Architecture Design

for

UIC Teamwork Contribution Assessment System

Version 1.0 approved

Prepared by Yuepeng LONG, Rongkai LIU, Yitao QIU and Hongyue SHEN

AIR

Sunday, April 5, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Yuepeng LONG, Rongkai LIU, Yitao QIU, Hongyue SHEN	2020-04-05	Initial version of this document	1.0

1. Overview

1.1 Project description

This project aims to bring UIC staffs and students a better and more convenient platform to do the record, management and assessment about teamwork contribution. With this application, teachers can easily specify the group allocation, manage the assessments uploaded by students and get the reference of teamwork contribution for grading students, while students can do submission of each assignment or group project, and assess their classmates' work clearly.

1.2 References

Y. P. Long, et al., "Software Requirements Specification for UIC Teamwork Contribution Assessment System v1.4" Sunday, April 5, 2020.

1.3 Design purpose

The design introduced in this document is used to divide our entire system into several subsystems with high cohesion and loose coupling for better division of labor. The capacities of each subsystem vary little and can be done independently and efficiently by different programmers with less communication.

2. Overall description

2.1 Use case diagram and class diagram

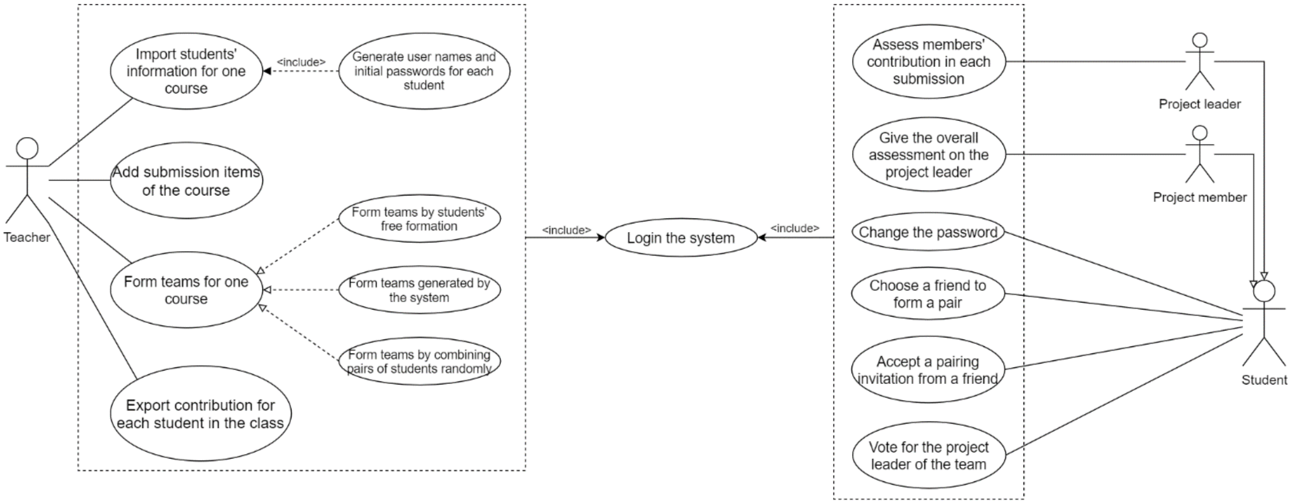

Import students' information for one course
<include>
Generate user names and initial passwords for each student
Teacher
Add submission items of the course
Form teams by students' free formation
Form teams for one course
Form teams generated by the system
Form teams by combining pairs of students randomly
Export contribution for each student in the class
<include>
Login the system
<include>
Assess members' contribution in each submission
Project leader
Give the overall assessment on the project leader
Project member
Change the password
Choose a friend to form a pair
Accept a pairing invitation from a friend
Vote for the project leader of the team
Student

