Architecture Design

for

Teamwork System

Version 1.0 approved

Prepared by Tian Zhanming, Zhang Qiaomu, Zhang Haibin, Li Jiajia

Software Development Workshop III

2020/4/4

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Zhang Qiaomu, Tian Zhanming, Zhang Haibin, Li Jiajia	2020/03/08	Prepare initial version	1.0

1. Overview

1.1 Project description

Teamwork System is an application that helps teachers to calculate the final assessment according to students' contribution. Besides, its automatically grouping function assists teachers and students a lot. The application allows teachers to export students' contribution, generate student accounts, set team forming, modify submissions and check to form. And it allows students to a group, choose a teammate, teammate invitation, assess others and vote a leader.

1.2 References

"Eagle_SRS_20200404_v1.4" April 6,2020.

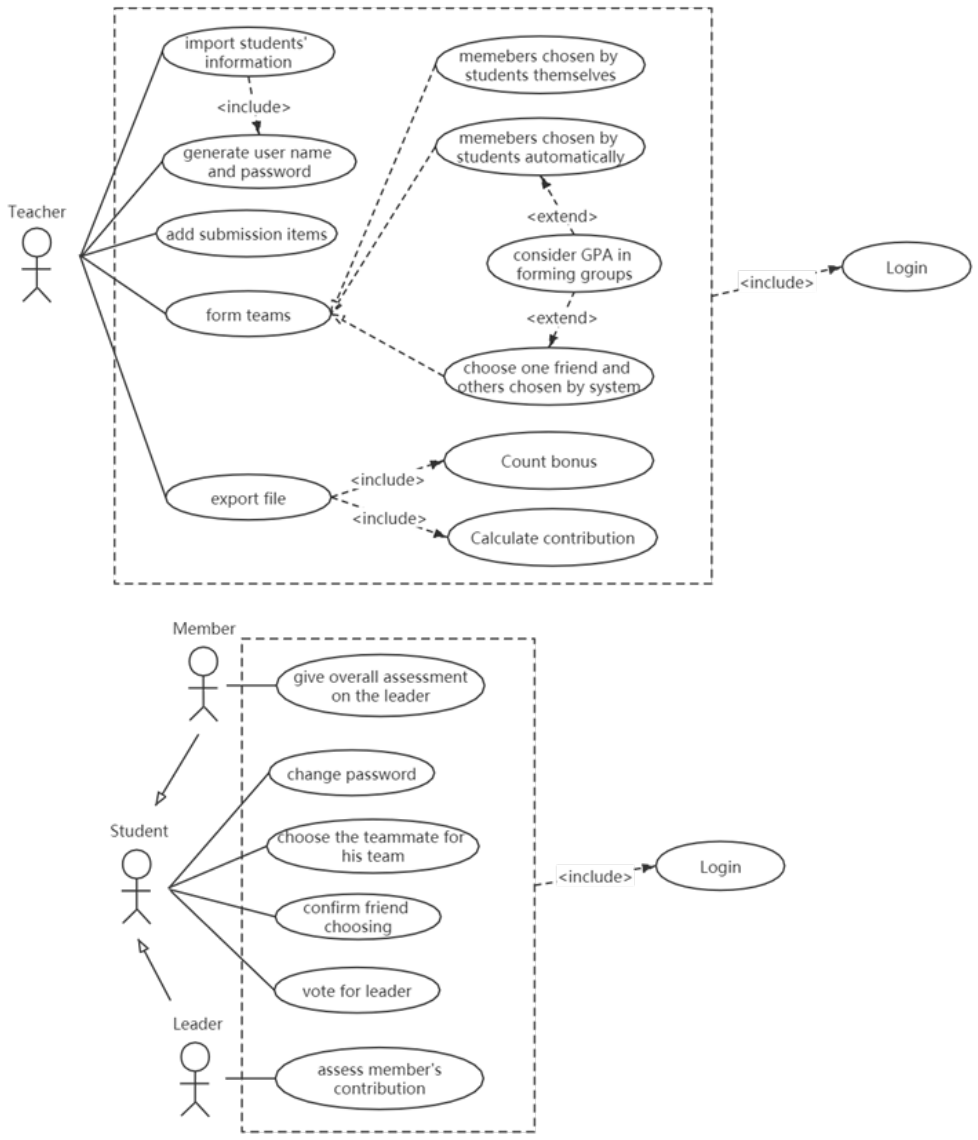
1.3 Design purpose

The purpose of this design is to divide the entire system into different subsystems in terms of use case. For each subsystem, Identify the persistent data, create corresponding interface subsystem and establish the database. In this way, the subsystems rise the cohesion and decrease the coupling to the system. The programmers in a team may have a more efficient way to cooperate and coordinate.

2. Overall description

2.1 Use case diagram and class diagram

Figure 1 Use case diagram


Teacher
import students' information
<include>
generate user name and password
add submission items
form teams
export file
memebers chosen by students themselves
memebers chosen by students automatically
<extend>
consider GPA in forming groups
<extend>
choose one friend and others chosen by system
Count bonus
<include>
<include>
Calculate contribution
<include>
Login
Member
give overall assessment on the leader
Student
change password
choose the teammate for his team
confirm friend choosing
vote for leader
Leader
assess member's contribution
<include>
Login

