

# **COMP 3111, 3111H - Software Engineering**

Fall 2022/2023

Leader:

Li Chun Tak

Members:

He Qihao

Sze Wing Kwan

# **Content**

<b>Content</b>	2
<b>Team Information</b>	3
<b>Part 1 - Documentation: Project Management</b>	4 - 12
211 Meeting Minutes	4 - 9
212 Gantt Chart	10
213 Burndown Chart	11
214 Git commit log on GitHub	12
<b>Part 2 - Documentation: Implementation &amp; Testing</b>	13 - 69
221 Report on the unit testing	13 - 19
222 Report on the coverage test	20
223 JavaDoc	21 - 69
<b>Part 3 - Program Execution and Screenshots</b>	70 - 78
3.1 Task A	70 - 74
3.2 Task C	75 - 78
<b>Part 4 - Modified Use Case Specification</b>	79 - 80

## Team Information

Team ID	15
Team Repo on GitHub	<a href="https://github.com/a1347539/comp3111-G15">https://github.com/a1347539/comp3111-G15</a>

Name (Member 1)	Li Chun Tak
GitHub ID	a1347539
Email ID	ctliaj@connect.ust.hk
Dev Branch	input
Task Assignment	Input

Name (Member 2)	He Qihao
GitHub ID	phyqh
Email ID	qheag@connect.ust.hk
Dev Branch	process
Task Assignment	Process

Name (Member 3)	Sze Wing Kwan
GitHub ID	szeu
Email ID	wkszeaa@connect.ust.hk
Dev Branch ID	output
Task Assignment	Output

Submitted by: \_\_\_\_\_ Li Chun Tak \_\_\_\_\_

Date of Submission: \_\_\_\_\_ 22/11/2022 \_\_\_\_\_

# PART 1: Documentation – Project Management

## 1.1. Team Meeting Minutes

### COMP 3111: Introduction to Software Engineering

#### Minutes of the 1st Project Scrum Meeting

ATU

**Date:** October 11, 2022

**Time:** 7:00 pm

**Place:** Zoom

**Attending:** Li Chun Tak, He Qihao, Sze Wing Kwan

**Absent:** NA

**Recorder:** Sze Wing Kwan

#### 1. Report on progress during the past week

Name	Tasks worked on in the past week	Total hours
Li Chun Tak	Set up Git repo and understand project requirement	0.5
He Qihao	Understand project requirement	0.5
Sze Wing Kwan	Understand project requirement	0.5

#### 2. Discussion of impediments and resolution

2.1. Li Chun Tak led the discussion of the project division.

2.2. Li Chun Tak pointed out that some features can only be accessed by the teaching team but not students.

2.3. Sze Wing Kwan suggested that the system can be divided into two windows with all restricted buttons only visible to the teaching team.

2.4. He Qihao suggested that a password would be required to identify the user.

2.5. He Qihao raised a question about the security of viewing the teaming up results and the criteria for selecting a recommended leader.

2.6. Sze Wing Kwan proposed that all students can search for another's teaming up results so students can swap with others with similar key energies easily.

2.7. Li Chun Tak suggested that if no one in the team indicated their preference to be the team leader, the one with highest K1 will be recommended to be the leader.

2.8. All group members reached a consensus on the system's overall UI design, and how TA and students will use it differently.

### **3. Goals for the coming week**

Name	<b>Tasks worked on in the past week</b>
Li Chun Tak	Draw class diagram, use case diagram and use case specification for Input
He Qihao	Draw class diagram, use case diagram and use case specification for Process
Sze Wing Kwan	Draw class diagram, use case diagram and use case specification for Output

### **4. Meeting adjournment and next meeting**

The meeting was adjourned at 10:00 p.m. The next project meeting will take place on next Sunday 11/6 at 2:00 p.m. on zoom.

# **COMP 3111: Introduction to Software Engineering**

## Minutes of the 2nd Project Scrum Meeting

**ATU**

**Date:** November 6, 2022

**Time:** 2:00 pm

**Place:** Zoom

**Attending:** Li Chun Tak, He Qihao, Sze Wing Kwan

**Absent:** NA

**Recorder:** Sze Wing Kwan

### **1. Report on progress during the past week**

Name	Tasks worked on in the past week	Total hours
Li Chun Tak	Created file for Gantt and Burndown charts and implemented UI code for Input	3
He Qihao	Finalized system UI design and implemented UI code for Process	1.5
Sze Wing Kwan	Implemented program code for necessary classes and UI code for Output	1.5

### **2. Discussion of impediments and resolution**

- 2.1. Sze Wing Kwan demonstrated the current implementation of classes and was agreed by all group mates.
- 2.2. He Qihao pointed out the problem in accessing the TA view due to password validation errors.
- 2.3. Sze Wing Kwan pointed out the data cumulation problem when the CSV file is loaded to the system more than one time.
- 2.4. Li Chun Tak addressed the question about the password validation problem and changed the input program code by clearing old records once a new CSV file is loaded.
- 2.5. Li Chun Tak asked about the output result under the situation when a duplicated student name is used for searching.
- 2.6. Sze Wing Kwan replied to the duplicated name problem that an alert to tell users to use student ID instead will pop out.

### **3. Goals for the coming week**

Name	Tasks worked on in the past week

Li Chun Tak	Finish the implementation of Input
He Qihao	Finish the implementation of Process
Sze Wing Kwan	Finish the implementation of Output

#### **4. Meeting adjournment and next meeting**

The meeting was adjourned at 4:00 p.m. The next project meeting will take place on next Tuesday 11/15 at 3:00 p.m.

# **COMP 3111: Introduction to Software Engineering**

## Minutes of the 3rd Project Scrum Meeting

**ATU**

**Date:** November 15, 2022

**Time:** 3:00 pm

**Place:** Library LG1

**Attending:** Li Chun Tak, He Qihao, Sze Wing Kwan

**Absent:** NA

**Recorder:** Sze Wing Kwan

### **1. Report on progress during the past week**

Name	Tasks worked on in the past week	Total hours
Li Chun Tak	Reviewed the application cycle and wrote test cases for unit testing	0.75
He Qihao	Wrote test cases for unit testing	0.5
Sze Wing Kwan	Finished merging three tasks and wrote comments for Javadoc	1

### **2. Discussion of impediments and resolution**

- 2.1. Sze Wing Kwan demonstrated the complete UI and functionality to all groupmates.
- 2.2. Li Chun Tak and He Qihao reported the application's potential bugs, and the responsible groupmates of each bug solved them during the meeting.
- 2.3. He Qihao discovered that some written test cases overlapped the same method and asked for a clearer division.
- 2.4. Sze Wing Kwan asked for clarification on some methods for writing comments.
- 2.5. Li Chun Tak scheduled an internal deadline to complete all documentation of the project.

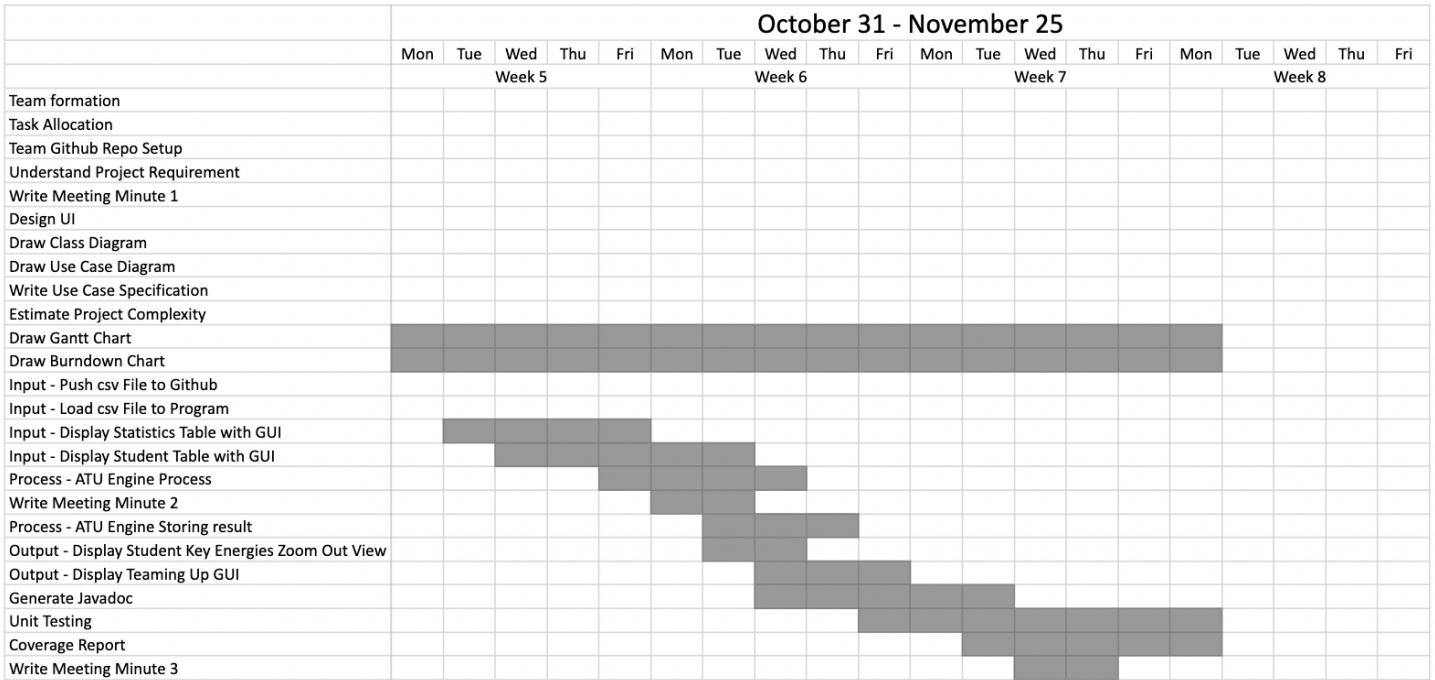
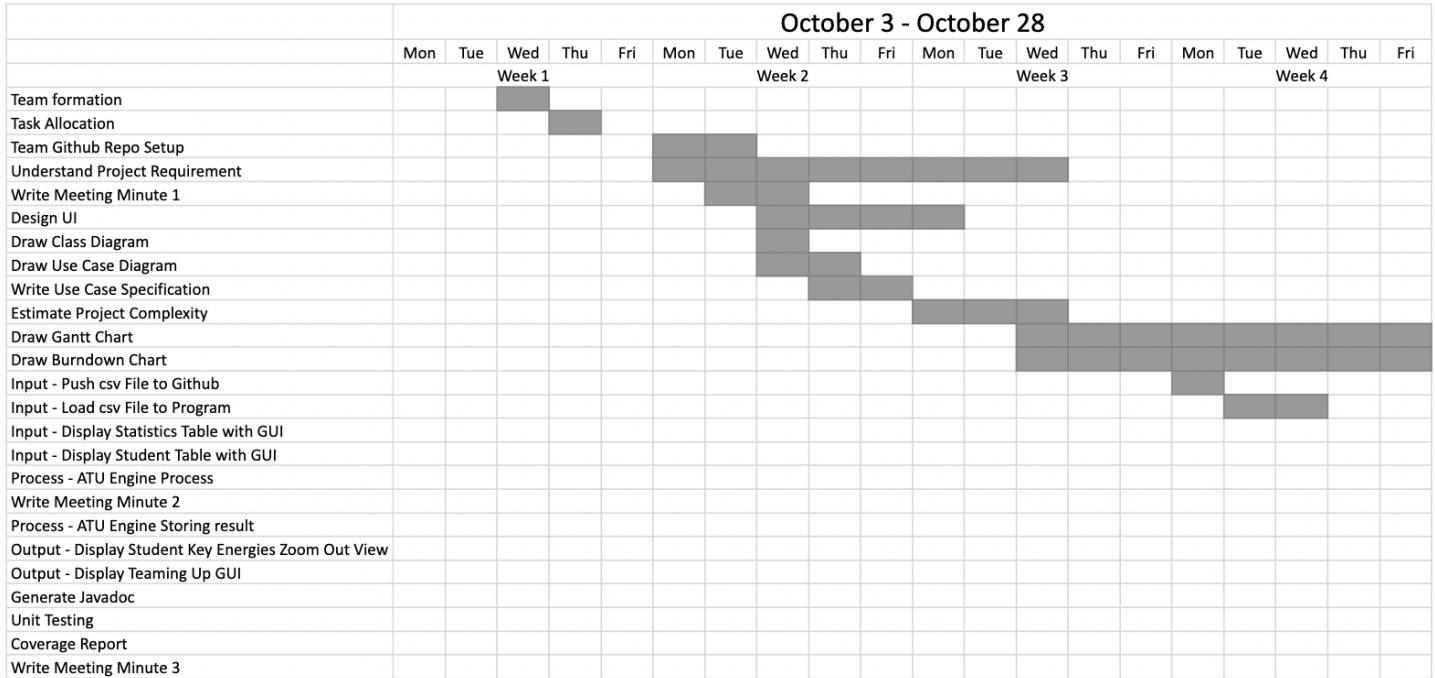
### **3. Goals for the coming week**

Name	Tasks worked on in the past week
Li Chun Tak	Integrate everything into one pdf file
HE Qihao	Generate unit testing and coverage report
SZE Wing Kwan	Generate Javadoc documentation

#### **4. Meeting adjournment and next meeting**

The meeting was adjourned at 5:00 p.m. If there is any major update by a groupmate, the next project meeting will take place on the coming Sunday 11/19 at 2:00 p.m. on zoom.

