



## SC2002 OBJECT ORIENTED DESIGN & PROGRAMMING

### FINAL YEAR PROJECT MANAGEMENT SYSTEM

#### Report of Project Structure Design & Functionality

**AY22/23 Sem 1 | A50, Group 6**

NAME	MATRICULATION NUMBER
Pu Fanyi	U2220175K
Jiang Jinyi	U2220259H
Jin Qingyang	U2220239A
Soo Ying Xi	U2220021D

**Project Demonstration Video:** <https://youtu.be/8FikWzfHILA>

**Project Document:** <https://pufanyi.github.io/FYPMS/>

**GitHub Main Page:** <https://github.com/pufanyi/FYPMS>

#### Declaration of Original Work for SC/CE/CZ2002 Assignment

We hereby declare that the attached group assignment has been researched, undertaken, completed, and submitted as a collective effort by the group members listed below.

We have honoured the principles of academic integrity and have upheld the Student Code of Academic Conduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

NAME	COURSE	LAB GROUP	SIGNATURE
Pu Fanyi	SC2002	A50	濮凡轶
Jiang Jinyi	SC2002	A50	蒋锦一
Jin Qingyang	SC2002	A50	晋清扬
Soo Ying Xi	SC2002	A50	苏英喜

# 1 DESIGN CONSIDERATIONS

FYPMS (Final Year Project Management System) is a Java console application designed with a focus on reusability, extensibility, and maintainability. It efficiently manages final year project settings and accommodates different user types and their requirements, allowing for easy upgrades and future development.

## 1.1 DESIGN APPROACH

The FYPMS was designed with a focus on high cohesion and loose coupling, with classes separated into three categories: *controllers*, *boundaries*, and *entities*. *Controllers* include “*Project Manager*”, “*Request Manager*”, *boundaries* include “*Student Main Page*”, “*Coordinator Main Page*”, *entities* include “*Students*”, “*Projects*” etc. When the user is using the system, he/she interacts with the boundaries, which then call the controller to perform requested operations such as making changes to the entity or retrieve information from entity to display etc. Each of these categories works together to complete our system while ensuring the dependency on each other is minimized. As such, our system is highly flexible, extendable and easy to maintain. Minimum effort is required when extending our system, for example when a new function in the system is introduced.

## 1.2 HIGHLIGHTS OF SOME DESIGNS

Reflection: The interface *Model* uses reflection to convert between classes and strings, enabling dynamic handling of model data without manual mapping.

Generic Repository Class: *Repository<Model>* class with generics allows for flexible data storage and retrieval for any model type, reducing duplication and improving maintainability.

SHA-3 Password Encryption: User passwords are encrypted using the SHA-3 algorithm for enhanced security.

Batch CSV Data Import: The system supports batch importing of initial data from CSV files, making it convenient to process large datasets in chunks.

Factory Design Pattern: The factory pattern is used to quickly generate requests based on different requirements, enhancing system scalability and adaptability.

JUnit 5: We used JUnit 5 to test our classes, it helps to ensure the correctness and robustness of our code.

## 1.3 APPLIED DESIGN PRINCIPLE

### 1.3.1 Single Responsibility Principle (SRP)

The Single Responsibility Principle (SRP) recommends that each class should have a clear and singular responsibility, avoiding unrelated tasks. By adhering to the SRP, we can minimize the ripple effect of changes,

simplifying the process of modifying, testing, and reusing code, resulting in more maintainable and robust software design.

### 1.3.2 Open/Closed Principle (OCP)

The Open/Closed Principle (OCP) states that classes should be open for extension but closed for modification, allowing for the addition of new functionality without changing existing code. OCP can be implemented through abstraction, inheritance, and polymorphism.