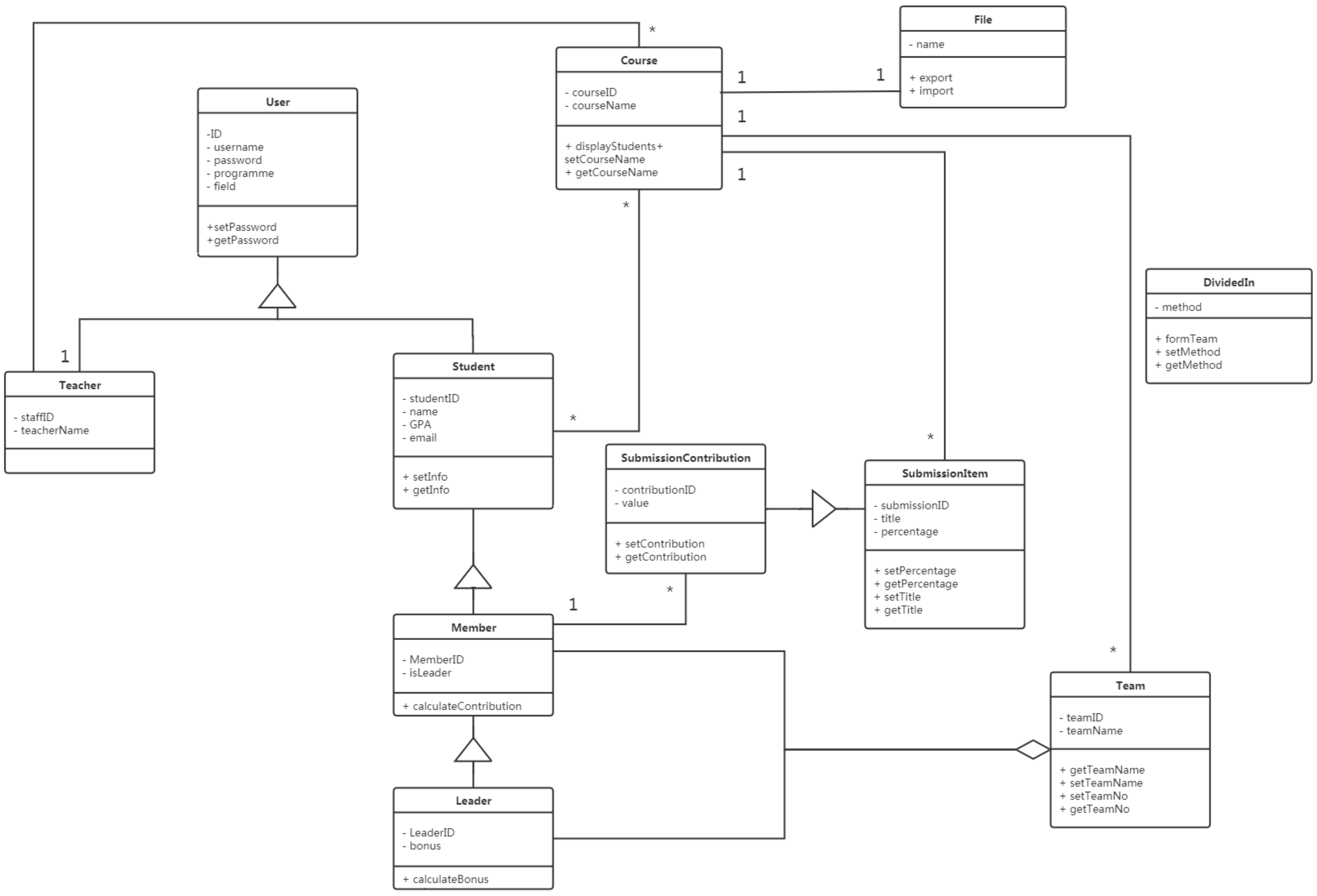


Figure 2 Class diagram

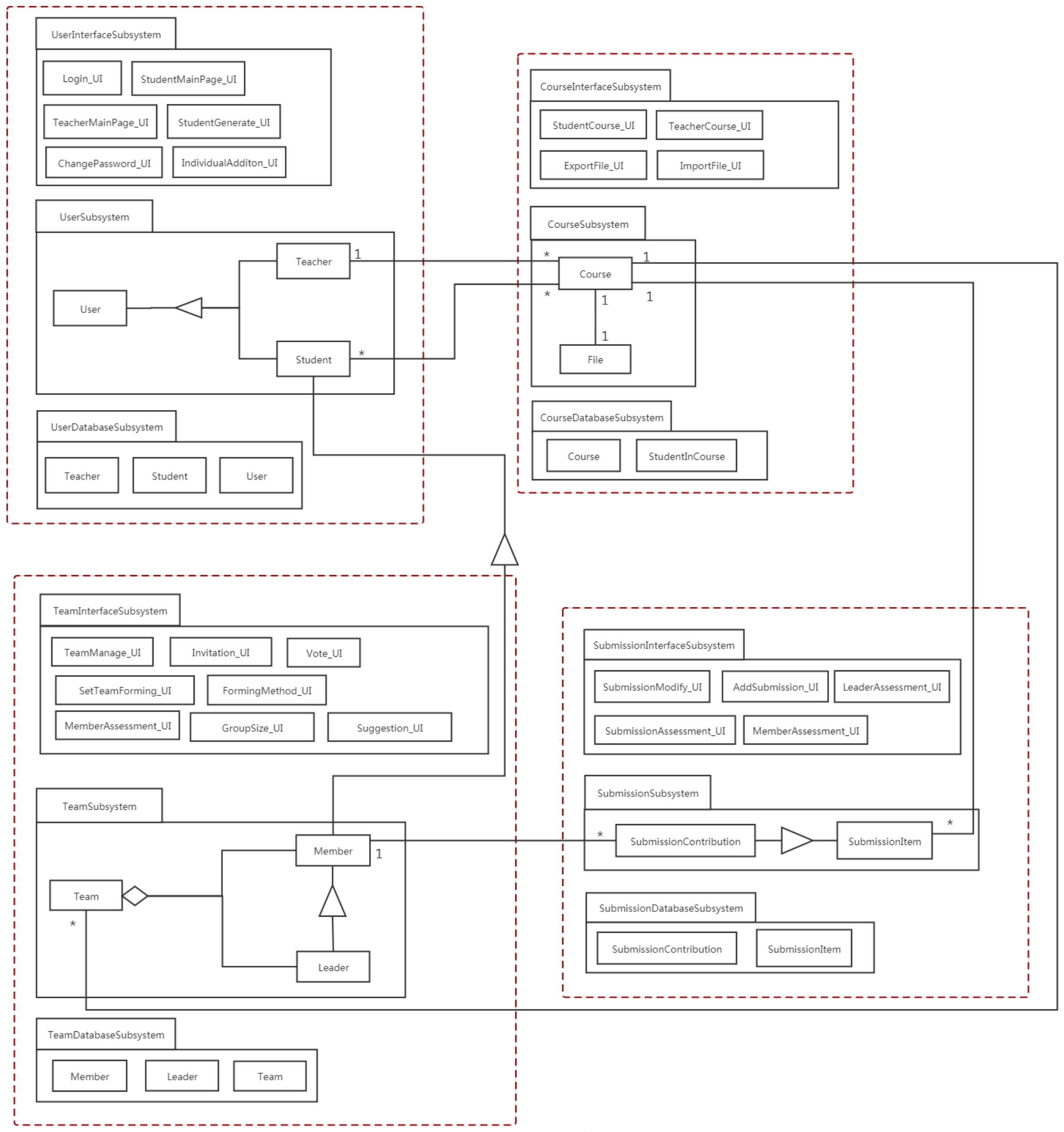
2.2 Design model

The system architecture model we used in this design is the MVT (Model-View-Template) model deriving from Django. The Model in MVT is for