

ASSIGNMENT A1

1. Objective

The objective of this assignment is to allow students to become familiar with architectural patterns.

2. Application Description

Use JAVA/C# API to design and implement an application a tennis club that organizes matches. Every match is between 2 players. A match is played in 'best of 5' format (first player that gets 3 sets, wins). This means that each match is composed of a maximum of 5 sets. For each set, the first player to reach minimum 6 games by a margin of 2.

The application should have two types of users:

- regular user represented by the player
- administrator user

Both kinds of uses have to provide an email and a password in order to access the application.

The regular user can perform the following operations:

- View Sets
- View Matches
- Update score of game (must be in the game, otherwise error)

The administrator user can perform the following operations:

- CRUD on player accounts
- CRUD on matches

3. Application Constraints

- The data will be stored in a database
- Use the Layers architectural pattern to organize your application.
- The Data Access Layer (DAL) will be implemented using an ORM framework.
- The application will use a config file from which the system administrator can set which DAL implementation will be used to read and write data to the database.
- All the inputs of the application will be validated against invalid data before submitting

the data and saving it in the database.

4. Requirements

- Create the analysis and design document (see the template).
- Implementation of the application
- Write at least one Unit Test for each method in the business layer

5. Deliverables

The following files will be uploaded on your personal github account in a new repository:

- Analysis and design document.
- Implementation source files.
- Build file (e.g. pom.xml)
- (Only if you added by hand in the database and not through code) SQL script populating the database with initial values.

The mentioned files are to be uploaded to the github classroom assignment.

Assignment link: <https://classroom.github.com/a/wiNWWDaX>

If, for some reason, this will not work, you will create a repository on your account and a link with said repository is to be sent to ardeleaneugenrichard@gmail.com after presenting the application at the laboratory.

In the absence of github classroom assignment and the email the grade of the assignment is 0, regardless of the circumstances.

6. Grading

Grade	Functionality
0	Starter Code
5	Analysis and design document 3-Tier Project Structure Admin: CRUD on players User: view sets - console/terminal User: view matches console/terminal
6	Login
7	Update Score (And detect Set & Match end) Create & View - Set & Match from UI
8	Unit Tests on business logic methods
9	Admin: CRUD on matches

10	Quality of Implementation and Documentation Match report
Bonus (+1)	Use Factory Method for multiple report formats (e.g. txt, pdf)

7. References

- 7.1. Martin Fowler et. al, Patterns of Enterprise Application Architecture, Addison Wesley, 2003 (Chapters 10 and 9)
- 7.2. https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture
- 7.3. https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
- 7.4. https://www.popsugar.co.uk/fitness/How-Keep-Score-Tennis-43702250?utm_medium=redirect&utm_campaign=US:RO&utm_source=www.google.com&fbclid=IwAR3xEmu53A87jv9RhPCiaU143hrvKuBp_N5F9pgRbLey_kJAXs2yqnEuETU