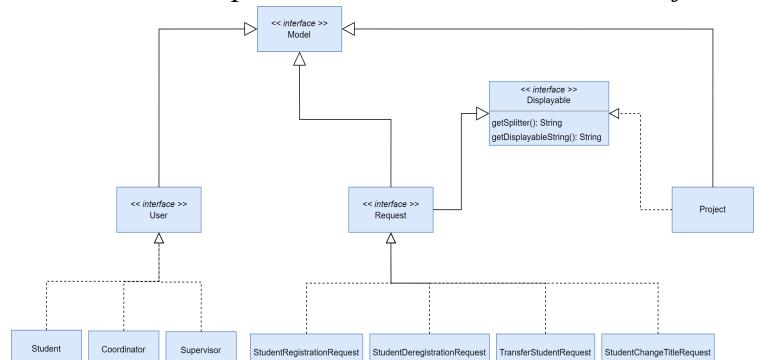
In our project, we apply OCP by creating an abstract class "Repository<Model>" that can be extended to create different types of Repositories such as "ProjectRepository" and "RequestRepository." Each subclass overrides the "getFilePath()" method to allow for easy extension of the Repository system. For requests, we use an interface "Request" to derive different types of requests such as "StudentRegistrationRequest" and "StudentDeregistrationRequest." Each subclass overrides the "get" methods for request details, enabling new request types to be added without changing the existing code.

### 1.3.3 Liskov Substitution Principle (LSP)

To put it simply, Liskov Substitution Principle states that the subtypes must be substitutable for the base types. In the case of our system, the usage of this principle is widely applied. One of the examples is the 'Request' class. Subclasses of the 'Request' include 'StudentRegistrationRequest', 'TransferStudentRequest' etc. All those subclasses are substitutable for the 'Request' class object while ensuring the methods behave correctly. When determining the request type of each different request, we may use different instances of requests to call the 'getRequestType' method, and the corresponding method in the subclass will be called and return the correct request type. Polymorphism is also applied here.

### 1.3.4 Interface Segregation Principle (ISP)

The interface segregation principle refers to that many specific interfaces are better than one general interface. In other words, we should always avoid designing a 'fat interface'. As such, when developing our system, we noticed that this is important to promote maintainability, flexibility and modularity. For instance, we found that the 'Model' must be further divided into separate interfaces such as *User*, *Project* and *Request* so that the different entities can implement the interfaces accordingly. By doing so, we can ensure that the entity classes don't have to implement methods that are not related and reduce the ripple effect when modifying our system.



Furthermore, we have applied the ISP in our project by incorporating the *Displayable* interface. This aligns with the principle of preferring specific interfaces over general ones, avoiding a bloated interface. The *Displayable* interface defines two methods, *getSplitter()* and *getdisplayableString()*, allowing objects to be formatted and displayed.

### 1.3.5 Dependency Injection Principle (DIP)

The dependency injection principle suggests that higher modules must not depend on lower modules, but both should depend on abstraction. In other words, instead of directly depending on the concrete class to perform some operation, we can depend on the interfaces, which are less likely to be changed. In our design, we highly focus on this principle. For example, when getting the ID of a student user, instead of depending on the '*Student*' concrete class, we depend on the *<User>* interface. This will allow us to add more users with the least effort needed in the future and make our system more extendable.

## 1.4 FURTHER ENHANCEMENT

For the further development of our system, we considered the situation that there may more than two students requesting to register for a project created by the same supervisor. When any two of the requests approved, the other requests will remain pending, and the students will be waiting forever for the request to be approved. As such, we designed a feature such that whenever a supervisor is supervising 2 projects, the other requests which are to register for the projects from the same supervisor will be rejected. By doing this, we can ensure that there is no miscommunication occurred, and students will be able to know their status and make their following decisions effectively.

On the other hand, we have made our system more friendly to the users. Every time the user is prompted to enter the ID of a project or request, we will first display the viewable projects or requests (depending on user type), then follow by asking the user to input the ID. With this, the users can interact better with our system's interface and thus increase its effectiveness.