interaction with the database, and View is the core part aiming to receive request and data then return results, Template is for letting the result show on the browser. This model provides programmers with a clear separating division of work, which makes the coordination more efficacious.

2.3 System architecture

5

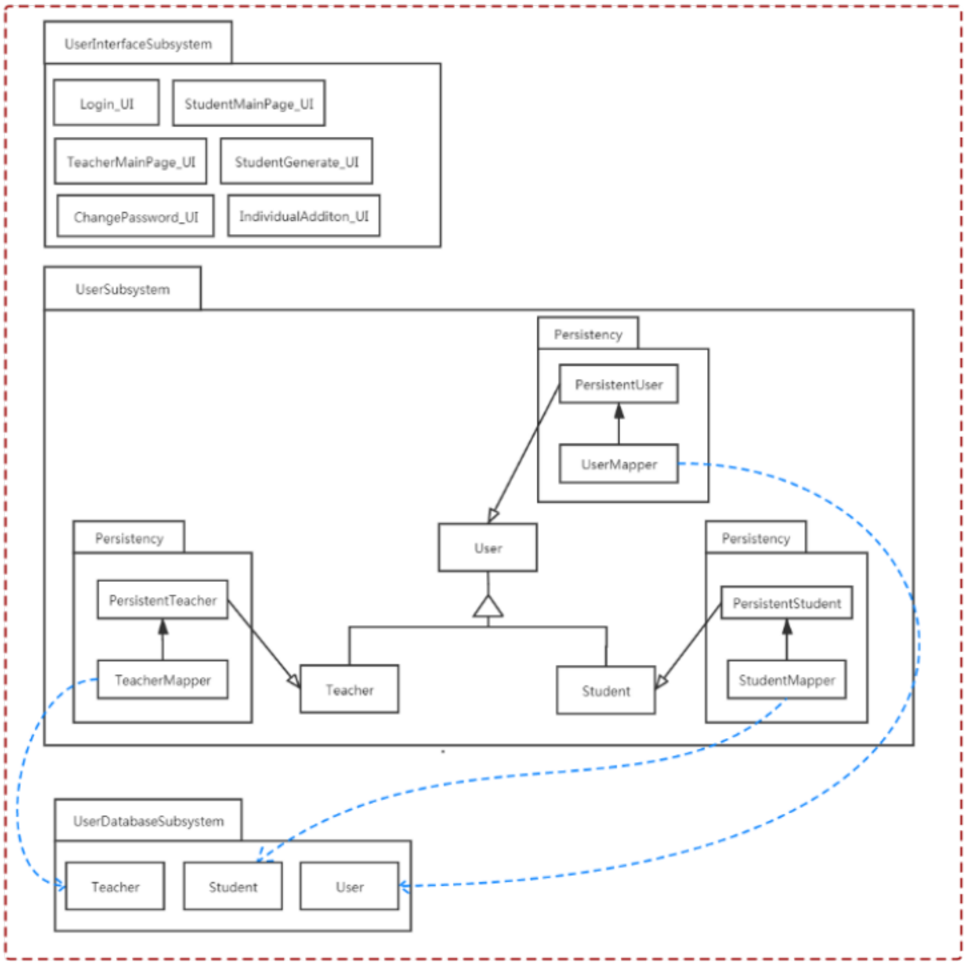
The system is divided into 4 subsystems: User Subsystem, Course Subsystem, Submission Subsystem/, Team Subsystem. The diagram shows the relationship between different subsystems.



