UNIVERSIDAD DE LAS FUERZAS ARMADAS

ESPE

Object-oriented programming

NRC: 9642

U2 Workshop N° 40

Topic: Final Project Inspection

Ing. Jorge Edison Lascano. PhD

Rubric:
Project GitHub /25 (Individual Contribution)
Project Review (Inspection) /25
        Requirements and modeling (updated and design patterns) /5
        Principles,CleanCode, Pitfalls /5
        Functional Requirements user test (No CRUD operations) /5
        MongoDB /5
        Design patterns code /5
Project Maintenance /25 (Time effort)
        new user / user type /10
        report screen /10
        print out /5
Project Letter /25 (Letter of acceptance or review)

Evidence:
Project GitHub
Project Review (Inspection)
        Requirements and modeling (updated and design patterns)
        Principles,CleanCode, Pitfalls
        Functional Requirements user test (No CRUD operations) 7:25
        MongoDB
        Design patterns code

Project Maintenance
Project Letter

Team 6: Code Warriors   **Leader:** Edison Verdesoto

1. **QUISHPE LLUMIQUINGA JOHN MATEO**
2. **TIPAN QUINTO LEONEL ALEJANDRO**
3. **VERDESOTO SEGOVIA EDISON DAMIAN**

Inspector: Team 1 Jsons   **Leader:** Yeshua Amador
Project: S&O HardwareStore
GitHub: https://github.com/EDVerdesoto/T06_SandO-System.git
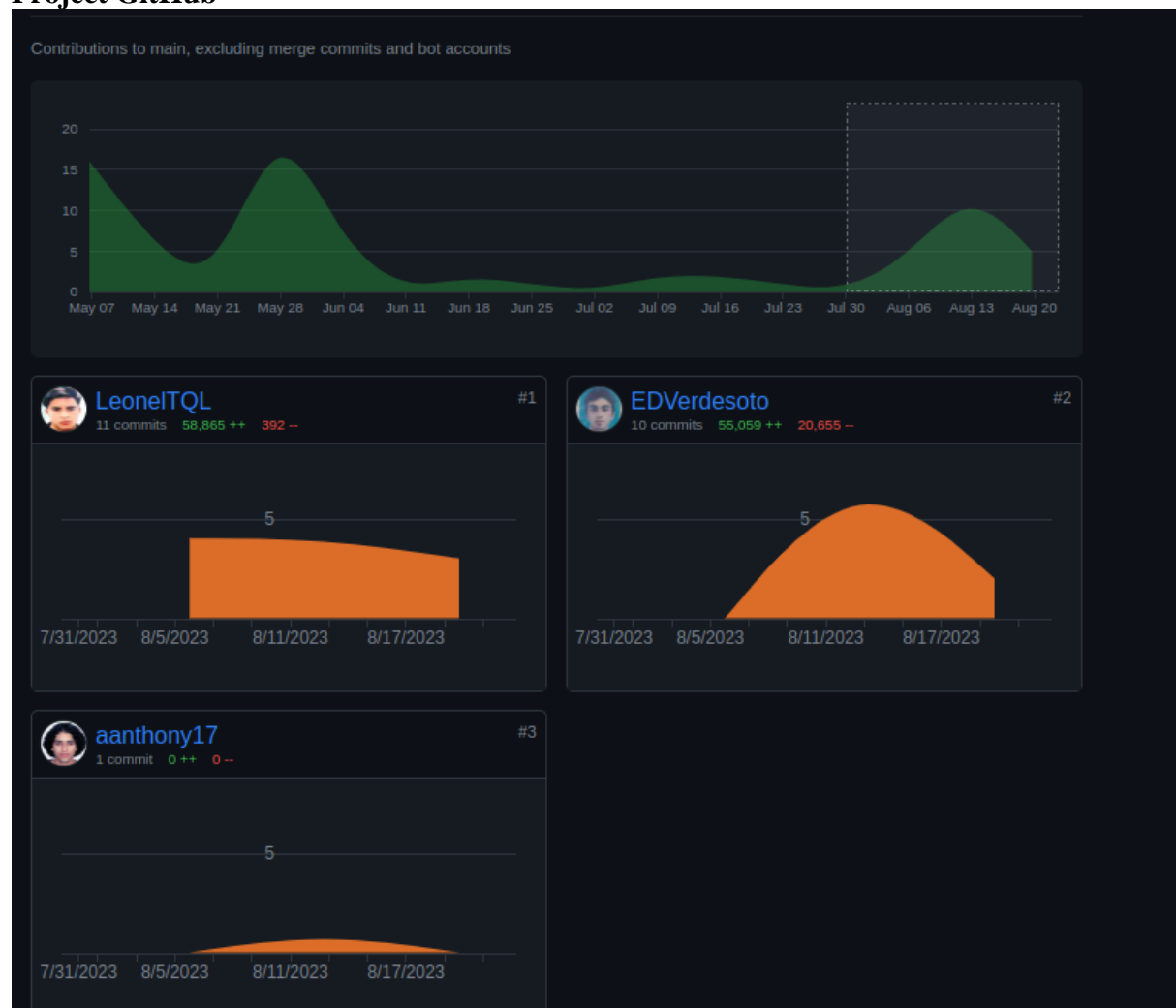**MongoDB Compass:**
mongodb+srv://latipan06:latipan06@cluster0.kawghe4.mongodb.net/?retryWrites=true&w=majority
Grade: **PR:** /25, **PM:** /25, **Letter:** /25   →44 /75

