



Ingegneria del Software e Progettazione Web
Progetto A.A. 2022/2023

SPORTIFY

Chiara D'Ambrogio
Matteo La Gioia

Indice

1. Software Requirement Specification	3
1.1. Introduction	3
1.1.1. The aim of Document	3
1.1.2. Overview of the defined system	3
1.1.3. Operational settings	3
1.1.4. Related Systems	4
1.2. User Stories	5
1.3. Functional requirements	5
1.4. Use cases: Overview Diagram	6
2. Storyboards	7
2.1. Format: HTML	7
2.2. Screens	7
3. Design	9
3.1. Class diagram	9
3.2. Design Patterns	11
3.3. Activity diagram	12
3.4. Sequence diagram	14
3.5. State diagram	15
4. Testing	16
4.1. Test Cases	16
4.2. Selenium GUI Test	16
4.3. Test Selenium API	16
5. Code	16
5.1. ~4K LOC	17
5.2. Similar functionality implemented with GUIs	17
5.3. Exceptions	17
5.4. Svn (or Git) + SonarCloud	17
5.5. DAO	18
6. Analytics	18
7. Video	21

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/it/uniroma2/dicii/ispw/sportify/MainAppLauncher.java](#)'.

1.1.4. Related Systems

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

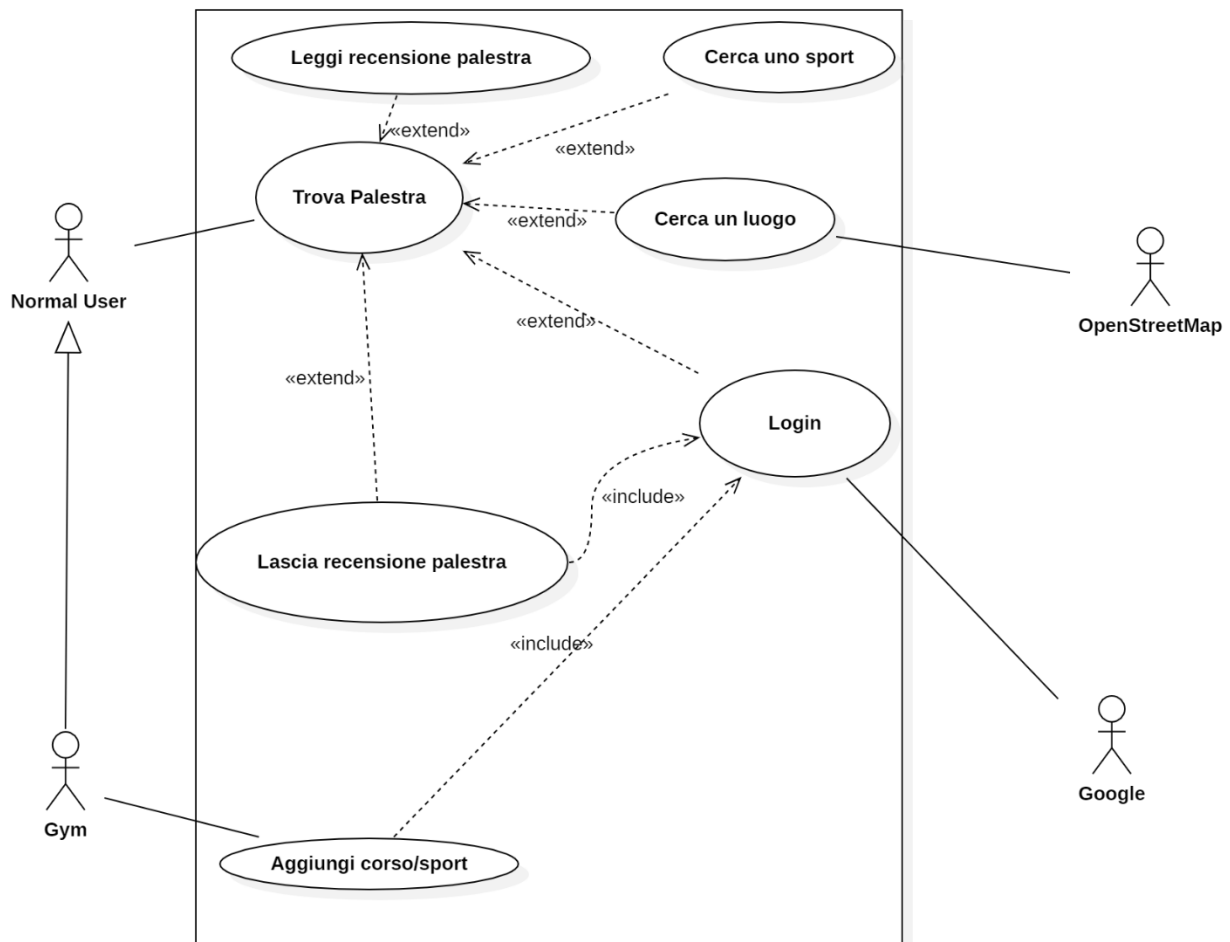
- As a new user, I want to discover the best suitable sport for me considering my needs so that I can start practice it.
- As a user, I want to review a gym based on my experience so that it could be useful to other users.
- As a gym owner, I want to know the ten most popular sports practiced in Italy, 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

- 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*).



