Software Engineering AE2

Name of team: *YHRLLL*

Group Number: *Group 29*

Team Members:

Chaoyi Yang   *2724184Y*

Haoshi Huang   *2635088H*

Shuai Ran   *2633609R*

Xintong Lin   *2824115L*

Yi Liu   *2807941L*

Zexuan Li   *2815250L*

Github link: *https://github.com/1RanShy/Software_Engeneering_Assement2.git*

Table Contents

Team members:

	Team Members	GUID	Contribution	Percentage Contribution
1	Chaoyi Yang	2724184Y	Staff.java Administrator.java PTT.java Requirement.java	16.7%
2	Haoshi Huang	2635088H	Validator.java Controller.java	16.7%
3	Shuai Ran	2633609R	View.java Controller.java	16.7%
4	Xintong Lin	2824115L	FileName.java FileIO.java	16.7%
5	Yi Liu	2807941L	Staff.java ClassDirector.java Course.java Requirement.java	16.7%
6	Zexuan Li	2815250L	Controller.java NumberGeneration.java	16.7%

User stories

In this software engineering design exercise, the main task of our group is to develop a teaching requirements list and a training list, and design and develop a system that involves three roles: Class director, administrator, and part-time teacher (PTT). Class director can submit and modify teaching requests and view all teaching requirements in the system. Administrators can allocate teaching requests to part-time teachers and arrange training plans. Part-time teachers can view teaching tasks and training plans. The specific user descriptions are as follows:

Class director: can view all existing teaching tasks and submit new teaching requests. When submitting a request, they need to fill in course information and time. Class director can also modify their own submitted teaching requests in the system.

Administrator: can view all teaching requests in the system and allocate appropriate teaching requirements to part-time teachers based on course information and teaching time. Administrators can also assign training tasks to part-time teachers and set the information of the training task.

Part-time teacher: can view their own teaching tasks in the system and complete corresponding teaching work. At the same time, teachers can also view the training tasks based on the allocated teaching requirements.

Figure 1.1: UserStory_1
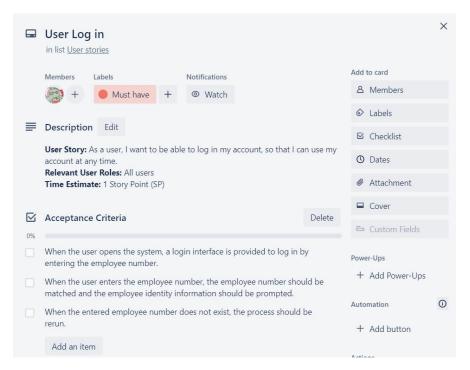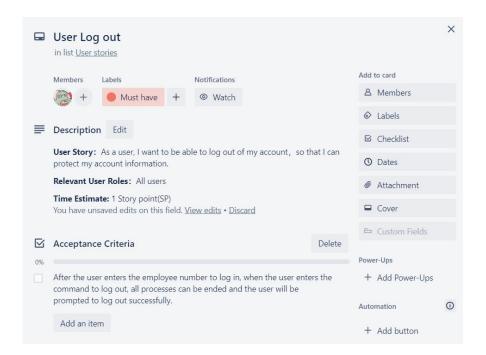

Figure 1.2: UserStory_2

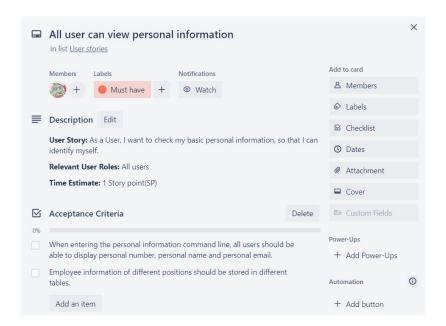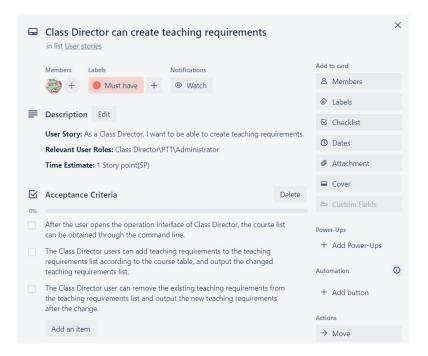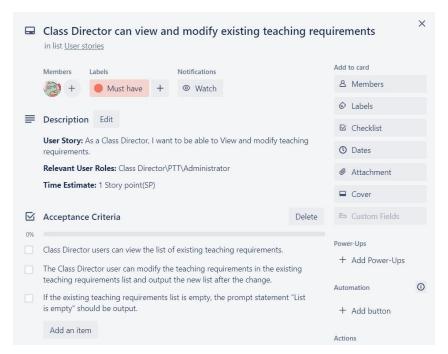Figure 1.3: UserStory_3
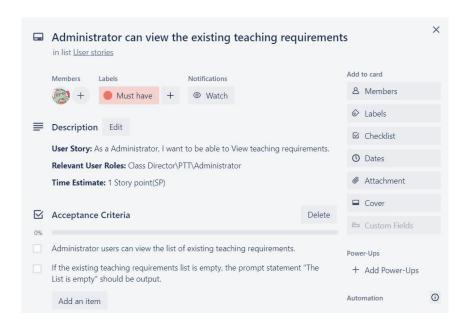

