TEAM PROJECT – REQUIREMENT SPECIFICATION

1.1 TEAM PROJECT

For COMP3111 course you may spend half of you time in learning various software platform and software tools, and the remaining time would be focusing on how you use the software tools with a series of documentation to "Manage a Project", "Develop a good Software Product". This comes up with a challenge: How to optimize the team power to achieve best performance in the Team Project?

Team Formation: A team project should have a standard team size of three members. Teammates might have their distinct capabilities in distinct areas to commit the team project e.g.:

[K1] Technical background: with strong skill sets & confidence in use of software tools;

[K2] Managerial Background: with strong conceptual mind sets in project management & documentation;

[K3] Other mindsets: Teammate who is lacking of confidence in either or both [K1] & [K2], but,

i) He/she is creative and willing to aggressively participating in team works, or

ii) He/she is willing to take more workloads that make contribution to the team.

1.1.1 TEAMING SUGGESTIONS

i) The best combination of forming a project team is [K1+K2+Any].

ii) A project team must have at least 1 x [K1] member whose K1 skill should be on or above standard requirement.

iii) The team power balance is focused on [K1]+[K2] (55%+45%). Project team combination with Strength[K1+K1+K1] is highly not recommended. This is an imbalanced arrangement that your team may be difficult to achieve the best performance of a team project.

iv) The Self-Team-Up strategy: Students are felt free to team up your own project team, and, submit your [Project Team Registration Form] (Appendix [A1.1]) in Excel format to TA office on or before 7th October, 2022. The Team up List template will be published in Canvas under Module -> Team_Project_F2022.

v) Student who was not invited by any Self-Temp-Up group may seek help from Teaching Assistant (TA). But he/she are required to fill in a questionnaire form (Appendix [A1.2]) providing data about your skill level of K1, K2, K3 in order to allow a fair matching.

vi) If a Self-Team-Up group is failed to form a team size of 3, i.e. 2. The team may also seek help from TA. But, the existing 2 members are also required to fill in the aforesaid questionnaire form (Appendix [A1.2]) for skill matching with the vacancy.

vii) There existed possibility the Maximum Team Size is 4 with the following conditions:
Remainder[X] = Total_number of_students % 3, where [X] could be in [0,1,2], then we need to append these 1 to 2 student(s) into any group of 3. If the team size is 4, the team may require 8% higher workloads or quality than standard requirement.

viii) Each project team should recommend a team leader to take in charge of the team project. The team leader is responsible to manage the project schedule & people, solving problems during any phases of the design, development & implementation cycles. He/she will be the representative of the team to consult instructor or TAs, and, presenting project outcome to the grading panel.

1.1.2 PROJECT THEME and Detail Requirement Specification

The COMP3111 F2022 Project Theme is [Automatic Teaming Up], Project ID: "**ATU**". The detail requirement specification of the team project is listed on **Appendix A2.0.** The ATU system is simply divided into three tasks:

A) INPUT:

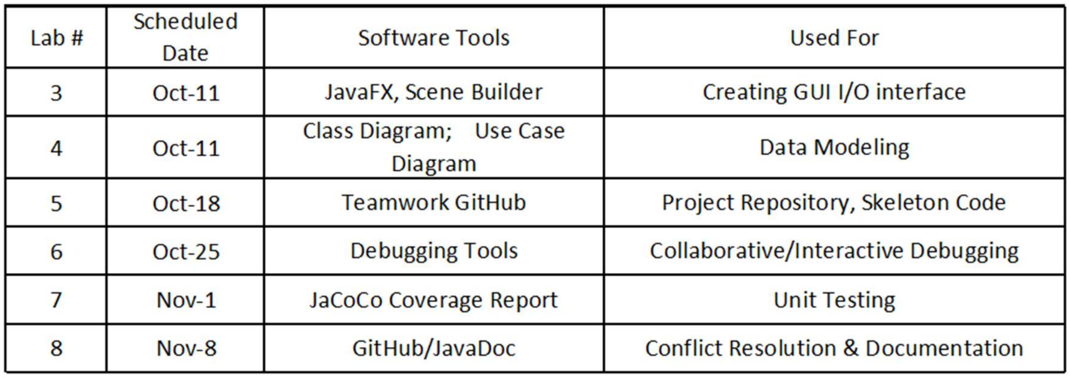
B) PROCESS:

C) OUTPUT:

There's an upgraded version of the project requirement for aggressive learners who choose to take challenging tasks to achieve highest performance of their works (**See Appendix A2.3**). Choosing these challenging tasks will made significant impacts to his/her team's total performance from evaluation point of "231" to "234" in 1.1.5 "Components of Project Assessment".

1.1.3 SELECTION OF SOFTWARE TOOLS & SET UP SCHEDULE

We expect students can utilize the software tools they have learned and practiced from lectures & labs respectively. Any special requests of exceptional software use in the Team Project must report to TA office for prior approval. Below table is the scheduled timeline for project team to get the software setup ready before you go further steps to develop the software. The setup schedule is correlated to the lab sessions, project team leader should have to monitor its progression status according to below scheduled date. TA office will counter check & grade on two milestone activities, but not individual software setup jobs.