In this architecture, it is unnecessary to put so detaild UI and database info

3. System architecture

3.1 User Subsystem



3.1.1 Description

For the User Subsystem, it consists of interface, control and database. In the control part, it can help teachers to generate students' name and passwords and provides changing password function for students.

3.1.2 Database

User				
ID	username	password	programme	field

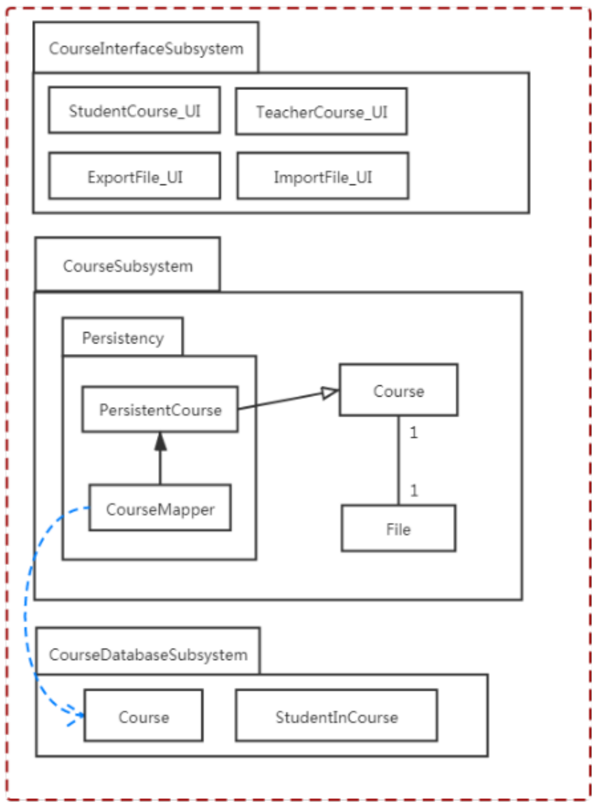
Student	