## 1.2. Gantt Chart



### 1.3. Burndown Chart

Task	Initial Estimate	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Hours Left
Documentation	6	0.5	2.5	0.5	0.25	1.5	0.5	2.5	0.25	-2.5
UI Design	2	0	2	1	0	0	0	0	0	-1
Input - UI	4	0	0	0	2	1.5	0	0	0	0.5
Input - File Reading	1	0	0	0	1	0.25	0	0	0	-0.25
Input - Statistics Table	1	0	0	0	0.5	0.5	0	0	0	0
Input - Student Table	2	0	0	0	0.25	1	0.5	0	0	0.25
Process - UI	1	0	0	0	0.5	0	0	0	0	0.5
Process - Data Processing	3	0	0	0	0	0.5	2.5	0	0	0
Process - Data Storing	2	0	0	0	0	0	2	0	0	0
Output - UI	3	0	0	0	1	1	1	0	0	0
Output - Energies Chart	1	0	0	0	0	0	1	0	0	0
Output - Team Result Table	2	0	0	0	0	0	1	1	0	0
Merging	3	0	0	0	0	0	0	1	0	2
Testing	3	0	0	0	0	0.5	1	1	0.5	0
Setting	Start	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	
Planned Hours		4.25	4.25	4.25	4.25	4.25	4.25	4.25	4.25	4.25
Actual Hours		0.5	4.5	1.5	5.5	6.75	9.5	5.5	0.75	
Remaining Effort	34	33.5	29	27.5	22	15.25	5.75	0.25	-0.5	
Ideal Burndown	34	29.75	25.5	21.25	17	12.75	8.5	4.25	0	



## 1.4. Representative Git commit log on GitHub

<https://github.com/a1347539/comp3111-G15/commit/fce5bc1eab2bdb93036b642c936bec92a2d64dc>

Merge pull request #30 from a1347539/process  
Process merged to main

main (#30)  
phyqh committed 6 days ago Verified

2 parents 5f1bf8b + aa3df2e commit fce5bc1eab2bdb93036b642c936bec92a2d64dc

[Browse files](#)

<https://github.com/a1347539/comp3111-G15/commit/9f18db3c80280b17de1ac848c012aeb7b696fc97>

Merge pull request #24 from a1347539/input  
Student Table UI layout and will load data successfully

main (#24)  
a1347539 committed 16 days ago Verified

2 parents 64f35ce + 9606da9 commit 9f18db3c80280b17de1ac848c012aeb7b696fc97

[Browse files](#)

<https://github.com/a1347539/comp3111-G15/commit/165edc8ff12735ba51512562b5a718c7ec4b1880>

Merge pull request #26 from a1347539/output  
Finish output teaming up result table set up

main (#26)  
szeu committed 16 days ago Verified

2 parents d34b970 + 8ff9ce2 commit 165edc8ff12735ba51512562b5a718c7ec4b1880

[Browse files](#)

Commits on Nov 14, 2022

- ATU Engine  
phyqh committed 8 days ago
- Merge pull request #28 from a1347539/main ...  
phyqh committed 8 days ago
- added some javadoc  
a1347539 committed 8 days ago
- Merge branch 'main' of https://github.com/a1347539/comp3111-G15  
a1347539 committed 8 days ago
- options for user level instead of password right away  
a1347539 committed 8 days ago
- minor bug fixes  
a1347539 committed 8 days ago

Commits on Nov 7, 2022

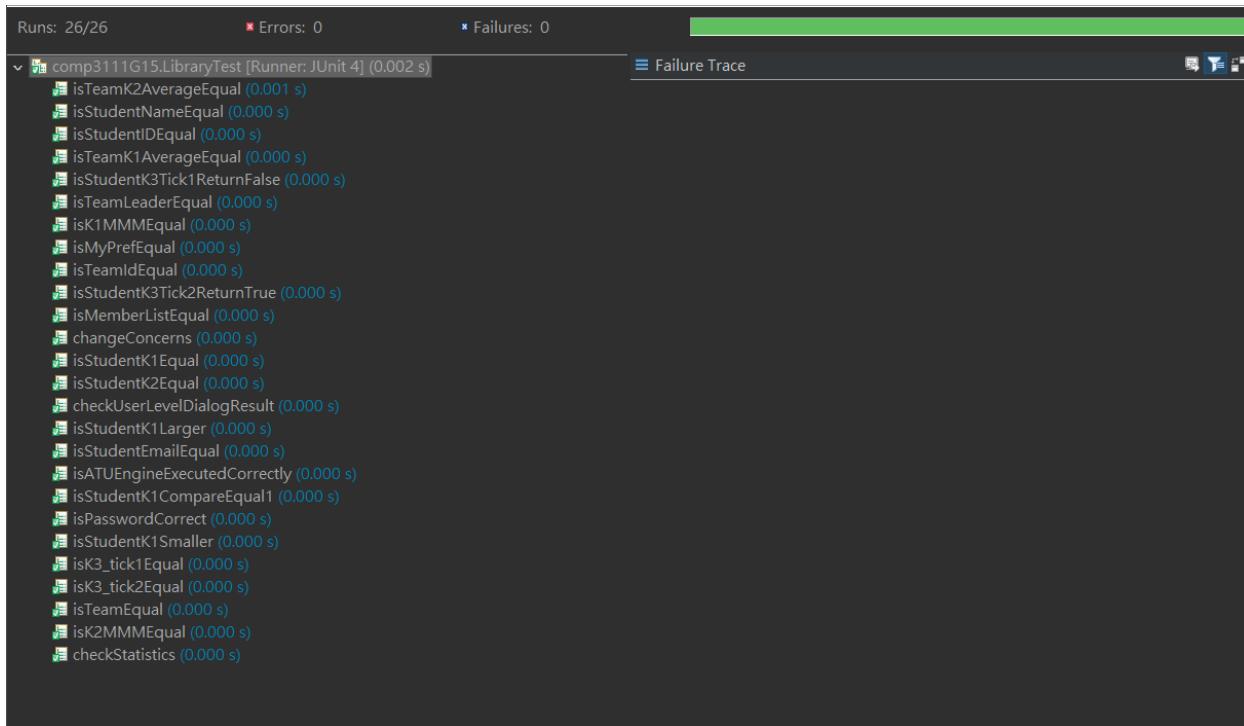
- Merge pull request #27 from a1347539/output ...  
szeu committed 15 days ago
- Remove quotation marks  
szeu committed 15 days ago

Commits on Nov 6, 2022

- Merge pull request #26 from a1347539/output ...  
szeu committed 16 days ago
- Finish output teaming up result table set up  
szeu committed 16 days ago
- Merge pull request #25 from a1347539/output ...  
szeu committed 16 days ago
- Show treetableview in display window  
szeu committed 16 days ago

## PART 2: Documentation – Implementation and Testing

### 2.1. Report on the unit testing for the implemented tasks (100% pass)



### Java Code:

```
/*
 * This Java source file was generated by the Gradle 'init' task.
 */
package comp3111G15;

import static org.junit.Assert.*;
import java.util.ArrayList;
import org.junit.Before;
import org.junit.Test;

/**
 *
 * The LibraryTest is used for unit testing
 * @author SzeWingKwan
 *
 */
public class LibraryTest {

    ArrayList<Student> student_data;
    Student student1;
    Student student2;
```

```

Student student3;
Student student4;
Student student5;
Student student6;
Student student7;
Student student8;
Team team;
ArrayList<Student> ATUStudent;
Team ATUTeam1;
Team ATUTeam2;
ATUEngine ATUEngine;
Student studentChangeable;

/**
 * Set up control variables
 * @throws Exception exception
 */
@Before
public void setUp() throws Exception {
    student_data = new ArrayList<Student>();
    ATUStudent = new ArrayList<Student>();

    student1 = new Student(0, "20004488", "SAFFRON", "Corgipoo", "CorgipooSAF@connect.ust.hk",
                          "26", "80", "0", "1", "0", "");
    student2 = new Student(1, "20023331", "HYSSOP", "Chamois", "ChamoisHYS@connect.ust.hk",
                          "27", "85", "0", "1", "1", "");
    student3 = new Student(2, "20067232", "CHRYSANTHEMUM", "Abelisaurus", "AbelisaurusCHR@connect.ust.hk",
                          "57", "60", "1", "0", "1", "");

    student4 = new Student(3, "2", "CHRYSANTHEMUM", "Abelisaurus", "AbelisaurusCHR@connect.ust.hk",
                          "1", "1", "1", "0", "1", "");
    student5 = new Student(4, "3", "CHRYSANTHEMUM", "Abelisaurus", "AbelisaurusCHR@connect.ust.hk",
                          "1", "1", "1", "0", "1", "");
    student6 = new Student(5, "4", "CHRYSANTHEMUM", "Abelisaurus", "AbelisaurusCHR@connect.ust.hk",
                          "1", "1", "1", "0", "1", "");
    student7 = new Student(6, "5", "CHRYSANTHEMUM", "Abelisaurus", "AbelisaurusCHR@connect.ust.hk",
                          "1", "1", "1", "0", "1", "");
    student8 = new Student(7, "5", "CHRYSANTHEMUM", "Abelisaurus", "AbelisaurusCHR@connect.ust.hk",
                          "14", "1", "1", "0", "1", "I hate freeriders");

    studentChangeable = new Student(100, "0", "A", "A", "x@connect.ust.hk",
                                   "0", "0", "0", "0", "0", "");
    studentChangeable.setStudentID("21112211");
    studentChangeable.setStudentName("Sal");
    studentChangeable.setStudentEmail("SalVa@connect.ust.hk");
    studentChangeable.setK1Energy("90");
    studentChangeable.setK2Energy("88");
    studentChangeable.setK3Tick1("1");
    studentChangeable.setK3Tick2("1");
    studentChangeable.setConcerns("I am really good!");
    studentChangeable.setIndex("101");

    student_data.add(student1);
    student_data.add(student2);
    student_data.add(student3);

    team = new Team(1, student_data, 0);

    ATUStudent.add(student3);
    ATUStudent.add(student2);
}

```

```

        ATUStudent.add(student1);

        ATUTeam1 = new Team(1, ATUStudent, 0);
    }

    /**
     * Test for team id
     */
    @Test
    public void isTeamIdEqual() {
        assertEquals(1, team.getID());
    }

    /**
     * Test for team recommended leader
     */
    @Test
    public void isTeamLeaderEqual() {
        assertEquals(student1, team.getLeader());
    }

    /**
     * Test for team K1
     */
    @Test
    public void isTeamK1AverageEqual() {
        assertEquals((int)(student1.getK1Energy_int()+student2.getK1Energy_int()+student3.getK1Energy_int())/(int),team.getK1Average());
    }

    /**
     * Test for team K2
     */
    @Test
    public void isTeamK2AverageEqual() {
        assertEquals((int)(student1.getK2Energy_int()+student2.getK2Energy_int()+student3.getK2Energy_int())/(int),team.getK2Average());
    }

    /**
     * Test for team member list
     */
    @Test
    public void isMemberListEqual() {
        assertEquals(student_data, team.getMemberList());
    }

    /**
     * Test for team
     */
    @Test
    public void isTeamEqual() {
        assertEquals(true, team.equals(team));
    }

    /**
     * Test for student name
     */
    @Test
    public void isStudentNameEqual() {
        assertEquals("SAFFRON, Corgipoo", student1.getStudentName());
    }
}

```

```

}

/**
 * Test for student id
 */
@Test
public void isStudentIDEqual() {
    assertEquals("20004488", student1.getStudentID());
}

/**
 * Test for student email
 */
@Test
public void isStudentEmailEqual() {
    assertEquals("CorgipooSAF@connect.ust.hk", student1.getStudentEmail());
}

/**
 * Test for student K1
 */
@Test
public void isStudentK1Equal() {
    assertEquals("26", student1.getK1Energy());
}

/**
 * Test for student K2
 */
@Test
public void isStudentK2Equal() {
    assertEquals("80", student1.getK2Energy());
}

/**
 * Test for student K1 compare - larger
 */
@Test
public void isStudentK1Larger() {
    assertEquals(-1, student2.compareTo(student1));
}

/**
 * Test for student K1 compare - smaller
 */
@Test
public void isStudentK1Smaller() {
    assertEquals(1, student2.compareTo(student3));
}

/**
 * Test for student K1 compare - equal
 */
@Test
public void isStudentK1CompareEqual1() {
    assertEquals(0, student2.compareTo(student2));
}

/**
 * Test for student K3 tick 1 - false
 */
@Test