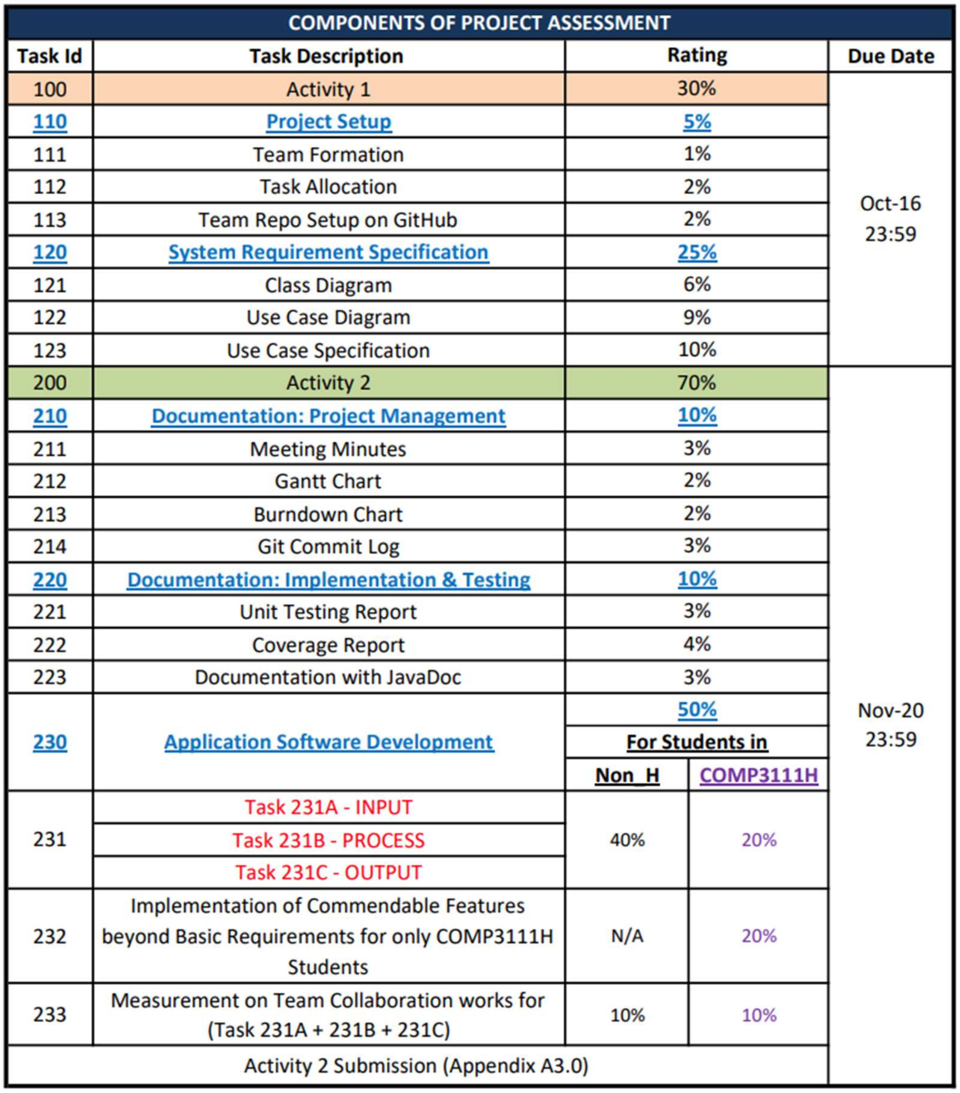
Lab #	Scheduled Date	Software Tools	Used For
3	Oct-11	JavaFX, Scene Builder	Creating GUI I/O interface
4	Oct-11	Class Diagram; Use Case Diagram	Data Modeling
5	Oct-18	Teamwork GitHub	Project Repository, Skeleton Code
6	Oct-25	Debugging Tools	Collaborative/Interactive Debugging
7	Nov-1	JaCoCo Coverage Report	Unit Testing
8	Nov-8	GitHub/JavaDoc	Conflict Resolution & Documentation

1.1.4 SYSTEM DEVELOPMENT PLATFORM

The requirement of System Development Platform is required to follow below standard, any exceptional use of platform components should be reported to TA office for prior approval:

- Programming Environment: Eclipse IDE for Java Developers
- Java SE Development Kit 15
- JavaFX as GUI development tool
- The project must use Gradle to manage (integrated in Eclipse)
- JUnit as testing suite (integrated in Eclipse)
- Jacoco as the test coverage measurement (integrated in Eclipse)
- A private GitHub repository for source control: In your private repository setting page, add your team members and all TAs as collaborators.

1.1.5 COMPONENTS OF PROJECT ASSESSMENT



COMPONENTS OF PROJECT ASSESSMENT

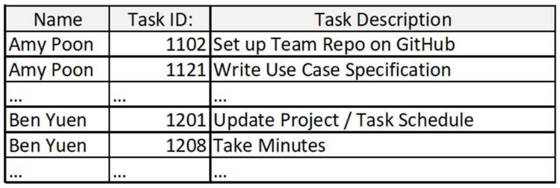
Task Id	Task Description	Rating		Due Date
100	Activity 1	30%		Oct-16 23:59
110	Project Setup	5%		
111	Team Formation	1%		
112	Task Allocation	2%		
113	Team Repo Setup on GitHub	2%		
120	System Requirement Specification	25%		
121	Class Diagram	6%		
122	Use Case Diagram	9%		
123	Use Case Specification	10%		
200	Activity 2	70%		Nov-20 23:59
210	Documentation: Project Management	10%		
211	Meeting Minutes	3%		
212	Gantt Chart	2%		
213	Burndown Chart	2%		
214	Git Commit Log	3%		
220	Documentation: Implementation & Testing	10%		
221	Unit Testing Report	3%		
222	Coverage Report	4%		
223	Documentation with JavaDoc	3%		
230	Application Software Development	50%		
		For Students in		
		Non_H	COMP3111H	
231	Task 231A - INPUT Task 231B - PROCESS Task 231C - OUTPUT	40%	20%	
232	Implementation of Commendable Features beyond Basic Requirements for only COMP3111H Students	N/A	20%	
233	Measurement on Team Collaboration works for (Task 231A + 231B + 231C)	10%	10%	
Activity 2 Submission (Appendix A3.0)				

