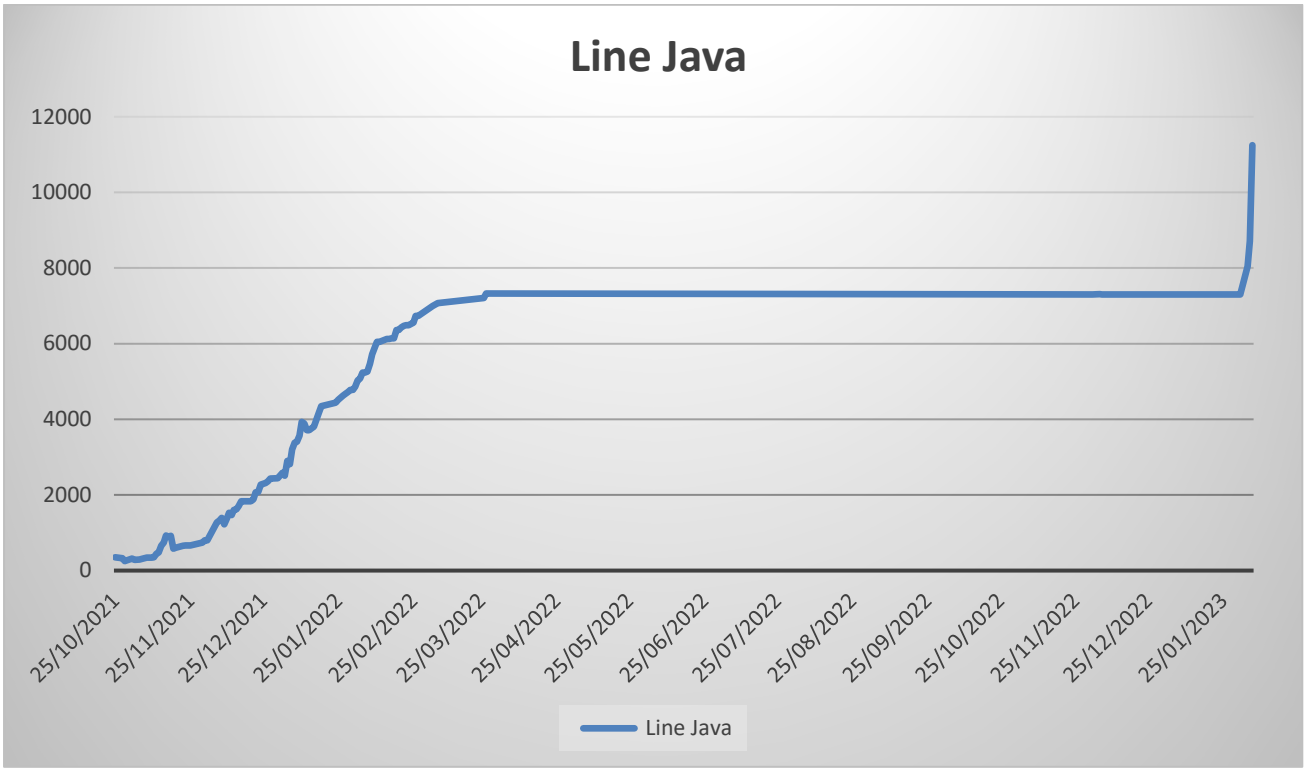
5.5. DAO

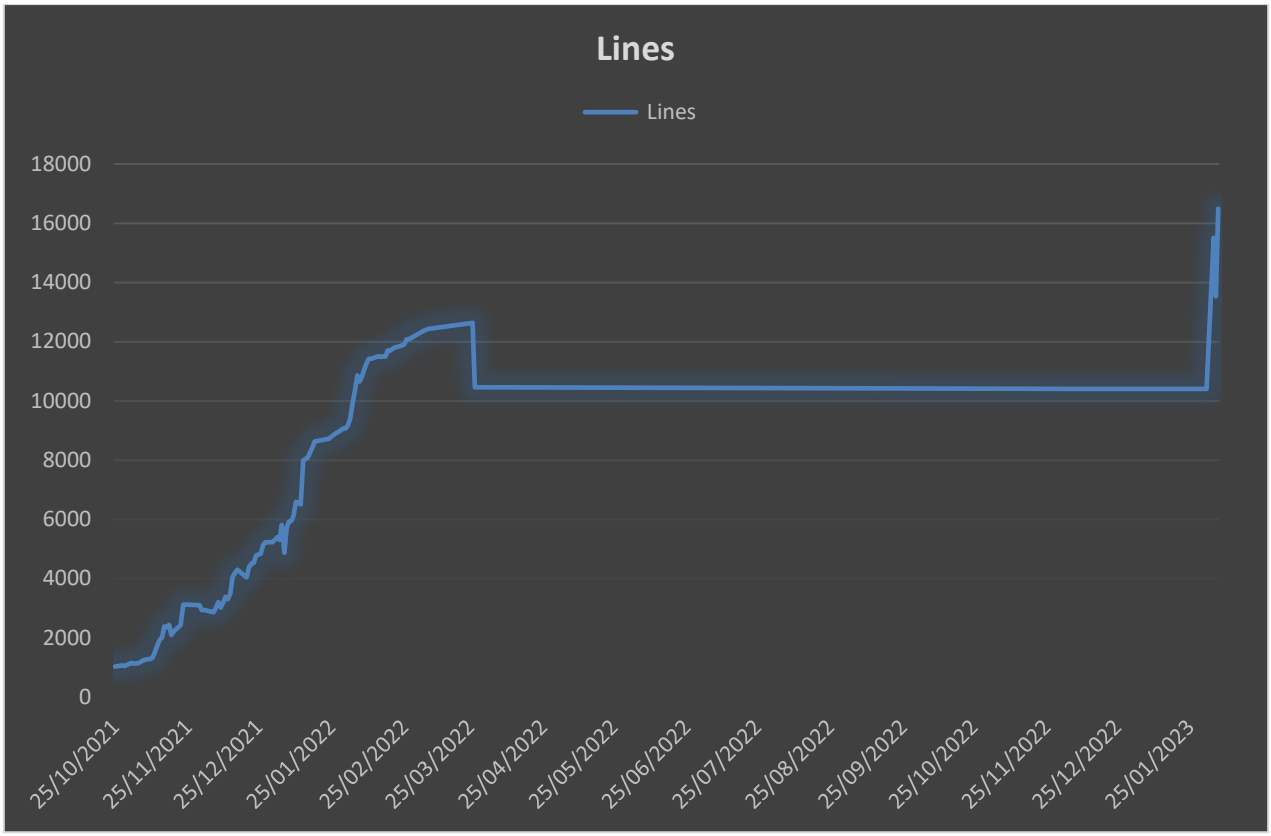
One DAO shall be provided in two versions, DMBS and file system.