```

```

public void isStudentK3Tick1ReturnFalse() {
    assertFalse(Integer.parseInt(student1.getK3Tick1()) == 1);
}

/**
 * Test for student K3 tick 2 - true
 */
@Test
public void isStudentK3Tick2ReturnTrue() {
    assertTrue(Integer.parseInt(student1.getK3Tick2()) == 1);
}

/**
 * Test for student concerns
 */
@Test
public void changeConcerns() {
    assertEquals(90, studentChangeable.getK1Energy_int());
    assertEquals(88, studentChangeable.getK2Energy_int());
    assertEquals("21112211", studentChangeable.getStudentID());
    assertEquals("Sal", studentChangeable.getStudentName());
    assertEquals("SalVa@connect.ust.hk", studentChangeable.getStudentEmail());
    assertEquals("I am really good!", studentChangeable.getConcerns());
    assertEquals("101", studentChangeable.getIndex());
    assertEquals(true, studentChangeable.getK3Tick1_bool());
    assertEquals(true, studentChangeable.getK3Tick2_bool());
}

/**
 * Test for InputManager statistic
 */
@Test
public void checkStatistics() {
    ArrayList<Statistics> stats = InputManager.getStatistics(student_data);
    assertEquals("3", stats.get(0).getValue());
    assertEquals("(36.7, 26, 57)", stats.get(1).getValue());
    assertEquals("(75.0, 60, 85)", stats.get(2).getValue());
    assertEquals("1", stats.get(3).getValue());
    assertEquals("2", stats.get(4).getValue());
    assertEquals("2", stats.get(5).getValue());
    assertEquals("0", stats.get(0).getIndex());
    assertEquals("K3_Tick2 = 1", stats.get(4).getEntry());
}

/**
 * Test for InputManager K1 mean, min, max
 */
@Test
public void isK1MMMEqual() {
    String[] k1 = InputManager.get_student_k1_mmm(student_data);
    assertEquals("36.7", k1[0]);
    assertEquals("26", k1[1]);
    assertEquals("57", k1[2]);
}

/**
 * Test for InputManager K2 mean, min, max
 */
@Test
public void isK2MMMEqual() {
    String[] k2 = InputManager.get_student_k2_mmm(student_data);
    assertEquals("75.0", k2[0]);
}

```

```

        assertEquals("60", k2[1]);
        assertEquals("85", k2[2]);
    }

    /**
     * Test for InputManager K3 tick 1
     */
    @Test
    public void isK3_tick1Equal() {
        String t = InputManager.get_k3_ticks(student_data)[0];
        assertEquals("1", t);
    }

    /**
     * Test for InputManager K3 tick 2
     */
    @Test
    public void isK3_tick2Equal() {
        String t = InputManager.get_k3_ticks(student_data)[1];
        assertEquals("2", t);
    }

    /**
     * Test for InputManager preference
     */
    @Test
    public void isMyPrefEqual() {
        String t = InputManager.get_k3_ticks(student_data)[2];
        assertEquals("2", t);
    }

    /**
     * Test for Security password
     */
    @Test
    public void isPasswordCorrect() {
        assertFalse(Security.checkPW("abcd"));
        assertTrue(Security.checkPW("1234"));
    }

    /**
     * Test for RequestWindowController
     */
    @Test
    public void checkUserLevelDialogResult() {
        ArrayList<Boolean> results1 =
RequestWindowController.onDialogGetResult(RequestWindowController.levels[0]);
        assertFalse(results1.get(0));
        assertFalse(results1.get(1));
        ArrayList<Boolean> result2 =
RequestWindowController.onDialogGetResult(RequestWindowController.levels[1]);
        assertTrue(result2.get(0));
    }

    /**
     * Test for ATUEngine
     */
    @Test
    public void isATUEngineExecutedCorrectly() {
        ArrayList<Student> temp = new ArrayList<Student>();
        temp.add(student1);
        temp.add(student2);
    }
}

```

```

        temp.add(student3);

        ATUEngine = new ATUEngine(temp);
        assertEquals(ATUTeam1.getMemberList().get(0).getStudentID(),
ATUTeam1.getMemberList().get(0).getStudentID());
        assertEquals(ATUTeam1.getMemberList().get(1).getStudentID(),
ATUTeam1.getMemberList().get(1).getStudentID());
        assertEquals(ATUTeam1.getMemberList().get(2).getStudentID(),
ATUTeam1.getMemberList().get(2).getStudentID());

        temp.add(student4);
        ATUEngine = new ATUEngine(temp);
        assertEquals(student1.getStudentID(),
ATUTeam1.getMemberList().get(3).getStudentID());

        temp.add(student5);
        temp.add(student6);
        temp.add(student7);
        temp.add(student8);
        ATUEngine = new ATUEngine(temp);
        assertEquals(student2.getStudentID(),
ATUTeam1.getMemberList().get(0).getStudentID());

        student8.setMyPreference("0");
        student3.setMyPreference("0");
        student2.setMyPreference("0");
        temp = new ArrayList<Student>();
        temp.add(student1);
        temp.add(student2);
        temp.add(student3);

        ATUEngine = new ATUEngine(temp);
        assertEquals(ATUTeam1.getMemberList().get(0).getStudentID(),
ATUTeam1.getMemberList().get(0).getStudentID());
        assertEquals(ATUTeam1.getMemberList().get(1).getStudentID(),
ATUTeam1.getMemberList().get(1).getStudentID());
        assertEquals(ATUTeam1.getMemberList().get(2).getStudentID(),
ATUTeam1.getMemberList().get(2).getStudentID());
        assertEquals(ATUTeam1.getMemberList().get(3).getStudentID(),
ATUTeam1.getMemberList().get(3).getStudentID());

        temp.add(student4);
        temp.add(student5);
        temp.add(student6);
        temp.add(student7);
        ATUEngine = new ATUEngine(temp);
        assertEquals(student2.getStudentID(),
ATUTeam1.getMemberList().get(0).getStudentID());
        assertEquals(ATUTeam1.getMemberList().get(1).getStudentID(),
ATUTeam1.getMemberList().get(1).getStudentID());
    }
}

```

## 2.2. Report on the coverage test (>65% branch coverage)

Element	Coverage	Uncovered Instructions	Covered Instructions	Total Instructions
comp3111G15	65.3 %	2,402	1,277	3,679
src/main/java	54.2 %	1,510	1,274	2,784
comp3111G15	54.2 %	1,510	1,274	2,784
RequestWindowController.java	10.3 %	54	470	524
DisplayWindowController.java	0.0 %	0	398	398
EnergyChartViewController.java	0.0 %	0	117	117
InputManager.java	88.4 %	608	80	688
StudentTableController.java	0.0 %	0	78	78
StatisticsTableController.java	0.0 %	0	36	36
UIApplication.java	0.0 %	0	34	34
ATUEngine.java	95.7 %	467	21	488
Statistics.java	69.4 %	34	15	49
Student.java	95.9 %	235	10	245
Library.java	0.0 %	0	6	6
Security.java	68.8 %	11	5	16
Team.java	96.2 %	101	4	105
src/test/java	99.7 %	892	3	895
comp3111G15	99.7 %	892	3	895
LibraryTest.java	99.7 %	892	3	895

## 2.3. Documentation on the implemented tasks using JavaDoc

### Package comp3111G15

Class Summary	
Class	Description
ATUEngine	The ATUEngine Produces Team-up results and have them stored in ATU_Team dataset.
DisplayWindowController	The DisplayWindowController describes the components used for the teaming up result table.
EnergyChartViewController	The EnergyChartViewController describes the components used for the Student Key Energies Zoom Out View
InputManager	The InputManager describes the resulting statistic data and student data
Library	The Library is the starting point of the application
LibraryTest	The LibraryTest is used for unit testing
RequestWindowController	The RequestWindowController describes the components used for the starting window of the ATU system.
Security	The Security checks password validation
Statistics	The Statistics shows table for students information
StatisticsTableController	The StatisticTableController controls the window for displaying statistics table
Student	The Student contains all information used for teaming up.
StudentTableController	The StudentTableController controls the window displaying the student information table
Team	The Team contains all information for one team
UIApplication	The UIApplication starts the UI of the ATU system

## Hierarchy For Package comp3111G15

### Class Hierarchy

- java.lang.Object
  - javafx.application.Application
    - comp3111G15.UIApplication
  - comp3111G15.ATUEngine
  - comp3111G15.DisplayWindowController (implements javafx.fxml.Initializable)
  - comp3111G15.EnergyChartViewController (implements javafx.fxml.Initializable)
  - comp3111G15.InputManager
  - comp3111G15.Library
  - comp3111G15.LibraryTest
  - comp3111G15.RequestWindowController
  - comp3111G15.Security
  - comp3111G15.Statistics
  - comp3111G15.StatisticsTableController
  - comp3111G15.Student (implements java.lang.Comparable<T>)
  - comp3111G15.StudentTableController
  - comp3111G15.Team

Package comp3111G15

## Class UIApplication

```
java.lang.Object
    javafx.application.Application
        comp3111G15.UIApplication
```

```
public class UIApplication
extends javafx.application.Application
```

The UIApplication starts the UI of the ATU system

Author:

SzeWingKwan

### Nested Class Summary

#### **Nested classes/interfaces inherited from class javafx.application.Application**

```
javafx.application.Application.Parameters
```

### Field Summary

#### **Fields inherited from class javafx.application.Application**

```
STYLESHEET_CASPIAN, STYLESHEET_MODENA
```

### Constructor Summary

#### Constructors

Constructor	Description
<code>UIApplication()</code>	

### Method Summary

#### All Methods

#### Static Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type	Method	Description
static void	<code>run(java.lang.String[] arg)</code>	UI entry point
void	<code>start(javafx.stage.Stage stage)</code>	Application entry point

#### **Methods inherited from class javafx.application.Application**

```
getHostServices, getParameters, getUserAgentStylesheet, init, launch, launch,
notifyPreloader, setUserAgentStylesheet, stop
```

## Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Details

### UIApplication

`public UIApplication()`

## Method Details

### start

`public void start(javafx.stage.Stage stage)  
throws java.lang.Exception`

Application entry point

Specified by:

`start in class javafx.application.Application`

Throws:

`java.lang.Exception`

### run

`public static void run(java.lang.String[] arg)`

UI entry point

Parameters:

`arg - argument array`

Package comp3111G15

## Class ATUEngine

java.lang.Object  
comp3111G15.ATUEngine

---

```
public class ATUEngine
extends java.lang.Object
```

The ATUEngine Produces Team-up results and have them stored in ATU\_Team dataset.

Author:

HE Qihao

### Field Summary

#### Fields

Modifier and Type	Field	Description
java.util.List<Team>	ATU_Team	List of team

### Constructor Summary

#### Constructors

Constructor	Description
ATUEngine (java.util.List<Student> studentData)	Class constructor, calls Create_Team() method to produce team-up results.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type	Method	Description
void	Create_Common_Team(int i)	Create normal teams with three students.
void	Create_Special_Team(int i)	Some finishing work on remaining students not allocated with a group in K3_list, this method create team with four students
void	Create_Team (java.util.List<Student> studentData)	Create teams and put students from K1_list, K2_list and K3_list in each of them.
java.util.List<Team>	getTeamlist()	Accessor that returns team-up results.

Modifier and Type	Method	Description
void	<code>Order_by_energies()</code>	Select top(Team_Size) student_id from student_data order by K1_Energy in descending order and store into K1_List, Then select top(Team_size) student_id from student_data order by K2_Energy and put in K2_list by ascending order, and then put rest students in K3_list with no order

### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait`

### Field Details

#### ATU\_Team

`public java.util.List<Team> ATU_Team`

List of team

### Constructor Details

#### ATUEngine

`public ATUEngine(java.util.List<Student> studentData)`

Class constructor, calls Create\_Team() method to produce team-up results.

**Parameters:**

`studentData` - for ATUEngine to start running

### Method Details

#### Order\_by\_energies

`public void Order_by_energies()`

Select top(Team\_Size) student\_id from student\_data order by K1\_Energy in descending order and store into K1\_List, Then select top(Team\_size) student\_id from student\_data order by K2\_Energy and put in K2\_list by ascending order, and then put rest students in K3\_list with no order

#### Create\_Common\_Team

```
public void Create_Common_Team(int i)
```

Create normal teams with three students.

**Parameters:**

i - for (team.id-1)

## Create\_Special\_Team

```
public void Create_Special_Team(int i)
```

Some finishing work on remaining students not allocated with a group in K3\_list, this method create team with four students

**Parameters:**

i - for (team.id-1)

## Create\_Team

```
public void Create_Team(java.util.List<Student> studentData)
```

Create teams and put students from K1\_list, K2\_list and K3\_list in each of them.

**Parameters:**

studentData - for teams to be created

## getTeamlist

```
public java.util.List<Team> getTeamlist()
```

Accessor that returns team-up results.