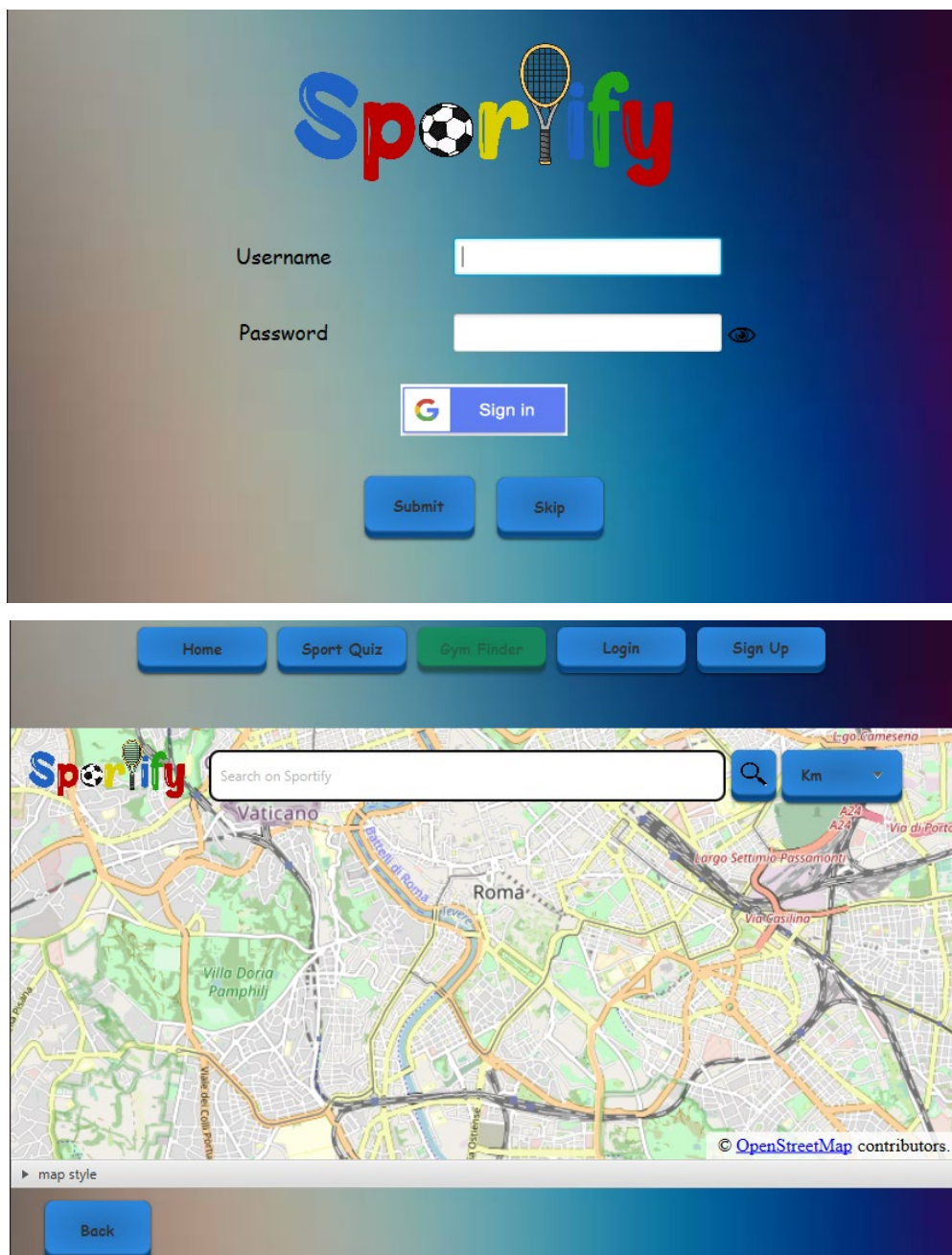
2. Storyboards

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

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





Home Sport Quiz Gym Finder Login Sign Up


Sporify

How old are you?

