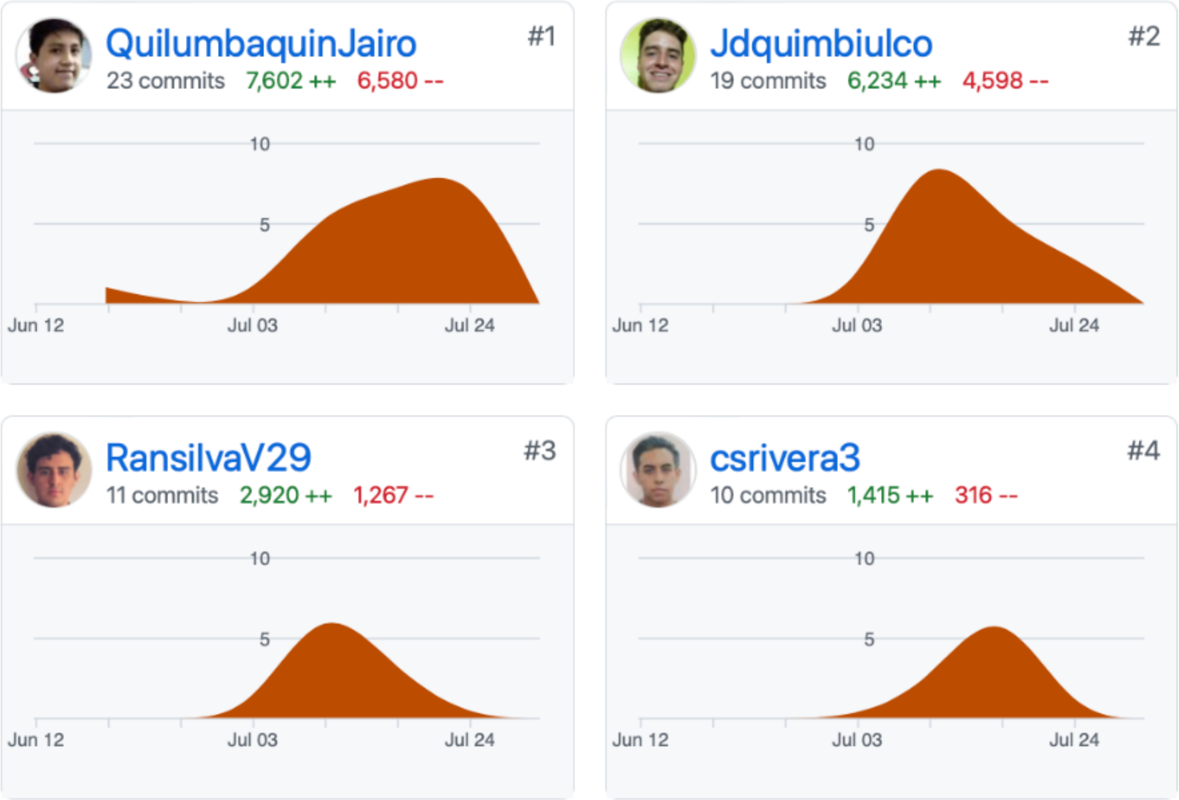
TEAM 06 CODEX++ INSPECTOR: TEAM 07 MyWayCode
(Daniela Tituaña)