Figure 1.4: UserStory_4

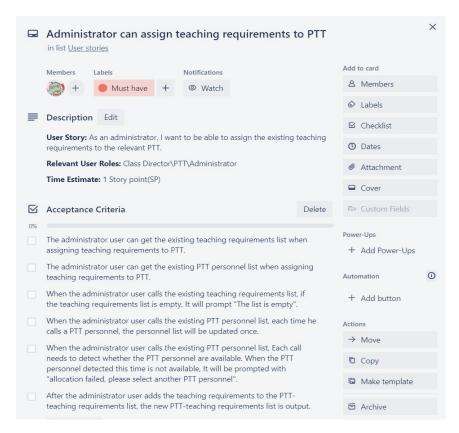Figure 1.4: UserStory_4

Figure 1.5: UserStory_5
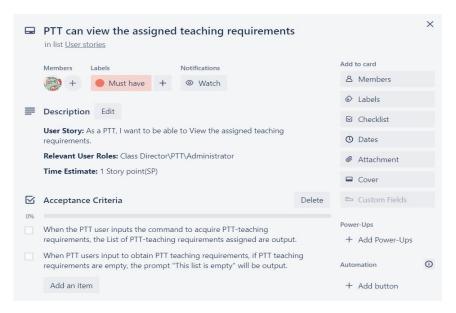
Figure 1.6: UserStory_6

Figure 1.7: UserStory_7

Figure 1.8: UserStory_8

Figure 1.9: UserStory_9

This is our user story, and we have basically completed most of the content required in the user story

Class structure diagram (UML)

Figure 2.1: UML Diagram

Staff

- staffID: String
- staffName: String
- email: String
- requirementIDList: List<String>

+ setStaffID(String staffID): void
+ setStaff(String Name): void
+ setEmail(String email): void
+ getStaffID( ): String
+ getStaffName( ): String
+ getEmail( ): String
+ getRequiremeIDList( ): List<String>
+ addRequirementID(String requirementID): void
+ toString( ): String

PTT

- canTeach: String
- trainingID: String

+ setCanTeach(String canTeach): void
+ setTrainingID(String trainingID): void
+ getCanTeach( ): String
+ addRequirementID(String requirementID): void
+ getIFAvailable( ): Boolean
+ String getTrainingID( ): String
+ toString( ): String

Class director

Adminstrator

Requirement

- requirementID: String
- createDate: String
- classDirectorID: String
- courseID: String
- reqStatus: int
- adminstratorID: String
- pttID: String
- trainingID: String

+ setPttID(String administratorID, String pttID, String trainingID): void
+ setReqStatus(String administratorID)
+ modifyCourseID(String staffID, String courseID): void
+ getRequirementID( ): String
+ getCreateDate( ): String
+ getCourseID( ): String
+ getPttID( ): String
+ getAdministratorID( ): String
+ getReqStatus( ): int
+ getclassDirectorID( ): String
+ toString( ): String

Course

- courseID: String
- courseName: String

+ toString( ): String
+ getCourseID( ): String

Training

- trainingID: string
- trainingName: String
- trainingTime: String
- pttIDList: List<String>
- maxPTT: int

+ addPTTID(String pttID): boolean
+ toString( ): String

1

1

1

1

Figure 2.2: Detailed UML Diagram 2

App

+ start( ): void

Validator

- scanner: Scanner

+ loginValidator(List<Staff> staffList): Staff
+ optionValidator(int maxNo): int
+ inputCourseID(List<Course> courseList): String

1

1

1

1

1

1

1

1

1

1

1

1

Controller

- fileIO: FileIO
- classDirectorList: List<ClassDirector>
- administratorList: List<Administrator>
- pttList: List<PTT>
- courseList: List<Course>
- trainingList: List<Training>
- requirementList: List<Requirement>
- currentUser: Staff
- isOpenSession: Boolean
- validator: Validator
- view: View

+ runClassDirectorSession( ): void
+ runAdministratorSession( ): void
+ runPTTSession( ): void
+ logVerify( ): String
+ getStaffList( ): List<Staff>
+ getRequirementList( ): List<Requirement>
+ modifyRequirement( ): void
+ requirementSearch(String searchKey): Requirement
+ serachAvailablePTT(Strng courseID): List<Staff>
+ searchAvailableTraining( ): List<Training>
+ allocateRequirement( ): void
+ saveData( ): void
+ converToFormatStaffData(List<T> staffList): <T extends Staff> List<ArrayList<String>>
+ converToFormatRequireData(List<Requirement> requirList): List<ArrayList<String>>
+ converToFormatCourseData(List<Course> courseList): List<ArrayList<String>>
+ converTOFormatTrainingData(ist<Training> trainingList): List<ArrayList<String>>