0 - 18 19 - 30

31 - 50 over 50

Back Next



Home Sport Quiz Gym Finder Login Sign Up

Sporify

*Username

First Name

Last Name

*Password

*Email

☐ Gym ☒ User

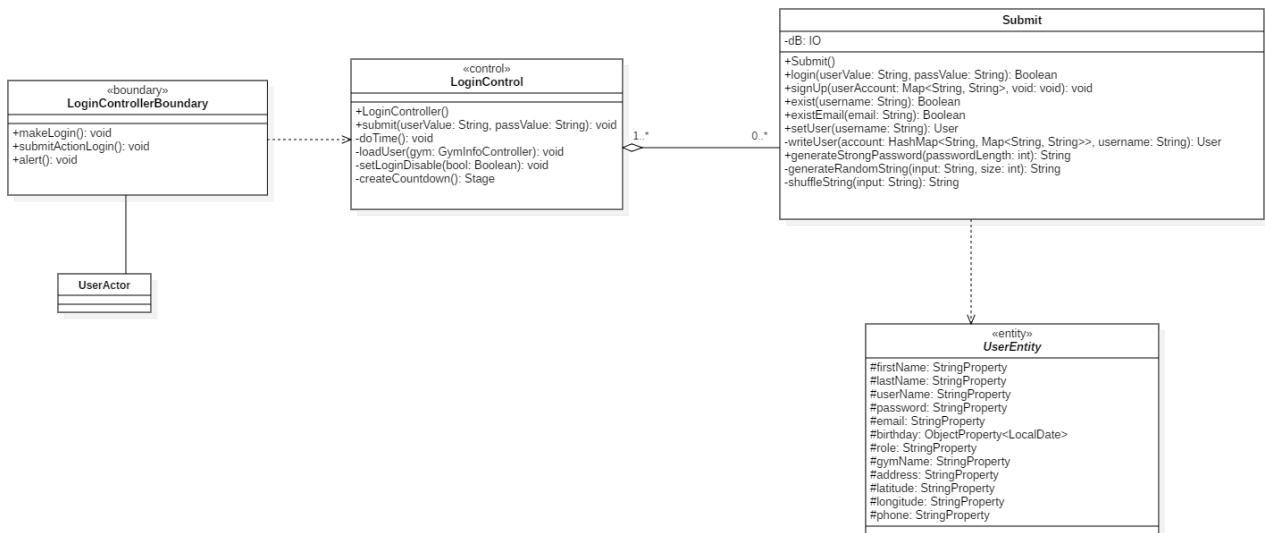
Submit Skip

3. Design

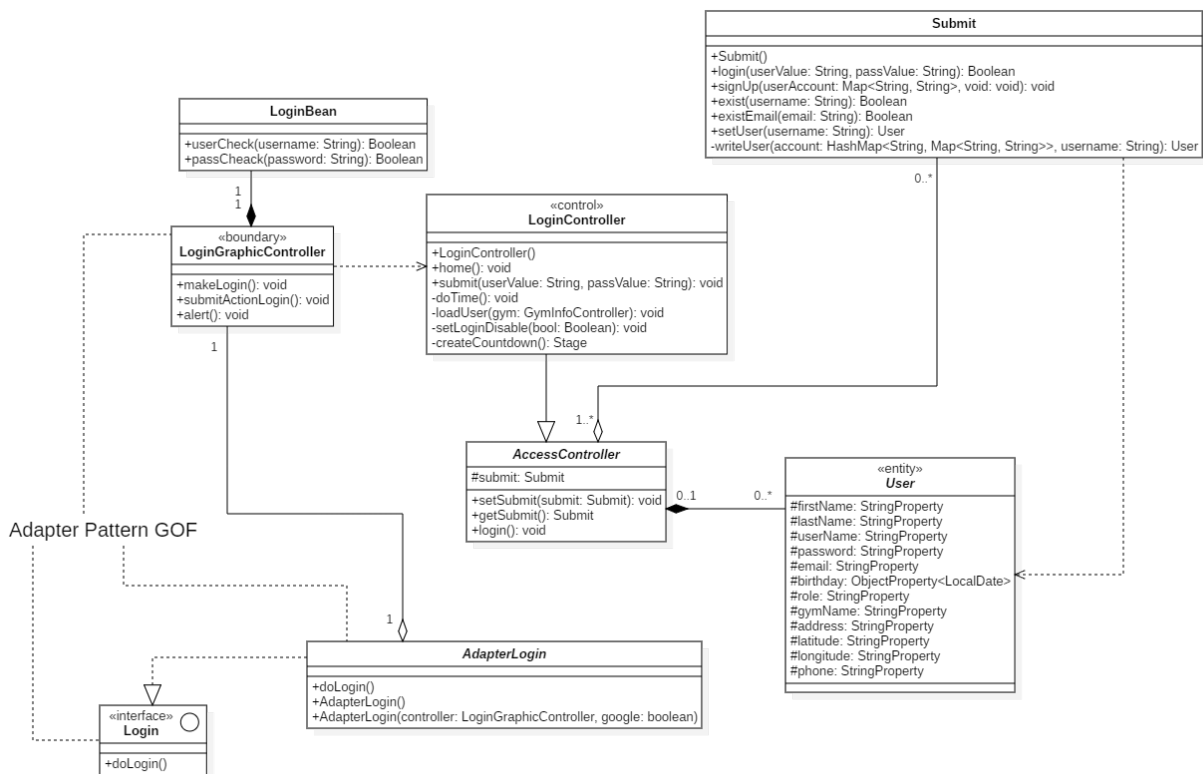
3.1. Class diagram

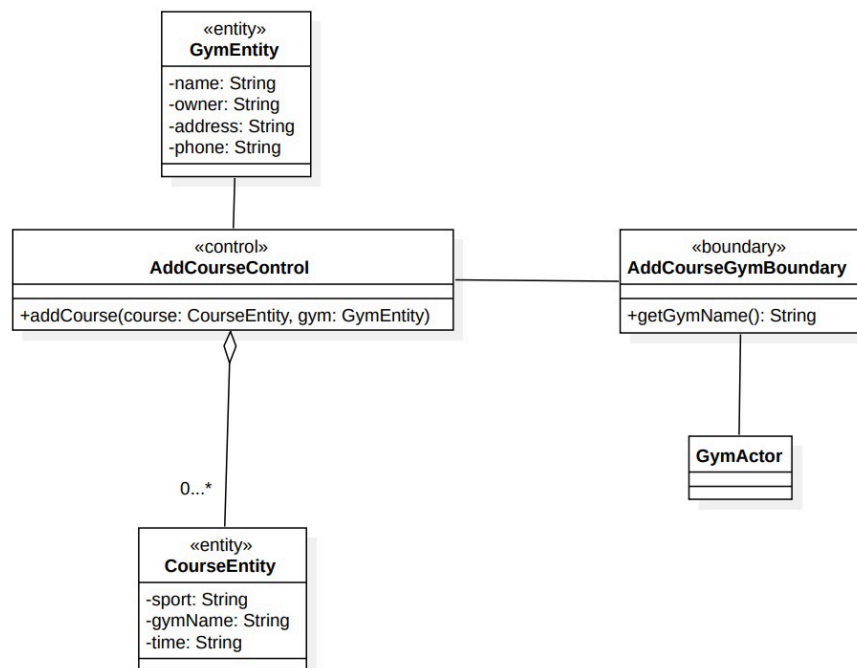
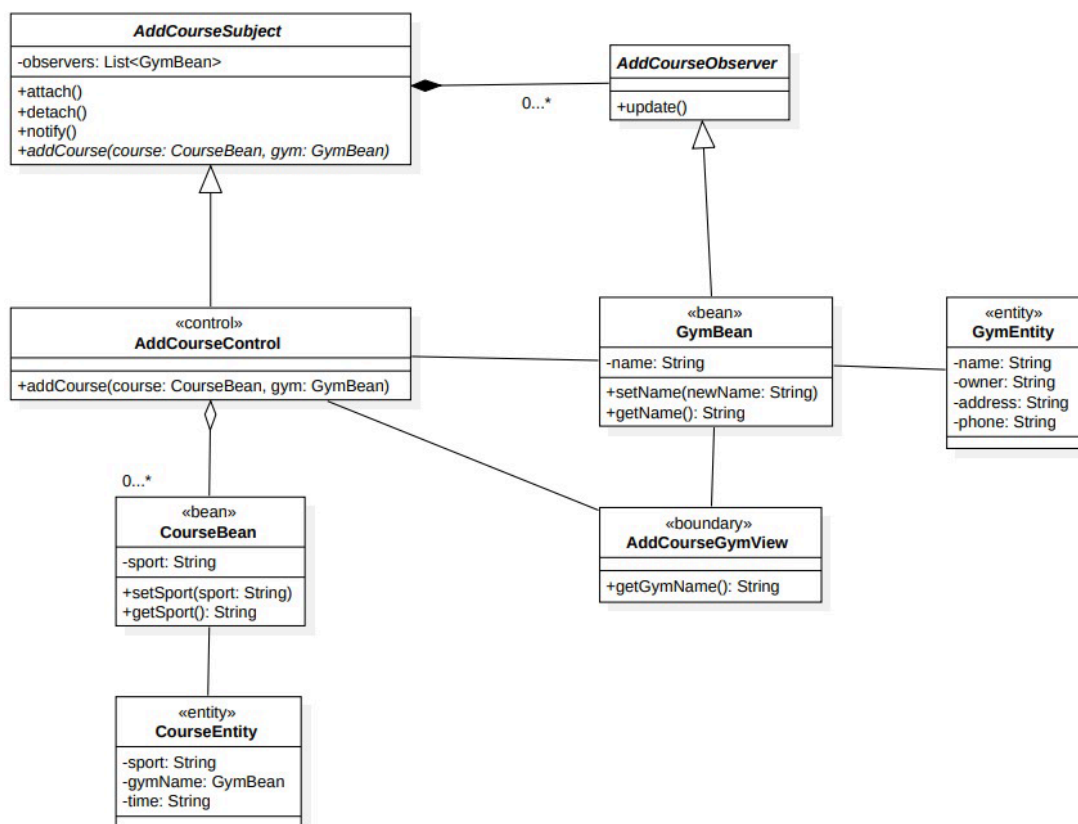
The class diagrams files are also available in pdf in the folder “[../Document/Documentation/Diagram](#)”