**Returns:**

list of resulting team

Package comp3111G15

## Class DisplayWindowController

java.lang.Object  
comp3111G15.DisplayWindowController

All Implemented Interfaces:

javafx.fxml.Initializable

---

```
public class DisplayWindowController
extends java.lang.Object
implements javafx.fxml.Initializable
```

The DisplayWindowController describes the components used for the teaming up result table.

Author:

SzeWingKwan

### Constructor Summary

#### Constructors

Constructor	Description
<code>DisplayWindowController()</code>	

### Method Summary

#### All Methods    Instance Methods    Concrete Methods

Modifier and Type	Method	Description
void	<code>initialize(java.net.URL location, java.util.ResourceBundle resources)</code>	

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait`

### Constructor Details

#### DisplayWindowController

```
public DisplayWindowController()
```

### Method Details

## initialize

```
public void initialize(java.net.URL location,  
                      java.util.ResourceBundle resources)
```

Specified by:

initialize in interface [javafx.fxml.Initializable](#)

Package comp3111G15

## Class EnergyChartViewController

```
java.lang.Object  
    comp3111G15.EnergyChartViewController
```

All Implemented Interfaces:

```
javafx.fxml.Initializable
```

---

```
public class EnergyChartViewController  
extends java.lang.Object  
implements javafx.fxml.Initializable
```

The EnergyChartViewController describes the components used for the Student Key Energies Zoom Out View

Author:

SzeWingKwan

### Constructor Summary

#### Constructors

Constructor	Description
<code>EnergyChartViewController()</code>	

### Method Summary

#### All Methods    Instance Methods    Concrete Methods

Modifier and Type	Method	Description
void	<code>initialize(java.net.URL url, java.util.ResourceBundle rb)</code>	

#### Methods inherited from class `java.lang.Object`

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

### Constructor Details

#### EnergyChartViewController

```
public EnergyChartViewController()
```

### Method Details

#### initialize

```
public void initialize(java.net.URL url,
                      java.util.ResourceBundle rb)
```

**Specified by:**

initialize in interface javafx.fxml.Initializable

Package comp3111G15

## Class InputManager

java.lang.Object  
comp3111G15.InputManager

```
public class InputManager  
extends java.lang.Object
```

The InputManager describes the resulting statistic data and student data

Author:

SzeWingKwan

### Field Summary

#### Fields

Modifier and Type	Field	Description
static java.lang.String	<code>delimiter</code>	Delimiter for student name
static javafx.collections.ObservableList<Statistics>	<code>stat_data</code>	List of statistics
static javafx.collections.ObservableList<Student>	<code>student_data</code>	List of student information

### Constructor Summary

#### Constructors

Constructor	Description
<code>InputManager()</code>	

### Method Summary

#### All Methods

#### Static Methods

#### Concrete Methods

Modifier and Type	Method	Description
static java.lang.String[]	<code>get_k3_ticks</code> (java.util.List<Student> student_data)	This function get the mean, min, max of all students' K3 ticks
static java.lang.String[]	<code>get_student_k1_mmm</code> (java.util.List<Student> student_data)	This function get the mean, min, max of all students' K1 value

Modifier and Type	Method	Description
static java.lang.String[]	<b>get_student_k2_mmm (java.util.List&lt;Student&gt; student_data)</b>	This function get the mean, min, max of all students' K2 value
static java.util.ArrayList<Statistics>	<b>getStatistics (java.util.List&lt;Student&gt; studentData)</b>	Populate the statistics stat_data, the ArrayList will contain number_of_student, K1mmm, K2mmm, K3_Tick1, K3_Tick2, My_preference, in order
static boolean	<b>read(java.lang.String csvFile)</b>	Read csv file

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait`

#### Field Details

##### **stat\_data**

`public static final javafx.collections.ObservableList<Statistics> stat_data`

List of statistics

##### **student\_data**

`public static final javafx.collections.ObservableList<Student> student_data`

List of student information

##### **delimiter**

`public static final java.lang.String delimiter`

Delimiter for student name

##### **See Also:**

Constant Field Values

#### Constructor Details

##### **InputManager**

```
public InputManager()
```

## Method Details

### get\_student\_k1\_mmm

```
public static java.lang.String[] get_student_k1_mmm(  
java.util.List<Student> student_data)
```

This function get the mean, min, max of all students' K1 value

**Parameters:**

`student_data` - A list of Student objects

**Returns:**

A string array of elements mean, min, max in ascending order.

### get\_student\_k2\_mmm

```
public static java.lang.String[] get_student_k2_mmm(  
java.util.List<Student> student_data)
```

This function get the mean, min, max of all students' K2 value

**Parameters:**

`student_data` - A list of Student objects

**Returns:**

A string array of elements mean, min, max in ascending order.

### get\_k3\_ticks

```
public static java.lang.String[] get_k3_ticks(java.util.List<Student> student_data)
```

This function get the mean, min, max of all students' K3 ticks

**Parameters:**

`student_data` - A list of Student objects

**Returns:**

A string array of elements tick1\_count, tick2\_count, my\_preference in ascending order.

### read

```
public static boolean read(java.lang.String csvFile)
```

Read csv file

**Parameters:**

`csvFile` - A String of file name in the parent directory

**Returns:**

true if the csv file is successfully read, false otherwise

## getStatistics

```
public static java.util.ArrayList<Statistics> getStatistics(  
java.util.List<Student> studentData)
```

Populate the statistics stat\_data, the ArrayList will contain number\_of\_student, K1mmm, K2mmm, K3\_Tick1, K3\_Tick2, My\_preference, in order

**Parameters:**

**studentData** - list of student data

**Returns:**

statistics

Package comp3111G15

## Class Library

java.lang.Object  
comp3111G15.Library

---

```
public class Library
extends java.lang.Object
```

The Library is the starting point of the application

Author:

SzeWingKwan

### Constructor Summary

#### Constructors

Constructor	Description
<a href="#">Library()</a>	

### Method Summary

#### All Methods    Static Methods    Concrete Methods

Modifier and Type	Method	Description
static void	<a href="#">main(java.lang.String[] args)</a>	The main method of the program

#### Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

### Constructor Details

#### Library

```
public Library()
```

### Method Details

#### main

```
public static void main(java.lang.String[] args)
```

The main method of the program

Parameters:

**args** - arguments

Package comp3111G15

## Class LibraryTest

java.lang.Object  
comp3111G15.LibraryTest

---

```
public class LibraryTest
extends java.lang.Object
```

The LibraryTest is used for unit testing

Author:

SzeWingKwan

### Constructor Summary

#### Constructors

Constructor	Description
<code>LibraryTest()</code>	

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type	Method	Description
void	<code>changeConcerns()</code>	Test for student concerns
void	<code>checkStatistics()</code>	Test for InputManager statistic
void	<code>checkUserLevelDialogResult()</code>	Test for RequestWindowController
void	<code>isATUEngineExecutedCorrectly()</code>	Test for ATUEngine
void	<code>isK1MMMEqual()</code>	Test for InputManager K1 mean, min, max
void	<code>isK2MMMEqual()</code>	Test for InputManager K2 mean, min, max
void	<code>isK3_tick1Equal()</code>	Test for InputManager K3 tick 1
void	<code>isK3_tick2Equal()</code>	Test for InputManager K3 tick 2
void	<code>isMemberListEqual()</code>	Test for team member list
void	<code>isMyPrefEqual()</code>	Test for InputManager preference
void	<code>isPasswordCorrect()</code>	Test for Security password
void	<code>isStudentEmailEqual()</code>	Test for student email
void	<code>isStudentIDEequal()</code>	Test for student id
void	<code>isStudentK1CompareEqual1()</code>	Test for student K1 compare - equal

Modifier and Type	Method	Description
void	<code>isStudentK1Equal()</code>	Test for student K1
void	<code>isStudentK1Larger()</code>	Test for student K1 compare - larger
void	<code>isStudentK1Smaller()</code>	Test for student K1 compare - smaller
void	<code>isStudentK2Equal()</code>	Test for student K2
void	<code>isStudentK3Tick1ReturnFalse()</code>	Test for student K3 tick 1 - false
void	<code>isStudentK3Tick2ReturnTrue()</code>	Test for student K3 tick 2 - true
void	<code>isStudentNameEqual()</code>	Test for student name
void	<code>isTeamEqual()</code>	Test for team
void	<code>isTeamIdEqual()</code>	Test for team id
void	<code>isTeamK1AverageEqual()</code>	Test for team K1
void	<code>isTeamK2AverageEqual()</code>	Test for team K2
void	<code>isTeamLeaderEqual()</code>	Test for team recommended leader
void	<code>setUp()</code>	Set up control variables

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

#### Constructor Details

##### LibraryTest

```
public LibraryTest()
```

#### Method Details

##### setUp

```
public void setUp()
    throws java.lang.Exception
```

Set up control variables

**Throws:**

`java.lang.Exception` - exception

##### isTeamIdEqual

```
public void isTeamIdEqual()
```

Test for team id

### **isTeamLeaderEqual**

```
public void isTeamLeaderEqual()
```

Test for team recommended leader

### **isTeamK1AverageEqual**

```
public void isTeamK1AverageEqual()
```

Test for team K1

### **isTeamK2AverageEqual**

```
public void isTeamK2AverageEqual()
```

Test for team K2

### **isMemberListEqual**

```
public void isMemberListEqual()
```

Test for team member list

### **isTeamEqual**

```
public void isTeamEqual()
```

Test for team

### **isStudentNameEqual**

```
public void isStudentNameEqual()
```

Test for student name

### **isStudentIDEqual**

```
public void isStudentIDEqual()
```

Test for student id

### **isStudentEmailEqual**

```
public void isStudentEmailEqual()
```

Test for student email

### **isStudentK1Equal**

```
public void isStudentK1Equal()
```

Test for student K1

### **isStudentK2Equal**

```
public void isStudentK2Equal()
```

Test for student K2

### **isStudentK1Larger**

```
public void isStudentK1Larger()
```

Test for student K1 compare - larger

### **isStudentK1Smaller**

```
public void isStudentK1Smaller()
```

Test for student K1 compare - smaller

### **isStudentK1CompareEqual1**

```
public void isStudentK1CompareEqual1()
```

Test for student K1 compare - equal

### **isStudentK3Tick1ReturnFalse**

```
public void isStudentK3Tick1ReturnFalse()
```

Test for student K3 tick 1 - false

### **isStudentK3Tick2ReturnTrue**

```
public void isStudentK3Tick2ReturnTrue()
```

Test for student K3 tick 2 - true

### **changeConcerns**

```
public void changeConcerns()
```

Test for student concerns

### **checkStatistics**

```
public void checkStatistics()
```

Test for InputManager statistic

### **isK1MMMEqual**

```
public void isK1MMMEqual()
```

Test for InputManager K1 mean, min, max

### **isK2MMMEqual**

```
public void isK2MMMEqual()
```

Test for InputManager K2 mean, min, max

### **isK3\_tick1Equal**

```
public void isK3_tick1Equal()
```

Test for InputManager K3 tick 1

### **isK3\_tick2Equal**

```
public void isK3_tick2Equal()
```

Test for InputManager K3 tick 2

### **isMyPrefEqual**

```
public void isMyPrefEqual()
```

Test for InputManager preference

### **isPasswordCorrect**

```
public void isPasswordCorrect()
```

Test for Security password

### **checkUserLevelDialogResult**

```
public void checkUserLevelDialogResult()
```

Test for RequestWindowController

### **isATUEngineExecutedCorrectly**

```
public void isATUEngineExecutedCorrectly()
```

Test for ATUEngine



Package comp3111G15

## Class RequestWindowController

java.lang.Object  
comp3111G15.RequestWindowController

```
public class RequestWindowController  
extends java.lang.Object
```

The RequestWindowController describes the components used for the starting window of the ATU system.

Author:

SzeWingKwan, LiChunTak, HE Qihao

### Field Summary

#### Fields

Modifier and Type	Field	Description
static java.lang.String[]	levels	two levels of users, Student or TA

### Constructor Summary

#### Constructors

Constructor	Description
RequestWindowController()	

### Method Summary

#### All Methods

#### Static Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type	Method	Description
static void	displayIncorrectFilenameDialog (java.lang.String filename)	Display a error dialog for when the .csv filename does not exist.
void	initialize()	Initialize the application with user authentication and display UI window
static java.util.ArrayList<java.lang.Boolean>	onDialogGetResult (java.lang.String result)	Get the result of the dialog which asks for user level

## Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Field Details

### levels

```
public static java.lang.String[] levels
```

two levels of users, Student or TA

## Constructor Details

### RequestWindowController

```
public RequestWindowController()
```

## Method Details

### initialize

```
public void initialize()
```

Initialize the application with user authentication and display UI window

### onDialogGetResult

```
public static java.util.ArrayList<java.lang.Boolean> onDialogGetResult(
    java.lang.String result)
```

Get the result of the dialog which asks for user level

**Parameters:**

**result** - from the dialog

**Returns:**

temporary boolean list, first element: If TA return true, else false, second element: If csv read successful return true, else false

### displayIncorrectFilenameDialog

```
public static void displayIncorrectFilenameDialog(java.lang.String filename)
```

Display a error dialog for when the .csv filename does not exist.