Figure 1 Use Case Diagram

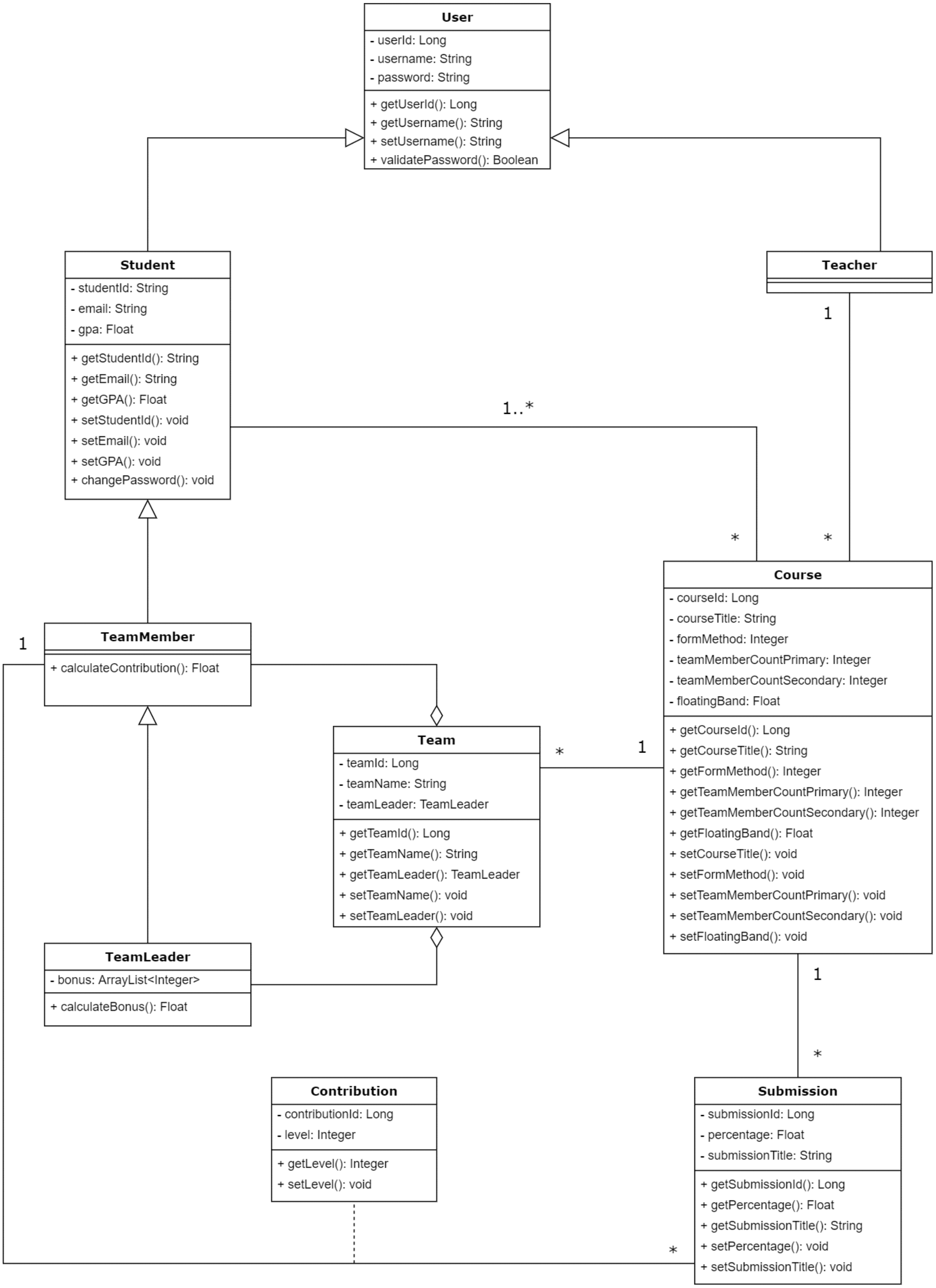


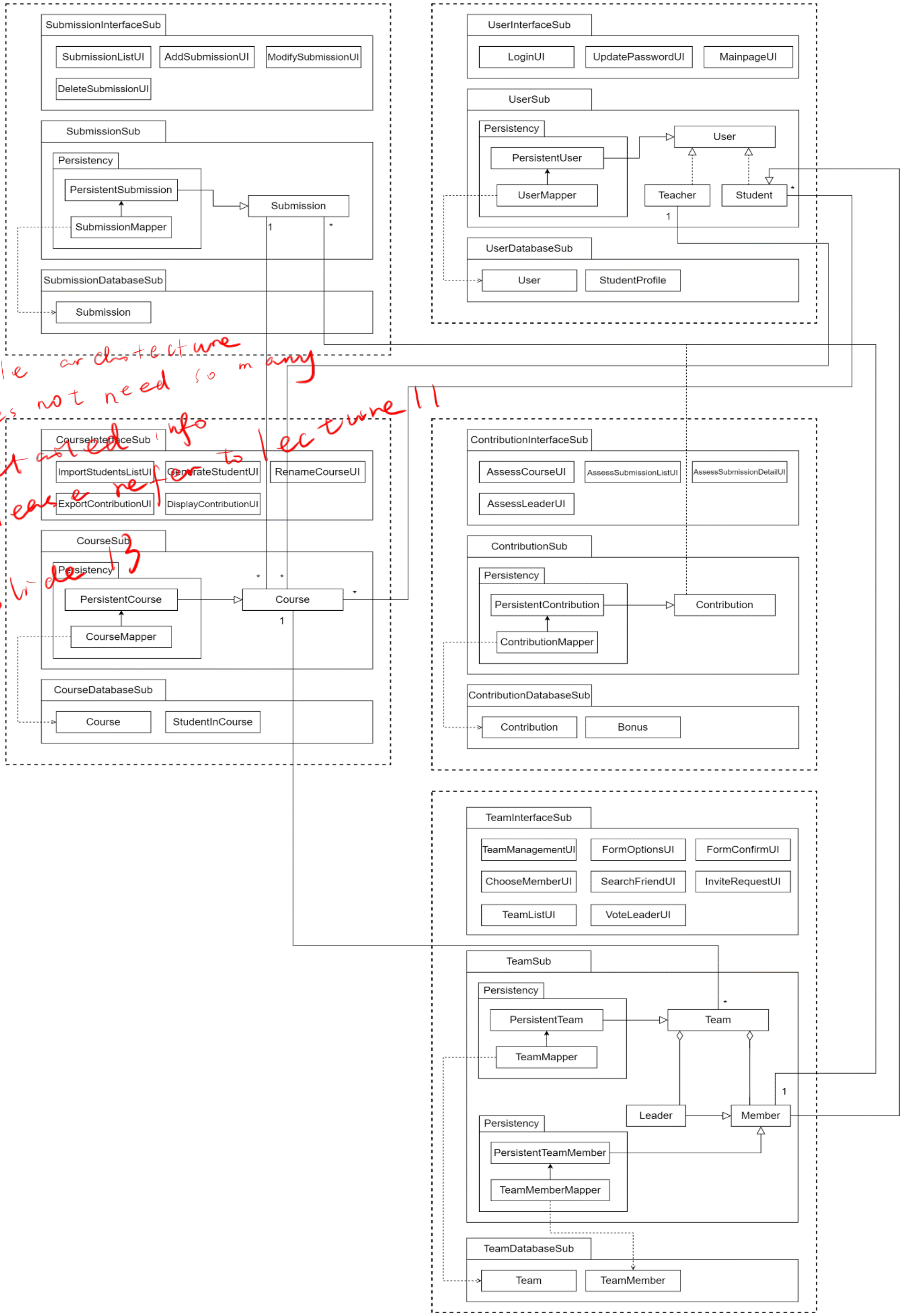
Figure 2 Class Diagram

2.2 Design model

The system is using MVT (a.k.a. Model-View-Template) three-tier architecture. This architecture model can help to divide the system into different parts (like database models, view functions and front-end templates) and allow developers to develop it separately and concurrently.

2.3 System architecture

The system is divided into 5 subsystems: User Subsystem, Course Subsystem, Submission Subsystem, Team Subsystem and Contribution Subsystem.
Diagram that indicates the relationship between subsystems is shown below.