View

+ welcomeView( ): void
+ staffInformationPrint(Staff staff): void
+ staffListPrint(List<Staff> staffList): void
+ requirementListPrint(List<Requirement> rqruirementList): void
+ courseListPrint(List<Course> courseList): void
+ trainingListPrint(List<Training> trainingList): void
+ welcomeStaffView(String staffID): void
+ classDirectorView( ): int
+ pttView( ): int
+ classDirecthirdView( ): int
+ administratorView( ): int
+ backToParentMenu( ): void
+ logOut( ): void
+ pageHeader( ): void
+ toContinue( ): void

FileIO

+ createClassDirectorList():List<ClassDirector>
+ createAdministratorList():List<Administrator>
+ createPTTList():List<PTT>
+ createCourseList():List<Course>
+ createTrainingList():List<Training>
+ createRequirementList():List<Requirement>
+ writeFile():void
+ convertStaffListToList(List<Staff>) : List<ArrayList<String>>
+ convertCourseListToList(List<Course>) : List<ArrayList<String>>
+ convertRequirementListToList(List<Requirement>) : List<ArrayList<String>>
+ convertTrainingListToList(List<Training>) : List<ArrayList<String>>
- readFile

NumberGeneration

+ requirementNum(List<String> requirementID): String

FileName

- DATA_DIR : String
- FILENAME_STAFFLIST : String
- FILENAME_CLASSDIRECTORLIST : String
- FILENAME_ADMINISTRATORLIST : String
- FILENAME_PTTLIST : String
- FILENAME_REQUIREMENTSLIST : String
- FILENAME_COURSELIST : String
- FILENAME_TRAININGLIST : String

+ getStaff() : String
+ getClassDirector() : String
+ getAdministrator() : String
+ getPTT() : String
+ getRequire() : String
+ getCourse() : String
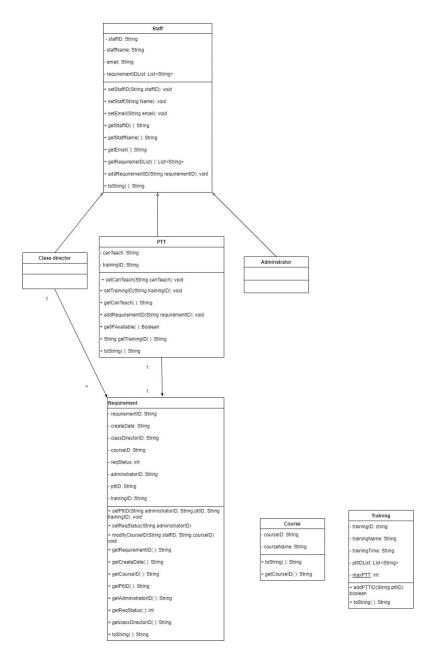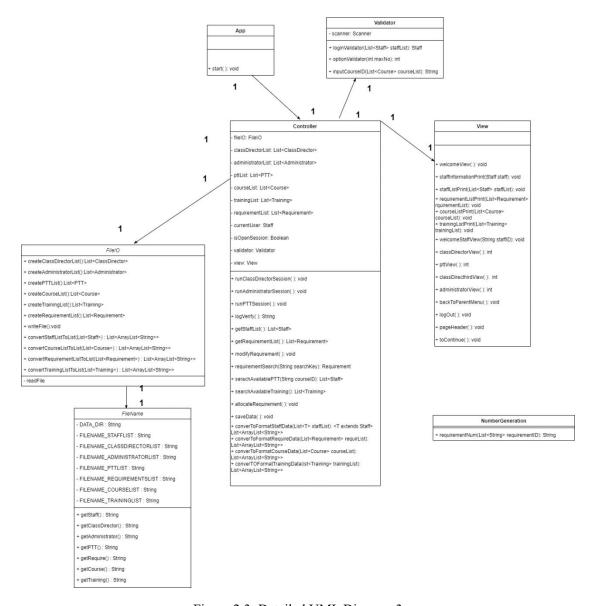+ getTraining() : String

---

Figure 2.3: Detailed UML Diagram 3

App class: used to launch the entire project, identify the current user type, and invoke different user session functions.

Controller package: contains only one Controller class, and calls other classes, such as View and Model, to control menu interaction processes and data changes for different types of users.

Service package: Implements various business logic, including the reading class FileIO for data files, the file name class FileName set as a global static resource, and

11

the number automatically generates a class (number generation) to generate a new number for the newly created instance class.

Model package: Defines various entity classes, including the Staff class that records basic employee information. The Administrator class, ClassDirector class, and PTT class inherit the Staff class as subclasses. The Course class records basic information about courses, the Training class records basic information about teacher training projects, and the Requirement class records information about teaching needs. "A table in a database corresponds to an entity class, and the attributes of the class correspond to the fields in the table one by one. However, due to time constraints, no code for the database version was written, so only the data was saved in a txt file.".