Login VOPC



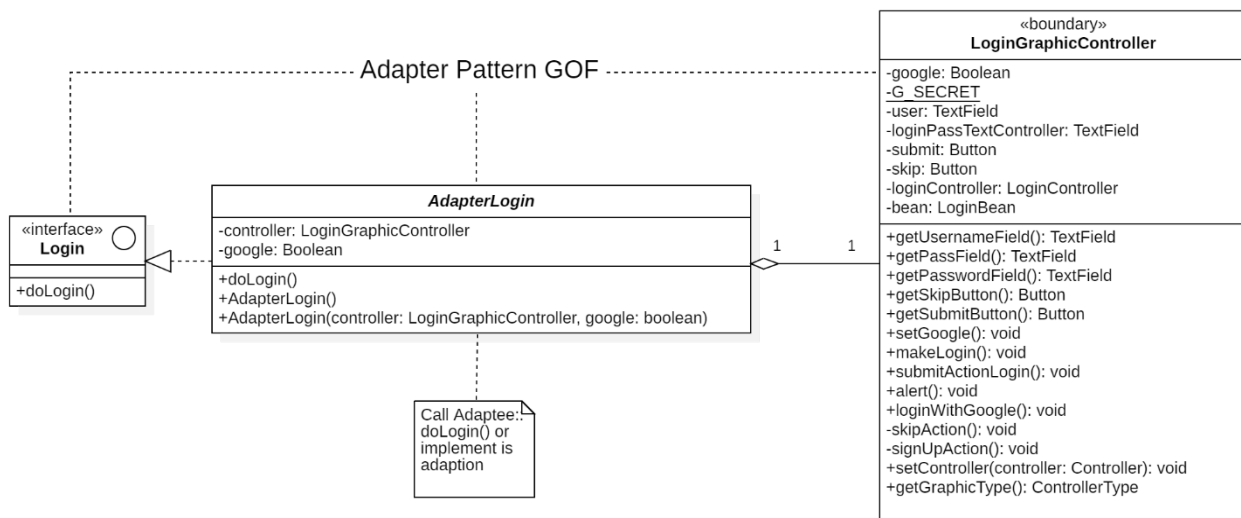
Login Design Level



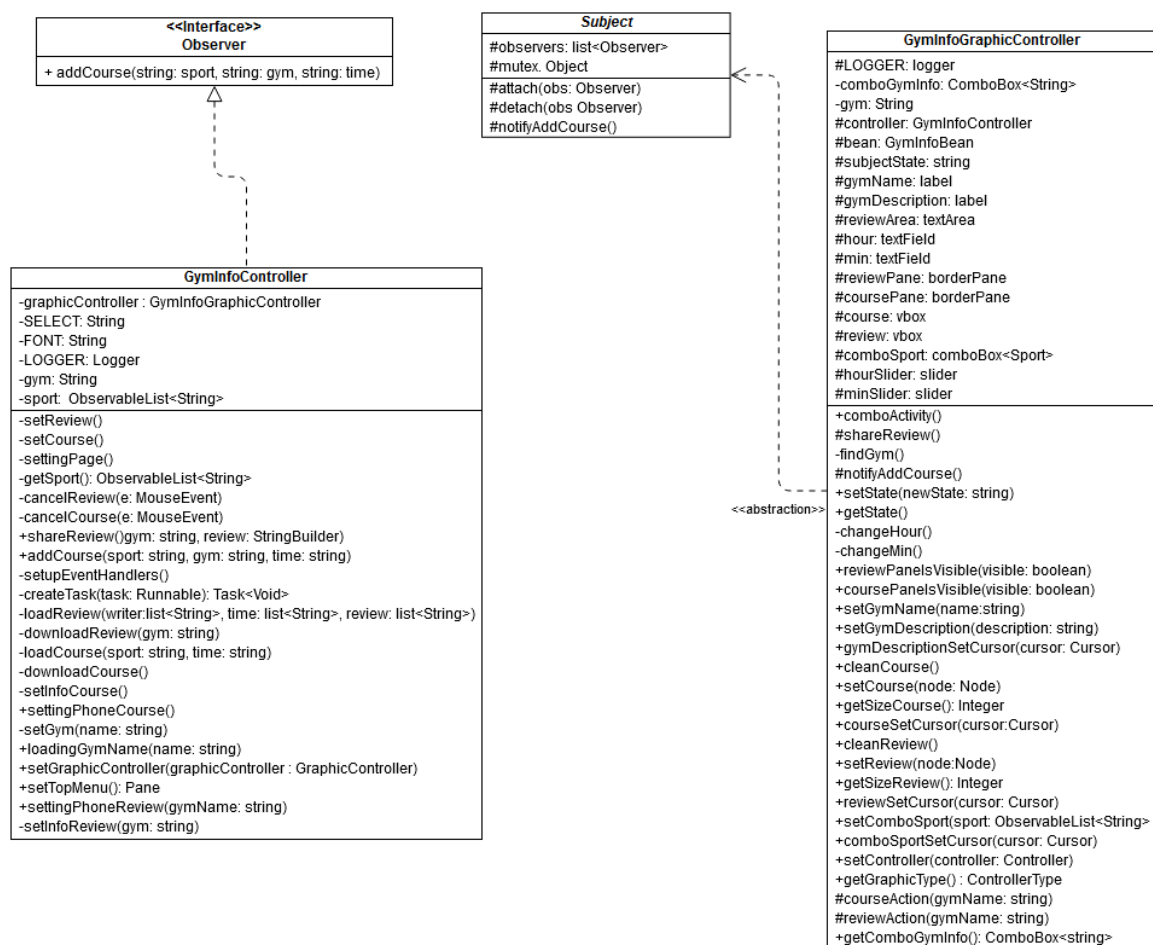
AddCourse VOPC*AddCourse Design Level*

3.2. Design Patterns

Adapter pattern GOF

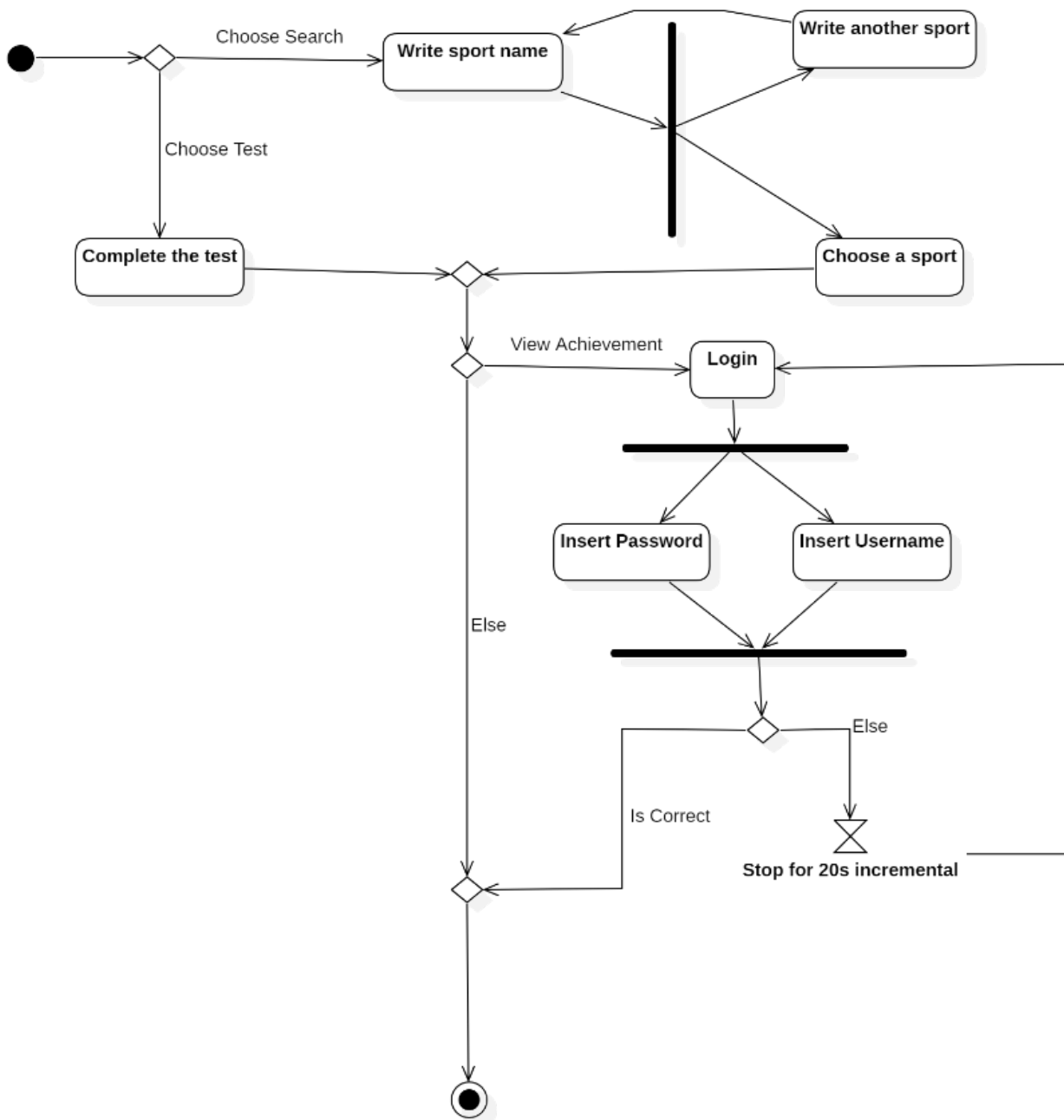


Observer pattern GOF

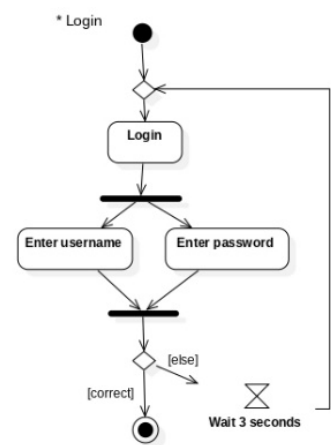
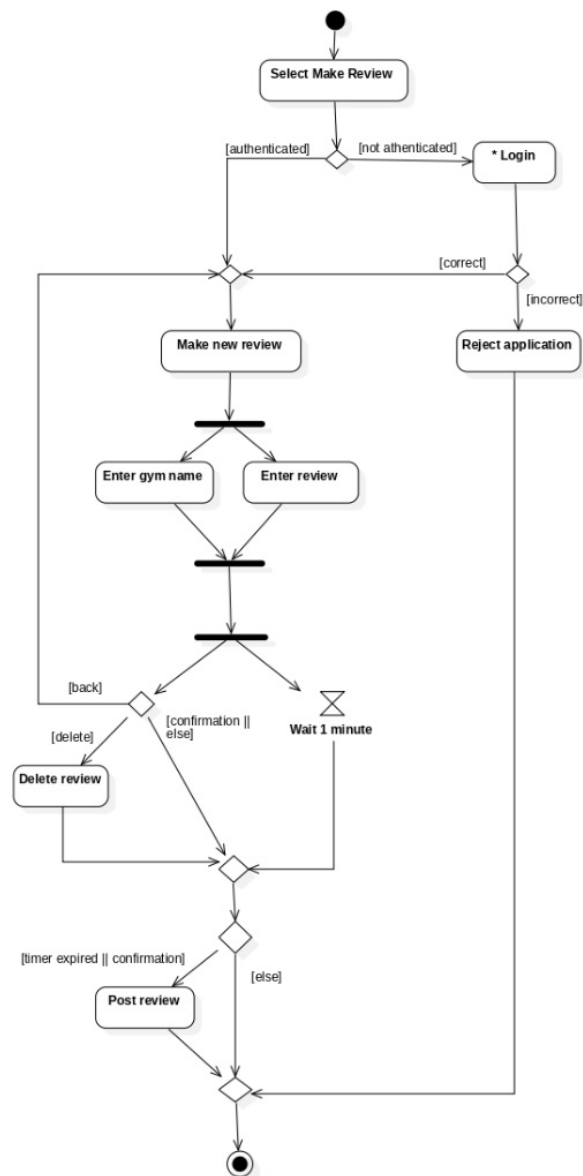