## 1.5 REFLECTION

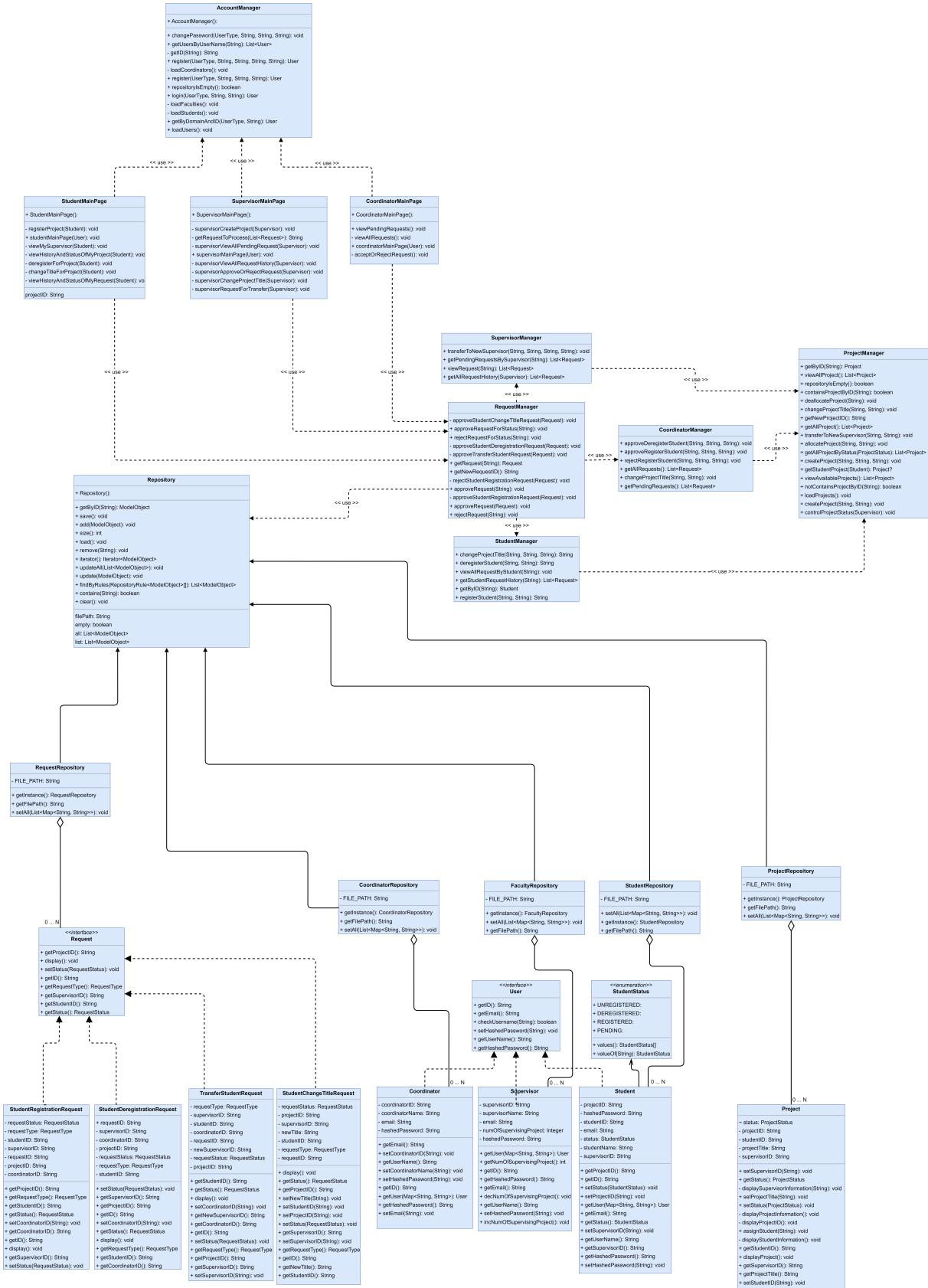
From this assignment, we have seen the importance of design principles through real application. At the beginning phase of this assignment, we found that a slight change in our code will trigger a ripple effect, causing that almost all other parts of our code must be changed. We then refer to the design principles and apply them widely in our assignment. Therefore, software with high cohesion and loose coupling is ultimately important, as to make it to have high flexibility, easily maintainable and extendable.

At the same time, we have learnt to design a software that fits its functions and real-world applications. Considering all users for the software, we kept modifying our design so that it takes into considerations different possible scenarios and ensures that there are no conflicts between the users.