View package: Implement user interaction functions. The View class is responsible for interface output, and the Validator class is responsible for obtaining user input and ensuring the correctness of input.

Software structure and design patterns

In our project design process, we chose MVC as our software architecture and divided the software application into three core parts: the Model, the View, and the Controller, each responsible for different tasks.

The Model is responsible for processing data and represents the business logic and data in the application, including operations such as data storage, retrieval, update, and deletion. The View is responsible for presenting data and is typically a part of the user interface, presenting data in a visual way to the user. The Controller is the interactive part of the application that receives input from the user and passes it on to the Model or View for processing. The Controller operates the Model and View based on the user's requests to ensure proper application operation.

In the MVC structure, each component can be developed and maintained independently, making MVC particularly suitable for large, complex applications because it improves code reusability, scalability, and maintainability.

The singleton pattern is a very useful creational design pattern that ensures a class has only one instance and provides a global access point for that instance, making it easy to access the unique instance. The singleton pattern is very useful when it is necessary to restrict the instantiation of a class, such as controlling resource access or ensuring that only one database connection instance exists.

In our design, we applied the singleton pattern to ensure that all function classes are singletons. This avoids recursion when users frequently switch between different functions, thus improving code performance and readability.

Additionally, we also use the singleton pattern to handle global variables, including

lists of all Staff, all teaching requests, filenames for storing permanent information, and the current logged-in user (Staff). By treating these global variables as singletons, we ensure that they have only one instance and can be easily called by other modules and functions, thus improving code reusability and maintainability.

When implementing the Controller class, the design pattern of lazyLoadingSingleton is used, which limits the controller class to be instantiated only once and provides global access points. This is because the controller class saves a large amount of data, and the singleton pattern can save resources.It is also possible to ensure that the controller class creates only one instance, which is useful if future functionality requires sharing data across multiple application modules.

Sequence diagrams for each user story

A sequence diagram for a user story is a graphical representation that displays the interaction process between objects in a user story. The use of sequence diagrams can help the development team better understand user requirements and expectations by displaying the interaction process between the user and the system. Team members can gain a clearer understanding of the user's actions and expected results through the sequence diagram, allowing for better development of system functionality that meets user needs. Therefore, we have designed a sequence diagram for each user story to guide and illustrate the program development process. The specific sequence diagram is shown below.

User Story: As a user, I want to be able to log in my account, so that I can use my account at any time.
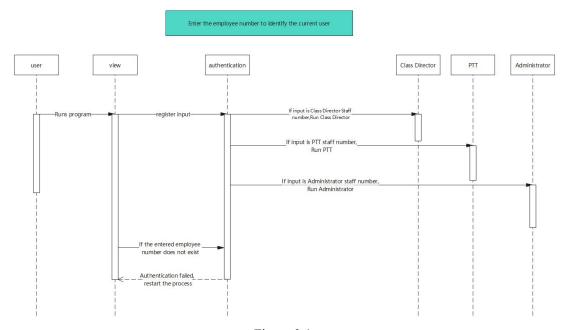

Figure 3.1

User Story： As a user, I want to be able to log out of my account, so that I can protect
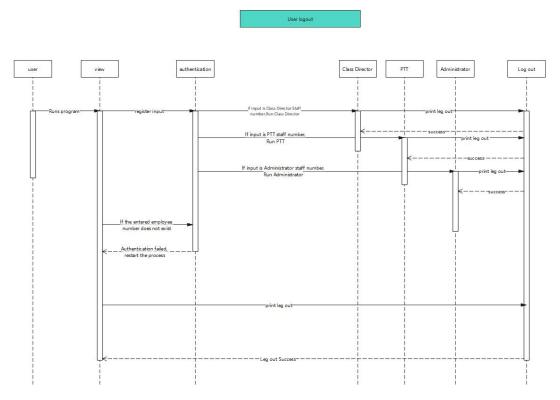
my account information.

Figure 3.2

User story： As a User, I want to check my basic personal information, so that I can identify myself.
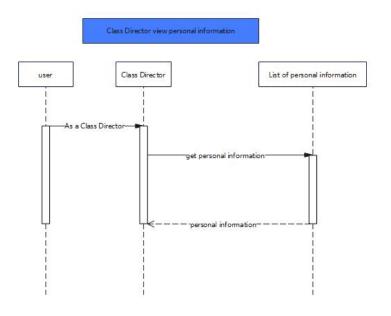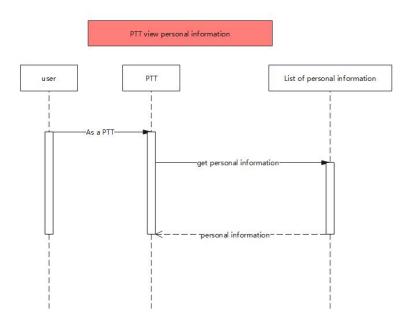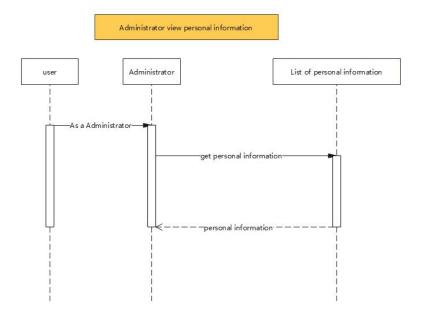