**Parameters:**

**filename** - incorrect filename

Package comp3111G15

## Class Security

java.lang.Object  
comp3111G15.Security

```
public class Security
extends java.lang.Object
```

The Security checks password validation

Author:

SzeWingKwan

### Constructor Summary

#### Constructors

Constructor	Description
<code>Security()</code>	

### Method Summary

#### All Methods    Static Methods    Concrete Methods

Modifier and Type	Method	Description
static boolean	<code>checkPW (java.lang.String input)</code>	Check input password against the default password

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Constructor Details

#### Security

```
public Security()
```

### Method Details

#### checkPW

```
public static boolean checkPW(java.lang.String input)
```

Check input password against the default password

**Parameters:**

`input` - String of input

**Returns:**

true if match, false otherwise

Package comp3111G15

## Class Statistics

java.lang.Object  
comp3111G15.Statistics

```
public class Statistics  
extends java.lang.Object
```

The Statistics shows table for students information

Author:

LiChunTak

### Constructor Summary

#### Constructors

Constructor	Description
<code>Statistics(int index, java.lang.String fName, java.lang.String lName)</code>	Constructor of Statistics

### Method Summary

#### All Methods    Instance Methods    Concrete Methods

Modifier and Type	Method	Description
java.lang.String	<code>getEntry()</code>	Get entry of Statistics in a string
java.lang.String	<code>getIndex()</code>	Get index of Statistics in a string
java.lang.String	<code>getValue()</code>	Get value of Statistics in a string
void	<code>setEntry(java.lang.String val)</code>	Set entry of Statistics
void	<code>setIndex(java.lang.String val)</code>	Set index of Statistics
void	<code>setValue(java.lang.String val)</code>	Set value of Statistics

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Constructor Details

#### Statistics

```
public Statistics(int index,  
                 java.lang.String fName,  
                 java.lang.String lName)
```

Constructor of Statistics

**Parameters:**

**index** - index

**fName** - first name

**lName** - last name

## **Method Details**

### **getIndex**

```
public java.lang.String getIndex()
```

Get index of Statistics in a string

**Returns:**

String

### **setIndex**

```
public void setIndex(java.lang.String val)
```

Set index of Statistics

**Parameters:**

**val** - index

### **getEntry**

```
public java.lang.String getEntry()
```

Get entry of Statistics in a string

**Returns:**

String

### **setEntry**

```
public void setEntry(java.lang.String val)
```

Set entry of Statistics

**Parameters:**

**val** - entry

### **getValue**

```
public java.lang.String getValue()
```

Get value of Statistics in a string

**Returns:**

String

### **setValue**

```
public void setValue(java.lang.String val)
```

Set value of Statistics

**Parameters:**

val - value

Package comp3111G15

## Class StatisticsTableController

java.lang.Object  
comp3111G15.StatisticsTableController

---

```
public class StatisticsTableController  
extends java.lang.Object
```

The StatisticTableController controls the window for displaying statistics table

Author:

SzeWingKwan

### ***Constructor Summary***

#### **Constructors**

Constructor	Description
<code>StatisticsTableController()</code>	

### ***Method Summary***

#### **Methods inherited from class java.lang.Object**

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

### ***Constructor Details***

#### **StatisticsTableController**

```
public StatisticsTableController()
```

Package comp3111G15

## Class Student

java.lang.Object  
comp3111G15.Student

All Implemented Interfaces:

java.lang.Comparable<Student>

---

```
public class Student
extends java.lang.Object
implements java.lang.Comparable<Student>
```

The Student contains all information used for teaming up.

Author:

SzeWingKwan

### Constructor Summary

#### Constructors

Constructor	Description
<code>Student(int index, java.lang.String student_id, java.lang.String student_name, java.lang.String email, java.lang.String k1_energy, java.lang.String k2_energy, java.lang.String k3_tick1, java.lang.String k3_tick2, java.lang.String my_preference, java.lang.String concerns)</code>	Class constructor

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type	Method	Description
int	<code>compareTo(Student other)</code>	The method used for sorting the students in a list in descending K1 order.
java.lang.String	<code>getConcerns()</code>	Gets concerns
java.lang.String	<code>getIndex()</code>	Gets index
java.lang.String	<code>getK1Energy()</code>	Gets K1 energy in String
int	<code>getK1Energy_int()</code>	Gets K1 energy in integer
java.lang.String	<code>getK2Energy()</code>	Gets K2 energy in String
int	<code>getK2Energy_int()</code>	Gets K2 energy in integer
java.lang.String	<code>getK3Tick1()</code>	Gets K3 tick 1 in String
boolean	<code>getK3Tick1_bool()</code>	Gets K3 tick 1

Modifier and Type	Method	Description
	<code>java.lang.String getK3Tick2()</code>	Gets K3 tick 2 in String
<code>boolean</code>	<code>getK3Tick2_bool()</code>	Gets K3 tick 2
	<code>java.lang.String getMyPreference()</code>	Gets my preference
	<code>java.lang.String getStudentEmail()</code>	Gets student email
	<code>java.lang.String getStudentID()</code>	Gets student ID
	<code>java.lang.String getStudentName()</code>	Gets student name
<code>void</code>	<code>setConcerns (java.lang.String val)</code>	Sets concerns
<code>void</code>	<code>setIndex (java.lang.String val)</code>	Sets index
<code>void</code>	<code>setK1Energy (java.lang.String val)</code>	Sets K1 energy
<code>void</code>	<code>setK2Energy (java.lang.String val)</code>	Sets K2 energy
<code>void</code>	<code>setK3Tick1 (java.lang.String val)</code>	Sets K3 tick 1
<code>void</code>	<code>setK3Tick2 (java.lang.String val)</code>	Sets K3 tick 2
<code>void</code>	<code>setMyPreference (java.lang.String val)</code>	Sets my preference
<code>void</code>	<code>setStudentEmail (java.lang.String val)</code>	Sets student email
<code>void</code>	<code>setStudentID (java.lang.String val)</code>	Sets student ID
<code>void</code>	<code>setStudentName (java.lang.String val)</code>	Sets student name

#### Methods inherited from class `java.lang.Object`

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

#### Constructor Details

`Student`

```
public Student(int index,
              java.lang.String student_id,
              java.lang.String student_name,
              java.lang.String email,
              java.lang.String k1_energy,
              java.lang.String k2_energy,
              java.lang.String k3_tick1,
              java.lang.String k3_tick2,
              java.lang.String my_preference,
              java.lang.String concerns)
```

Class constructor

**Parameters:**

**index** - index

**student\_id** - student ID

**student\_name** - student name

**email** - student email

**k1\_energy** - K1 energy

**k2\_energy** - K2 energy

**k3\_tick1** - whether is creative and participate aggressively

**k3\_tick2** - whether is willing to take more workloads

**my\_preference** - my preference to be the leader

**concerns** - concerns/comments

## **Method Details**

### **compareTo**

```
public int compareTo(Student other)
```

The method used for sorting the students in a list in descending K1 order.

**Specified by:**

`compareTo` in interface `java.lang.Comparable<Student>`

### **getK1Energy\_int**

```
public int getK1Energy_int()
```

Gets K1 energy in integer

**Returns:**

K1 energy

### **getK2Energy\_int**

```
public int getK2Energy_int()
```

Gets K<sub>2</sub> energy in integer

**Returns:**

K<sub>2</sub> energy

### **getK3Tick1\_bool**

```
public boolean getK3Tick1_bool()
```

Gets K<sub>3</sub> tick 1

**Returns:**

true/false

### **getK3Tick2\_bool**

```
public boolean getK3Tick2_bool()
```

Gets K<sub>3</sub> tick 2

**Returns:**

true/false

### **getIndex**

```
public java.lang.String getIndex()
```

Gets index

**Returns:**

index

### **setIndex**

```
public void setIndex(java.lang.String val)
```

Sets index

**Parameters:**

val - for index

### **getStudentID**

```
public java.lang.String getStudentID()
```

Gets student ID

**Returns:**

student ID

## **setStudentID**

```
public void setStudentID(java.lang.String val)
```

Sets student ID

**Parameters:**

**val** - for student ID

## **getStudentName**

```
public java.lang.String getStudentName()
```

Gets student name

**Returns:**

student name

## **setStudentName**

```
public void setStudentName(java.lang.String val)
```

Sets student name

**Parameters:**

**val** - for student name

## **getStudentEmail**

```
public java.lang.String getStudentEmail()
```

Gets student email

**Returns:**

student email

## **setStudentEmail**

```
public void setStudentEmail(java.lang.String val)
```

Sets student email

**Parameters:**

**val** - for student email

## **getK1Energy**

```
public java.lang.String getK1Energy()
```

Gets K1 energy in String

**Returns:**

K1 energy

### setK1Energy

```
public void setK1Energy(java.lang.String val)
```

Sets K1 energy

**Parameters:**

val - for K1 energy

### getK2Energy

```
public java.lang.String getK2Energy()
```

Gets K2 energy in String

**Returns:**

K2 energy

### setK2Energy

```
public void setK2Energy(java.lang.String val)
```

Sets K2 energy

**Parameters:**

val - for K2 energy

### getK3Tick1

```
public java.lang.String getK3Tick1()
```

Gets K3 tick 1 in String

**Returns:**

1/0

### setK3Tick1

```
public void setK3Tick1(java.lang.String val)
```

Sets K3 tick 1

**Parameters:**

val - for K3 tick 1

### getK3Tick2

```
public java.lang.String getK3Tick2()
```

Gets K3 tick 2 in String

**Returns:**

1/o

### setK3Tick2

```
public void setK3Tick2(java.lang.String val)
```

Sets K3 tick 2

**Parameters:**

val - for K3 tick 2

### getMyPreference

```
public java.lang.String getMyPreference()
```

Gets my preference

**Returns:**

1/o

### setMyPreference

```
public void setMyPreference(java.lang.String val)
```

Sets my preference

**Parameters:**

val - for my preference

### getConcerns

```
public java.lang.String getConcerns()
```

Gets concerns

**Returns:**

concerns

### setConcerns

```
public void setConcerns(java.lang.String val)
```

Sets concerns

**Parameters:**

val - for concerns



Package comp3111G15

## Class StudentTableController

```
java.lang.Object  
    comp3111G15.StudentTableController
```

```
public class StudentTableController  
extends java.lang.Object
```

The StudentTableController controls the window displaying the student information table

Author:

SzeWingKwan

### **Constructor Summary**

#### **Constructors**

Constructor	Description
<code>StudentTableController()</code>	

### **Method Summary**

#### **Methods inherited from class java.lang.Object**

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait
```

### **Constructor Details**

#### **StudentTableController**

```
public StudentTableController()
```

Package comp3111G15

## Class Team

java.lang.Object  
comp3111G15.Team

```
public class Team  
extends java.lang.Object
```

The Team contains all information for one team

Author:

SzeWingKwan

### Constructor Summary

#### Constructors

Constructor	Description
<code>Team(int id, java.util.List&lt;Student&gt; members, int leader)</code>	Class constructor

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type	Method	Description
int	<code>getID()</code>	Gets team ID
double	<code>getK1Average()</code>	Gets average K1 energy
double	<code>getK2Average()</code>	Gets average K2 energy
<code>Student</code>	<code>getLeader()</code>	Gets recommended leader
<code>java.util.List&lt;Student&gt;</code>	<code>getMemberList()</code>	Gets team member list
void	<code>sortMember()</code>	The method sorts the team member list in descending K1 order.

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait`

### Constructor Details

#### Team

```
public Team(int id,  
           java.util.List<Student> members,  
           int leader)
```

Class constructor

**Parameters:**

`id` - team ID

`members` - team member list

`leader` - recommended leader

## **Method Details**

### **sortMember**

```
public void sortMember()
```

The method sorts the team member list in descending K<sub>1</sub> order.

### **getID**

```
public int getID()
```

Gets team ID

**Returns:**

team ID

### **getMemberList**

```
public java.util.List<Student> getMemberList()
```

Gets team member list

**Returns:**

team member list

### **getLeader**

```
public Student getLeader()
```

Gets recommended leader

**Returns:**

recommended leader

### **getK1Average**

```
public double getK1Average()
```

Gets average K<sub>1</sub> energy

**Returns:**

average K1 energy

### **getK2Average**