Notation:

110 Project Setup

111 Team Formation: Submission of Team Registration Form

112 Task Allocation: Provide check list for each teammate, what detail tasks they're going to execute



Name	Task ID:	Task Description
Amy Poon	1102	Set up Team Repo on GitHub
Amy Poon	1121	Write Use Case Specification
…	…	…
Ben Yuen	1201	Update Project / Task Schedule
Ben Yuen	1208	Take Minutes
…	…	…

113 Team Repo setup on GitHub:

i) private repo with all TAs added as collaborators;

ii) readme.md file with corresponding information of all team members (name, email Id, branch Id, etc.)

iii) readme.md file with information on the assignment of tasks and links to the related documentation (class diagram, use case diagram, use-case specifications, etc.)

120 System Requirement Specification:

Class Diagram, Use Case Diagram, Use Case Specification for three major tasks (**Appendix A2.2**) of

i) INPUT

ii) PROCESS

iii) OUTPUT

210 Documentation: Project Management:

211 At least 3 Meeting Minutes

212 Gantt Chart for showing project timeline and work breakdown (Task List)

213 Burndown Chart for showing the Works In Progress.

214 Representative Git commit log on GitHub: At least three "non-trivial" commits and one "non-trivial" pull request (per team member)

220 Documentation: Implementation & Testing

221 Report on the unit testing for the implemented tasks (should 100% pass)

222 Report on the coverage test (>65% branch coverage)

223 Documentation on the implemented tasks using JavaDoc

230 Software Development

231 INPUT / PROCESS / OUTPUT (see **Appendix A2.2**)

232 Implementation of Commendable Features beyond standard requirements (see **Appendix A2.3**) is mandatory for each COMP3111H students. The essential user requirements have been clearly stated **in Appendix A2.2**, which mostly cover the aspects of functionality and usability of a software system. However, innovative and competent software developers should always aim at delivering higher quality of software product and services, that users would be getting more excited about. That would be the WOW Factor in software product development and software engineering. Detail requirement specification of these WOW factors are listed on

Appendix A2.3. In other words, unlike standard requirement, COMP3111H students have to think more and analyze the problem statement from client, then design a system to solve the problem(s).

233 Task 233 is a measurement of final outcome of the application software product. The judgement of result is theoretically depending on how members in a project team work collaboratively with each other. Let's takes an example: if 3 students got their individual marks = 6, 5, 7 for 231A, 231B, 231C respectively, as a result the team average mark is 6. Under this situation, if we can see they have done a good communication & project management works according to their system design document, even they've made some data errors in the task of 231B and hence affected the unideal output result of 231C, we would consider to markup the final average grade from 6 up to 7 or 8. However, if free rider existed in team and the team leader didn't report promptly & take appropriate reaction, a mark down deduction would be introduced.

1.1.6 GRADING POLICY

A) Grading Policy

1) On Table 1.1.5 [Components of Project Assessment], ALL TASKS except task ID 231 & 232 are measured by team performance, i.e. one marking result for all teammates, there'll have no individual marking points.

2) For Task ID 231 & 232 there'll have three breakdown components of 231A (INPUT), 231B (PROCESS) & 231C (OUTPUT), three break down tasks should be fully distributed to three teammates respectively. If the team size is 4, the 4th teammate can choose one of the breakdown tasks as repeated task in the team. Any copying works of repeated tasks within a team or cross team is strictly prohibited. All works of Task ID 231A, 231B & 231C in 231 & 232 are measured by individual performance.

3) Late submission (within 5 days after due date) with penalty is enable for Activity 1 only.

B) Notes & Marking Scheme for Activity 1

You should work with your teammates collaboratively to produce the diagrams and documentation according to the project description. At this stage, there is no need to include any "commendable features" beyond the basic requirements.

111 Team Formation

- 1 mark : Successfully submitted the [Team Registration Form] with completed information on or before the due date.
- 0 mark : otherwise

112 Task Allocation

- 2 marks : Clearly states detail tasks & description with sufficient information provided.
- 1 mark : Unclearly states detail tasks & description without sufficient information provided.
- 0 mark : No submission
- Late submission is accepted, 50% of penalty will be incurred on the final result of works.

113 Team Repo Setup on GitHub

- 2 marks: Successfully setup the Team Repo in GitHub for team & TA's sharing & work properly.
- 0 mark: Failed to build the GitHub for team sharing.
- Late submission is accepted, 50% of penalty will be incurred on the final result of works.

121 Class Diagram (max: 6): Present your data model for the overall system as a Class Diagram. Only the essential classes should be included. Maximum mark is 6. Late submission is accepted, 30% of penalty will be incurred on the final result of works.

122 Use Case Diagram (max: 9): There should be one use case diagram for the overall system which consists of the 3 major tasks 231A, 231B, & 231C. No generalization should be used. Maximum mark is 9. Late submission is accepted, 30% of penalty will be incurred on the final result of works.

123 Use Case Specification (max: 10): You should provide a brief description with the corresponding use case diagram and elaborate on the basic flow as well as the alternate flow. Maximum mark is 10. Late submission is accepted, 30% of penalty will be incurred on the final result of works.