Rubric:
Project Review (Inspection) 17.5/25

Requirements and modeling (updated and design patterns)   **4/5**

Principles,CleanCode, Pitfalls   **4/5**
Functional Requirements user test (No CRUD operations)   **4.5/5**
MongoDB   **5/5**
Design patterns code   **0/5**
Project Maintenance   **10/25 (Time effort)**
    new user / user type   **0/10**
    report screen   **5/10**
    print out   **5/5**

Project GitHub   **/25 (Individual Contribution)**

Project Letter   **15/25 (Letter of acceptance or review)**
    missing contact information
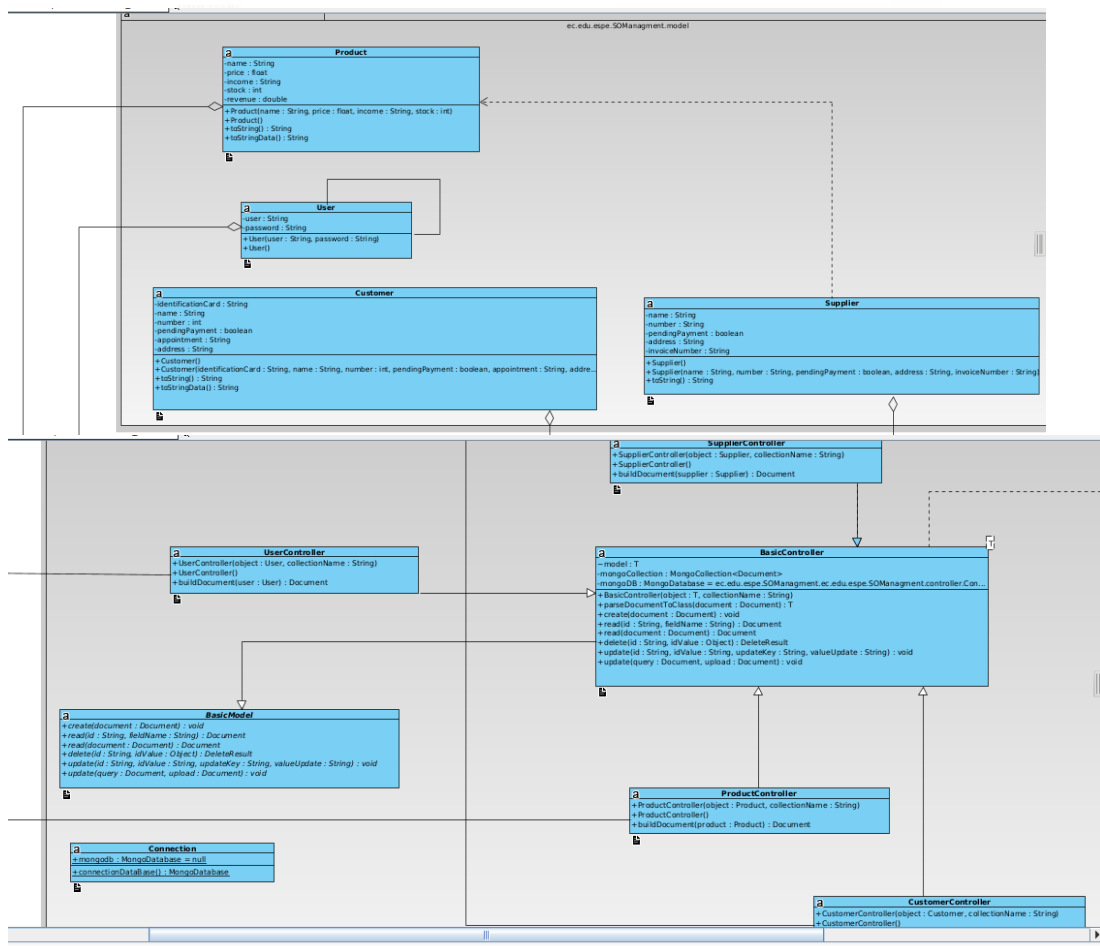
Evidence:
Project GitHub

