

SC2002 第一次讨论

pufanyi

目录

1 Overview	3
1.1 Techniques We Use	3
1.2 本次讨论目的	3
1.3 Use Case Diagram	3
2 Codes	4
2.1 Location	4
2.2 Model	4
2.2.1 getID()	4
2.2.2 toMap()	4
2.2.3 fromMap(Map<String, String> map)	5
2.2.4 Supervisor(Map<String, String> map)	5
2.3 Repository	5
2.3.1 getFilePath()	6
2.3.2 seeAll(List<Map<String, String>> data)	6
2.3.3 FacultyRepository()	6
2.3.4 FacultyRepository getInstance()	6
2.3.5 getByID	6
2.3.6 contains	7
2.3.7 add	7
2.3.8 remove	7
2.3.9 isEmpty	7
2.3.10 size	7
2.3.11 update	7
2.3.12 updateAll	7
2.3.13 load	7
2.3.14 save	8
2.3.15 findByRules	8
2.3.16 遍历 Repository	8
2.3.17 调用 Repository	8
2.4 JUnit	9
2.5 @NotNull	9

1 Overview

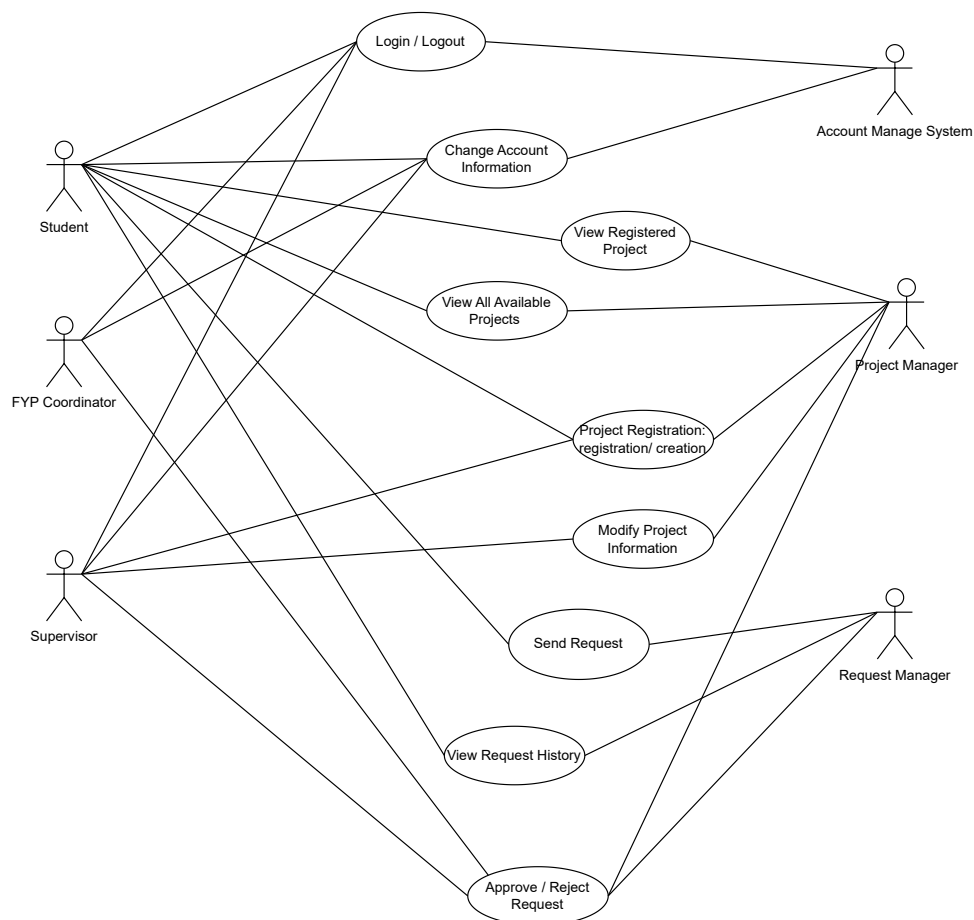
1.1 Techniques We Use

- GitHub
- JUnit 5.8.1
- GitHub Copilot & ChatGPT
- JetBrains IntelliJ IDEA

1.2 本次讨论目的

- 介绍一下我们的 project 的目的
- 介绍一下我们的 project 的大致架构
- 进行 Model 和 Repository 部分的分类与分工
- 大致理解 Controller 部分，完整读完题目

1.3 Use Case Diagram



2 Codes

2.1 Location

/src/main/util/config/Location.java

每个人都需要修改一下

代码可以复制 /src/main/util/config/Location.java.example

具体的，需要把其中的 LOCATION 改成自己整个 project 的位置。

```
1 package main.utils.config;
2
3 public class Location {
4     public static final String LOCATION = "D:\\NTU\\Y1S2\\SC2002\\FYPMS";
5 }
```

2.2 Model

/src/main/model/Model.java

具体详细的例子可以参考 /src/main/model/user/Supervisor.java

需要实现的 method:

- @Override public String getID()
- @Override public Map<String, String> toMap()
- @Override public void fromMap(Map<String, String> map)
- public Supervisor(Map<String, String> map)

2.2.1 getID()

返回的是 Model 的 ID，用于在 Repository 中进行查找。

对于同一类型的 Model，ID 必须是唯一的。

2.2.2 toMap()

返回的是 Model 的 Map，用于在 Repository 中进行存储。

Map 类似于 python 中的 dict，可以通过 key 来获取 value。

其中我们把 Model 的 attribute 的 name 作为 key，attribute 的 value 作为 value。

```
1 @Override
2 public Map<String, String> toMap() {
3     Map<String, String> ansMap = new HashMap<>();
4     ansMap.put("supervisorID", this.supervisorID);
5     ansMap.put("supervisorName", this.supervisorName);
6     ansMap.put("email", this.email);
7 }
```

```
7     ansMap.put("password", getPassword());
8     return ansMap;
9 }
```