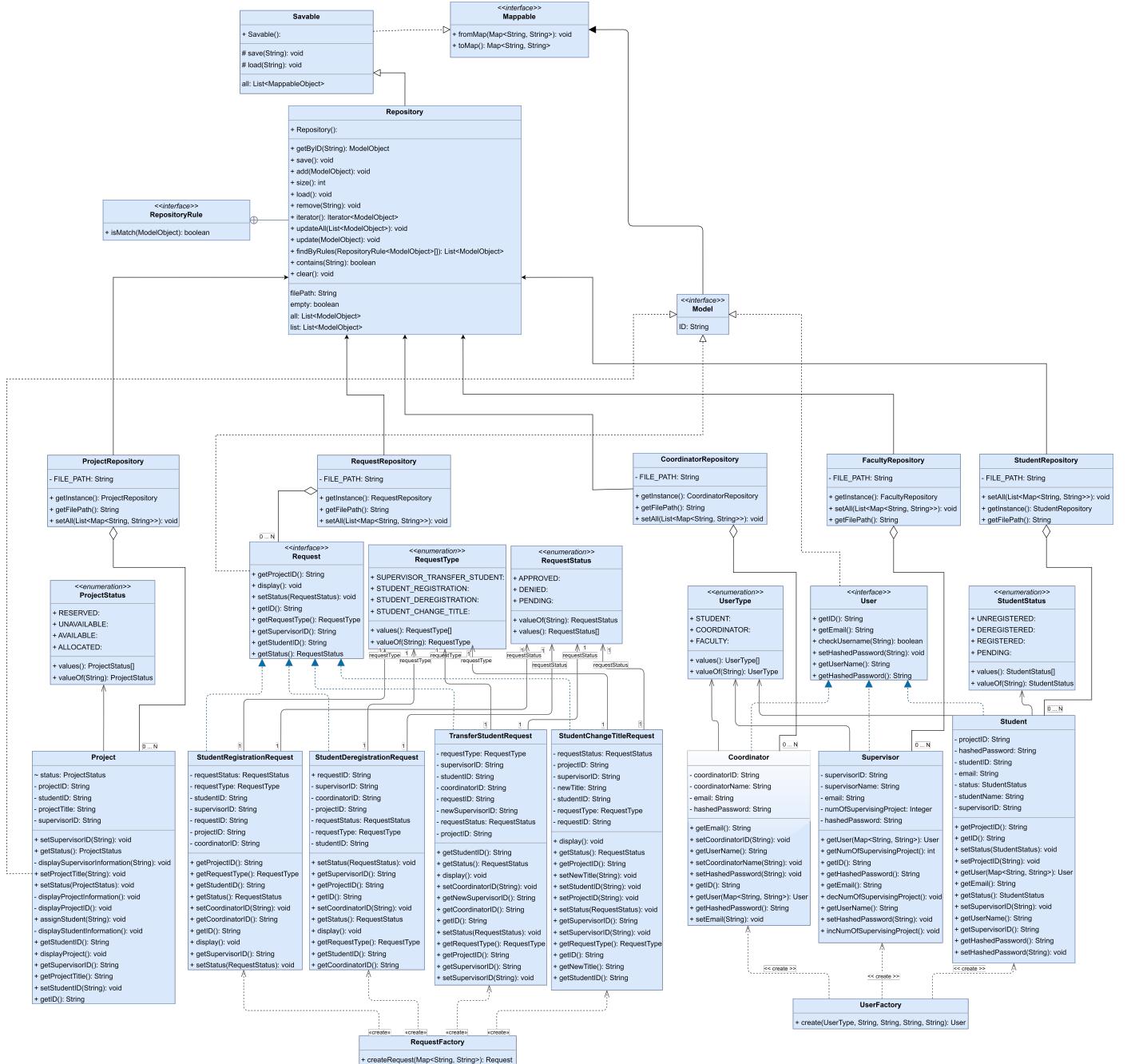
## 2 DETAILED /UML CLASS DIAGRAM

Please Refer to the <UMLClassDiagram> for further details.