Figure 3.3

Figure 3.4

Figure 3.5

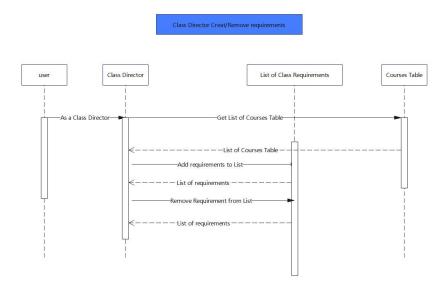**User Story**: As a Class Director, I want to be able to create teaching requirements.

Figure 3.6

User Story： Class Director can view and modify existing teaching requirements
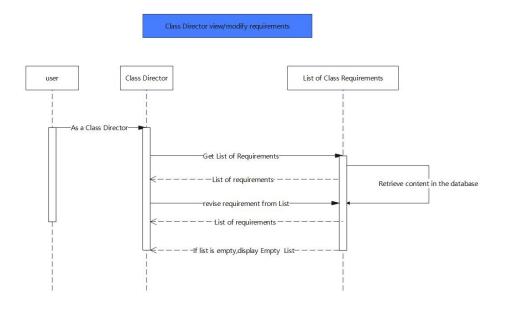

Figure 3.7

User Story: As a Administrator, I want to be able to View teaching requirements.

Figure 3.8

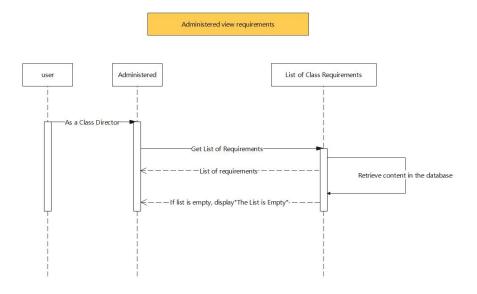**User Story:** As an administrator, I want to be able to assign the existing teaching requirements to the relevant PTT.
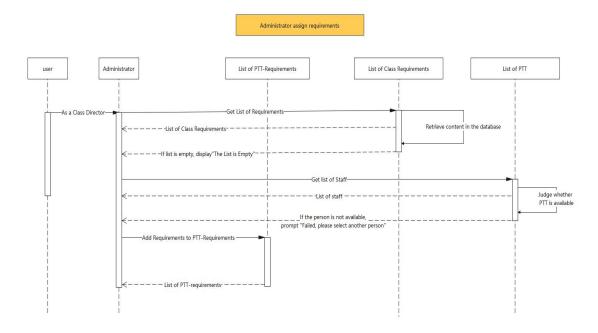

Figure 3.9

User Story: As an administrator, I want to be able to assign the existing training to the relevant PTT.
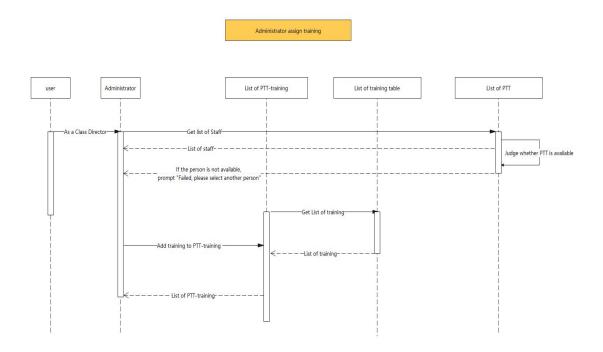

Figure 3.10

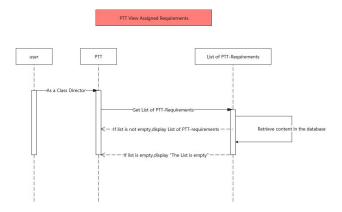**User Story:** As a PTT, I want to be able to View the assigned teaching requirements.

Figure 3.11

User Story: As a PTT, I want to be able to View the assigned trainings.
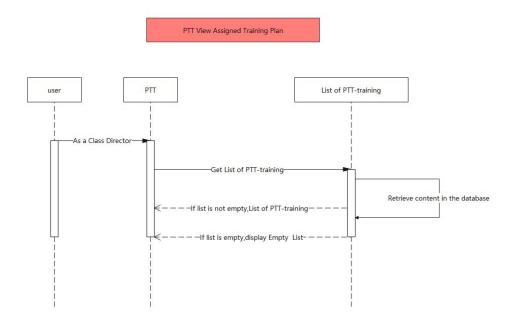

Figure 3.12

System operation results

Each role class has a corresponding txt file which stores the relevant data for that class. First, we read the data from the txt file and create the corresponding class based on the data that was read. Then, we follow the prompts in the menu to select the corresponding options and test whether each menu option can provide the correct output.