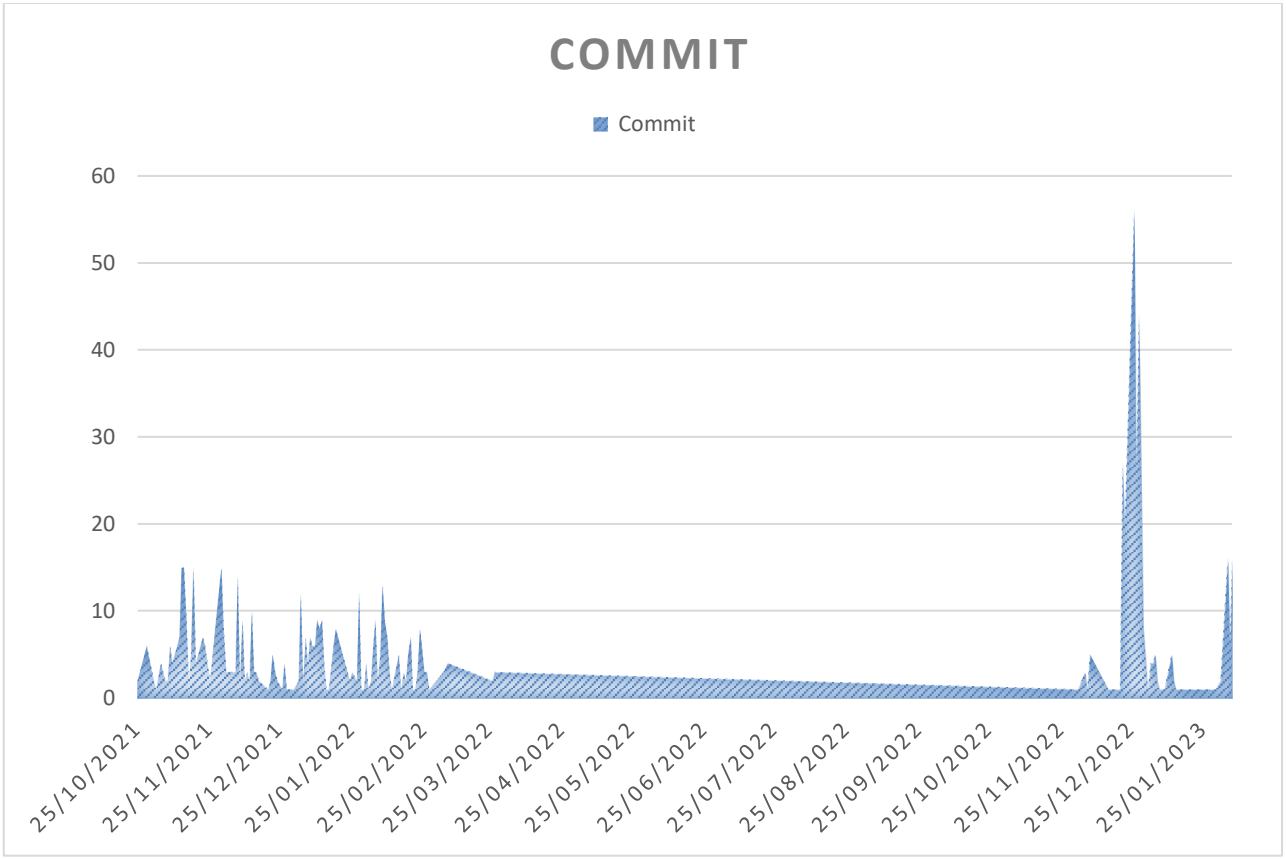
The files with which the database has been set up are present in the folder '*.. /DB*' and the database is controlled by the application through the *DAO*, *IO* and *DBConnection* class.

6. Analytics

Provide a process control chart as explained in the slides.




Lines
Lines
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
25/10/2021
25/11/2021
25/12/2021
25/01/2022
25/02/2022
25/03/2022
25/04/2022
25/05/2022
25/06/2022
25/07/2022
25/08/2022
25/09/2022
25/10/2022
25/11/2022
25/12/2022
25/01/2023


COMMIT
Commit
60
50
40
30
20
10
0
25/10/2021
25/11/2021
25/12/2021
25/01/2022
25/02/2022
25/03/2022
25/04/2022
25/05/2022
25/06/2022
25/07/2022
25/08/2022
25/09/2022
25/10/2022
25/11/2022
25/12/2022
25/01/2023

7. Video

A 1- to 2-minute recorded video of the developed system performing the expected functions.
**.mpeg*

In the "*..\Document\Documentation*" folder, there are two videos, each related to a graphical interface showing the same functionality.