By design, there's no single correct answer or perfect solution for Activity 1. Assessment is based on correctness and completeness (information included must be accurate and appropriate).

The teaching team will also check your team project repo on GitHub to see if there's a proper README.md file showing the Team ID, members' personal particulars, and individual task assignment.

C) Notes & Marking Scheme for Activity 2

211 Meeting Minutes (max: 3)

- 3 marks : for providing at least 3 meeting minutes.
- 1.5 marks : for providing less than 3 meeting minutes.
- 0 mark : for providing no meeting minutes.

212 & 213 Gantt Chart & Burndown Chart (max: 4)

- 4 marks : for providing Gantt chart (showing project timeline & work breakdown) and Burndown chart (showing the work progress) with sufficient information.
- 2 marks : for providing Gantt Chart (showing project timeline & work breakdown) and Burndown chart (showing the work progress) with insufficient information.
- 0 marks : for not providing Gantt Chart (showing project timeline & work breakdown) nor Burndown chart (showing the work progress).

214 Git Commit Log (max: 3)

- 3 marks : for showing at least 3 "non-trivial" commits and 3 "non-trivial" pull requests (or, at least one from each team member).
- 1.5 marks : for showing at least 3 "non-trivial" commits and 3 "non-trivial" pull requests (or, at less than one from each team member).
- 0 mark : for not showing any commits nor pull.

221 Unit Testing Report (max: 3)

- 3 marks : for providing report on the unit testing for the implemented tasks (100% pass).
- 1.5 marks : for providing report on the unit testing for implemented tasks (less than 100% pass).
- 0 mark : for not providing report on the unit testing for implemented tasks.

222 Coverage Report (max: 4)

- 4 marks : for providing report on the coverage test for the implemented tasks (>65% branch coverage)

- 2 marks : for providing report on the coverage test for the implemented tasks (<=65% branch coverage).
- 0 mark : for not providing report on the coverage test for the implemented tasks.

223 Javadoc (max: 3)

- 3 marks : for providing sufficient documentation on the implemented tasks using Javadoc.
- 1.5 marks : for providing insufficient documentation on the implemented tasks using Javadoc.
- 0 mark : for not providing documentation on the implemented tasks using Javadoc.

231A INPUT (max: 40/20 for Non_H/COMP3111H)

231A-1 Push the csv file into GitHub (max: 4/2)

- 4/2 marks : Successfully push the "StuPi.csv" file into GitHub & convert into a dataset for team sharing.
- 0 mark : otherwise

231A-2 Skeleton Codebase in java to load the data (max: 8/4)

- 8/4 marks : Successfully derived a skeleton codebase in Java to load the data, and it can be ready to show some basic statistics.
- 0 mark : otherwise

231A-3 Display the statistics in the format of a table on a GUI (max: 28/14)

- 28/14 marks : Successfully showed the statistics in 231A-2 in the format of a table on GUI by JavaFX and Scene Builder. Partial marks from 16 to full mark 22 will be given depending on the accuracy of statistical data & the design layout of the GUI interface.
- 20/10 marks : for partially completed the task according to the requirement specification and can show sufficient efforts you have done in the project.
- 12/6 marks : for partially completed the task according to the requirement specification but cannot show sufficient contributions you have done in the project.
- 0 mark : for zero submission or otherwise.

231B PROCESS (max: 40/20 for Non_H/COMP3111H)

- 40/20 marks : Successfully built the team dataset according to the requirement specification and your design in data modeling. Partial marks from 21 to full mark 30 will be given depending on the data integrity and precision and the robustness of the ATU Engine.
- 28/14 marks : for partially completed the task according to the requirement specification and can show significant efforts you have done in the project.
- 18/9 marks : for partially completed the task according to the requirement specification but can not show sufficient contributions you have done in the project.
- 0 mark : for zero submission or otherwise.

231C OUTPUT (max: 40/20 for Non_H/COMP3111H)

- 40/20 marks : Successfully showed ONE online student inquiry service, and, plotted ONE 2D Line Chart to show the statistical curve of either student average or team average in class. Partial marks from 21 to full mark 30 will be given depending on the accuracy and feasibility of the output.
- 28/14 marks : for partially completed the task according to the requirement specification and can show significant efforts you have done in the project.
- 18/9 marks : for partially completed the task according to the requirement specification but cannot show sufficient contributions you have done in the project.
- 0 mark : for zero submission or otherwise.

232 The WOW Factors: Implementation of Commendable Features (max: 20)

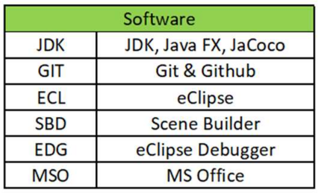
- 12 – 20 Marks : Successfully completed Appendix A2.3-1 [INPUT] – "Creation of Simulation" or
Successfully completed Appendix A2.3-2 [PROCESS] – OUTPUT ENHANCEMENT or
Successfully completed Appendix A2.3-2 [OUTPUT] – PROCESS ENHANCEMENT;
- 5 – 12 Marks : Partially completed Appendix A2.3-2 [INPUT] – "Creation of Simulation" or
Partially completed Appendix A2.3-2 [PROCESS] – OUTPUT ENHANCEMENT or
Partially completed Appendix A2.3-2 [OUTPUT] – PROCESS ENHANCEMENT;
- 0 Marks : Failed implement or No submission
- There's no generalization marking scheme for commendable features. The judgement of marking result is proposed by a panel of TAs, and, to be final reviewed by Prof. Leung for approval.