Administrator Test

After logging in to the system with the ID number of a staff, determine the type and display the functions that he can use. The following figure shows the function of detecting and viewing the basic information of the current user.

```
Please enter your staff number:
sa001

------ Welcome back, Bob Don ------

--------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Allocate requirements to PTT

------ 3. Check Requirements

------ 4. Log Out

------ 5. Exit

Please enter number from 1 to 5
1


--------------------------------------------
This is your information :
staffID:sa001    staffName:Bob Don    eamil:Bob@gmail.com

Please press enter to continue
```

Figure: Administrator Test_1

Assign the appropriate PTT for the outstanding requirement and select the Training for the PTT with remaining places.If there is no suitable PTT and Training, both prompt and return to the main menu.

```
Please enter your staff number:
sa001

------ Welcome back, Bob Don ------

---------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Allocate requirements to PTT

------ 3. Check Requirements

------ 4. Log Out

------ 5. Exit

Please enter number from 1 to 5
2


---------------------------------------------
1. This is the requirement : Completed
requiredmentID:r0001                                    classDirectorId:sc001
The created time:10-02-2023    courseId:cm001    pttId:sp001    administrator:sa002    trainingID:t0001

2. This is the requirement : Incompelete
requiredmentID:r0002                                    classDirectorId:sc002
The created time:11-02-2023    courseId:cm001    pttId:null    administrator:sa002    trainingID:null

3. This is the requirement : Incompelete
requiredmentID:r0003                                    classDirectorId:sc001
The created time:12-02-2023    courseId:cp001    pttId:null    administrator:null    trainingID:null

4. This is the requirement : Completed
requiredmentID:r0004                                    classDirectorId:sc001
The created time:15-03-2023    courseId:cm002    pttId:sp004    administrator:sa001    trainingID:t0001

Please press enter to continue
Please enter number from 1 to 4
2
There is no matched teacher in system!
r0002 Requirement need to be modified! There are no suitable teachers.
Please press enter to continue


---------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Allocate requirements to PTT
```

Figure: Administrator Test_2

```
Please enter number from 1 to 5
2


---------------------------------------------
1. This is the requirement : Completed
requiredmentID:r0001                                    classDirectorId:sc001
The created time:10-02-2023    courseId:cm001    pttId:sp001    administrator:sa002    trainingID:t0001

2. This is the requirement : Need to be modified
requiredmentID:r0002                                    classDirectorId:sc002
The created time:11-02-2023    courseId:cm001    pttId:null    administrator:sa001    trainingID:null

3. This is the requirement : Incompelete
requiredmentID:r0003                                    classDirectorId:sc001
The created time:12-02-2023    courseId:cp001    pttId:null    administrator:null    trainingID:null

4. This is the requirement : Completed
requiredmentID:r0004                                    classDirectorId:sc001
The created time:15-03-2023    courseId:cm002    pttId:sp004    administrator:sa001    trainingID:t0001

Please press enter to continue
Please enter number from 1 to 4
3


---------------------------------------------
1. This is your information :
staffID:sp002    staffName:Joe Barry    eamil:Joe@gmail.com    canTeach:p

Please press enter to continue
Please enter number from 1 to 1
1


---------------------------------------------
1. TrainingID:t0001    trainingName:t1    trainingTime:10911

All the PTTS who take this training : sp001sp004
The remaining PTT number : 1

2. TrainingID:t0002    trainingName:t2    trainingTime:21214

All the PTTS who take this training :
The remaining PTT number : 3

3. TrainingID:t0003    trainingName:t3    trainingTime:31416

All the PTTS who take this training :
The remaining PTT number : 3

Please press enter to continue
Please enter number from 1 to 3
1
Allocate Joe Barry(sp002) to r0003 Requirement successfully!
Please press enter to continue
```

Figure: Administrator Test_3

View existing requirements.

```
-------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Allocate requirements to PTT

------ 3. Check Requirements

------ 4. Log Out

------ 5. Exit

Please enter number from 1 to 5
3


-------------------------------------------
1. This is the requirement : Completed
requiredmentID:r0001                              classDirectorId:sc001
The created time:10-02-2023   courseId:cm001   pttId:sp001   administrator:sa002   trainingID:t0001

2. This is the requirement : Need to be modified
requiredmentID:r0002                              classDirectorId:sc002
The created time:11-02-2023   courseId:cm001   pttId:null   administrator:sa001   trainingID:null

3. This is the requirement : Completed
requiredmentID:r0003                              classDirectorId:sc001
The created time:12-02-2023   courseId:cp001   pttId:sp002   administrator:sa001   trainingID:t0001

4. This is the requirement : Completed
requiredmentID:r0004                              classDirectorId:sc001
The created time:15-03-2023   courseId:cm002   pttId:sp004   administrator:sa001   trainingID:t0001

Please press enter to continue
```

