**SW2 – Project Evaluation Form**

* **Each team must submit the following Documentation that contains:**

**- Project Description in detail.**

**- Class Diagram. And Database Schema.**

* **Each team must submit the project via GitHub:**
* **Source Code.**
* **Video Demo for running ( 2 – 5 Minutes ).**
* **Documentation and Evaluation Form.**
* **The Evaluation will start with giving all teams 30 marks then check the following criteria:**

**Full violated –**They violate the principle in all cases in the Code.

**Medium Violated –** They apply the principle in some cases and violate it in some cases (in total the student applies the principle on average).

**Small violated –** They Apply the principle in most cases except for a very few cases, no more than one or two.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Violation Level** | **Full** | **Medium** | **Small** | **Grade** |
| **Documentation** | **-5** | **-2** | **-1** |  |
| **Not Apply MVC (it does not Separate Business logic from GUI).**  **Example of violation: write the implantation for a method such as an inset item into the database inside the Button Action method)** | **-6** | **- 3** | **-1** |  |
| **Violate clean code – Variables** | **-2** | **-1** | **-.05** |  |
| **Violate clean code – Functions** | **-2** | **-1** | **-.05** |  |
| **Violate Single-responsibility Principle** | **-2** | **-1** | **-.05** |  |
| **Violate Open-closed Principle** | **-2** | **-1** | **-.05** |  |
| **Violate the Liskov Substitution Principle** | **-2** | **-1** | **-.05** |  |
| **Violate Interface Segregation Principle** | **-2** | **-1** | **-.05** |  |
| **Violate Dependency Inversion Principle** | **-2** | **-1** | **-.05** |  |
| **Not Upload code to GitHub** | **-1** | | |  |
| **Only One Branch Without Merge (GitHub)** | **-2** | | |  |
| **Only One Contribution (GitHub)** | **-2** | | |  |
| **Total Minus from Grade** |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Design Pattern Bounce** | **+4** |  |
| **Bounce on Overall Work** | **+2** |  |
| **Total Team Grade / 30** |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name (Arabic)** | **ID** | **Individual Bounce +2** | **Grade** |
| **محمد خالد مصطفي السيد** | **201900665** |  |  |
| **عمرو محمد عطاالله** | **201900542** |  |  |
| **يوسف علاء الدين احمد** | **201900989** |  |  |
| **محمد مختار محمد متبولى** | **201900731** |  |  |
| **عمر محمد سعد محمد** | **201900526** |  |  |
| **عيد خالد عاشور عبدالمعز** | **201900545** |  |  |
| **مصطفى ناجح فتحي ثابت** | **201900835** |  |  |

**Project Description**

**Bug Tracking System**

**Project consists of 4 module :-**

**1-Tester Module**

a. Enables tester to define bug name, bug type, priority, bug level, project name, bug date, status.

b. Enables tester to assign bug to developer.

c. Enables tester to attach screenshot of bug.

d. Enables tester to monitor open and closed bugs

e. System will send email automatically to assigned developer with bug details.

**2-Developer Module**

a. Enables developer to view assigned bugs

b. Enables developer to change status of bug after finishing it.

c. System will send email automatically to tester when developer changed status of bug to complete