3.3. Activity diagram

Login Activity Diagram

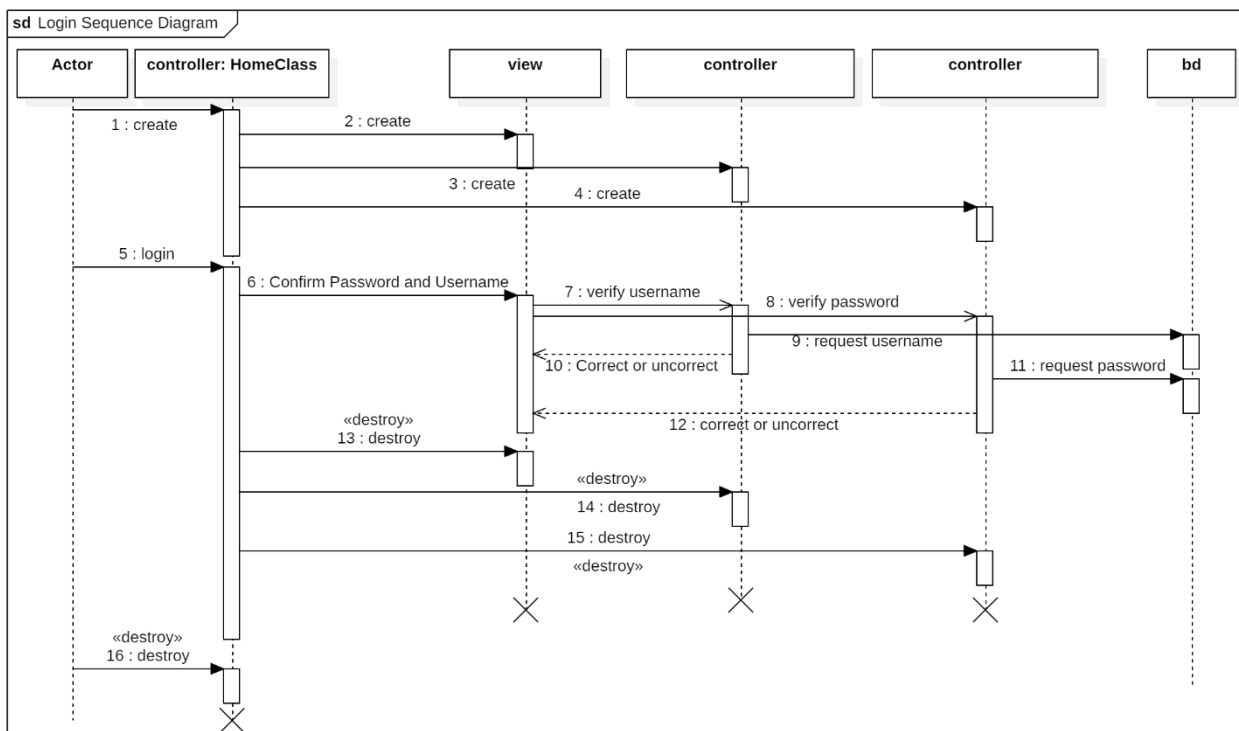


Make Review Activity Diagram

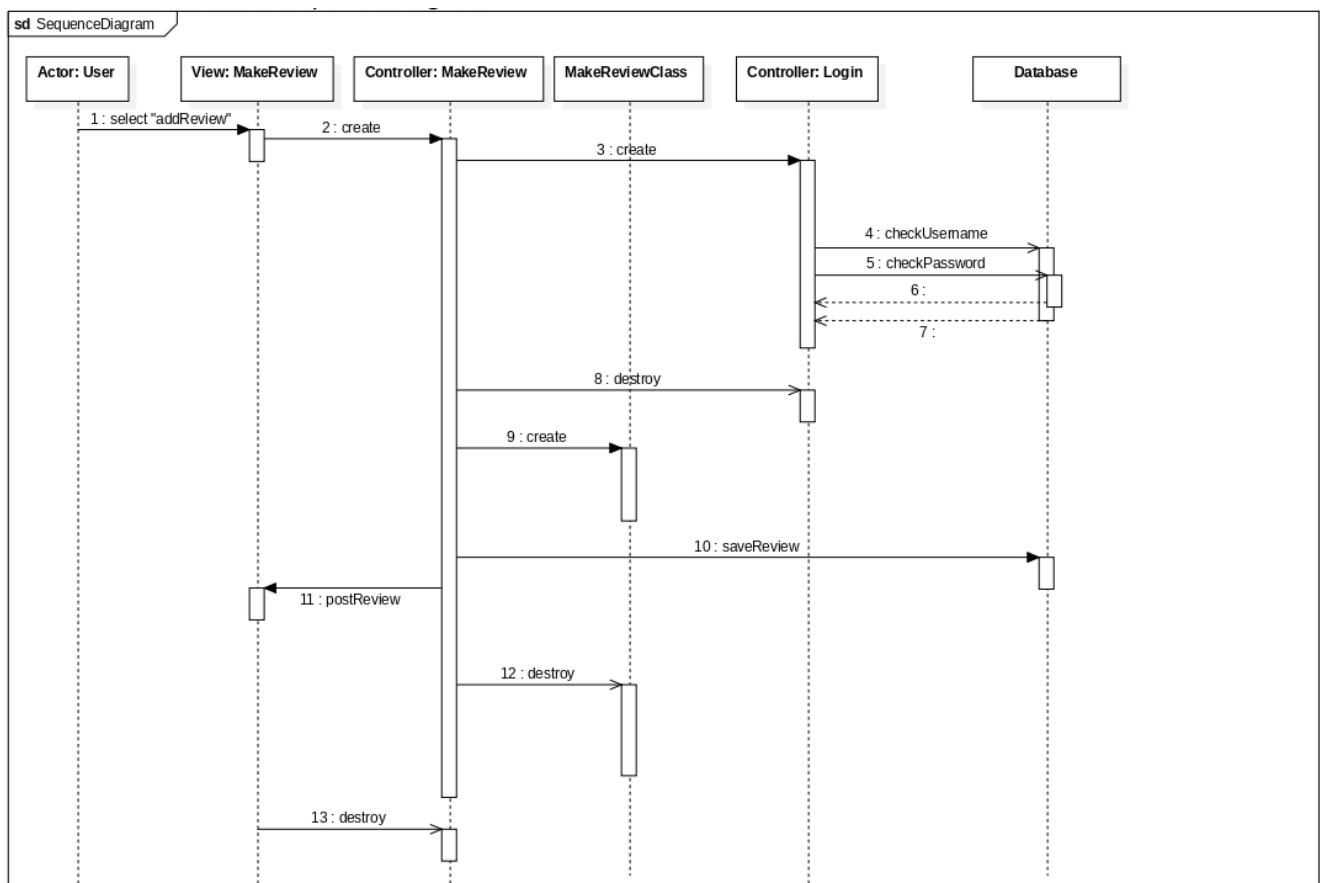


3.4. Sequence diagram

Login Sequence Diagram

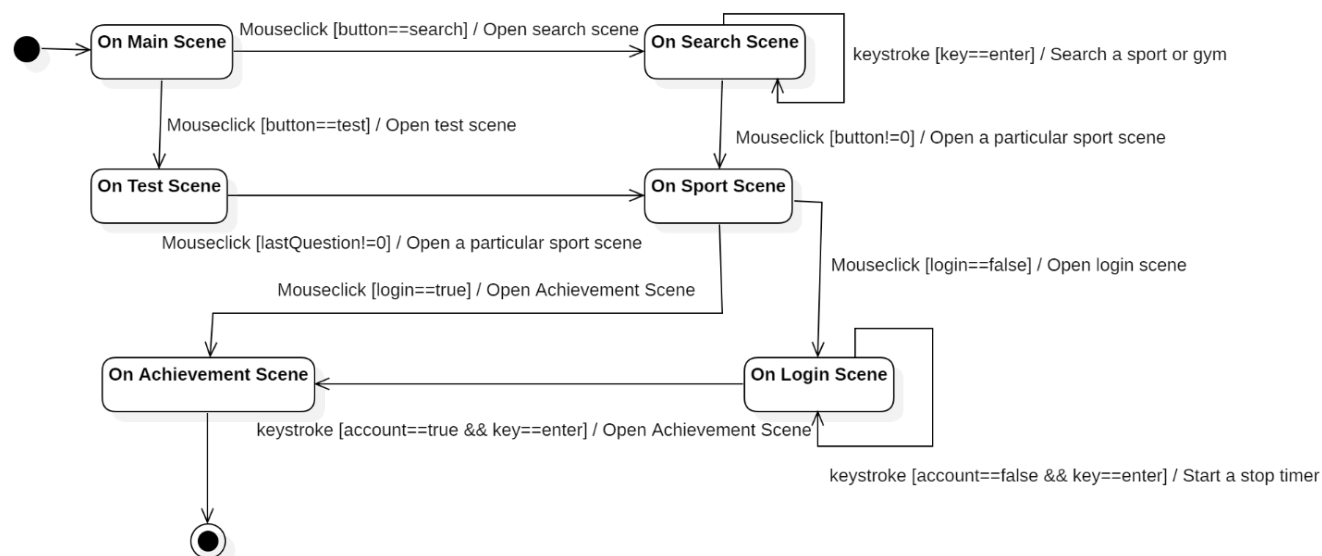


Make Review Sequence Diagram

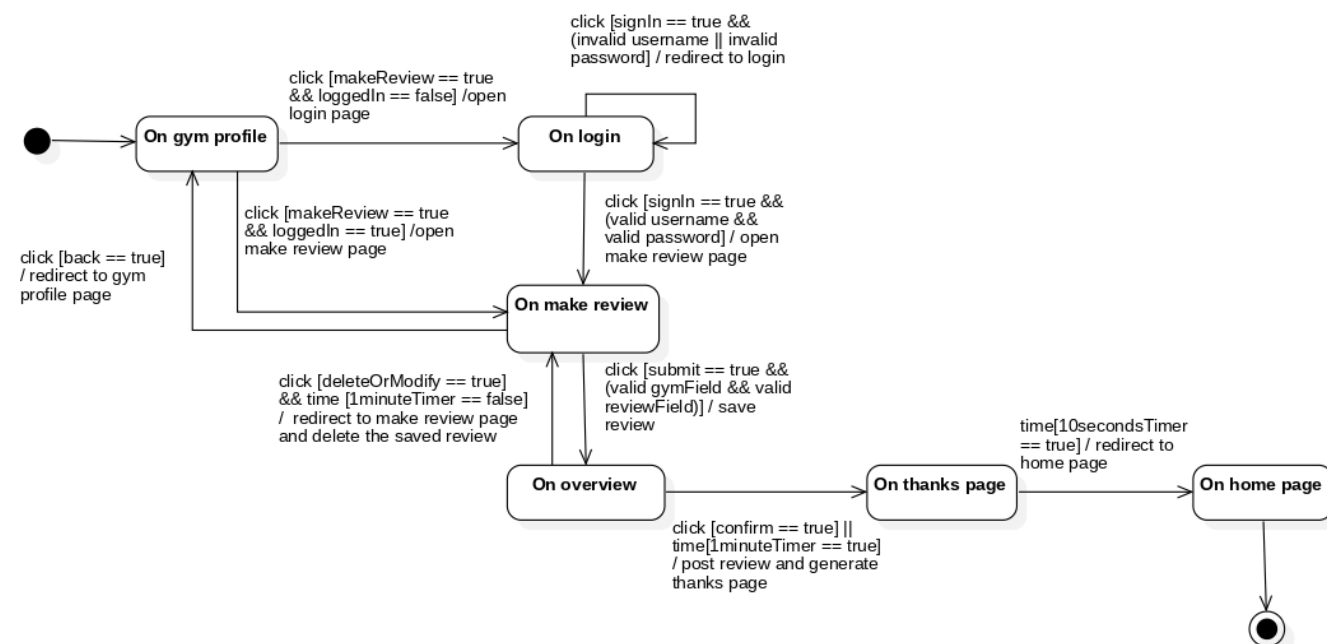


3.5. State diagram

Sport State Diagram



Make Review State Diagram



4. Testing

4.1. Test cases

The tests we have implemented are:

['src/test/java/it/uniroma2/dicii/ispw/sportify/HomePhoneTest.java'](#),

['src/test/java/it/uniroma2/dicii/ispw/sportify/LoginTest.java'](#),