```
public double getK2Average()
```

Gets average K2 energy

**Returns:**

average K2 energy

## Uses of Package comp3111G15

### Classes in comp3111G15 used by comp3111G15

Class	Description
Statistics	The Statistics shows table for students information
Student	The Student contains all information used for teaming up.
Team	The Team contains all information for one team

## Uses of Class comp3111G15.Statistics

### Uses of Statistics in comp3111G15

#### Fields in comp3111G15 with type parameters of type Statistics

Modifier and Type	Field	Description
static javafx.collections.ObservableList<Statistics>	InputManager.stat_data	List of statistics

#### Methods in comp3111G15 that return types with arguments of type Statistics

Modifier and Type	Method	Description
static java.util.ArrayList<Statistics> (java.util.List<Student> studentData)	InputManager.getStatistics	Populate the statistics stat_data, the ArrayList will contain number_of_student, K1mmm, K2mmm, K3_Tick1, K3_Tick2, My_preference, in order

## Uses of Class comp3111G15.Student

### Uses of Student in comp3111G15

#### Fields in comp3111G15 with type parameters of type Student

Modifier and Type	Field	Description
static javafx.collections.ObservableList<Student>	InputManager.student_data	List of student information

#### Methods in comp3111G15 that return Student

Modifier and Type	Method	Description
Student	Team.getLeader()	Gets recommended leader

#### Methods in comp3111G15 that return types with arguments of type Student

Modifier and Type	Method	Description
java.util.List<Student>	Team.getMemberList()	Gets team member list

#### Methods in comp3111G15 with parameters of type Student

Modifier and Type	Method	Description
int	Student.compareTo (Student other)	The method used for sorting the students in a list in descending K1 order.

#### Method parameters in comp3111G15 with type arguments of type Student

Modifier and Type	Method	Description
void	ATUEngine.Create_Team (java.util.List<Student> studentData)	Create teams and put students from K1_list, K2_list and K3_list in each of them.
static java.lang.String[]	InputManager.get_k3_ticks (java.util.List<Student> student_data)	This function get the mean, min, max of all students' K3 ticks
static java.lang.String[]	InputManager.get_student_k1_mmm (java.util.List<Student> student_data)	This function get the mean, min, max of all students' K1 value
static java.lang.String[]	InputManager.get_student_k2_mmm (java.util.List<Student> student_data)	This function get the mean, min, max of all students' K2 value

Modifier and Type	Method	Description
static java.util.ArrayList<Statistics> (java.util.List<Student> studentData)	InputManager.getStatistics	Populate the statistics stat_data, the ArrayList will contain number_of_student, K1mmm, K2mmm, K3_Tick1, K3_Tick2, My_preference, in order

#### Constructor parameters in comp3111G15 with type arguments of type Student

Constructor	Description
<b>ATUEngine</b> (java.util.List<Student> studentData)	Class constructor, calls Create_Team() method to produce team-up results.
<b>Team</b> (int id, java.util.List<Student> members, int leader)	Class constructor

## Uses of Class comp3111G15.Team

### *Uses of Team in comp3111G15*

#### Fields in comp3111G15 with type parameters of type Team

Modifier and Type	Field	Description
java.util.List<Team>	<b>ATUEngine.ATU_Team</b>	List of team

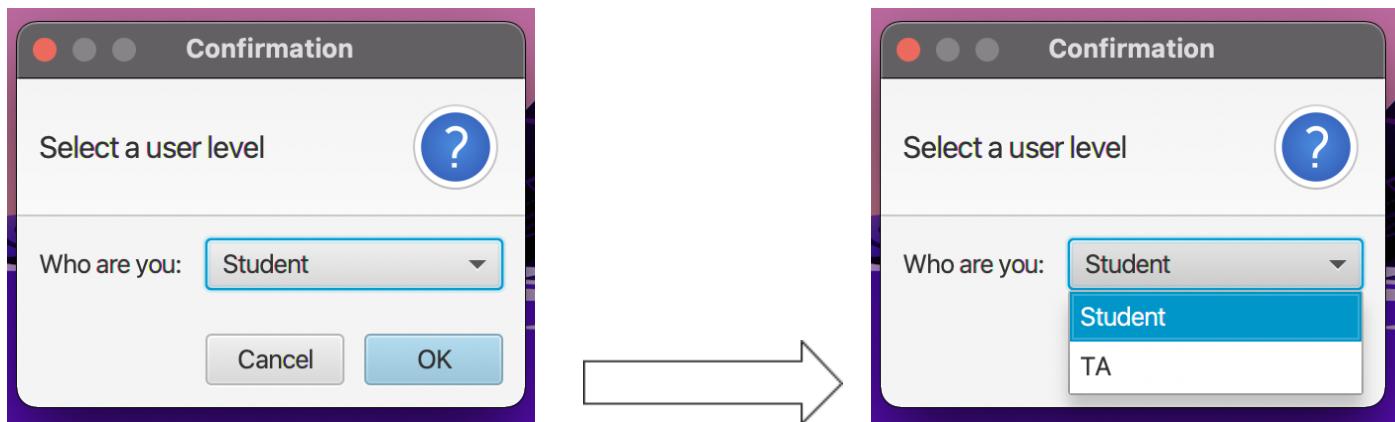
#### Methods in comp3111G15 that return types with arguments of type Team

Modifier and Type	Method	Description
java.util.List<Team>	<b>ATUEngine.getTeamlist()</b>	Accessor that returns team-up results.

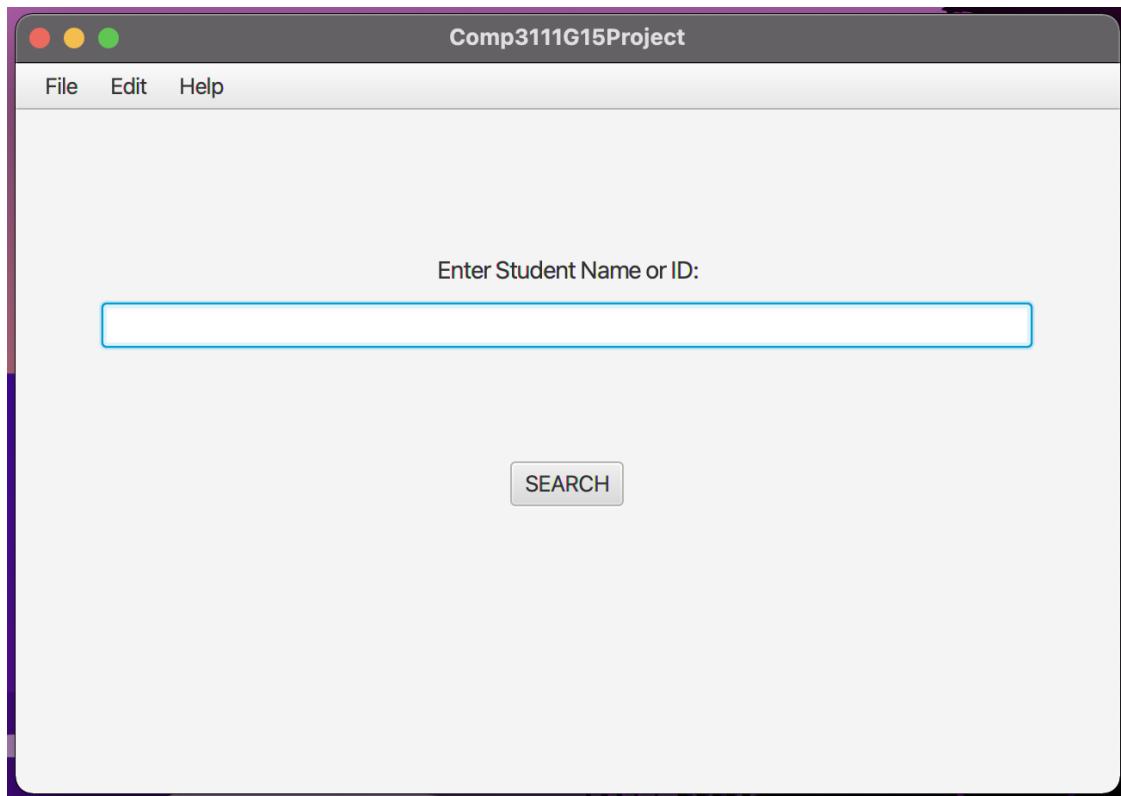
## PART 3: Program Execution and Screenshots

### 3.1. Screenshots of the execution of the application showing sample inputs and outputs of your choice (Task 231A)

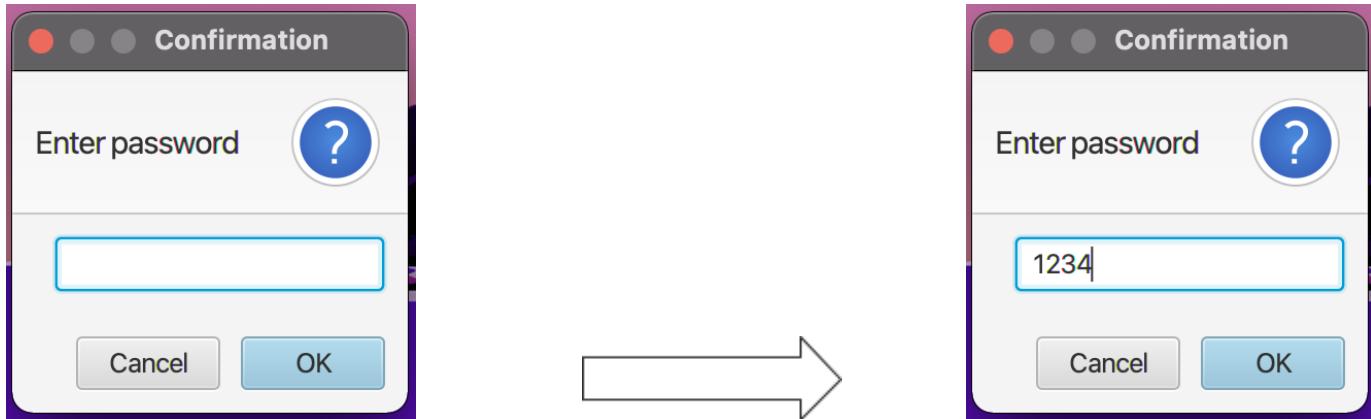
Two type of users:



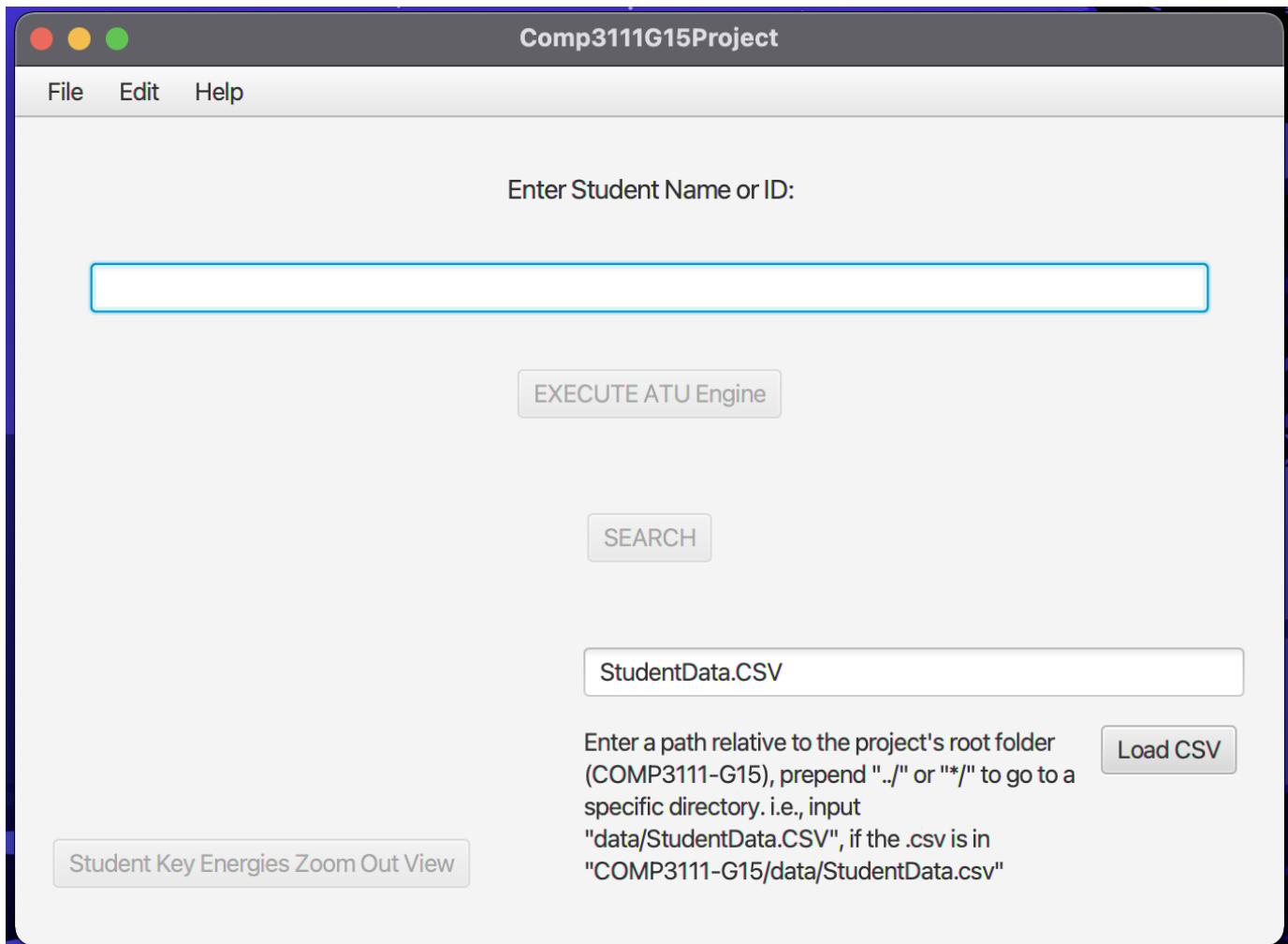
When “student” is chosen:



When “TA” is chosen:



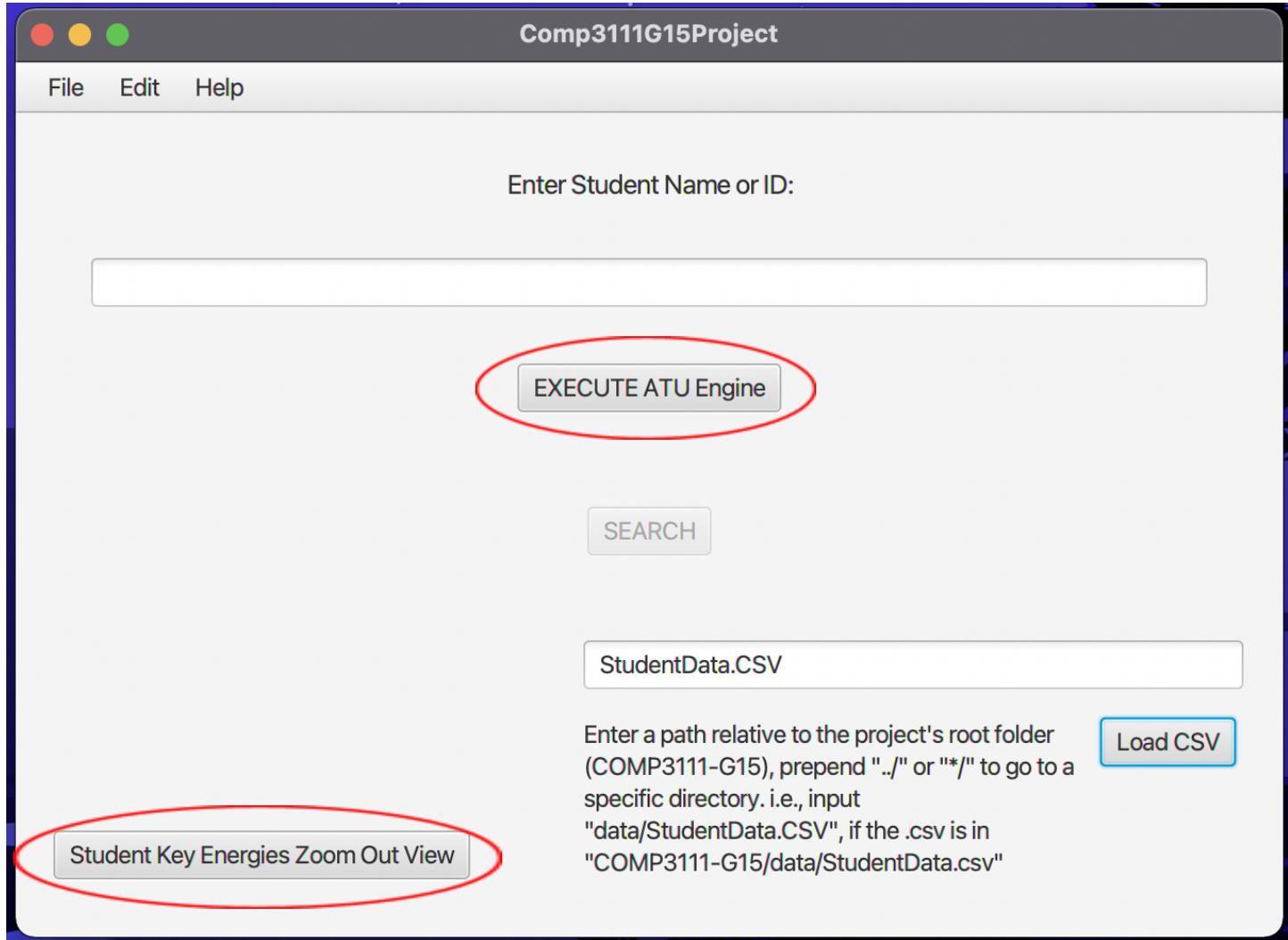
Original View for TA (correct password):