233 Measurement of Team Collaboration Works on Task 231A + 231B +231C (max: 10)

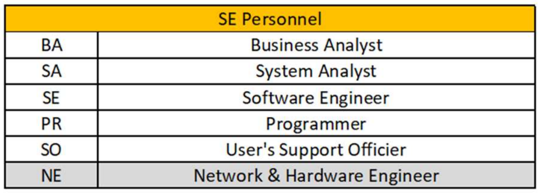
- 5-10 marks : Depends on the team participation & collaborative works.
- 0 mark : All Tasks 231A & 231B &231C were no submission or failed to complete.

1.1.7 RESOURCES ALLOCATION IN THE PROJECT

1.1.5A Resources of Software & Personnel in Software Engineering Team

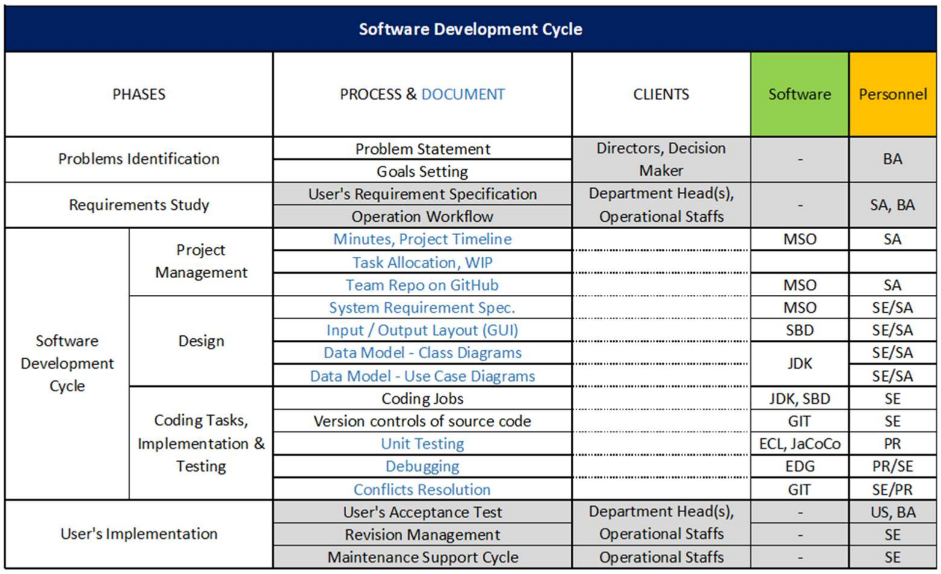


Software	
JDK	JDK, Java FX, JaCoco
GIT	Git & Github
ECL	eClipse
SBD	Scene Builder
EDG	eClipse Debugger
MSO	MS Office



SE Personnel	
BA	Business Analyst
SA	System Analyst
SE	Software Engineer
PR	Programmer
SO	User's Support Officier
NE	Network & Hardware Engineer

As refer to Appendix A1.3 "Personnel involved in Software Development, it introduces the software engineering personnel participating in a software development cycle to play their different roles dealing with their clients, their end users, and their cooperative teammates to commit a project.



