26. SARAGURO CHAVEZ LEIDY LISBETH **10**
MongoDB URL: mongodb+srv://saraguro:saraguro@cluster0.5nyjils.mongodb.net/
Inspector: Sigcha Kenned
Rubric

A	1.2/1
B	1.2/1
C	1/1
D	1/1
E	1/1
F	1.2/1
H	1/1
G	1/1
J	1/1
main	0.5/1

Evidence

A

```
public class A {         //0.5
    private ArrayList<A> as;    //0.5

    //0.2

    /**
     * @return the as
     */
    public ArrayList<A> getAs() {
        return as;
    }

    /**
     * @param as the as to set
     */
    public void setAs(ArrayList<A> as) {
        this.as = as;
    }

    public A(ArrayList<A> as) {
        this.as = as;
    }

    @Override
    public String toString() {
        return "A{" + "as=" + as + '}';
    }


}
```

B

```
public class B extends A {      //0.5

    private ArrayList<H> hs;          //0.5
    //0.2
    public B(ArrayList<A> as) {
        super(as);
        this.hs = hs;
    }

    @Override
    public String toString() {
        return "B{" + "hs=" + getHs() + '}';
    }

    /**
     * @return the hs
     */
    public ArrayList<H> getHs() {
        return hs;
    }

    /**
     * @param hs the hs to set
     */
    public void setHs(ArrayList<H> hs) {
        this.hs = hs;
    }
}
```

C

```
public class C extends A { //0.5

    ArrayList<E> cs; //0.5

    public C(ArrayList<A> as) {
        super(as);
    }
}
```

D

```
public class D extends A{ //0.5

    private E[] es = new E[5]; //0.25
    private ArrayList<F> fs;   //0.25

    public D(ArrayList<A> as) {
        super(as);
    }
}
```

E

```
public class E { //1
    private List<C> relationC;
    private List<D> relationD;

    public E() {
        this.relationC = new ArrayList<>();
        this.relationD = new ArrayList<>();
    }
     public void addRealationC(C c) {
        relationC.add(c);
    }

    public void addRelationD(D d) {
        relationD.add(d);
    }
}
```

F

```
public class F {  //1

    //0.2
    @Override
    public String toString() {
        return "F{" + '}';
    }

}
```

H

```
public interface  H { //1
}
```

G

```
public class G implements H{ //1

}
```

J

```
public class J { //1

}
```

main

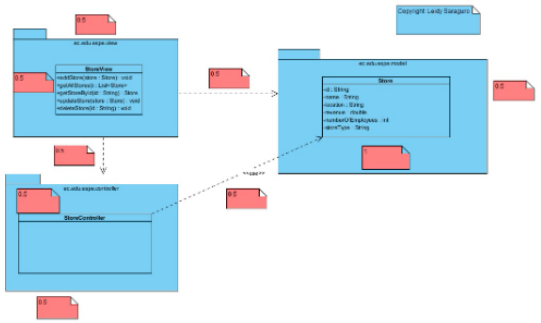
```
public class ViewClass {   //0.5

    public static void main(String[] args) {

    }

}
```

Rubric

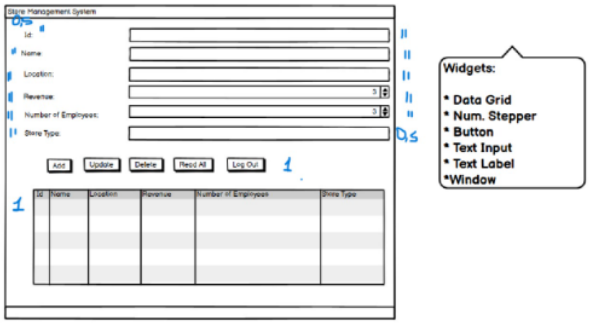
120. MVC(design) 5/5
121. GUI(design) 5/5
122. MVC implementation: 5/5 pts
123. GUI implementation:. 5/5 pts (Create: good use of widgets, read all: good use of tables, Find: showing results)
124. DB implementation: 5/5 pts.
125. Data persistence: 10/10 pts. (your application inserts into MongoDB Atlas, your application reads from MongoDB Atlas)
126. Clean code: 5/5 pts.

Evidence

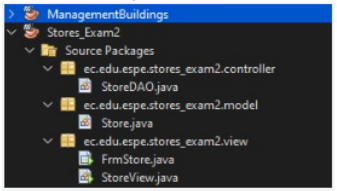
107. MVC(design)



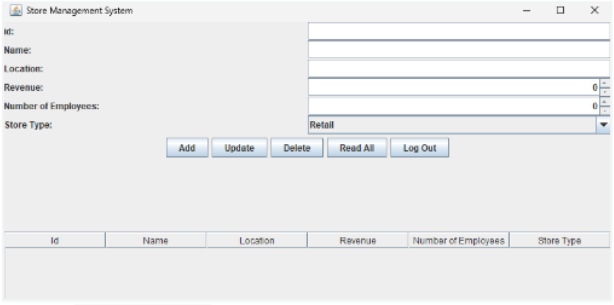
108. GUI(design)



109. MVC implementation

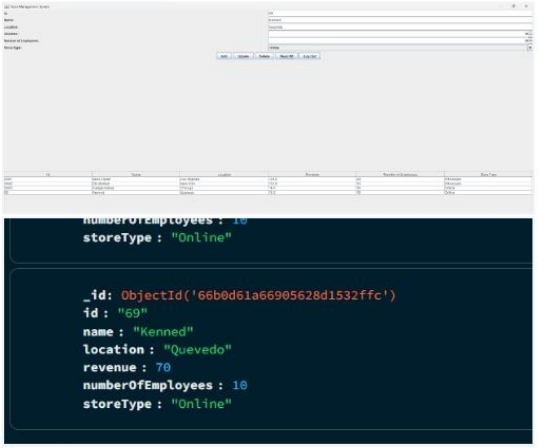


110. GUI implementation



111. DB implementation

112. Data persistence



Clean code

```java
public class Store {
    private String id;
    private String name;
    private String location;
    private double revenue;
    private int numberOfEmployees;
    private String storeType;

    public Store(String id, String name, String location, double revenue, int numberOfEmployees, String storeType) {
        this.id = id;
        this.name = name;
        this.location = location;
        this.revenue = revenue;
        this.numberOfEmployees = numberOfEmployees;
        this.storeType = storeType;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getLocation() {
        return location;
    }
```