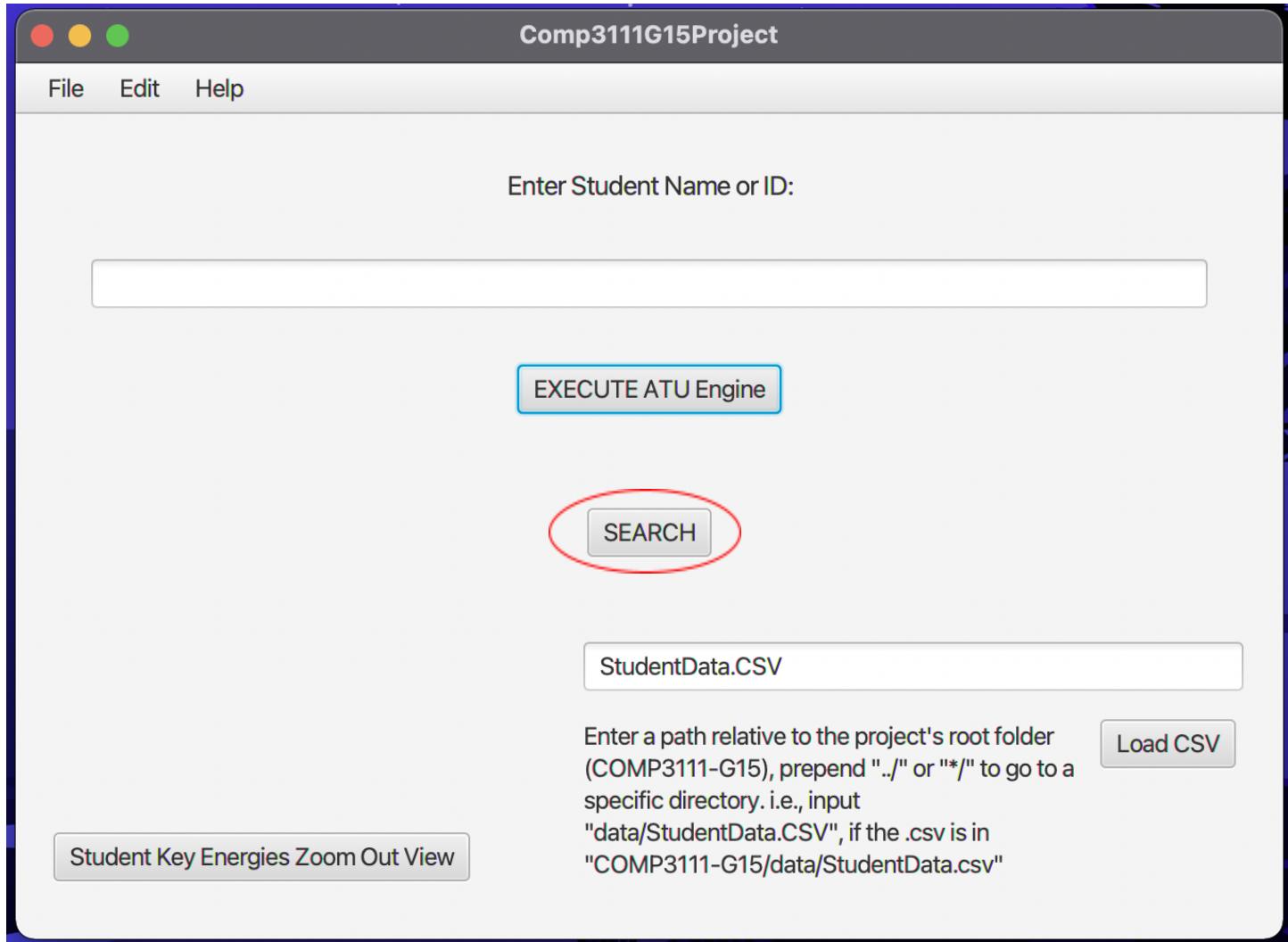
When a .csv file is loaded successfully:

Table of students' personal data									
Students									
Row_Index	Student_ID	Student_Name	Student_Email	K1_Energy	K2_Energy	K3_Tick1	K3_Tick2	My_Preference	Concerns
0	20004488	SAFFRON Co...	CorgipooSAF...	26	80	0	0	1	...
1	20023331	HYSSOP Cha...	ChamoisHYS...	27	85	0	0	0	...
2	20043679	LEEKs Beetle	BeetleLEE@c...	71	40	0	0	0	...
3	20067232	CHRYSANTH...	AbelisaurusC...	57	60	0	0	0	...
4	20076931	CELERIAC Aniu	AniuCEL@co...	54	80	0	0	1	...
5	20089887	ANGELICA D...	DaisyANG@c...	44	65	1	1	0	...
6	20097861	CHICORY Igu...	IguanodonC...	58	45	0	0	0	...
7	20109368	SPEARMINT ...	KenerdySPE...	44	70	0	0	0	I like to work ...
8	20121416	LAVENDER V...	VenusLAV@...	55	65	0	0	0	...
9	20133348	VANILLA Aph...	AphidsVAN@...	47	65	1	0	0	...
10	20136565	CUMINPOW...	AddaxCUM@...	47	60	1	1	0	...
11	20152854	PARSLEY Ath...	AthenaPAR@...	59	60	0	0	0	...
12	20153166	ORCHID Uran...	UranusORC...	89	45	0	0	0	...
13	20155801	ROSEMARY ...	HephaestusR...	35	70	0	0	1	...
14	20167346	OREGANO G...	GeorgeORE...	83	60	0	0	0	...

Table of statistics		
Statistics		
Row_Index	Entry	Value
0	Total Number of ...	100
1	K1_Energy(Aver...)	(55.1, 10, 100)
2	K2_Energy(Aver...)	(65.6, 30, 90)
3	K3_Tick1 = 1	12
4	K3_Tick2 = 1	5
5	My_Preference ...	19



When “EXECUTE ATU Engine” button is pressed:



### 3.2. Screenshots of the execution of the application showing sample inputs and outputs of your choice (Task 231C)

Teaming up result (press “SEARCH” with an input)

1. Valid student name or ID:

- Case 1 - One of the teammates is the recommended leader

Search Result

File Edit Help

Teaming Up Result

Student Information:	My Teammates	Recommended Leader
WASSABI Python 20734171	<p>▼ SAFFRON Corgipoo</p> <p>Student ID: 20004488</p> <p>Email: CorgipooSAF@connect.ust.hk</p> <p>K1 Energy: 26</p> <p>K2 Energy: 80</p> <p>K3 Tick 1: No</p> <p>K3 Tick 2: No</p> <p>Preference: Yes</p> <p>Concerns: ...</p>	Yes
My K1 Energy: 79		
My K2 Energy: 60		
My Team Number: 14		
Average K1 Energy: 46.0		
Average K2 Energy: 65.0		
	<p>▼ PERUVIAN Florestina</p> <p>Student ID: 20972736</p> <p>Email: FlorestinaPER@connect.ust.hk</p> <p>K1 Energy: 33</p> <p>K2 Energy: 55</p> <p>K3 Tick 1: No</p> <p>K3 Tick 2: Yes</p>	No

If no recommended leader is shown, you are the recommended leader :).

The screenshot shows a Java Swing application window titled "Search Result". The main title bar has three colored circles (red, yellow, green) on the left and the title "Search Result" in the center. Below the title bar is a menu bar with "File", "Edit", and "Help" items. The main content area is titled "Teaming Up Result". On the left, there is a sidebar with student information: "WASSABI Python" and "20734171". Below that are "My K1 Energy: 79", "My K2 Energy: 60", "My Team Number: 14", "Average K1 Energy: 46.0", and "Average K2 Energy: 65.0". The main content area contains a table with three columns: "Student Information:", "My Teammates", and "Recommended Leader". The "My Teammates" column lists two teammates: "SAFFRON Corgipoo" and "PERUVIAN Florestina". For "SAFFRON Corgipoo", the "Recommended Leader" is marked as "Yes". For "PERUVIAN Florestina", the "Recommended Leader" is marked as "No". The "My Teammates" table rows have a blue border. At the bottom of the window, a message says "If no recommended leader is shown, you are the recommended leader :) .".

- Case 2 - The student being searched is the recommended leader

Search Result

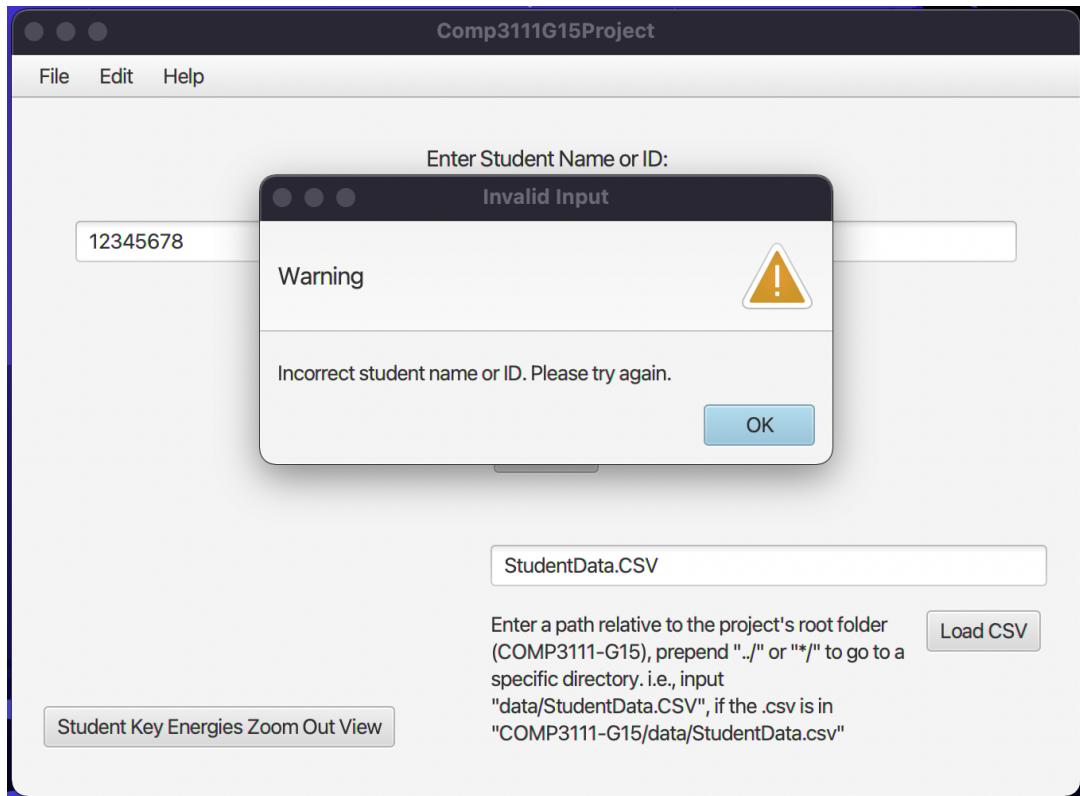
File Edit Help

Teaming Up Result

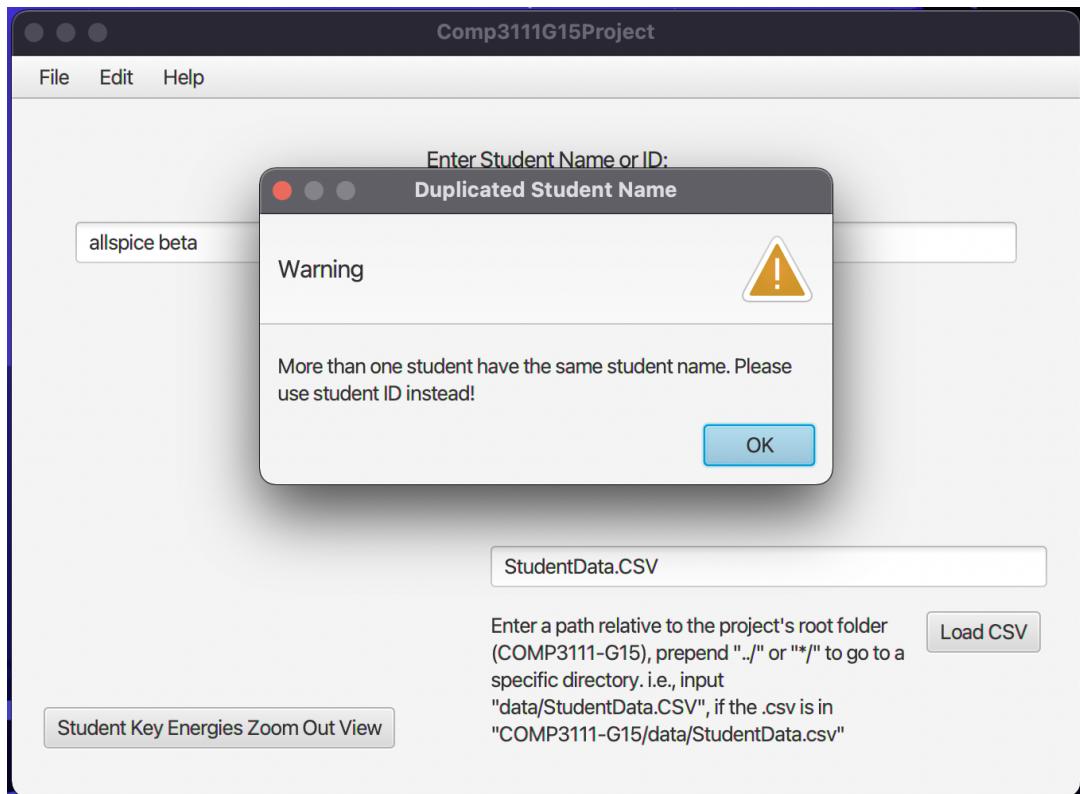
Student Information:	My Teammates	Recommended Leader
SAFFRON Corgipoo 20004488	<b>WASSABI Python</b> Student ID: 20734171 Email: PythonWAS@connect.ust.hk K1 Energy: 79 K2 Energy: 60 K3 Tick 1: No K3 Tick 2: No Preference: No Concerns: ...	No
My K1 Energy: 26  My K2 Energy: 80  My Team Number: 14  Average K1 Energy: 46.0  Average K2 Energy: 65.0	<b>PERUVIAN Florestina</b> Student ID: 20972736 Email: FlorestinaPER@connect.ust.hk K1 Energy: 33 K2 Energy: 55 K3 Tick 1: No K3 Tick 2: Yes	No

