Team Information

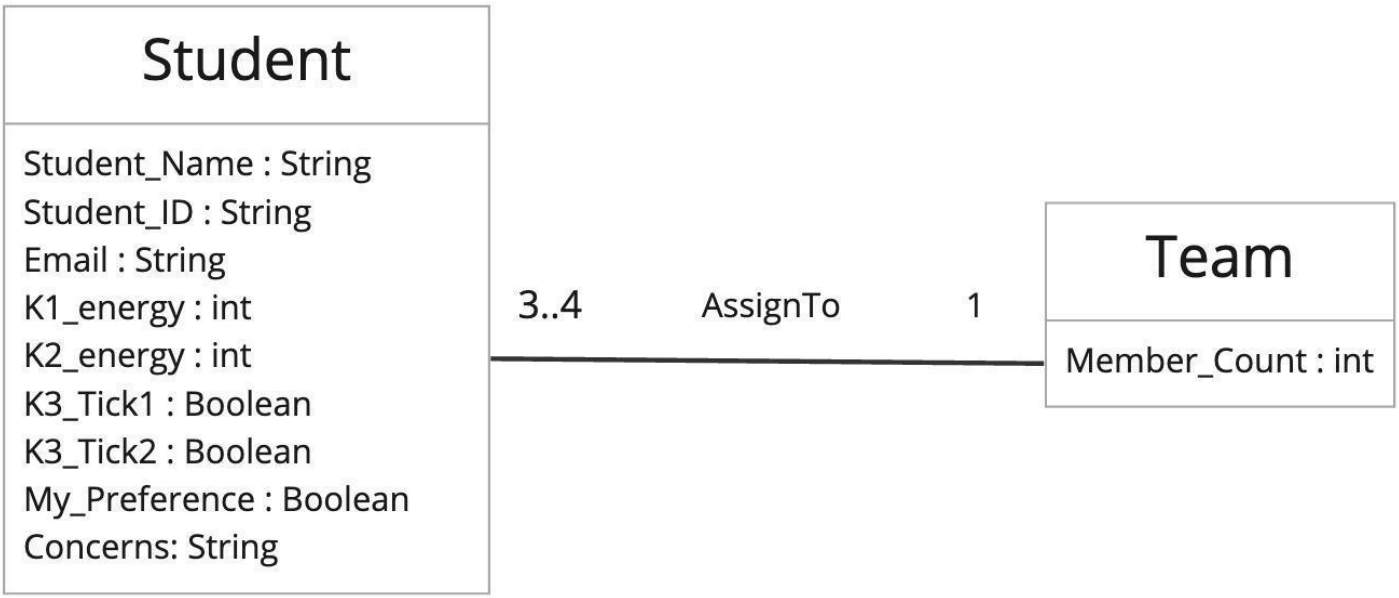
Team ID	**Group 15**
Team Repo on GitHub	**https://github.com/a1347539/comp3111Project**

Name (Member 1)	**Li Chun Tak**
GitHub ID	**a1347539**
Email ID	**1347539@gmail.com**
Dev Branch	**dev/input**
Task Assignment	**INPUT**