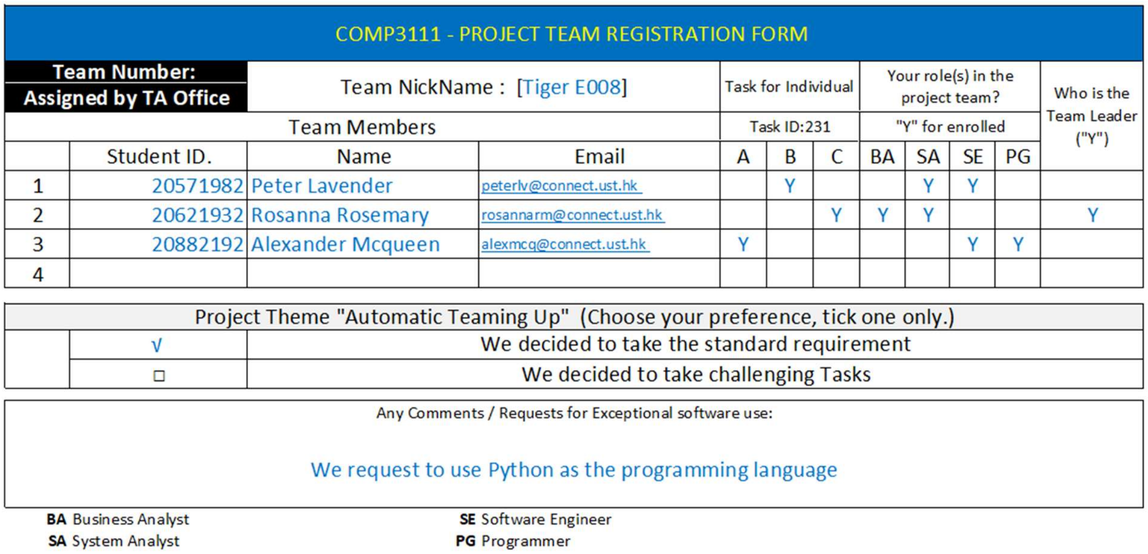
Software Development Cycle

PHASES		PROCESS & DOCUMENT	CLIENTS	Software	Personnel
Problems Identification		Problem Statement	Directors, Decision Maker	-	BA
		Goals Setting			
Requirements Study		User's Requirement Specification	Department Head(s), Operational Staffs	-	SA, BA
		Operation Workflow			
Software Development Cycle	Project Management	Minutes, Project Timeline		MSO	SA
		Task Allocation, WIP			
		Team Repo on GitHub		MSO	SA
	Design	System Requirement Spec.		MSO	SE/SA
		Input / Output Layout (GUI)		SBD	SE/SA
		Data Model - Class Diagrams		JDK	SE/SA
		Data Model - Use Case Diagrams			SE/SA
	Coding Tasks, Implementation & Testing	Coding Jobs		JDK, SBD	SE
		Version controls of source code		GIT	SE
		Unit Testing		ECL, JaCoCo	PR
		Debugging		EDG	PR/SE
		Conflicts Resolution		GIT	SE/PR
User's Implementation		User's Acceptance Test	Department Head(s), Operational Staffs	-	US, BA
		Revision Management		-	SE
		Maintenance Support Cycle	Operational Staffs	-	SE

For all teammates in the team project, everyone should clearly define what role(s) they are going to play. A software product consists of 6 key steps: To understand the problems, to find a solution, to design the solution, to manage the project, to execute the jobs, to launch the products and providing supports to end users. Teammates have to fill in their roles in the [Project Team Registration Form] (Appendix [A1.1]). Those processes & documents marked with grey in color are ignored in this project. Problem Statement and Goal Setting in the row of “Problems Identification” are given in Appendix A2.1, project team may copy and modify it as part of your project document.

APPENDICES

A1.1 PROJECT TEAM REGISTRATION FORM



COMP3111 - PROJECT TEAM REGISTRATION FORM											
Team Number: Assigned by TA Office		Team NickName : [Tiger E008]		Task for Individual			Your role(s) in the project team?				Who is the Team Leader ("Y")
Team Members				Task ID:231			"Y" for enrolled				
	Student ID.	Name	Email	A	B	C	BA	SA	SE	PG	
1	20571982	Peter Lavender	peterlv@connect.ust.hk		Y			Y	Y		
2	20621932	Rosanna Rosemary	rosannarm@connect.ust.hk			Y	Y	Y			Y
3	20882192	Alexander Mcqueen	alexmcq@connect.ust.hk	Y					Y	Y	
4											

Project Theme "Automatic Teaming Up" (Choose your preference, tick one only.)		
	√	We decided to take the standard requirement
	□	We decided to take challenging Tasks

Any Comments / Requests for Exceptional software use:

We request to use Python as the programming language

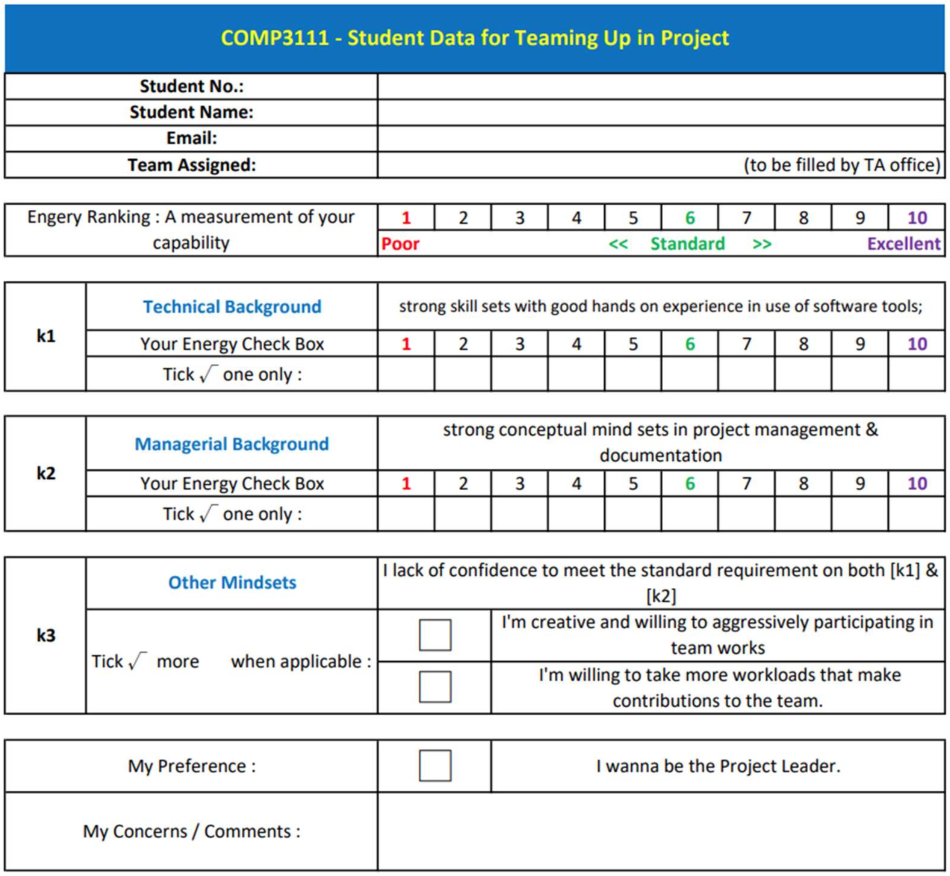
BA Business Analyst

SA System Analyst

SE Software Engineer

PG Programmer

A1.2 STUDENT DATA FOR TEAMING UP IN PROJECT



COMP3111 - Student Data for Teaming Up in Project

Student No.:	
Student Name:	
Email:	
Team Assigned:	(to be filled by TA office)

Engery Ranking : A measurement of your capability	1	2	3	4	5	6	7	8	9	10
	Poor				<<	Standard	>>			Excellent

k1	**Technical Background**	strong skill sets with good hands on experience in use of software tools;									
	Your Energy Check Box	1	2	3	4	5	6	7	8	9	10
	Tick √ one only :										

k2	**Managerial Background**	strong conceptual mind sets in project management & documentation									
	Your Energy Check Box	1	2	3	4	5	6	7	8	9	10
	Tick √ one only :										

k3	**Other Mindsets**	I lack of confidence to meet the standard requirement on both [k1] & [k2]	
	Tick √ more when applicable :	□	I'm creative and willing to aggressively participating in team works
		□	I'm willing to take more workloads that make contributions to the team.