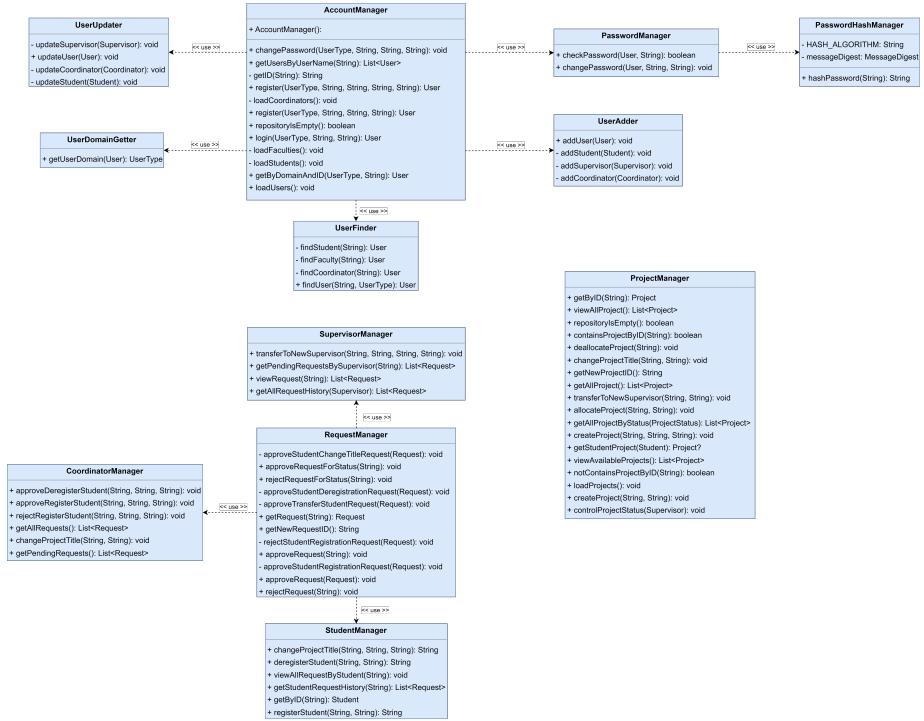
### 2.1 MAIN DIAGRAM



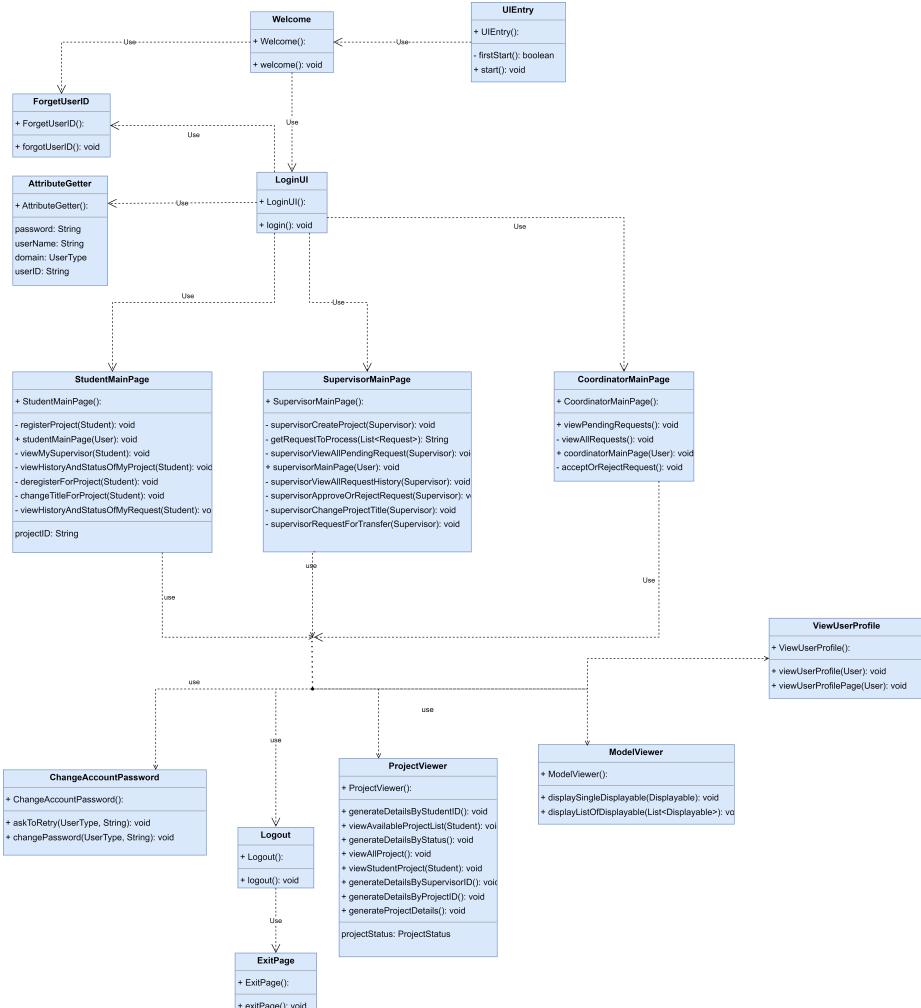
## 2.2 ENTITY SUB DIAGRAM



## 2.3 CONTROLLER SUB DIAGRAM



## 2.4 BOUNDARY SUB DIAGRAM



### 3 TESTING

#### 3.1 WELCOME PAGE

##### 3.1.1 Login Page



Welcome to the Final Year Project Management System!