staffID	teacherName

Both teacher and User have inherited User info

Teacher			
studentID	studentName	GPA	email

So no need to save user here

3.2 Course Subsystem



3.2.1 Description

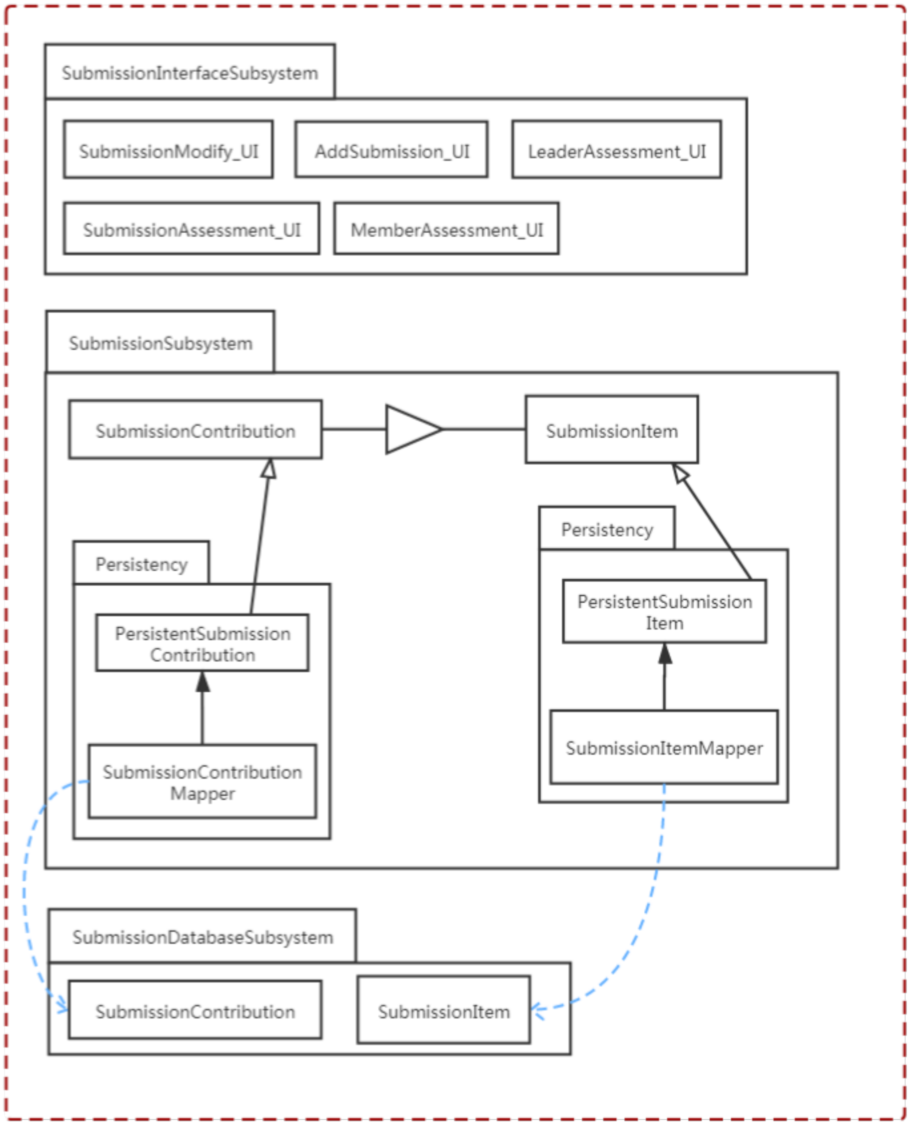
For the Course Subsystem, it consists of interface, control and database. In the main control part, it helps the teacher to add/delete submissions, assist teachers and students to do further operations in the course and help import/export files.