If no recommended leader is shown, you are the recommended leader :).

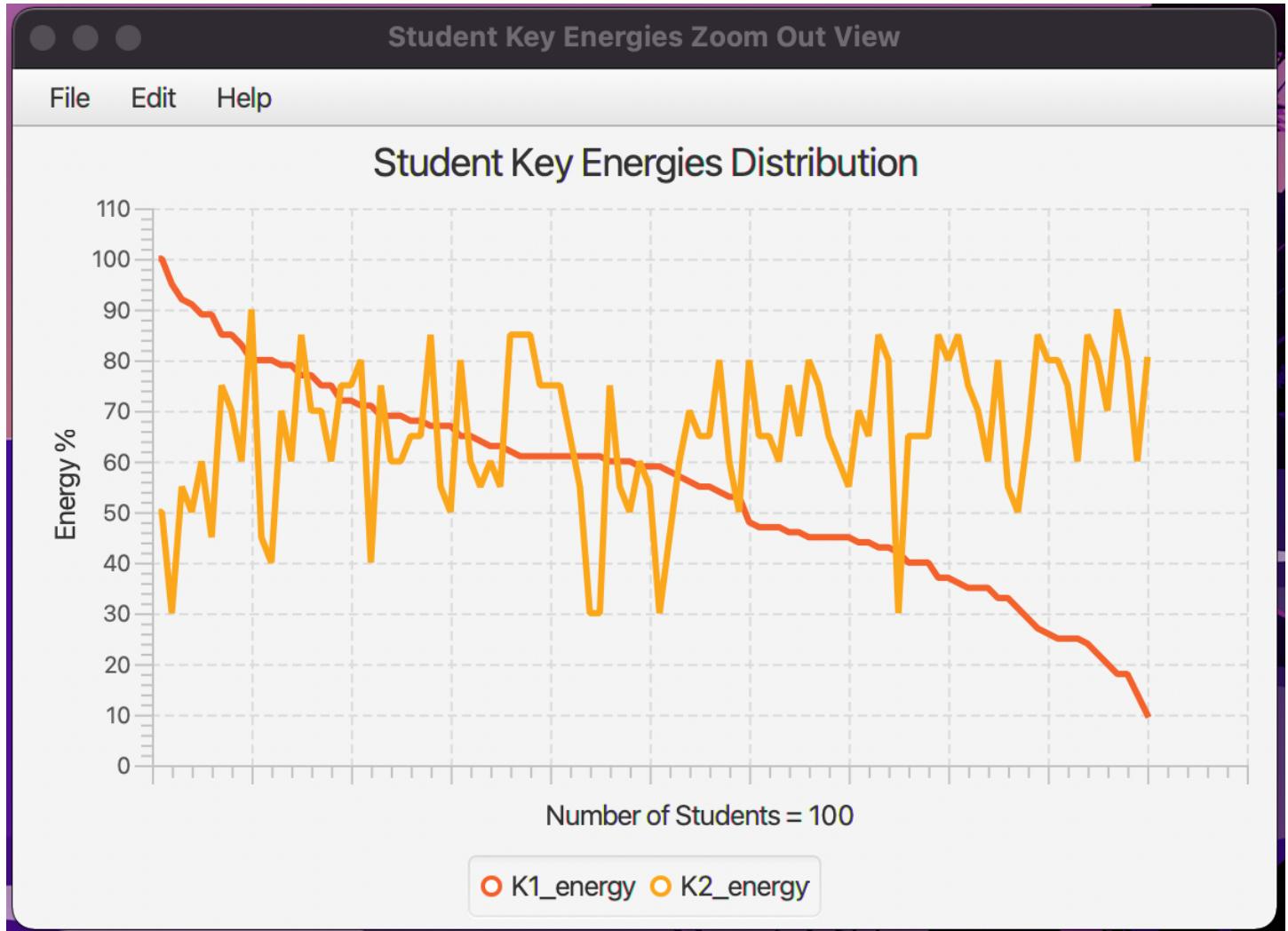
## 2. Invalid student name or ID



## 3. Duplicated student name



Student Key Energies Zoom out View (press the button after a .csv file is loaded):



# Modified Use Case Specifications

## Administrative Use Case Specification 1 (Updated)

Use case	User validation
Brief description	This use case identifies the user's privilege, and decide rather the system should display the buttons for performing extra actions.
Participating actors	TeachingTeamMember, Student
Preconditions	The system has a pre-defined passcode.
Flow of events	<p>Basic flow:</p> <ol style="list-style-type: none"><li>1. The use case begins when the user get onto the system.</li><li>2. <b>The system creates a pop-up window that allows the users to select a user-level.</b></li><li>3. The system listens to the user input.</li><li>4. {Select User-level}</li><li>5. The use case ends.</li></ol> <p>Alternative flow:</p> <p>A1. "Student" user-level selected. At {Select User-level} if the "Student" user-level is selected, 1. The system recognizes the user as a non-Teaching Team Member and hides the buttons for performing the extra actions which include read-chart, upload csv. file, and run algorithm. 2. The use case ends.</p> <p>A2. "TA" user-level selected. At {Select User-level} if the "TA" user-level is selected, 1. {Enter passcode} 2. The system Creates a pop-up window that allows the users to enter a password. 3. The system listens to the user input. 4. {Process passcode} 5. The use case ends.</p> <p>A3. Correct passcode At {Process passcode} if the user input is equal to the predefined passcode, 1. The system recognizes the user as a Teaching Team Member and shows the buttons for performing the extra actions which include read-chart, upload csv. file, and run algorithm. 2. The use case ends.</p> <p>A4. Incorrect passcode At {Process passcode} if the user input is not equal to the predefined passcode, 1. The system creates a pop-up window to inform the user that the passcode is incorrect. 2. The flow of events is resumed at {Enter passcode}.</p>
Postcondition	if selected the "TA" user-level and entered the passcode correctly, the system shows the buttons for performing the extra actions, else if selected the "Student" user-level, the system hides the buttons for performing the extra actions.

miro

# INPUT Use Case Specification 1 (**Updated**)

Use case	Upload student information.
Brief description	The system receives the csv. file containing all the student's information. Then it structures the data.
Participating actors	Teaching Team Member
Preconditions	<p>The system shows the corresponding button for this action.</p> <p>The data in csv. has all the student information.</p> <p>The input data is in the correct format.</p>
Flow of events	<p>Basic flow:</p> <ol style="list-style-type: none"> <li>1. The use case begins when the UPLOAD button is pressed.</li> <li>2. The system retrieves the user entered csv. file's path relative to the project's root folder + file name from the corresponding text field.</li> <li>{Find csv. file in user specified directory}</li> <li>3. The system loads the csv. file from the given path.</li> <li>4. The system receives the student information.</li> <li>5. The system represents each student information with a self-defined data structure and puts all the records in a container.</li> <li>6. The system responds 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 the overall K1 and K2 Energy.</li> <li>7. The use case ends.</li> </ol> <p>Alternative flow:</p> <p>A1. Incorrect directory</p> <p>At {Find csv. file in user specified directory} if there does not exist a csv. file with the inputted file name.</p> <ol style="list-style-type: none"> <li>1. The system displays error message to tell the user type again.</li> <li>2. The use case ends.</li> </ol>
Postconditions	The correct csv. file is uploaded and each of the student information is correctly stored in an internal data structure.

miro