Please enter your choice to continue.

1. Login  
2. Forget UserID  
3. Exit

Your choice (1-3): 1

=====

1. Student  
2. Supervisor  
3. FYP Coordinator

Please select your domain (1-3): 1

Please enter your UserID (Press enter if you forget): JINQ0003

[Please enter your password:

=====

Welcome to Student Main Page

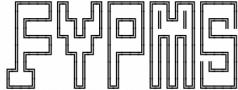
Hello, Jin Qingyang!

1. View my profile  
2. Change my password  
3. View project list  
4. View my project  
5. View my supervisor  
6. Register for a project  
7. Deregister for a project  
8. Change title for a project  
9. View history and status of my request  
10. Logout

=====

Here we use students as an example.

##### 3.1.2 Forgot User ID



Welcome to the Final Year Project Management System!

Please enter your choice to continue.

1. Login  
2. Forget UserID  
3. Exit

Your choice (1-3): 2

Please enter your username: Jin Qingyang

Found 1 user(s) with name Jin Qingyang.

The list of UserID associated with Jin Qingyang is:

User ID	User Domain
JINQ0003	STUDENT

Press Enter key to go back.

If the user chooses the option of *forget UserID*, the system will return the UserID of the user.

##### 3.1.3 Change Password

Welcome to Change Student Password

[Please enter your old password:  
[Please enter your new password:  
[Please enter your new password again:  
Password changed successfully.  
Press [Enter] to go back to the main page.

Successfully change password

Welcome to Change Student Password [Please enter your old password: [Please enter your new password: [Please enter your new password again: Password must be at least 8 characters long. Enter [b] to go back, or any other key to try again.	Welcome to Change Student Password [Please enter your old password: [Please enter your new password: [Please enter your new password again: Two passwords are not the same. Enter [b] to go back, or any other key to try again.
--	---

Entering a new password with fewer than 8 characters or incorrectly entering the password twice is not permitted.

### 3.1.4 Logout

```
Thank you for using our system!
```



## 3.2 Student Register and Deregister

### 3.2.1 Student sends a request to register a project

```
Here is the list of available projects:
```

Machine Learning-based Interference Mitigation in a Multi-tier Networks	
Project ID	P1
Supervisor Name	A S Madhukumar
Supervisor Email Address	ASMAHDUKUMAR@NTU.EDU.SG
Project Status	AVAILABLE
Deep Learning-Driven Edge Caching for 5G-and-Beyond Industrial IoT Networks	
Project ID	P2
Supervisor Name	A S Madhukumar
Supervisor Email Address	ASMAHDUKUMAR@NTU.EDU.SG
Project Status	AVAILABLE
Sonification of geometry 1	
Project ID	P3
Supervisor Name	Alexei Sourin
Supervisor Email Address	ASSOURIN@ntu.edu.sg
Project Status	AVAILABLE
Edge/Cloud Resource Management for Time-Sensitive Applications (2)	
Project ID	P4
Supervisor Name	Arvind Easwaran
Supervisor Email Address	ARVINEDE@NTU.EDU.SG
Project Status	AVAILABLE

```
Here is the project information:
```

Build Software Agents for Power Trading Agent Competition	
Project ID	P6
Supervisor Name	Bo An
Supervisor Email Address	BOAN@NTU.EDU.SG
Project Status	AVAILABLE

```
Are you sure you want to register for this project? (y/[n]): y  
Request submitted!  
Press Enter to go back.
```

The system will first show a list of available projects to choose from. Student then chooses a project, the system will ask student to confirm then submit the request

### 3.2.2 Coordinator approve registration request

```
Welcome to Coordinator Main Page  
Hello, Li Fang!
```

1. View my profile
2. Change my password
3. View all projects
4. View pending requests
5. View all requests' history and status
6. Accept or reject requests NEW!
7. Generate project details
8. Logout

```
Please enter your choice: 6
```

The coordinator can choose to approve/reject request inside the coordinator main page

Accept or Reject Requests	
Request ID   R1	Request Type   STUDENT_REGISTRATION
Request Status   PENDING	Project ID   p6
Supervisor ID   BOAN	Supervisor ID   BOAN
Student ID   JINQ0003	Student ID   JINQ0003