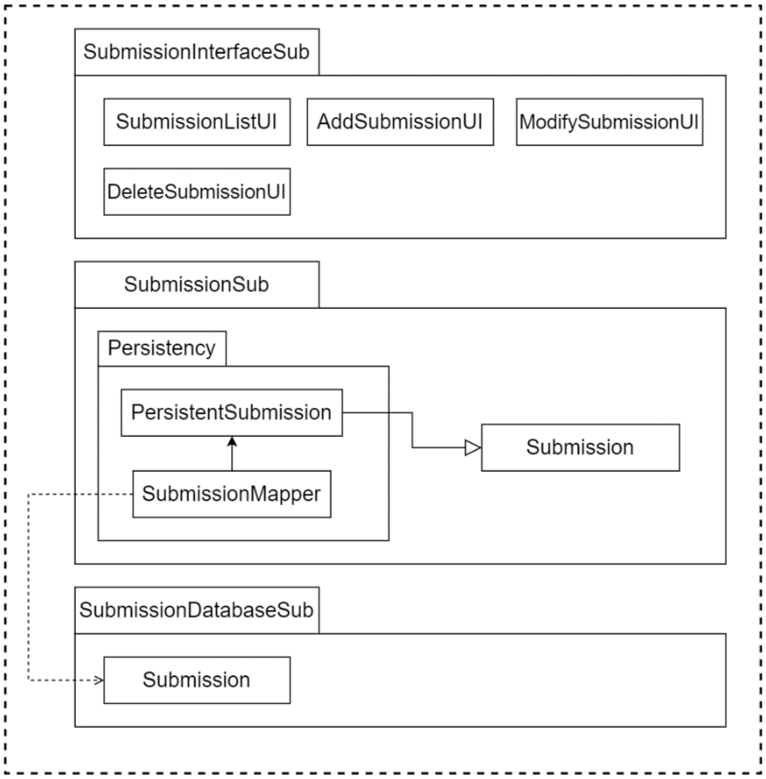


whole architecture does not need so many detailed info please refer to lecture 11 slide 13

Figure 3 Architecture Design Diagram

3. System architecture

3.1 Submission Subsystem



3.1.1 Description

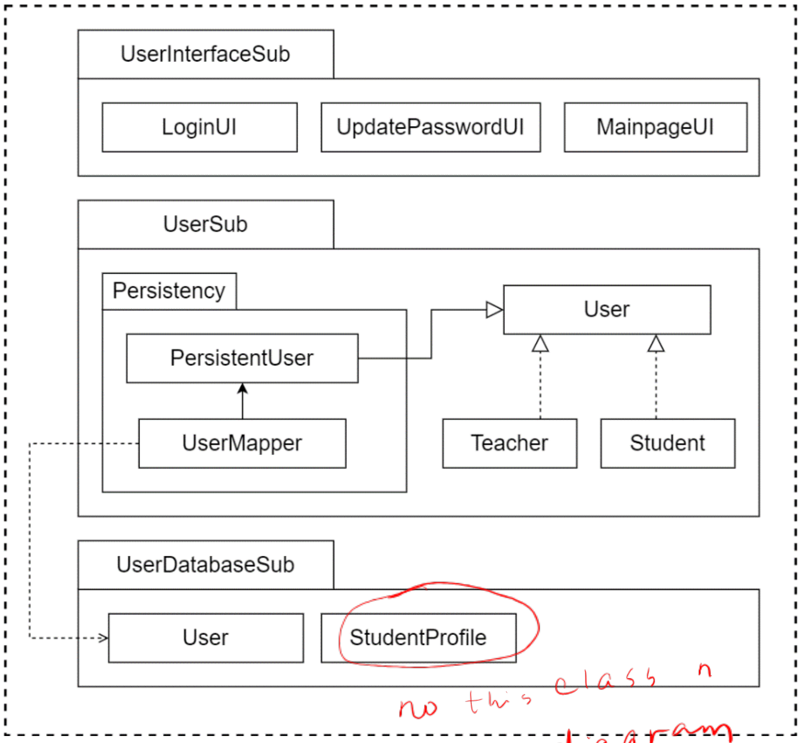
The subsystem contains 3 parts: Submission interface, application logic and storage part. All components in the “Submission interface” are for editing the submissions of the course, including listing all submissions, adding new submissions, modifying existing submissions and deleting submissions. In the application logic part, the “Submission” component is for getting and setting the properties of every submission. Since the data in “Submission” component is persistent, it will store its data to the “Submission” component in storage part through the “SubmissionMapper” component. In the storage part, the “Submission” component realizes the storage, retrieval and query of the data it received.

3.1.2 Database

There is one database relation table in the Submission Subsystem. The “Submission” table is used to record the properties of every submissions. The fields of the relation are listed below.

