UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

CLASS NAME: OBJECT ORIENTED PROGRAMMING

TEACHER: EDISON LASCANO

NRC:14575

WORKSHOP #: 41

TOPIC: Exam Inspection

NAME: CARLOS ÑATO

MBDURI:
mongodb+srv://olivier:olivier@cluster0.gguuezt.mongodb.net/?retryWrites=true&w=majority
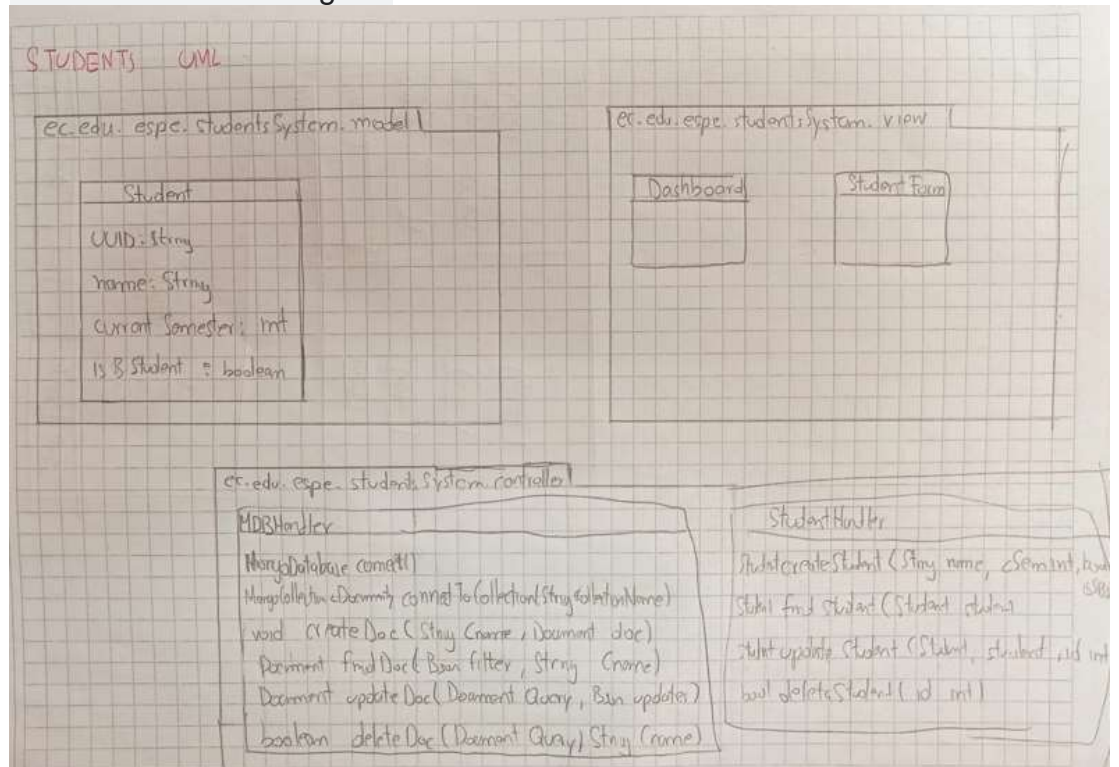DB: DailyDevDB
COL: Students
TOTAL   39/40

Rubric:

MVC:	UML	5/5 pts
GUI design:	Mockup	5/5 pts.
MVC:	Code	5/5 pts
GUI implementation:	Code	5/5 pts.
DB implementation:	MongoDB	5/5 pts.
Data persistence:	Run	10/10 pts.
Clean code:	Code	4/5 pts.(code commented)