['src/test/java/it/uniroma2/dicii/ispw/sportify/MainAppTest.java'](#),

['src/test/java/it/uniroma2/dicii/ispw/sportify/MenuTest.java'](#),

['src/test/java/it/uniroma2/dicii/ispw/sportify/QuizPhoneTest.java'](#),

['src/test/java/it/uniroma2/dicii/ispw/sportify/QuizTest.java'](#),

which test the corresponding functions of the screens.

4.2. Selenium GUI Test

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

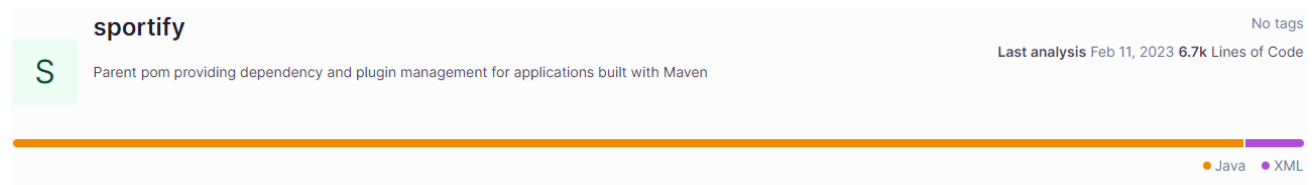
4.3. Test Selenium API

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

5. Code

5.1. ~4K LOC

6.7K LOC has been programmed for the entire program.