Quilumbaquin	100%
Quimbiulco	95%
Rivera	75%
Silva	80%



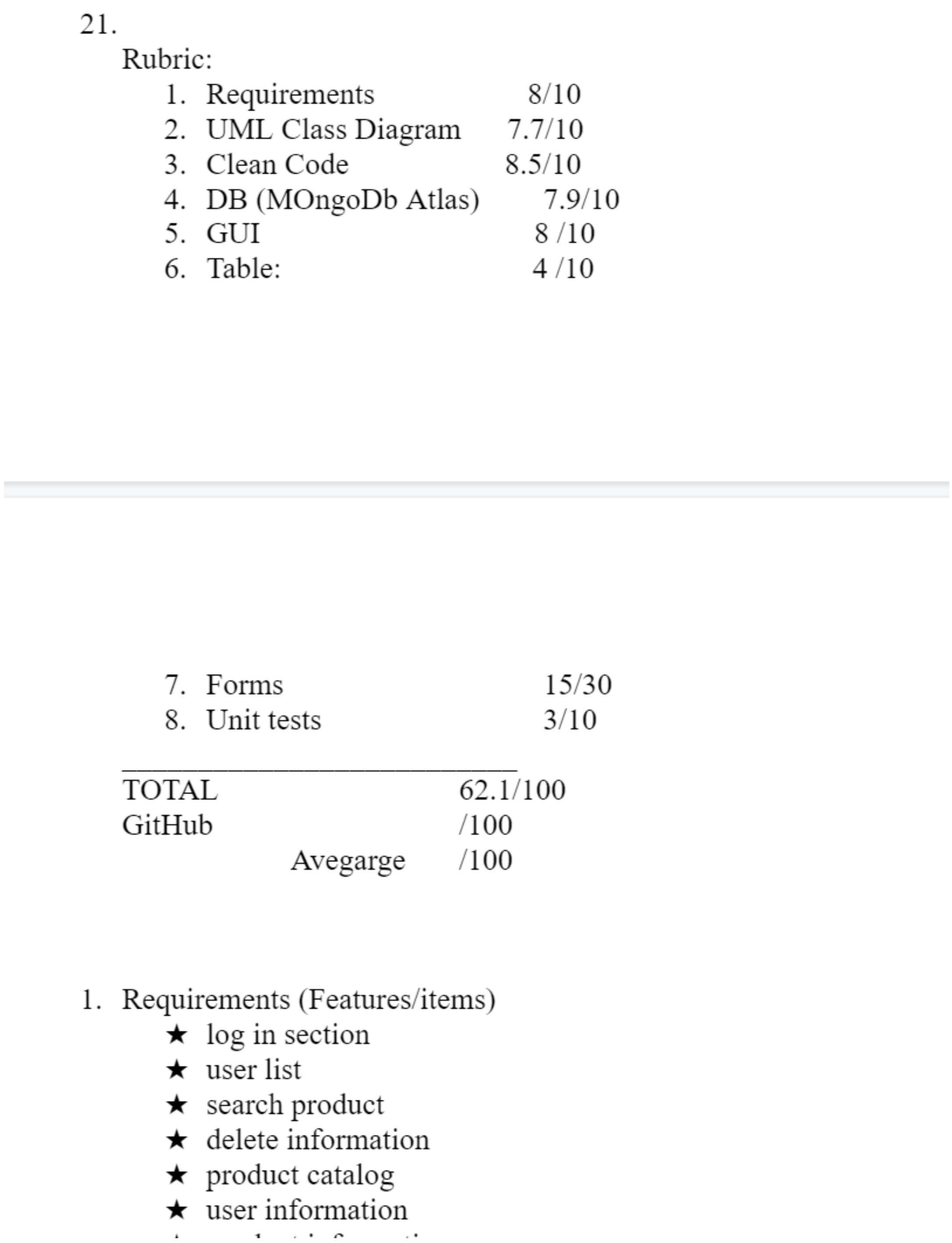
QuilumbaquinJairo #1
23 commits 7,602 ++ 6,580 --

Jdquimbiulco #2
19 commits 6,234 ++ 4,598 --

RansilvaV29 #3
11 commits 2,920 ++ 1,267 --

csrivera3 #4
10 commits 1,415 ++ 316 --

Github:https://github.com/Jdquimbiulco/TC_RooKify.git



21.