Submission			
Submission ID	Related course	Percentage	Title

3.2 User Subsystem



no this class n Class diagram

3.2.1 Description

The subsystem contains 3 parts: interface part, application logic part, and database part. The interface part of the system contains Login interface, update password interface, and main page. The Login interface gives users the form to fill, enabling them to log in the system for further operation. In update password interface, students can change their passwords into new ones. In main page, all the accessible functions will be displayed for users to access according to their roles. The data of user class is persistent, which will be store in the database. The application logic part of the system contains “User”. With the “User” component, all the teachers and students have identification in the system to do further operations and access other parts in the system. In the database part, a table called User stores the common part of data of users’ information (There will be 2 extended classes of User called Student and Teacher). Another table called “StudentProfile” stores the properties of every student.

3.2.2 Database

User			
UserID	Username	Password	Related Student Profile

StudentProfile			
Profile ID	Student ID	Email	GPA

3.3 Course Subsystem

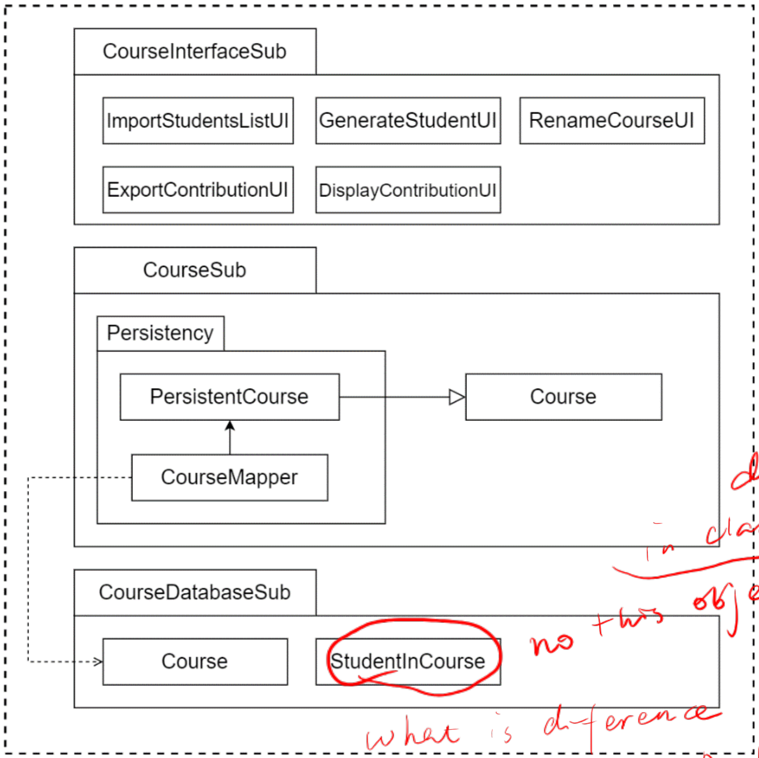


diagram in class

no this object

what is diference between student Profile and Student In Course

3.3.1 Description

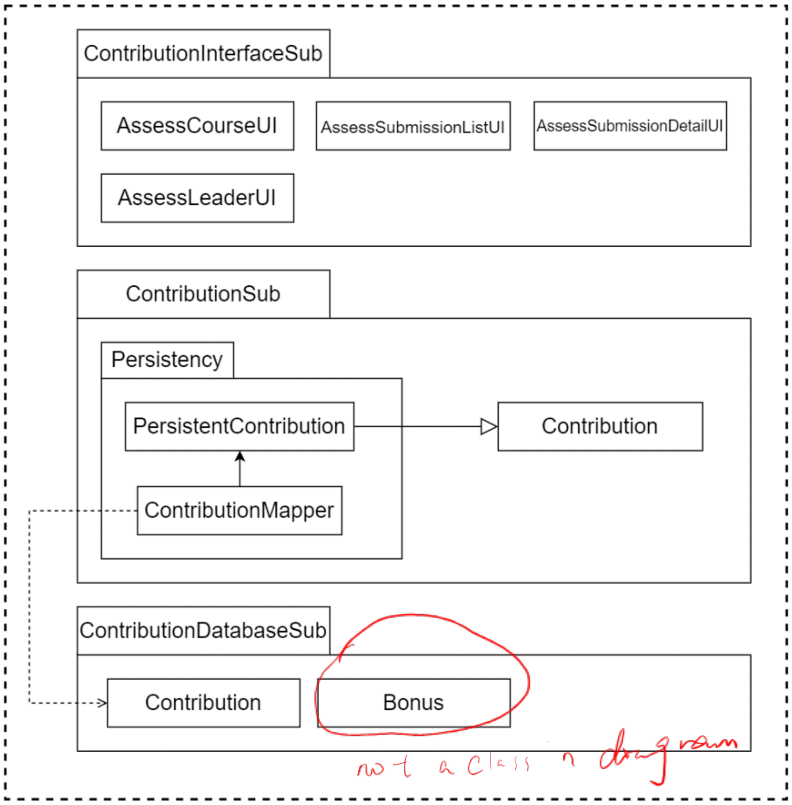
The subsystem contains 3 parts: Interface part, application logic, and database part. In interface part, “import students” interface enables user to import all the students in one course into the system for course registration. “Generate student” interface is used to import student individually. “Rename course” interface is used to rename the course name. In “Export contribution” interface, user can export the contribution records list as a file. As for the application logic, there is a class called Course. With Course, users can do the management of courses, students and also teams in the system. The data of Course class is persistent, storing in the database part.