5.2. Similar functionality implemented with GUIs.

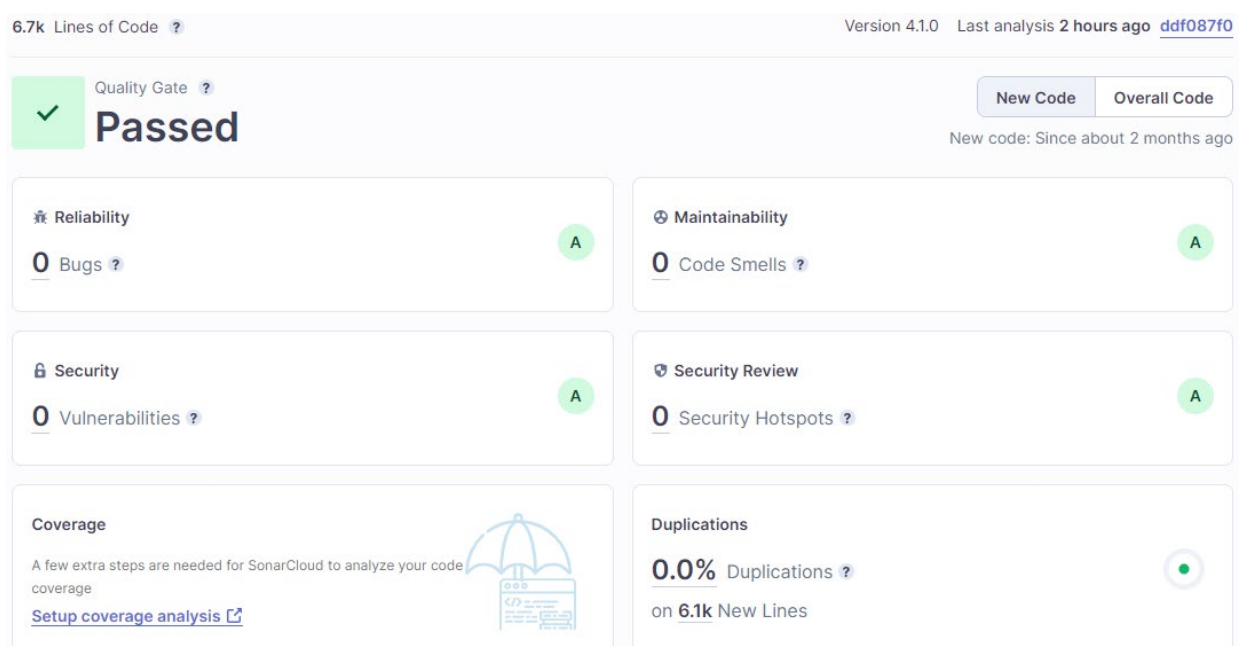
5.3. Exceptions

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

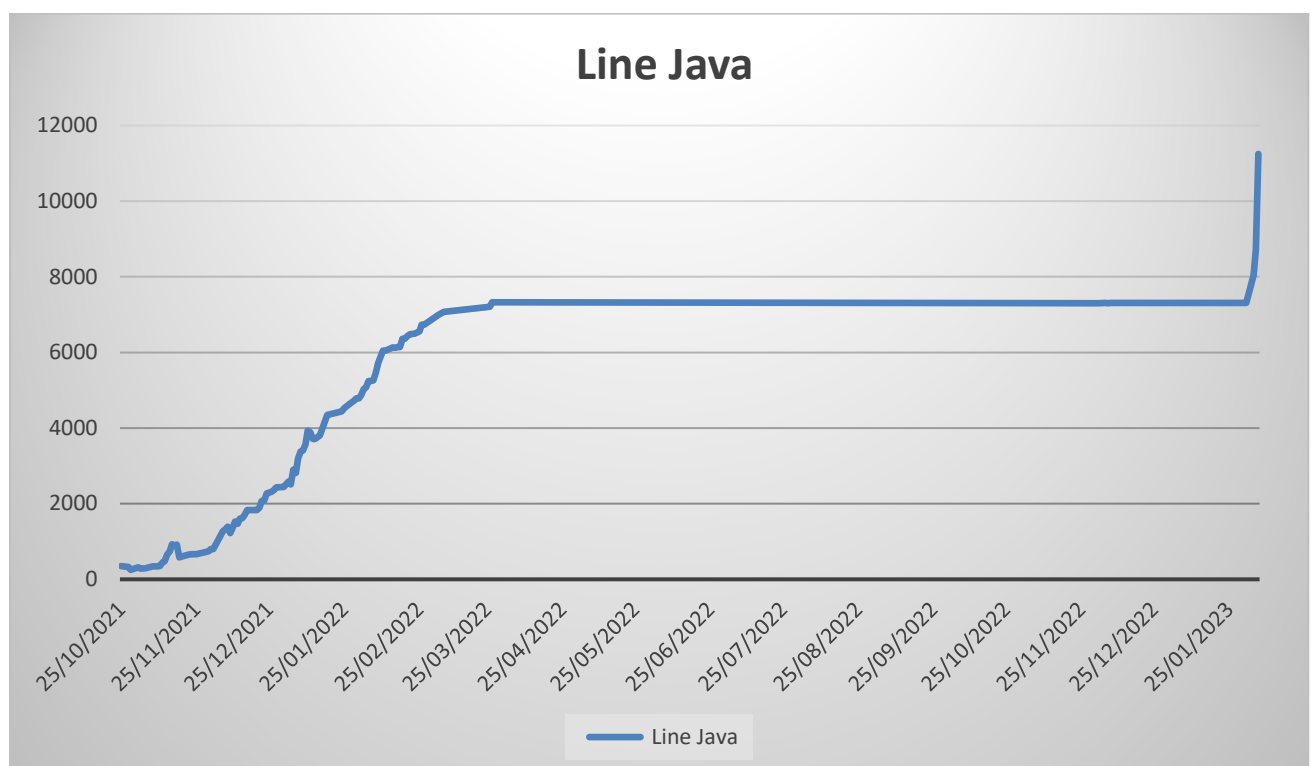
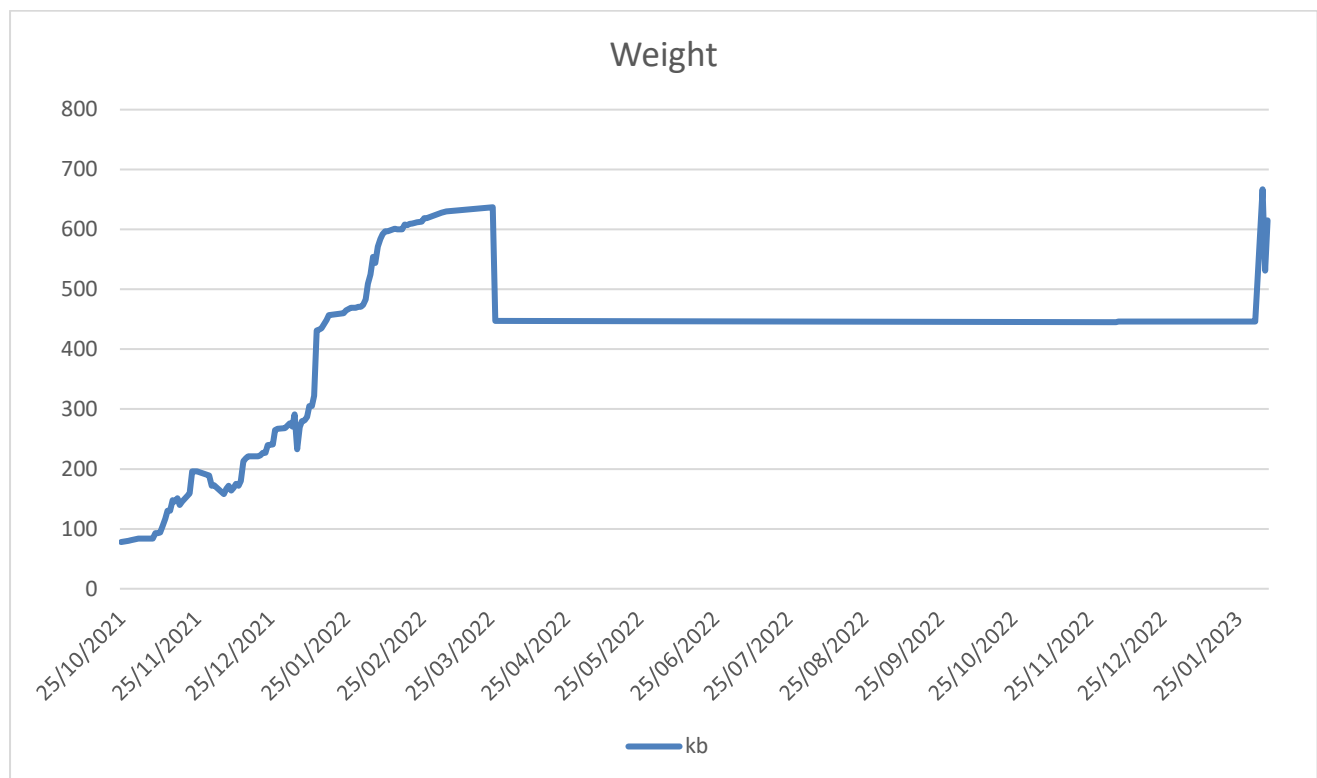
The project after analyzing [SonarCloud](#) does not present code smells, vulnerabilities, or bugs.