Please enter the ID of the request you want to accept or reject. (Enter 0 to go back.)  
Please enter your choice: R1

Accept or Reject Requests	
Request ID   R1	Request Type   STUDENT_REGISTRATION
Request Status   PENDING	Project ID   p6
Supervisor ID   BOAN	Supervisor ID   BOAN
Student ID   JINQ0003	Student ID   JINQ0003

1. Approve  
2. Reject  
3. Go back  
Please enter your choice:  
1

Here is the updated request:	
Request ID   R1	Request Type   STUDENT_REGISTRATION
Request Status   APPROVED	Project ID   p6
Supervisor ID   BOAN	Supervisor ID   BOAN
Student ID   JINQ0003	Student ID   JINQ0003

Press enter to go back.

The system will display all the pending request, and the coordinator can choose a request by the RequestID to approve/reject.

### 3.2.3 Student requests to deregister

Welcome to Student Main Page	
Hello, Jin Qingyang!	Your current project is:
1. View my profile 2. Change my password 3. View project list 4. View my project 5. View my supervisor 6. Register for a project 7. Deregister for a project 8. Change title for a project 9. View history and status of my request 10. Logout	Build Software Agents for Power Trading Agent Competition
Please enter your choice: 7	Project ID   P6   Supervisor Name   Bo An   Supervisor Email Address   BOAN@NTU.EDU.SG   Student Name   Jin Qingyang   Student Email Address   JINQ0003@e.ntu.edu.sg   Project Status   ALLOCATED
	Are you sure you want to deregister from this project? (y/[n]) y supervisorID = BOAN Successfully sent a request to deregister Press Enter to go back.

Student select the option to deregister. Student confirm to deregister.

Welcome to Student Main Page	
Hello, Jin Qingyang!	You are already registered/deregistered for a project. Press Enter to go back.
1. View my profile 2. Change my password 3. View project list 4. View my project 5. View my supervisor 6. Register for a project 7. Deregister for a project 8. Change title for a project 9. View history and status of my request 10. Logout	
Please enter your choice: 6	

After the coordinator approves the deregistration request, the student is not allowed to choose other projects.

### 3.2.4 Three students request one supervisor

Viewing all projects....	
Deep Reinforcement Learning for Complex Environment	
Project ID   P5	Supervisor Name   Bo An
Supervisor Email Address   BOAN@NTU.EDU.SG	Project Status   UNAVAILABLE
Build Software Agents for Power Trading Agent Competition	
Project ID   P6	Supervisor Name   Bo An
Supervisor Email Address   BOAN@NTU.EDU.SG	Student Name   Jin Qingyang
Student Email Address   JINQ0003@e.ntu.edu.sg	Project Status   ALLOCATED
Designing Negotiation Agents to Parcipitate in International Competition	
Project ID   P7	Supervisor Name   Bo An
Supervisor Email Address   BOAN@NTU.EDU.SG	Student Name   Pu Fanyi
Student Email Address   FPU001@e.ntu.edu.sg	Project Status   ALLOCATED

Enter enter to go back

When two students have registered for the projects of a supervisor, the supervisor's remaining project will become unavailable.

## 3.3 Change Title

### 3.3.1 Supervisor change title

```
Changing the title of project....  
Here are all your projects:  
=====| Deep Reinforcement Learning for Complex Environment |  
| Project ID | P5  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Project Status | AVAILABLE  
=====| Build Software Agents for Power Trading Agent Competition |  
| Project ID | P6  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Project Status | AVAILABLE  
=====| Designing Negotiation Agents to Participate in International Competition |  
| Project ID | P7  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Project Status | AVAILABLE  
=====  
Enter the project ID to change: P5
```

```
Here is the project:  
=====| Deep Reinforcement Learning for Complex Environment |  
| Project ID | P5  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Project Status | AVAILABLE  
=====
```

```
Enter the new title:  
New Title
```

```
Here is the new project after changing the title:  
=====| New Title |  
| Project ID | P5  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Project Status | AVAILABLE  
=====
```

```
Are you sure you want to change the title? (Y/[N])
```

```
y  
Project title changed successfully!
```

```
Enter enter to continue
```

The supervisor can choose a project from list of his projects to change the title. The system will then prompt the user to enter new title. The system will change the title after confirming with the supervisor.