My Preference :	□	I wanna be the Project Leader.
My Concerns / Comments :		

A1.3 PERSONNEL INVOLVED IN SOFTWARE DEVELOPMENT

Software Development is an innovation and an integration of technology and management. Demands may come from industrial/commercial activities, civil public facilities, our daily life, and research purposes etc. Precondition of demand is, in short, used to solve a real-world problem or, on the other hand, an innovative research.

Students may take this as reference in your study curve, your team does not require to appoint full list of below roles in your project, because the project is a comparatively very small scale. But, you need to clearly identify what roles listed below you're going to play in the project.

A1.3.1 **DEMAND** is coming from **Client** with below key personnel:

a) **Decision maker** of a business unit or an organization who invests in the software development. He/She raises up his/her concerns/interests/pains and asks for a solution with desired outcome.

b) Client has its **operational teams** consists of managerial and operational members that they can show you the detail of their existing problems and would be dedicated in the system implementation phases when final software product is released.

A1.3.2 **SUPPLY:** The Structure of Personnel in Software Development Team

Software Development Team could be a part of the Client's departmental unit, or a supplier of the client. It consists of below personnel:

1) **BA** – Business Analyst

   i) BA should have strong business sense with the significant domain knowhow that related to the client's business.

   ii) BA requires strong analytical skills to realize the bigger scale of a problem or request and able to decompose and identify it into multiple smaller size of problems or requests.

   iii) BA should be able to consolidate the existing operational work flow of the client, or, derive the break down components of a new demand.

   iv) BA requires the minimum technology knowledge (Here's software tools, all CS students should already have).

   v) BA is responsible to write the [User Requirement Specification] document according to client's existing work flow and the statement of target output.

   vi) BA is responsible to write the [User Acceptance Test] document before launch the software product to client.

2) **SA** – System Analyst

   i) SA should have higher (Medium) level knowledge in technology (software tools). He/she does not require to do hands on technical works e.g. hardware set up and software coding.

   ii) SA should have strong analytical skills able to convert [User Requirement Specification] and work with Software Engineer **SE** into [SYSTEM DESIGN DOCUMENT] such as {Class Diagram}, {Case Diagram}, System Architecture, Database Schema etc.

iii) SA is responsible to take the lead of project management, able to manage people, work schedule, and other resource required in tasks execution.

iv) SA is responsible to prepare [Project Management document] including the planning data of project schedule, time table, task detail, and resource allocation.

3) **SE** – Software Engineer

i) SE should have highest level knowledge in technology (software tools). He/she is responsible to take in charge of programming team to code the software according to the [SYSTEM DESIGN DOCUMENT], and to manage the debugging & unit testing jobs.

ii) SE is responsible to monitor the progressions status in project schedule according to the [Project Management Document].

4) **PG** – Programmer

i) PG should have deepest knowledge and hands on experience in job required software tools.

ii) PG is the shop floor level (production line) member to conduct the daily routine works in coding according to the predefined [SYSTEM DESIGN DOCUMENT] to produce the components of the software product.

iii) PG may also require to participate in Debugging and Unit Testing work cycle.

5) **US** User Support Officer

i) US would receive sufficient training in operating the software product.

ii) He/she is responsible to conduct the “User Acceptance Test” with Client’s operation team during the phase of implementation of the software product.

iii) He/she is responsible to backup users for all queries of daily operation problems found in running the software products, during the phase of system maintenance period.

6) **NE** – Network & Hardware Engineer

i) Responsible for overall computer hardware & data communication networking….

A2.0 PROJECT THEME:

Project ID : ATU

Project Name : Automatic Teaming Up

A2.1 Project Description

a) Prefix

A client requests a software product from a software development team to solve a "Teaming Up" problem in an academic Team Project. The client is the instructor of a course. Take this COMP3111 course as an example, we have about 200 students to form N project teams with a standard team size of 3 teammates. Students may come from mostly Computer Science Engineering (CSE) background while a small portion of students may come from Management School or other NON-CSE background. The Team Project requires two specific areas of domain expertise:

i) [K1] Strong technical background in using software tools, from cumulative experience or learning in class COMP3111.

ii) [K2] Strong conceptual background from cumulative experience or learning in class COMP3111 in the particular areas of Project Management & Documentation.

The ratio of the effort contribution for [K1] : [K2] assumes 5.5 : 4.5

b) Problem Statement

1) Course Instructor & Teaching Assistants (TA) were suffering in handling team up problems whether or not allow students to teaming up by themselves, with the consideration of next concerns;

2) Allow students to self-teaming up may not be a good approach, because

   i) A friendship group is not a good academic training, because this won't be happened in students' future real world working environment. students need to overcome the difficulties of people management especially if he/she is appointed as the team leader.

   ii) If three teammates are all outstanding in [k1] but poor in [k2], they probably won't be able to achieve the best performance of the project;

   iii) If three teammates all lack of [k1] skills, that would be a big problem for them to complete the project properly.