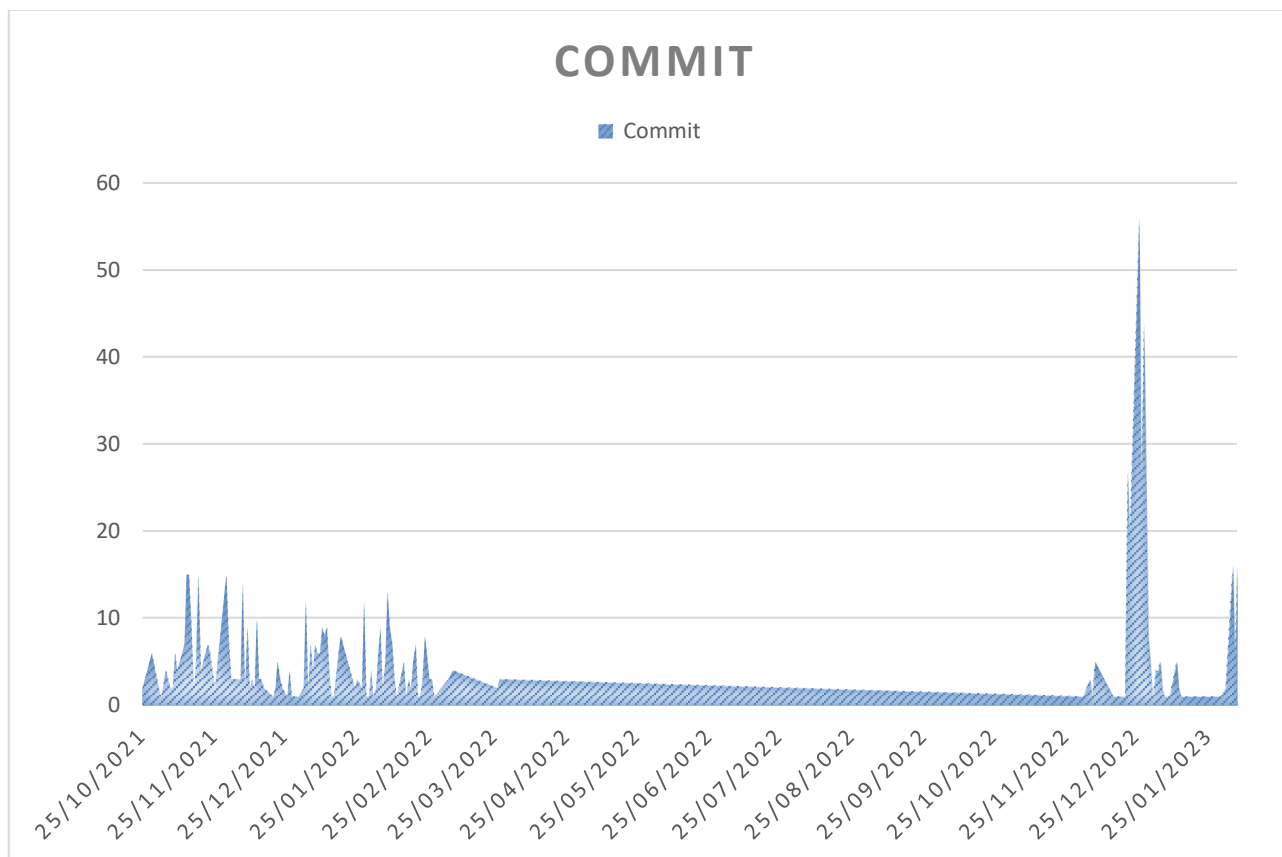
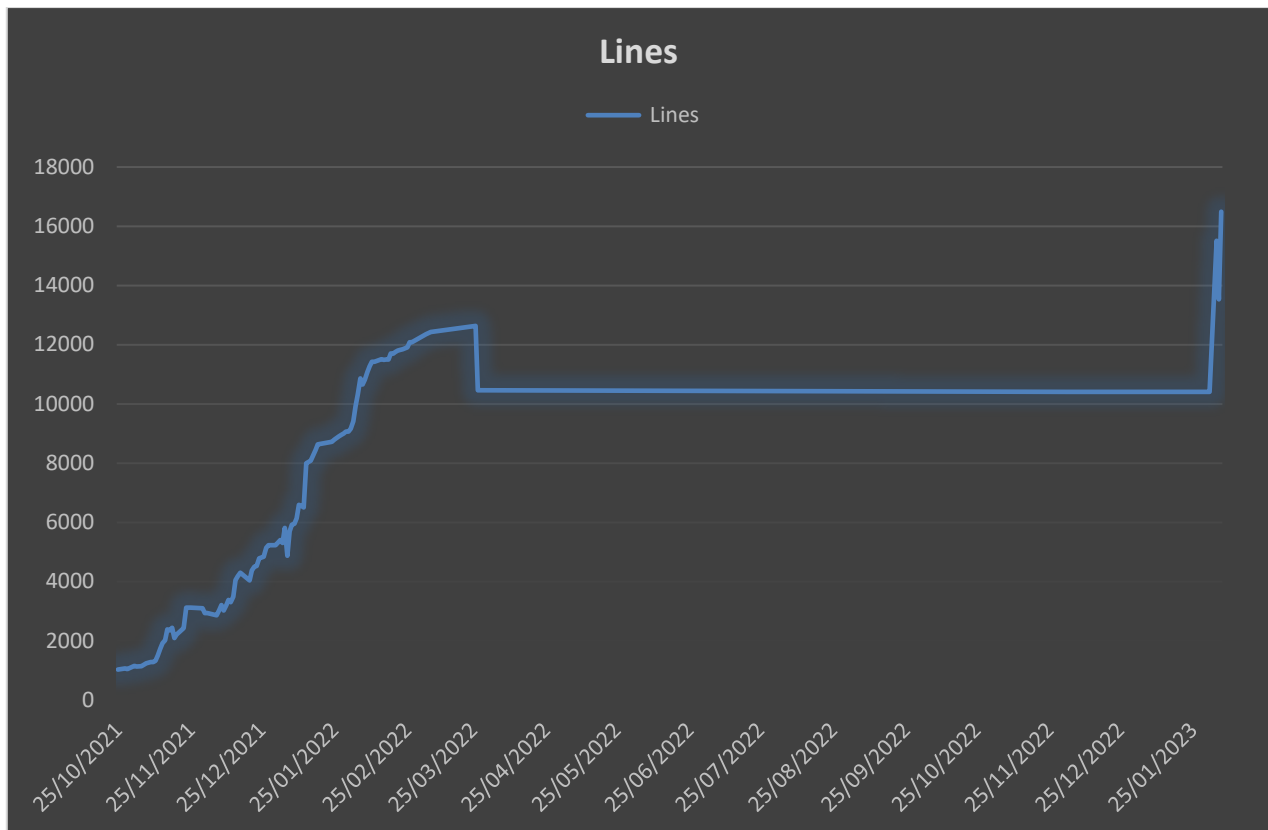


5.5. DAO

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 all the [DAO](#) class, [IO](#) and [DBConnection](#) class.

6. Analytics





7. Video

In the “[..\Document\Documentation\Video](#)” folder, there are two videos, each related to a graphical interface showing the same functionality.