Evidence:
MVC: UML Class Diagram

Class 0.2/0.2
5 attributes 0.4/0.4
5 methods 0.4/0.4
Student Controller 0.2/0.2
methods: 0.8/0.8
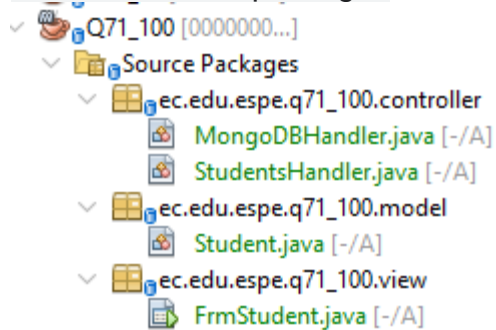Student Class 0.2/0.2
Attributes: 0.8/0.8
Dependency: 1.0
Package : 1.0

TOTAL: 5

GUI: Mockup tool or piece of paper

MVC: Netbeans packages
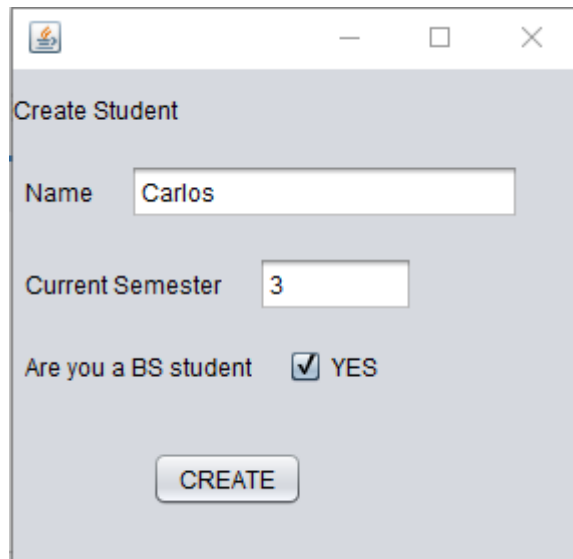

GUI implementation: JForm

DB implementation:Use Compass to show the data and the structure.

```
_id: ObjectId('65c3a9a92591a354e0b39b55')
id : UUID('a6e1b6f6-56a4-461f-961e-80d0b8866765')
name : "olivier"
currentSemester : "1"
isBSStudent : true
```

```
_id: ObjectId('65c3a9b42591a354e0b39b57')
id : UUID('805a814e-6ac6-4fc6-afce-1c1b79d23e0b')
name : "olivier"
currentSemester : "1"
isBSStudent : true
```

```
_id: ObjectId('65c63967a402e80a7ad97e98')
id : UUID('d5f9e60e-4f25-43a0-824c-477050185495')
name : "Olivier Paspuel"
currentSemester : 2
major : "Software Engineering"
semestersLeft : 6
```

Data persistence: Run and see data using Compass.

```
id : UUID('d5f9e60e-4f25-43a0-824c-477050185495')
name : "Olivier Paspuel"
currentSemester : 2
major : "Software Engineering"
semestersLeft : 6
```

```
_id: ObjectId('65ccef8e4a3d981dc0c996df')
id : UUID('b9bd9bc8-a6cc-4dbe-8426-e4366fbe1e63')
name : "Carlos"
currentSemester : "3"
isBSStudent : true
```

Clean code: Code Inspection

```java
    public Document findDocWithFilter(Bson filter, String collectionName) {
        MongoCollection<Document> collection = connectToCollection(collectionName);
        return collection.find(bson: filter).first();
    }

    public static FindIterable<Document> findAllDocs(Bson filter, String collectionName) {
        MongoCollection<Document> collection = connectToCollection(collectionName);
        return collection.find(bson: filter);
    }

//    private static Task convertDocumentToTask(Document document) {
//        // Implement logic to convert Document to Task object
//        Gson gson = new Gson();
//
//        String task = gson.toJson(document);
//
//        return gson.fromJson(task, Task.class);
//        // Example: return new Task(document.getString("taskId"), document.getString("taskName"), ...);
//    }

    public static boolean createDocument(Document document, String collectionName) {
        MongoCollection<Document> collection = connectToCollection(collectionName);
        if (collection == null) {
            System.err.println("Failed to connect to collection: " + collectionName);
            return false;
        }

        try {
            collection.insertOne(ad: document);
            System.out.println(x: "Document created successfully");
```