**3-Project Manager Module**

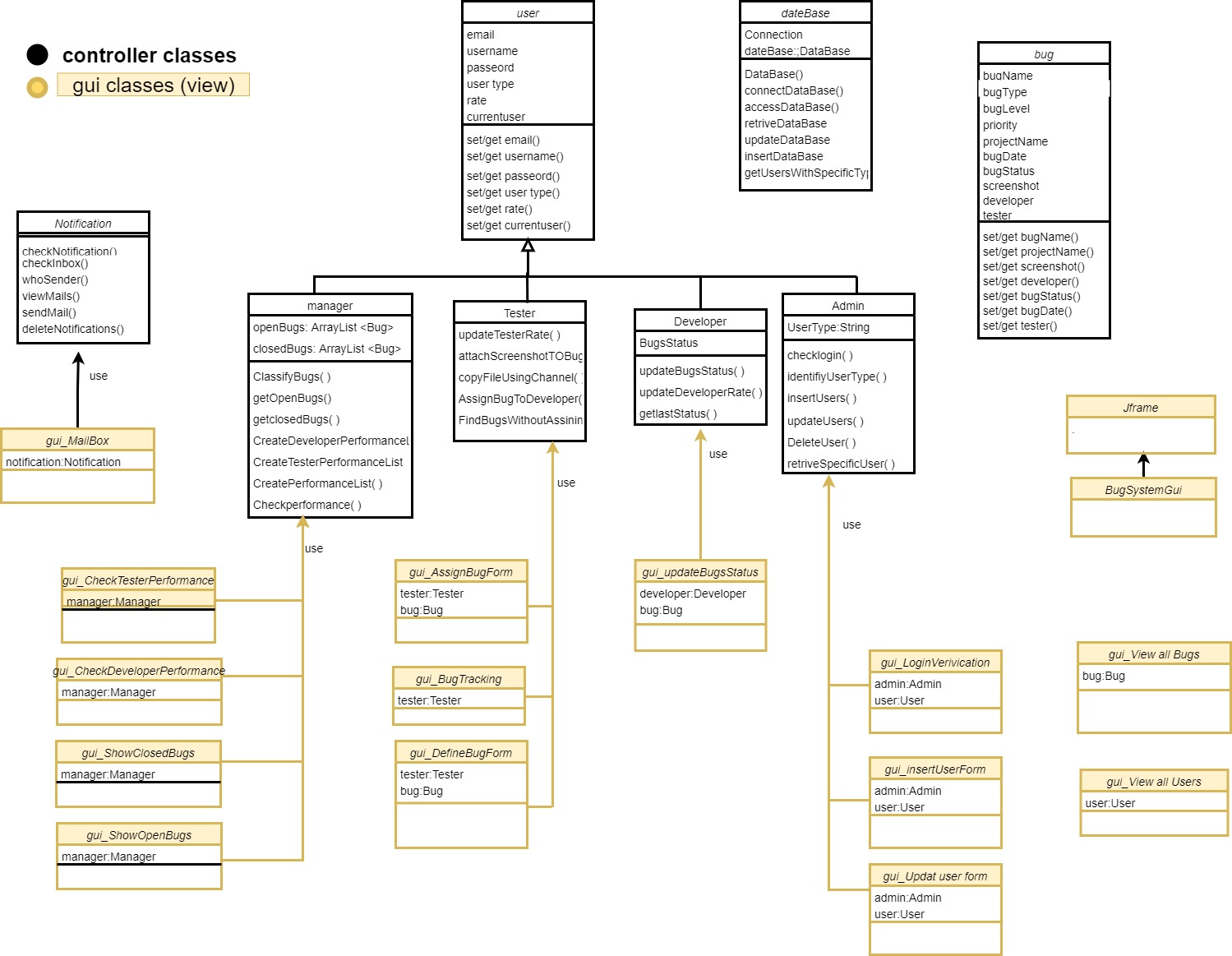
a. Enables PM to check performance of developers and testers.

b. Enables PM to monitor open and closed bugs.

**4- Admin Module**

a. Enables admin to view all bugs of project.

b. Enables admin to add / update / delete users with different roles.

**Class diagram**

**Diagram

Description automatically generated**

**Selected design pattern**

**Creational : singleton pattern**

**Context:**

For class should only one instance exist ( singleton ) . .

**Problem:**

How do you ensure that it is never possible to create more than one instance of a (ADMIN CLASS) And provide a global point of access to it .

**Forces:**

\*The use of a private constructor cannot guarantee that no more than one instance willcreated

\*The singleton instance must also be accessible to all classes that require it , therefore it must often be public .

**Solution :**

**•**Have the admin class constructor private to ensure that no other class will be able to create an instance of the admin class .

• Define a public static method called GetInstance() , The first time this method is called , it creates the single instance of the class " Admin " and stores a reference to that object in a static private variable CALEED admin

**Diagram

Description automatically generated**