### 3.3.2 Student change title

```
Here is your project:  
=====| Build Software Agents for Power Trading Agent Competition |  
| Project ID | P6  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Student Name | Jin Qingyang  
| Student Email Address | JINQ0003@e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====  
Are you sure you want to change the title of this project?  
Enter [y] to confirm, or press enter to go back.  
y  
Please enter the new title:  
New Title
```

```
Your new project is:  
=====| New Title |  
| Project ID | P6  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Student Name | Jin Qingyang  
| Student Email Address | JINQ0003@e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====
```

```
Are you sure you want to change the title of this project?
```

```
Enter [y] to confirm, or press enter to go back.
```

```
y  
Successfully sent a request to change title  
Press Enter to go back.
```

Student can choose to modify his registered project's title and send the request to the supervisor

```
Approving or rejecting a request....  
Here are all pending requests:  
=====| Request ID | R2  
| Request Type | STUDENT_CHANGE_TITLE  
| Request Status | PENDING  
| Project ID | P6  
| Supervisor ID | BOAN  
| Student ID | JINQ0003  
| New Title | New Title  
=====  
Enter the request ID to approve or reject (or [0] to go back): R2
```

```
Here is the request:  
=====| Request ID | R2  
| Request Type | STUDENT_CHANGE_TITLE  
| Request Status | PENDING  
| Project ID | P6  
| Supervisor ID | BOAN  
| Student ID | JINQ0003  
| New Title | New Title  
=====  
Enter the status to change to APPROVED (A) / REJECTED (R)  
A
```

Only the supervisor of the project have the permission to process the request

```
View Student Project  
=====| New Title |  
| Project ID | P6  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Student Name | Jin Qingyang  
| Student Email Address | JINQ0003@e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====  
Press Enter to go back.
```

The new project title is successfully updated

### 3.4 Transfer Student

```
Processing to transfer....  
Below are all your projects ready for transfer:  
=====  
New Title  
=====  
| Project ID | P6  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Student Name | Jin Qingyang  
| Student Email Address | JINQ00030e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====  
| Designing Negotiation Agents to Participate in  
| International Competition  
=====  
| Project ID | P7  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Student Name | Pu Fanyi  
| Student Email Address | FPU0010e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====  
Enter the project ID to transfer:  
P7  
Here is the project:  
=====  
| Designing Negotiation Agents to Participate in  
| International Competition  
=====  
| Project ID | P7  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Student Name | Pu Fanyi  
| Student Email Address | FPU0010e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====
```

Professor An has two allocated project, he decided to transfer one of the student to other supervisor.

Here is the new project after changing the supervisor:

```
=====  
| Designing Negotiation Agents to Participate in  
| International Competition  
=====  
| Project ID | P7  
| Supervisor Name | Li Fang  
| Supervisor Email Address | ASFLI@NTU.EDU.SG  
| Student Name | Pu Fanyi  
| Student Email Address | FPU0010e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====
```

Are you sure you want to change the supervisor? (Y/[N])  
y

Successfully sent request for transfer!  
Enter enter to continue

View Student Project

```
=====  
| Designing Negotiation Agents to Participate in  
| International Competition  
=====  
| Project ID | P7  
| Supervisor Name | Li Fang  
| Supervisor Email Address | ASFLI@NTU.EDU.SG  
| Student Name | Pu Fanyi  
| Student Email Address | FPU0010e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====
```

Press Enter to go back.

After the Coordinator approve the request, the supervisor of the student will be successfully changed to the new supervisor.

### 3.5 Supervisor create project

```
Viewing all projects....  
=====  
| Deep Reinforcement Learning for Complex Environment  
|  
| Project ID | P5  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Project Status | UNAVAILABLE  
=====  
| Build Software Agents for Power Trading Agent Competition  
|  
| Project ID | P6  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Student Name | Jin Qingyang  
| Student Email Address | JINQ00030e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====  
| Designing Negotiation Agents to Parcipitate in  
| International Competition  
|  
| Project ID | P7  
| Supervisor Name | Bo An  
| Supervisor Email Address | BOAN@NTU.EDU.SG  
| Student Name | Pu Fanyi  
| Student Email Address | FPU0010e.ntu.edu.sg  
| Project Status | ALLOCATED  
=====
```

Enter enter to go back

When two students have registered for the projects of a supervisor, the supervisor's remaining project will become unavailable.