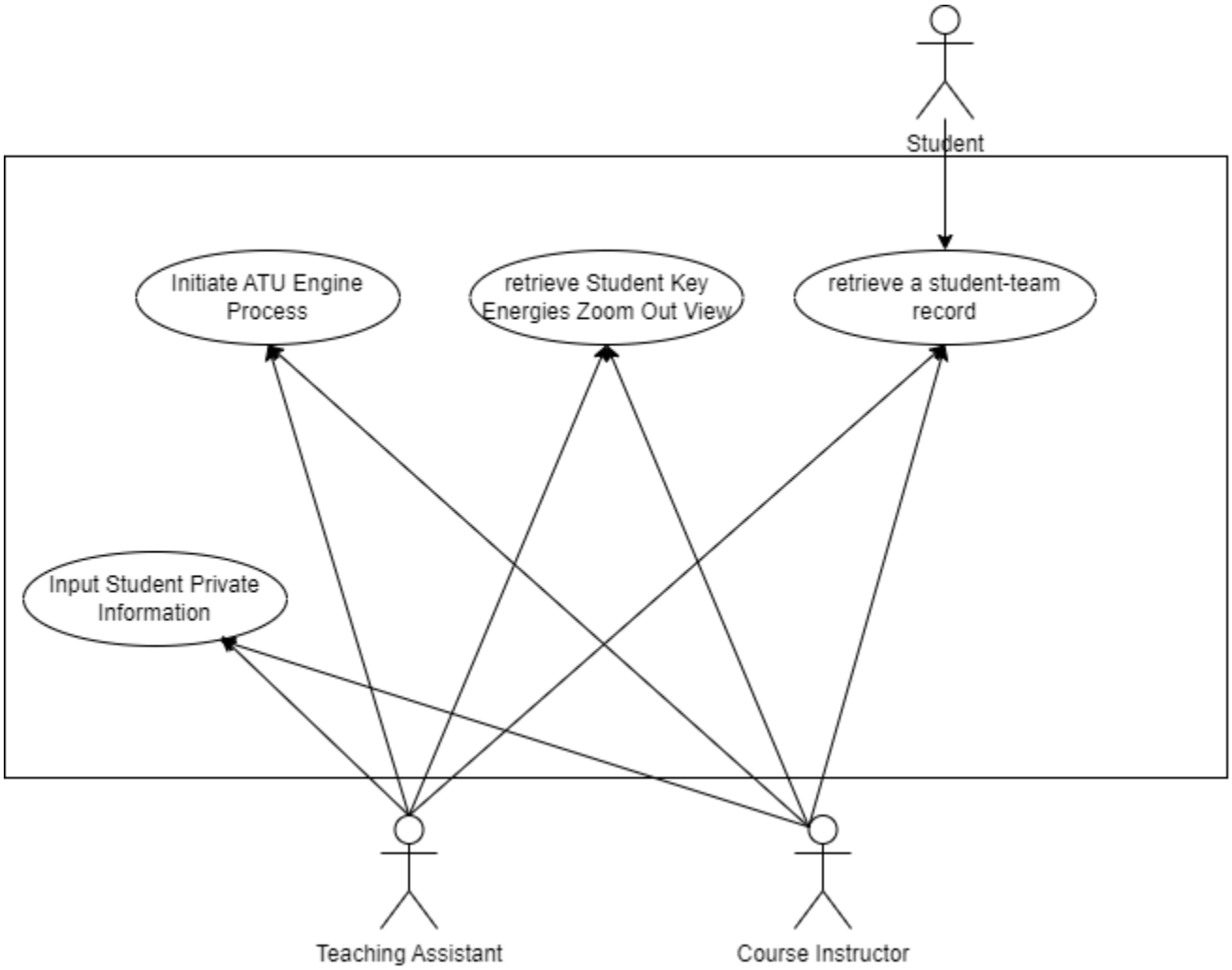
Name (Member 2)	**HE Qihao**
GitHub ID	**phyqh**
Email ID	**qheag@connect.ust.hk**
Dev Branch	**dev/process**
Task Assignment	**PROCESS**

Name (Member 3)	**SZE Wing Kwan**
GitHub ID	**szeu**
Email ID	**wkszeaa@connect.ust.hk**
Dev Branch ID	**dev/output**
Task Assignment	**OUTPUT**

Class Diagram (for overall system)



Use Case Diagram (for overall system)



Use Case Specification: Task (INPUT or PROCESS or OUTPUT)

INPUT

Use-case	Input personal information
Brief description	The TeachingTeam feeds the csv. to the system which contains all the student's information including Student_ID, Student_Name, email, K1_Energy, K2_Energy, K3_Tick1, K3_Tick2, My_Preferene, and Concerns.
Participating actors	Teaching Assistant
Preconditions	The data in csv. has all the student information. The input data is in the correct format.
Flow of events	Basic flow: 1. The use-case begins with the TA getting onto the system. {Input directory} 2. The user input a specific directory that store the csv. file. 3. If the UPLOAD activity is selected. {Find csv. file in user specified directory} 4. The system loads the csv. file put in the specific directory. 5. The system receives the student information 6. The system represents each student information in a data structure and put all the processed information in a container. 7. The system response with a pop-up window showing all the students. information stored in the system, count of the number of student and K3_ticks, and min/max/mean of overall K1 and K2 Energy. 8. The use-case ends. Alternative flow: A1. Incorrect directory At {Find csv. file in user specified directory} if there does not exist a csv. file. 1. The system displays error message to tell the user type again. 2. The flow of event is resumed at {Input directory}. A2. Mutliple files detected At {Find csv. file in user specified directory} if there exists multiple csv. file. 1. The system displays error message to tell the user type again. 2. The flow of event is resumed at {Input directory}.
Postconditions	The correct csv. file is uploaded and each of the student information is correctly stored in an internal data structure.

