Cairo University Faculty of Computers and Information



**CS352 – Software Engineering II**

**Phase 1 Template**

**2017**

**Project Team**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20140377 | Omar AbdulazizSaeed | Omer.alqubati@gmail.com | 01204996961 |
| 20140375 | Hameedmohammedabdullah | Hameed203083@gmail.com | 01020714213 |
| 20140341 | ZiadAmerAbdulhai | Ziadaamer9@gmail.com | 01090946201 |
| 20140367 | Asseel Ahmed Mohamed | Asseel7723@gmail.com | 01118956026 |
| 20140339 | Wafa'a Samir Alahmar | fofoalah@gmail.com |  |
|  |  |  |  |

**Staff:**

**DrAmrKamel** [a.kamel@fci-cu.edu.eg](mailto:a.kamel@fci-cu.edu.eg)

**DrKhadiga Mohamed kelbedweihy@fci-cu.edu.eg**

**[Omar Khalid]**

**Design and Code Checklist**

**Design Principles**

1. **Does the design follow SOLID principles?** ∏ What %60

Single responsibility Principle is not applied to: Database, Game, and UI

As they are both have so many functions as we can see from the class diagram.

Open close and Liskov’s substitution principles: as we can see we have only one Inheritance relation between User class as a parent and 2 children class Student and Teacher, we in subclasses never change or modify any property that is in the base class, we can substitute the subclass (User, Teacher) to be a base class, also adding new subtype never hurt the base class or modified it.

We didn’t use any interfaces in our design, so the interface segregation principle doesn’t apply.

We didn’t take into account the Dependency Inversion principle.

1. **Does the design follow OOP rules?** ∏ What % 65

Encapsulation rule is applied: in every class we use assessors and mutators to get or to set the values in that class, but most of the data variables are public.

Polymorphism: is used in the User (as a base class) and student, teacher (as their subtypes), also we didn’t use any interface or abstraction.

1. **Is the design simple and easy to modify?** ∏ What % 85

The Design is easyto some extent and can be modified without having issues, as we used the MVC design which divide the classes into 3 different categories and updating one doesn’t affect the others so much (small effect ,only related items to be changed) as they are divided.

**Coding Standards**

1. **Is the code understandable and readable?** ∏ What % 78

Readability and Understandability as we said earlier we use MVC design which makes the code understandable, and the code can be easily readable for someone if he first understand the design, else it will not be that easy.

1. **Does the code follow Java Coding Style?**  ∏ What % 65

* all Classes Files use the suffix .java
* Comments does not used the /\* \*/ c-style comments
* no class implementation comments
* The methods are grouped by functionality
* not all classes have constructors
* most of the lines lengths are small
* Wrapping lines are not used as if the function parameters are too many they are written on the same line also long if conditions on the same line
* comments are not used so much also aren’t used properly
* One declaration per line is the dominant declarations ,also are at the beginning of each block
* open braces ‘{‘anent placed at same line of the methods declaration
* almost no two statements on the same line but long statements are
* if-else doesn’t follow the convention way of writing nor the for

1. **Is indentation used properly?**  ∏ What % 90

Is used properly in percentage of 90%

1. **Do variable have good names?**  ∏ What % 90

Most of them are meaningful

**Comments**

1. **Is the code commented enough?** ∏ What % 10

The code is not commented that well

1. **Is every class and method commented?**  ∏ What % 10

No ,not all of the them

1. **Do comments follow Javadoc style?** ∏ What % 10

No, comments do not follow the java coding style

1. **Is Javadoc generated for all the code?** ∏ What % 10

The javadoc is not used at all

1. **Are there useless / wrong comments?** ∏ What % 10

There is no useless / wrong comment, we didn’t use comment

**Code Structure**

1. **Does the code follow the design precisely?** ∏ What % 85

The code follows the design in almost all classes

1. **Are there very long classes or methods?** ∏ What % 10

There is only one very long class is the database class

1. **Is there repeated code? (put put in a function)** ∏ What % 20

There is less repeated in our code about only some function like search we implement it twice.

**Error Handling**

1. **Does the code handle errors and exceptions**? ∏ What % 95

Yes, we handle error and exception in our code

1. **Is defensive programming used to avoid errors?** ∏ What % ……

As we handle the errors and the extreme cases, we try to avoid any cases of the system being down or crash based on any wrong input or misuse of the program.

**Logic**

1. **Do loops have correct conditions and bounds**? ∏ What % ……

Yes, all loops have correct conditions and bounds

1. **Do loops always terminate?**

Yes, all loops always terminate there is no infinite loop

**Overall**

1. **Are the design and code of good quality?** ∏ What % 85

Based on the above the quality is relatively good

* 1. **UserTestingclass**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Testing function** | **Description** | **Result** |
| **1.** | **Login (name , email , pass , type)** | **Testing function for Login**  **function in user entity. This test case test the normal login scenario**  **Assumption: name is unique for each user** | **Passed** |
| **2.** | **SignUp(name , email , pass , type)** | **Testing function for sign up function. This test case test the normal SignUpscenario** | **Passed** |
| **3.** | **AvailableGame(name )** | **Testing function for AvailableGamefunction. This test case testswhether a game is available before or not.**  **Scenario**  **Assumption: game's name is unique for each game.** | **Passed** |
| **4.** | **Load (name)** | **Testing function for Load function. This function test case will test whether to load the data from file or not** | **Passed** |

# Git repository link