2.2.3 fromMap(Map<String, String> map)

从 Map 中读取 Model 的 attribute。

```
1 @Override
2 public void fromMap(Map<String, String> map) {
3     this.supervisorID = map.get("supervisorID");
4     this.supervisorName = map.get("supervisorName");
5     this.email = map.get("email");
6     setPassword(map.get("password"));
7 }
```

2.2.4 Supervisor(Map<String, String> map)

通过 Map 来构造 Model。

```
1 public Supervisor(Map<String, String> map) {
2     fromMap(map);
3 }
```

2.3 Repository

具体详细的例子可以参考 `/src/main/repository/user/FacultyRepository.java`

`Repository<ModelType>` 中的 `ModelType` 是要存储的 Model 的类型，必须继承于 Model 类型。

```
1 import main.repository.Repository;
2
3 public class FacultyRepository extends Repository<Supervisor> {
4     ...
5 }
```

需要实现的 method:

- `@Override public String getFilePath()`
- `@Override public void seeAll(List<Map<String, String>> data)`
- `public FacultyRepository()`
- `public static FacultyRepository getInstance()`

其他方法不需要再实现，直接用即可。

2.3.1 getFilePath()

返回的是存储仓库的绝对路径。

```
1 final private static String FILE_PATH = "/data/user/faculty.txt";
2
3 @Override
4 public String getFilePath() {
5     return Location.LOCATION + FILE_PATH;
6 }
```

2.3.2 seeAll(List<Map<String, String>> data)

给你的一个 List<Map<String, String>>, 你需要把它转换成你需要的类型。
具体的转法见 Model.toMap(Map<String, String>)。

```
1 @Override
2 public void setAll(List<Map<String, String>> listOfMappableObjects) {
3     for (Map<String, String> map : listOfMappableObjects) {
4         getAll().add(new Supervisor(map));
5     }
6 }
```

2.3.3 FacultyRepository()

仿照以下方法写就好了。

```
1 public FacultyRepository() {
2     super();
3     load();
4 }
```

2.3.4 FacultyRepository getInstance()

仿照以下方法写就好了。

```
1 public static FacultyRepository getInstance() {
2     return new FacultyRepository();
3 }
```

2.3.5 getByID

```
1 public ModelObject getByID(String modelObjectID) throws
    NoSuchElementException
```

2.3.6 contains

```
1 public boolean contains(String modelObjectID)
```

2.3.7 add

```
1 public void add(String modelObjectID) throws IllegalArgumentException
```

抛出 `IllegalArgumentException` 的原因是因为 `modelObject` 的 ID 已经存在了。

2.3.8 remove

```
1 public void remove(String modelObjectID) throws NoSuchElementException
```

抛出 `NoSuchElementException` 的原因是因为 `modelObject` 的 ID 不存在。

2.3.9 isEmpty

```
1 public boolean isEmpty()
```

2.3.10 size

```
1 public int size()
```

2.3.11 update

```
1 public void update(ModelObject) throws NoSuchElementException
```

将原来 ID 为 `modelObject` 的 `Model` 替换成 `modelObject`。

抛出 `NoSuchElementException` 的原因是因为 `modelObject` 的 ID 不存在。

2.3.12 updateAll

```
1 public void updateAll(List<ModelObject> modelObjects)
```

2.3.13 load

```
1 public void load()
```

从仓库中重新加载数据。

2.3.14 save

```
1 public void save()
```

将数据保存到仓库中。

需要注意的是，凡是已完成的方法，如果数据库中的数据发生了变化，都已经自动调用 `save()` 方法。

2.3.15 findByRules

不给出定义了，只说一下怎么用：

```
1 facultyRepository.findByRules(  
2     supervisor -> supervisor.getID().equals("A1234567A")  
3 );  
4 facultyRepository.findByRules(  
5     supervisor -> supervisor.getID().equals("A1234567A"),  
6     supervisor -> supervisor.getID().equals("12345")  
7 );
```

传入参数是 `modelObject -> expression` 或者 `modelObject -> { statements; }`，返回值是 `List<ModelObject>`。

`expression` 是一个 `boolean` 类型的表达式，`statements` 是一系列的语句，最终需要 `return` 一个 `boolean` 类型的值，类似于 `method`。

返回的东西均满足 `expression` 或者 `statements`。

如果传入多个参数，那么返回的东西均满足所有的 `expression` 或者 `statements`。

2.3.16 遍历 Repository

```
1 for (Supervisor supervisor : facultyRepository) {  
2     ...  
3 }
```

2.3.17 调用 Repository

使用 `getInstance()` 方法来获取 `Repository` 的实例。

该实例已经被初始化和数据库同步好了，可以直接使用。

```
1 FacultyRepository facultyRepository = FacultyRepository.getInstance();
```

例如，在代码中要找到所有叫 `Li Yi` 的 `Supervisor`，可以这样写：

```
1 List<Supervisor> manyLiYi = FacultyRepository.getInstance().findByRules(  
2     supervisor -> supervisor.getName().equals("Li Yi");
```



```
3   ));  
4   for (Supervisor supervisor : manyLiYi) {  
5       System.out.println(supervisor.getID());  
6   }
```

2.4 JUnit

重点：一切交给 GitHub Copilot / ChatGPT!!!

例子看 `test` 目录下的代码。

2.5 @NotNull

如果像保证一个函数的传入参数不为 `null`，可以这样写：

```
1   public void add(@NotNull String modelObjectID) {  
2       ...  
3   }
```

如果传入 `modelObjectID` 为 `null`，那么会抛出 `NullPointerException`。