Rubric:

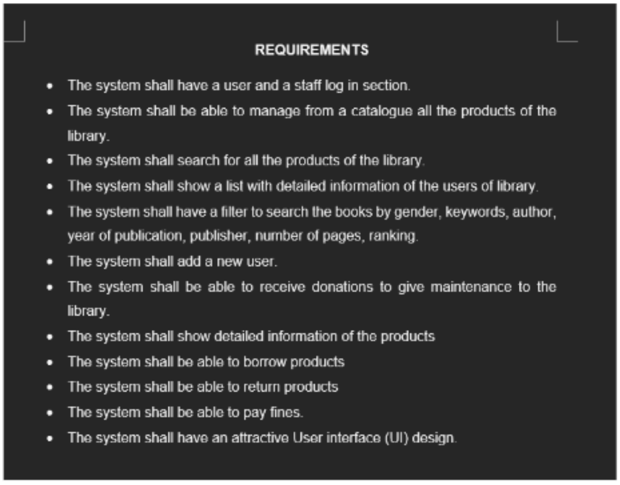
1.	Requirements	8/10
2.	UML Class Diagram	7.7/10
3.	Clean Code	8.5/10
4.	DB (MOngoDb Atlas)	7.9/10
5.	GUI	8 /10
6.	Table:	4 /10
7.	Forms	15/30
8.	Unit tests	3/10

____

TOTAL	62.1/100
GitHub	/100
Avegarge	/100

1. Requirements (Features/items)
   - ★ log in section
   - ★ user list
   - ★ search product
   - ★ delete information
   - ★ product catalog
   - ★ user information

1. Requirements (Features/items)
    - ★ log in section
    - ★ user list
    - ★ search product
    - ★ delete information
    - ★ product catalog
    - ★ user information
    - ★ product information
    - ★ book filter
    - ★ payment of fines



REQUIREMENTS

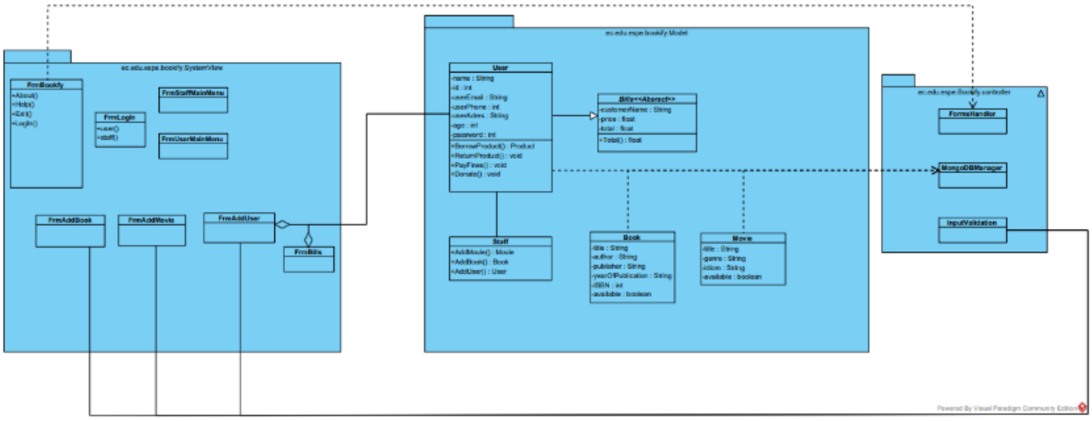
- The system shall have a user and a staff log in section.
- The system shall be able to manage from a catalogue all the products of the library.
- The system shall search for all the products of the library.
- The system shall show a list with detailed information of the users of library.
- The system shall have a filter to search the books by gender, keywords, author, year of publication, publisher, number of pages, ranking.
- The system shall add a new user.
- The system shall be able to receive donations to give maintenance to the library.
- The system shall show detailed information of the products
- The system shall be able to borrow products
- The system shall be able to return products
- The system shall be able to pay fines.
- The system shall have an attractive User interface (UI) design.