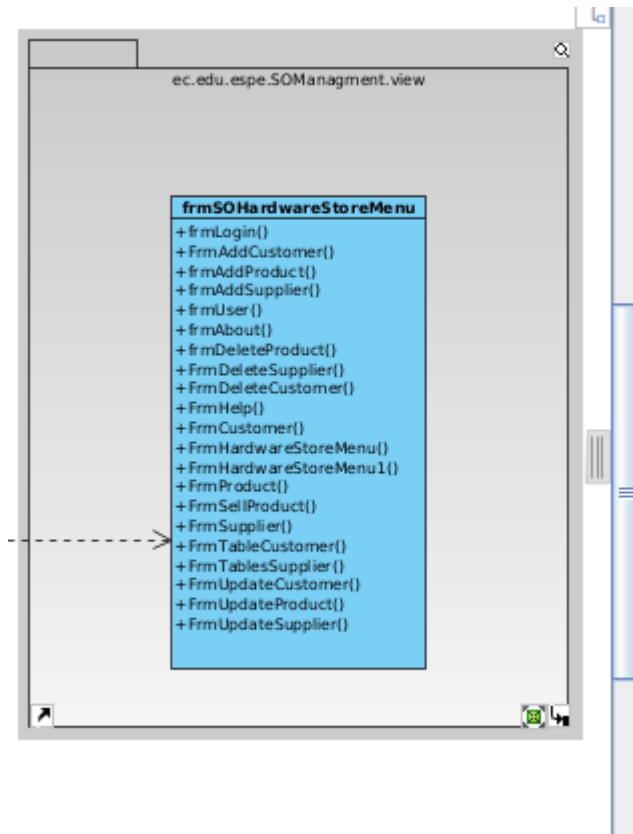

Project Review (Inspection)
    Requirements and modeling (updated and design patterns)

3.2. Functions

The software shall have the following functions:

- **Inventory management:**
- Add new inventory items: The user shall be able to add new items to the inventory database.
- Delete inventory items: The user shall be able to delete items for the inventory database.
- Update inventory items: The user shall be able to update the information of existing items in the inventory database.
- View inventory status: The user shall be able to view the current status of the inventory, including the quantity in stock and the location of the inventory.
- Calculate taxes and profits: The software shall automatically calculate the IVA and 30% of the products to determine the store's profit.
- User management: The program must allow more than one user.
- Sell Products: Update the BD by selling products but don't show the bill.

ec.edu.espe.SOManagment.model

Product
-name : String
-price : float
-income : String
-stock : int
-revenue : double
+Product(name : String, price : float, income : String, stock : int)
+Product()
+toString() : String
+toStringData() : String

User
-user : String
-password : String
+User(user : String, password : String)
+User()

Customer
-identificationCard : String
-name : String
-number : int
-pendingPayment : boolean
-appointment : String
-address : String
+Customer()
+Customer(identificationCard : String, name : String, number : int, pendingPayment : boolean, appointment : String, addre...
+toString() : String
+toStringData() : String

Supplier
-name : String
-number : String
-pendingPayment : boolean
-address : String
-invoiceNumber : String
+Supplier()
+Supplier(name : String, number : String, pendingPayment : boolean, address : String, invoiceNumber : String)
+toString() : String

SupplierController
+SupplierController(object : Supplier, collectionName : String)
+SupplierController()
+buildDocument(supplier : Supplier) : Document