PROCESS

Use-case	Initiate the processing of ATU Engine
Brief description	This use case describes the system interaction when the Course Instructor or Teaching Assistant(s) start the processing of the ATU Engine
Participating actors	Course Instructor, Teaching Assistant
Preconditions	The system has accurate Student Information prepared and ready to be retrieved by the ATU Engine.
Basic Flow	1. The use case begins when the Instructor or the TA clicks a START button that initiates ATU Engine. 2. The ATU Engine retrieves and processes the Student Information to form teams. 3. The ATU Engine stores team-up results in a dataset with records of team id related to student id. 4. The use case ends.
Postcondition	All students have been assigned a team, where the team information is stored in a new dataset with record of Team and Student relationship. Each Team will have at least 3 at most 4 students. Each Team will have at least one student whose K1_energy >= average_k1_energy. K1_Energy & K2_Energy for each team is balanced.

OUTPUT

OUTPUT Use Case Specification 1

Use-case	Check Teaming Up Result (individual student)
Brief description	This use case describes the system interaction when a user check the teaming up result using a student ID or student name
Participating actors	Course Instructor, Teaching Assistant, Student
Preconditions	The system has finished processing all information and generated teaming up result.
Flow of events	Basic flow: 1. The use case begins when Check Teaming Up Result button is pressed. 2. The system displays the interface with text entries for inputing student name or student id. {Enter Student Information} 3. The user enter student name or student id in the corresponding text field {Input validation} 4. The system check the validation of the input. 5. The system displays the corresponding team information 6. The use case ends. Alternative flow: A1. Invalid Input At {Input validation} if the input is invalid, meaning that the gave student ID or the student name does not exist in the system. 1. The system displays error message to tell the user type again 2. The flow of event is resumed at {EnterStudent Information}
Postconditions	The resulting team contains the student with the given ID or name.

OUTPUT Use Case Specification 2

Use-case	Check Teaming Up Result (Chart result)
Brief description	This use case describes the system interaction when the Course Instructor or Teaching Assistant(s) check student key energies displayed on a 2D-Line Chart.
Participating actors	Course Instructor, Teaching Assistant
Preconditions	The system has all students' records (with valid energies) stored.
Flow of events	Basic flow: 1.The use case begins when the Instructor or the TA actor selects the Check Chart Result acticity. 2.The system perform subflow *Display Chart Result.* 3.The use case ends. Subflow: S1: Display Chart Result 1. The system retrieves each student's record and get their K1 and K2 energy 2. The system generates the Student Key Energies Zoom Out View Chart. 3. The system displays the chart in a pop-up window

Task Allocation:

Name:	Task ID:	Task Description:
Li Chun Tak	113	Set up Team Repo on GitHub
Li Chun Tak	121	Draw a class diagram
HE QiHao	122	Draw the use case diagram
Li Chun Tak	123-1	Use case specification - Input
HE QiHao	123-2	Use case specification - Process
SZE Wing Kwan	123-3	Use case specification - Output
SZE Wing Kwan	211-1	Meeting Minutes - 1
Li Chun Tak	211-2	Meeting Minutes - 2
HE QiHao	211-3	Meeting Minutes - 3
Li Chun Tak	212	Draw Gantt chart
Li Chun Tak	213	Draw Burndown chart
Li Chun Tak	214-1	Write at least one meaningful Git commit log - 1
HE QiHao	214-2	Write at least one meaningful Git commit log - 2
SZE Wing Kwan	214-3	Write at least one meaningful Git commit log - 3
HE QiHao	221	Unit testing report

HE QiHao	**222**	**Coverage report**
SZE Wing Kwan	**223**	**Documentation with JavaDoc**
Li Chun Tak	**231A**	**Task INPUT**
HE QiHao	**231B**	**Task PROCESS**
SZE Wing Kwan	**231C**	**Task OUTPUT**
HE QiHao	**233-1**	**Perform unit testing**
SZE Wing Kwan	**233-2**	**Integrate all tasks**
Li Chun Tak	**233-3**	**Perform integration testing**