3.3.2 Database

Course			
CourseID	CourseTitle	FormMethod	TeamMemberCountPrimary
TeamMemberCountSecondary	FloatingBand	Instructor	

StudentInCourse	
CourseID	Student User ID

3.4 Contribution Subsystem



not a class in diagram

3.4.1 Description

The subsystem contains 3 parts: interface part, application logic part and database part. The interface part of the system contains: Assess Course interface, Assess Submission List interface, Assess Submission Detail interface and Assess Leader interface. In Assess Course interface, the students can see all the courses which he/she is in. In Choose Submission interface the team leader can see all the submission items and choose to edit one of them. In the Assess Submission Detail interface, the leader can edit the level of the contribution for each member. In the Assess Leader interface, the members other than team leader can make assessment for the leader.
The data of Contribution class is persistent, which will be stored into the database.
The application logic part of the system contains "Contribution". With Contribution, the user of the system can know the contribution of each team member.
In the database part, a table called Contribution stores the data of all contribution's information. Another table "Bonus" stores all the bonus for the team leaders.

3.4.2 Database

From here, you can see Contribution as association

strange

Contribution			
ContributionID	Level	Related Submission	Related TeamMemeber

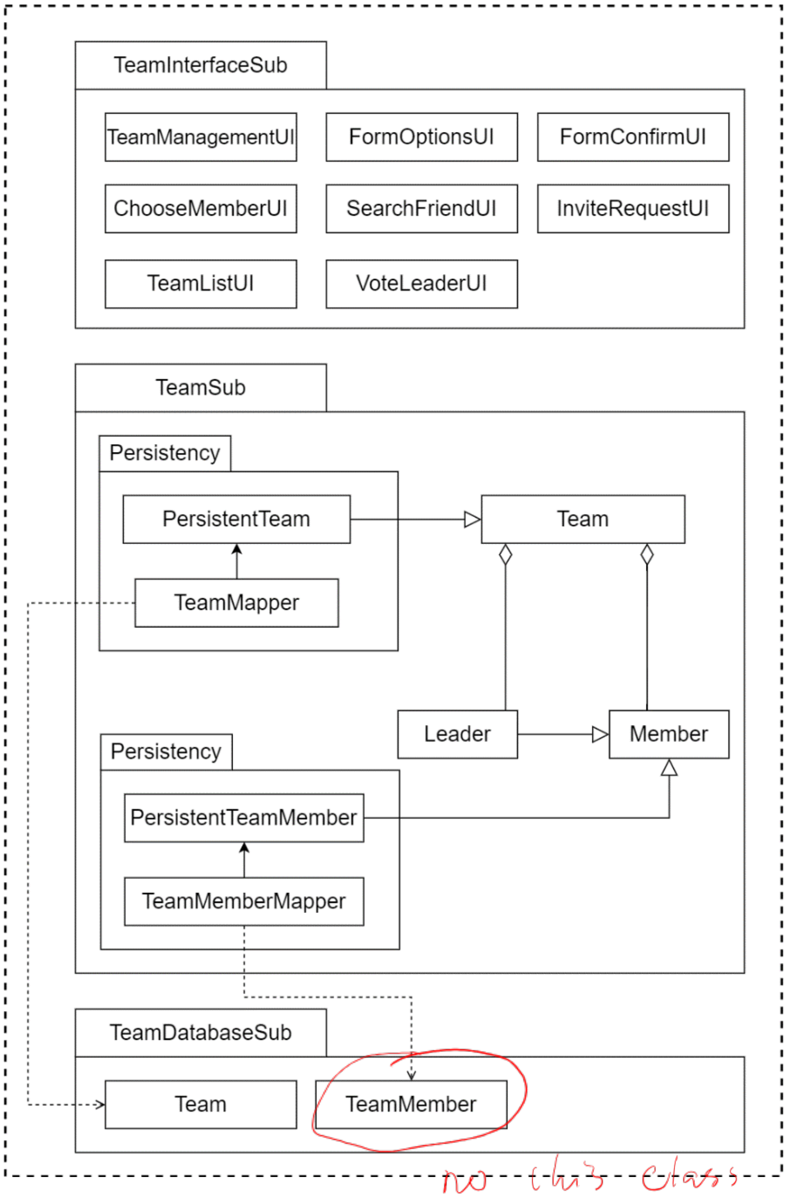
is

Good

Bonus			
BonusID	Related Team	Source Member	Level

strange

3.5 Team Subsystem



3.5.1 Description

The subsystem contains 3 parts: interface part, application logic part and database part. The interface part of the system contains: Team Management interface, Form options interface, Form Confirm interface, Choose Member interface, Search Friend interface, Invite request interface, Team List interface and Vote Leader interface. The teacher can see the team number of all the Courses in team management interface, he/she can confirm the formation of the team or change the Form team options. In the Form options interface, the teacher can choose the forming method of the team, according to the forming method, he/she can enter the primary number, secondary number and average GPA for the further setting. In the Form Confirm interface, the teacher can see all the formed teams, according to the forming method, the interface may show the students who are not in any team. In the Choose Member interface, the students can see all the course he/she is in, he/she can click one of the courses to choose team member. In the Search Friend interface, the student can search his/her friend by student ID and then invite them to his/her team. In the Invite request interface, the student can see all the requests he/she has received, which the student can accept or refuse. In the Team List interface, the student can see all the teams he/she has already in, by clicking one of the teams, the student can vote for a team leader for the team. In Vote team leader interface, the student can see all the team members and vote one of them to be the team leader.