Figure: Administrator Test_4

Classdirector Test

Like administrator, log in and display the menu for class director

```
------ Welcome to PTT Management System ------
Please enter your staff number:
sc003

------ Welcome back, Alice Amy ------

--------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Create Requirements

------ 3. Modify Requirements

------ 4. Log Out

------ 5. Exit

Please enter number from 1 to 5
```

Figure : ClassDirector Test_1

Display the basic information of the current user.

```
--------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Create Requirements

------ 3. Modify Requirements

------ 4. Log Out

------ 5. Exit

Please enter number from 1 to 5
1


--------------------------------------------
This is your information :
staffID:sc003    staffName:Alice Amy    eamil:Lan@gmail.com

Please press enter to continue
```

Figure : ClassDirector Test_2

Create a new requirement and select one of the existing courses as the course requirement.

```
Please enter number from 1 to 5
2


--------------------------------------------
1. This is the requirement : Completed
requiredmentID:r0001                                    classDirectorId:sc001
The created time:10-02-2023   courseId:cm001   pttId:sp001   administrator:sa002   trainingID:t0001

2. This is the requirement : Need to be modified
requiredmentID:r0002                                    classDirectorId:sc002
The created time:11-02-2023   courseId:cc001   pttId:null   administrator:sa002   trainingID:null

3. This is the requirement : Incompelete
requiredmentID:r0003                                    classDirectorId:sc001
The created time:12-02-2023   courseId:cp001   pttId:null   administrator:null   trainingID:null

4. This is the requirement : Completed
requiredmentID:r0004                                    classDirectorId:sc001
The created time:15-03-2023   courseId:cm002   pttId:sp004   administrator:sa001   trainingID:t0001

Please press enter to continue


--------------------------------------------
------ Couse Information ------
1. courseID: cm001, courseName: statistics

2. courseID: cp001, courseName: physics

3. courseID: cc001, courseName: chemistry

4. courseID: cm002, courseName: math

Please enter number from 1 to 4
2
Create r0005 Requirement successfully!
Please press enter to continue
```

Figure : ClassDirector Test_3

Modify the course information of requirement.First determine whether it is created by yourself. If so, all the courses that can be replaced will be given. You must choose a different course to complete the modification.

```
Please enter number from 1 to 5
3


-------------------------------------------
1. This is the requirement : Completed
requiredmentID:r0001                        classDirectorId:sc001
The created time:10-02-2023   courseId:cm001   pttId:sp001   administrator:sa002   trainingID:t0001

2. This is the requirement : Need to be modified
requiredmentID:r0002                        classDirectorId:sc002
The created time:11-02-2023   courseId:cc001   pttId:null   administrator:sa002   trainingID:null

3. This is the requirement : Incompelete
requiredmentID:r0003                        classDirectorId:sc001
The created time:12-02-2023   courseId:cp001   pttId:null   administrator:null   trainingID:null

4. This is the requirement : Completed
requiredmentID:r0004                        classDirectorId:sc001
The created time:15-03-2023   courseId:cm002   pttId:sp004   administrator:sa001   trainingID:t0001

5. This is the requirement : Incompelete
requiredmentID:r0005                        classDirectorId:sc003
The created time:16-03-2023   courseId:cp001   pttId:null   administrator:null   trainingID:null

Please press enter to continue
------ Please enter a number below :

------ 1. Choose a requirement number to modify

------ 2. Return the previous content

Please enter number from 1 to 2
2
Back to Parent Menu!



-------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Create Requirements

------ 3. Modify Requirements

------ 4. Log Out

------ 5. Exit
```

Figure : ClassDirector Test_4

```
Please enter number from 1 to 5
3


-------------------------------------------
1. This is the requirement : Completed
requiredmentID:r0001                              classDirectorId:sc001
The created time:10-02-2023   courseId:cm001   pttId:sp001   administrator:sa002   trainingID:t0001

2. This is the requirement : Need to be modified
requiredmentID:r0002                              classDirectorId:sc002
The created time:11-02-2023   courseId:cc001   pttId:null   administrator:sa002   trainingID:null

3. This is the requirement : Incompelete
requiredmentID:r0003                              classDirectorId:sc001
The created time:12-02-2023   courseId:cp001   pttId:null   administrator:null   trainingID:null

4. This is the requirement : Completed
requiredmentID:r0004                              classDirectorId:sc001
The created time:15-03-2023   courseId:cm002   pttId:sp004   administrator:sa001   trainingID:t0001

5. This is the requirement : Incompelete
requiredmentID:r0005                              classDirectorId:sc003
The created time:16-03-2023   courseId:cp001   pttId:null   administrator:null   trainingID:null

Please press enter to continue
------ Please enter a number below :

------ 1. Choose a requirement number to modify

------ 2. Return the previous content

Please enter number from 1 to 2
1

Please choose a requirement number:
Please enter number from 1 to 5
2
You have no right to modify other's requiremnet. Please choose another one.
Back to Parent Menu!
```