UserController
+UserController(object : User, collectionName : String)
+UserController()
+buildDocument(user : User) : Document

BasicController
-model : T
-mongoCollection : MongoCollection<Document>
-mongoDB : MongoDatabase = ec.edu.espe.SOManagment.ec.edu.espe.SOManagment.controller.Con...
+BasicController(object : T, collectionName : String)
+parseDocumentToClass(document : Document) : T
+create(document : Document) : void
+read(id : String, fieldName : String) : Document
+read(document : Document) : Document
+delete(id : String, idValue : Object) : DeleteResult
+update(id : String, idValue : String, updateKey : String, valueUpdate : String) : void
+update(query : Document, upload : Document) : void

BasicModel
+create(document : Document) : void
+read(id : String, fieldName : String) : Document
+read(document : Document) : Document
+delete(id : String, idValue : Object) : DeleteResult
+update(id : String, idValue : String, updateKey : String, valueUpdate : String) : void
+update(query : Document, upload : Document) : void

ProductController
+ProductController(object : Product, collectionName : String)
+ProductController()
+buildDocument(product : Product) : Document

Connection
+mongodb : MongoDatabase = null
+connectionDataBase() : MongoDatabase

CustomerController
+CustomerController(object : Customer, collectionName : String)
+CustomerController()

```
 * @author Edison Verdesoto, Code Warriors, DCCO-ESPE
 */
public class Decimals {
    public static Double roundToTwoTenths(Double amountToRound){
        Double amountRounded;
        int dollars = (int) (amountToRound/1) ;
        Double cents = (amountToRound%1)*100;

        if(cents == 0){
            amountRounded = amountToRound;
        }
        else {
            cents = (double) Math.round(a: cents);
            amountRounded = dollars + (cents/100);
        }

        return amountRounded;
```

Principles,CleanCode, Pitfalls

```
    public static boolean validateID(String identificationCard) {

        int suma = 0;

        if (identificationCard.length() != 10) {

            return false;
        }

        for (int i = 0; i < 9; i++) {
            int digit = Integer.parseInt(identificationCard.charAt(index: i) + "");
            if (i % 2 == 0) {
                digit = digit * 2;
                if (digit > 9) {
                    digit -= 9;
                }
```

```
        String identificationCard;
    String name;
    int number;
    boolean pendigPayment;
    boolean validateId = true;
    String appointment;
    String address;

    identificationCard = txtId.getText();
    name = txtName.getText();
    number = Integer.parseInt(s: txtNumber.getText());
    pendigPayment = false;
    appointment = formDate.format(date: txtDateService.getDate());
    address = txtAddress.getText();

    validateId = IdentificationCardController.validateID(identificationCard);
```

Functional Requirements user test (No CRUD operations)

tio
nd modeling (updated and design patterns)

–Second functional requirement: PVP with IVA and 30% of the initial price

Producto

Nombre	Tijeras
Fecha de ingreso	24 ago 2023
Precio	15
Stock	10

Agregar   Volver

Message

Successfully created

OK

Productos

Nombre	Fecha de ingreso	Costo	Stock	Precio de venta
Brochas lizas	11-07-2023	5.0	29	6.5
Serrucho	16-07-2023	30.0	20	39.0
Pintura en spray	14-07-2023	20.0	17	26.0
Cenemto de cont...	17-07-2023	4.0	64	5.2
Tornillo en cruz	12-08-2023	14.0	18	18.2
Tornillo plano	25-07-2023	50.0	89	65.0
Destornillador	25-07-2023	20.5	83	26.65
Martillo	25-07-2023	20.5	89	26.65
Cinta de embalaje	25-07-2023	20.5	23	26.65
piintura	11-08-2023	23.0	3	29.9
Canastas	22-08-2023	20.0	20	26.0
uwu	19-08-2023	1.0	0	1.3
Tijeras	24-08-2023	15.0	10	19.5

Creating Users Modified:

-Second functional requirement updated in database

```
_id: ObjectId('64e7525af195e276e75953ed')
name: "Tijeras"
price: 15
income: "24-08-2023"
stock: 10
PVP: 19.5
```

-Third functional requirement updated in database
CRUD modifications in MongoDB

Design patterns code   0/5

Project Maintenance
Project Letter