The data of Team and Member class is persistent, which will be store into the database.

The application logic part of the system contains Team, Leader and Member. The team contains member and leader. With the team, the user of the system can know the members' and leaders' information in each team and do further operations.

In the database part, a table called Team stores the data of all the team's information. Another table "TeamMember" stores all the relations between students and teams, representing the "team member" relationship.

3.5.2 Database

Team			
Team ID	Related Course	Team Name	Team Leader

TeamMember		
Student User ID	Team ID	Is Leader

where to store submission item and percentage ?

4. Assessment

4.1 Stability

The system based on this architecture is stable. The whole system is divided by applying the most reasonable method for the subsystem differentiation, with minimizing the connection between the subsystem which will cross subsystem boundaries and implement the functional cohesion within each subsystem. Therefore, if some components are changed, we only need to modify the interface of very few objects.

4.2 Reusability

The components of the system can be easily extended with little change. Each of our five subsystems can independently implement their respective responsibilities while building data links with other individual subsystems. Such as: Submission subsystem achieve all add, delete or modify the submission of the work; All operations about team implementation are related to the Team subsystem; Contribution subsystem is a very independent subsystems, just as each member state of the submission of the provider. By comparison, Course subsystem and User subsystem seem to be slightly more associated with other subsystems, such as Course subsystem is responsible for all users who join in the course and generate student which as one kind of the user. However, this does not mean that User system must change its structures and functions provides with the change of Course subsystem. More importantly, these systems only need to extract data from other subsystems rather than the functions of other subsystems, which means that all the components of the subsystem are highly cohesive.

4.3 Scalability

This system has good scalability, because this system solves the complexity problem of the system. Throughout the system, each component has clear input and output requirement, thus it is easy to extend the architecture.

5. Alternative design (optional)

N/A

6. More considerations

In order to better understand and imagine details of our system design, it is recommended that readers can refer the User Interface part of our “ Software Requirements Specification for UIC Teamwork Contribution Assessment System v1.3” as a reference.

7. Appendix

N/A