3.2.2 Database

Course			
courseID	courseName	formMethod	staffID

StudentInCourse	
courseID	studentID

3.3 Submission Subsystem



3.3.1 Description

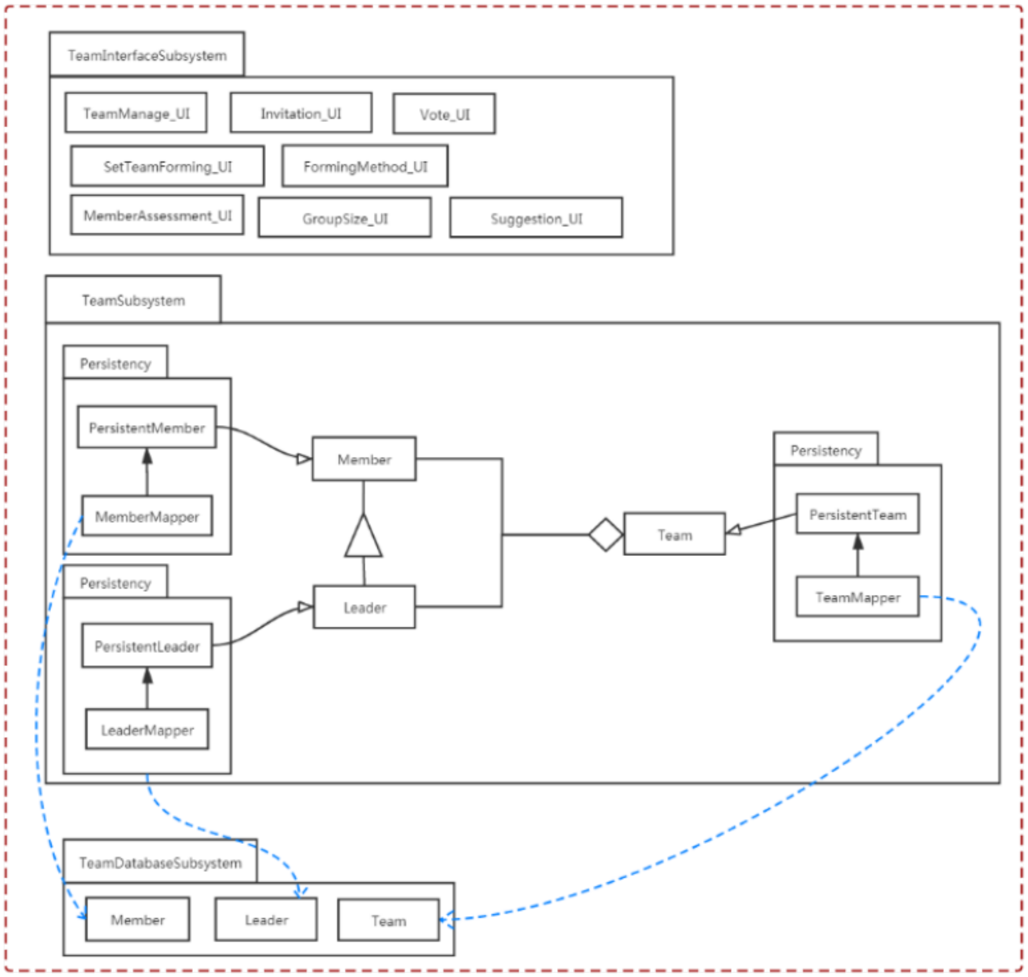
For the Submission Subsystem, it consists of interface, control and database. In the main control part, it helps teachers to add new submissions and provides assessment functions for team leaders and members.

3.3.2 Database

SubmissionItem			
submissionID	Title	Percentage	courseID

SubmissionContribution			
ContributionID	Value	SubmissionID	MemberID

3.4 Team Subsystem



3.4.1 Description

For the Team Subsystem, it consists of interface, control and database. In the main control part, it allows teachers to choose a team forming method, helps divided students into teams, provides friend choosing function before team forming finish and vote and assessment functions after team forming for students.

3.4.2 Database

where is member's contribution

Member		
MemberID	TeamID	isLeader

what is difference between memberID, LeadID, studentID

Leader		
LeaderID	TeanID	bonus

Team		
teamID	TeamName	CourseID

Member and leader Inherits students Should have student info otherwise

4. Assessment

4.1 Stability

This system based on the architecture is stable, since the communication between two subsystems via parameter passing, which has been highly reduced to a small number. All the design is based on the model, so if some changes are made to specific components, only the change parts will be influenced. Because each component within a subsystem works mutual independence.

4.2 Reusability

This system can be reused in other systems without change. Since Django framework in python can let programmer decouple the various components of an application to create different subsystem and template elements. Those separate applications can be reused and be immediately installed by other programmers. Additionally, for subsystems in our design, the subsystems can be reused to create an additional function without changes in itself.

4.3 Scalability

This system is easy to extend. Because in Django MVT model: View increases application flexibility and configurability. View can be used to create applications and readily to be extended to more functions. Besides, programmers can use the model for manipulation on the database according to the user's needs and then create the appropriate Template subsystems to display the processing results to the user.

5. Alternative design (optional)

None.

6. More considerations

For more details, please refer to our document “Eagle_SRS_20200404_v1.4”.

7. Appendix

None.