2. UML Class Diagram

    Requirements (Features/items)
    UML Class Diagram

    - Classes 2
    - Attributes 2
    - Methods 2
    - Abstract Classes/Interfaces 2
    - Abstract methods 2
    - Polymorphism (overridden methods) 0
    - Inheritance 2
    - MVC -> packages model (POJO classes), view(GUI), controller (Abstract C./Concrete C./Interfaces)(getters and setters (-.01),

- Inheritance 2
- MVC -> packages model (POJO classes), view(GUI), controller (Abstract C./Concrete C./Interfaces)(getters and setters (-.01), labels(-0.1), panels(-0.1), no dependencies in the view package (0.1) 1.7/2



➔

Observations:

➔ There are no methods in the controller package.

➔ The methods must not be in the model package.

➔ Missing relationships in package model.

➔ Methods must have parameters.

3.Clean Code

Pitfalls/Code Smells (0.1)

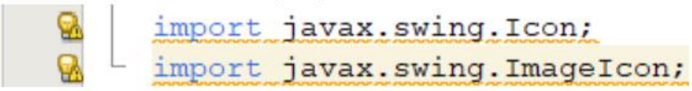
➔

Observations:

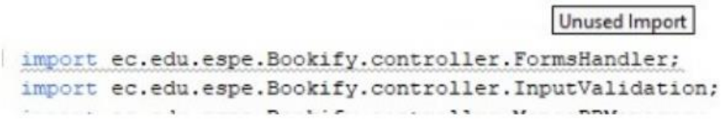
➔ There are no methods in the controller package.
➔ The methods must not be in the model package.
➔ Missing relationships in package model.
➔ Methods must have parameters.

3.Clean Code

Pitfalls/Code Smells (0.1)



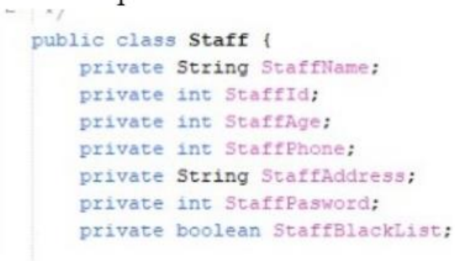
```
import javax.swing.Icon;
import javax.swing.ImageIcon;
```



Unused Import

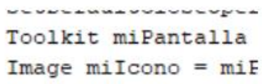
```
import ec.edu.espe.Bookify.controller.FormsHandler;
import ec.edu.espe.Bookify.controller.InputValidation;
```

unused imports -0.1



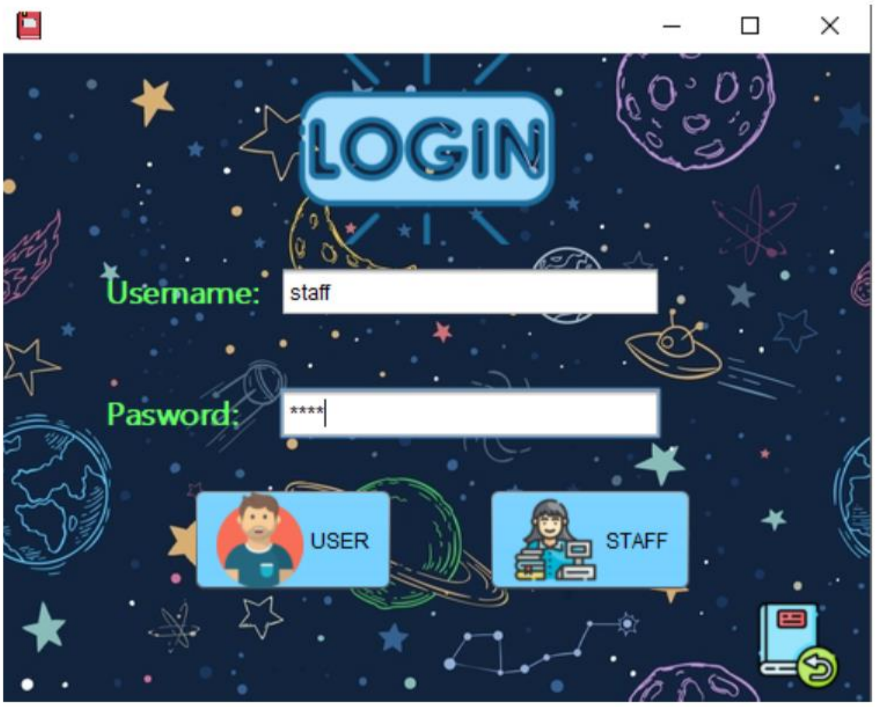
```
 */
public class Staff {
    private String StaffName;
    private int StaffId;
    private int StaffAge;
    private int StaffPhone;
    private String StaffAddress;
    private int StaffPasword;
    private boolean StaffBlackList;
```