Figure : ClassDirector Test_5

```
------ Welcome to PTT Management System ------
Please enter your staff number:
sc002

------ Welcome back, Donald Nick ------

--------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Create Requirements

------ 3. Modify Requirements

------ 4. Log Out

------ 5. Exit

Please enter number from 1 to 5
3


--------------------------------------------
1. This is the requirement : Completed
requiredmentID:r0001                          classDirectorId:sc001
The created time:10-02-2023    courseId:cm001   pttId:sp001   administrator:sa002   trainingID:t0001

2. This is the requirement : Need to be modified
requiredmentID:r0002                          classDirectorId:sc002
The created time:11-02-2023    courseId:cc001   pttId:null   administrator:sa002   trainingID:null

3. This is the requirement : Incompelete
requiredmentID:r0003                          classDirectorId:sc001
The created time:12-02-2023    courseId:cp001   pttId:null   administrator:null   trainingID:null

4. This is the requirement : Completed
requiredmentID:r0004                          classDirectorId:sc001
The created time:15-03-2023    courseId:cm002   pttId:sp004   administrator:sa001   trainingID:t0001

Please press enter to continue
------ Please enter a number below :

------ 1. Choose a requirement number to modify

------ 2. Return the previous content

Please enter number from 1 to 2
1

Please choose a requirement number:
Please enter number from 1 to 4
2


--------------------------------------------
------ Couse Information ------
1. courseID: cm001, courseName: statistics

2. courseID: cp001, courseName: physics

3. courseID: cc001, courseName: chemistry

4. courseID: cm002, courseName: math

Please enter number from 1 to 4
3
The courseID is the same. Please choose another one.


--------------------------------------------
------ Couse Information ------
1. courseID: cm001, courseName: statistics

2. courseID: cp001, courseName: physics

3. courseID: cc001, courseName: chemistry

4. courseID: cm002, courseName: math

Please enter number from 1 to 4
1
Modified successfully!
Back to Parent Menu!


--------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Create Requirements

------ 3. Modify Requirements

------ 4. Log Out

------ 5. Exit

Please enter number from 1 to 5

```

Figure : ClassDirector Test_6

All users can log out of the system and re-enter the employee number to log in.

```
-------------------------------------------
------ Please enter a number below :

------ 1. Check your own information

------ 2. Create Requirements

------ 3. Modify Requirements

------ 4. Log Out

------ 5. Exit

Please enter number from 1 to 5
4
Log out successfully!

Please enter your staff number:
█
```

Figure : ClassDirector Test_7

Part Time Teacher

View the requirements and training of the current PTT.

```
Please enter your staff number:
sp002

------ Welcome back, Joe Barry ------

-------------------------------------------
------ Please enter a number below

------ 1. Check your own information

------ 2. Check your requirement and Training

------ 3. Log Out

------ 4. Exit

Please enter number from 1 to 4
2


-------------------------------------------
This is the requirement : Completed
requiredmentID:r0003                                    classDirectorId:sc001
The created time:12-02-2023   courseId:cp001   pttId:sp002   administrator:sa001   trainingID:t0001

Please press enter to continue
```

Figure : PTT Check_Requirement Test

Output to the txt file

When the user selects exit to exit the program, all information added by the current modification is stored in the relevant file.

```
1    RequirementID,createDate,classDirectorID,courseID,stuats,administratorID,pttID,trainingID
2    r0001,10-02-2023,sc001,cm001,1,sa002,sp001,t0001
3    r0002,11-02-2023,sc002,cm001,0,sa002,sp003,t0001
4    r0003,12-02-2023,sc001,cp001,0
5    r0004,15-03-2023,sc001,cm002,1,sa001,sp004,t0001
6    r0005,16-03-2023,sc001,cc001,0
7    r0006,16-03-2023,sc001,cm001,0
8    r0007,16-03-2023,sc002,cm002,0
9
```

Figure : Data Output Test

Retrospective

Fortunately, everyone on our team is willing to share the work and be responsible for it. Therefore, our team is very cohesive and happy to cooperate. I think this is the main advantage of our team. However, it has to be mentioned that the user story is determined and conceived at the early stage of the project, and it is not clear how to implement the content of the project design according to the requirements at the early stage of the project design. Therefore, in the second stage of project design, we spent a lot of time discussing how to realize user stories, how to determine the final scale of the project, and how to filter our final user stories. After confirming the user story, there are some differences in how to implement each function. Some members want to use a more centralized and simple way to achieve it, while others want to ensure the diversity and reliability of the final product, and different ideas determine the amount of code in the implementation process.

Due to the size of the project, we encountered some difficulties in determining the distribution of each member's class. Therefore, we chose to actively hold a group meeting to ensure that the understanding and cooperation between our groups are strengthened and that the final task is smoothly distributed among members. Positive communication and cooperation saved us a lot of time for later modification and adaptation. When confirming the structure level of the project, due to the lack of discussion on the relationship between classes in the early stage of our project design, it is difficult and inconsistent to determine a reasonable design pattern and structure level.

The final code implements the basic functions of our selected user stories. In the actual future iterations of the plan, the basic functions may be enriched and improved, and the diversity will be increased.