variables that start with a capital letter -0.1



```
Toolkit miPantalla
Image miIcono = miP
```

inconsistent variables



Username: staff

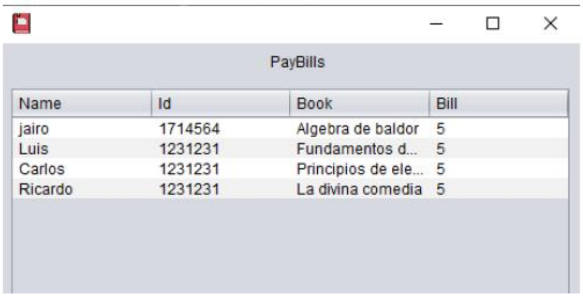
Pasword: ****

LogIn information it´s a predesignated password and user (-0.1)

6.Tables:

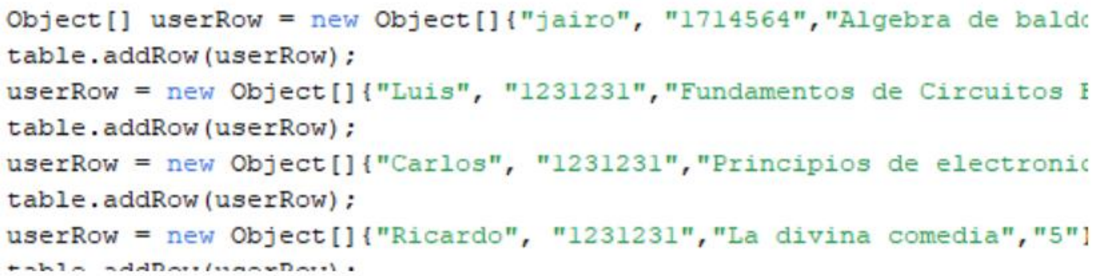
data come from DB 5

print data to printer 5



PayBills

Name	Id	Book	Bill
jairo	1714564	Algebra de baldor	5
Luis	1231231	Fundamentos d...	5
Carlos	1231231	Principios de ele...	5
Ricardo	1231231	La divina comedia	5



```
Object[] userRow = new Object[]{"jairo", "1714564","Algebra de bald
table.addRow(userRow);
userRow = new Object[]{"Luis", "1231231","Fundamentos de Circuitos E
table.addRow(userRow);
userRow = new Object[]{"Carlos", "1231231","Principios de electronic
table.addRow(userRow);
userRow = new Object[]{"Ricardo", "1231231","La divina comedia","5"}
table.addRow(userRow);
```

The system does not display database information.

7.Forms

at least, there must be 6 different forms, every form is worth 5 points

- Login Form
- 5 Forms for Information (CRUD operations, 4 business rules)
  Every form
    - Every input (JTextField) is validated
    - There are different types of inputs (not everything is a JTextFields) -1 for each JTextField
    - Buttons' events, no code is here for CRUD operations or business rules, i.e., these functions will call some controller method, and will use some model class.

every mistake: -1
less than 5 forms -> /50%

Unit Test