3) Passing the teaming up jobs to TAs

   i) Heavy workloads would be incurred to TAs after team formation;

   ii) Arguments between teammates would be occurred during grading period;

   iii) Tough to avoid errors from human judgement as a result conflicts & complains increased.

c) Goals Setting

1) The client believes if we can get significant data about students' skill profile and/or interests and/or preferences and/or personality trait etc., with a measurement of their strengths & weaknesses prior to the formation of a Project Team, a scientific software product might be able to provide a solution that helps to automatically teaming up students in Project Teams. The solution should be a fair organizer to optimize the grouping

arrangement result with the balance of strength & weakness matching algorithm. The client has given a name to the solution called “Automatic Teaming Up”.

2) Assumption: We may assume that students would provide their skill information before team’s formation.

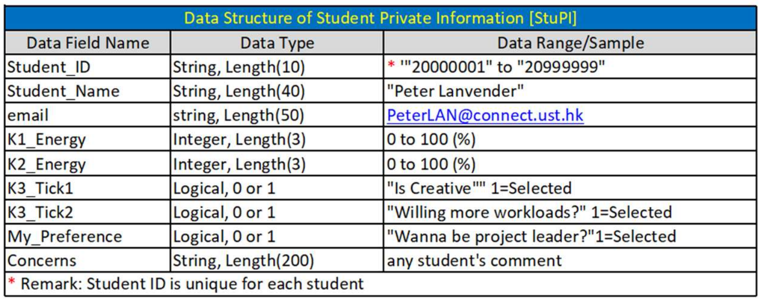
3) Making Teaching Team Relax: The client believes that smart student always making complains to weak teammates might be minimized or more ease to handle, because, we got comparative student data from team registration and the actual contribution when project submitted. Why? Think about any student if he/she provide true/false declaration during registration.

4) Making Student Happy: Only a small group of smart students they lose the benefit from self-teaming up to get the highest academic performance. Most of the team would receive the fair deal competition with each other’s, since the software product maintain the average power of skills among teams. The challenging values are: who can manage the project better depending on the devotion of every teammate.

A2.2 Detail Project Requirement Specification (Standard Functions for non-H students)

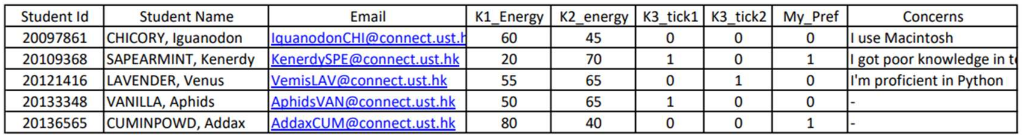
1) INPUT (Task ID: 231A)

i) We assume students will provide his personal data for you to run the ATU program as refer to Appendix A1.2. Data Structure is therefore look like this:



Data Structure of Student Private Information [StuPI]		
Data Field Name	Data Type	Data Range/Sample
Student_ID	String, Length(10)	* '"20000001" to "20999999"
Student_Name	String, Length(40)	"Peter Lanvender"
email	string, Length(50)	PeterLAN@connect.ust.hk
K1_Energy	Integer, Length(3)	0 to 100 (%)
K2_Energy	Integer, Length(3)	0 to 100 (%)
K3_Tick1	Logical, 0 or 1	"Is Creative"" 1=Selected
K3_Tick2	Logical, 0 or 1	"Willing more workloads?" 1=Selected
My_Preference	Logical, 0 or 1	"Wanna be project leader?"1=Selected
Concerns	String, Length(200)	any student's comment
* Remark: Student ID is unique for each student		

Sample of data



Student Id	Student Name	Email	K1_Energy	K2_energy	K3_tick1	K3_tick2	My_Pref	Concerns
20097861	CHICORY, Iguanodon	IguanodonCHI@connect.ust.h	60	45	0	0	0	I use Macintosh
20109368	SAPEARMINT, Kenerdy	KenerdySPE@connect.ust.hk	20	70	1	0	1	I got poor knowledge in t
20121416	LAVENDER, Venus	VemisLAV@connect.ust.hk	55	65	0	1	0	I'm proficient in Python
20133348	VANILLA, Aphids	AphidsVAN@connect.ust.hk	50	65	1	0	0	-
20136565	CUMINPOWD, Addax	AddaxCUM@connect.ust.hk	80	40	0	0	1	-

You’ll be given a set of data “StuPi” in csv file format with a record size of 100 student records. The project team requires to

a) Push the csv file into GitHub for team sharing;

b) Student who takes this task has to think about how to represent a student or a list of students in the program.

c) Derive a skeleton codebase in Java to load the data, and show some basic statistics, e.g.

*	Total Number of Students	: 100
*	K1_Energy(Average, Min, Max)	: (59.8, 10, 80)
*	K2_Energy(Average, Min, Max)	: (62.3, 40, 85)
*	K3_Tick1 = 1	: 12

*	K3_Tick2 = 1	: 3
*	My_Preference = 1	: 19

d) Display the statistics in (b) in the format of a table on a GUI. You are required to develop the GUI with JavaFX and SceneBuilder taught in Lab 3 (you are recommended to use JavaFX's TableView (https://openjfx.io/javadoc/17/javafx.controls/javafx/scene/control/TableView.html) API to construct the GUI). Below shows some examples of TableView (here is a tutorial (https://docs.oracle.com/javafx/2/ui_controls/table-view.htm) to reproduce the example). Note that you are free to design the format of the table, as long as all the statistics in (b) are shown.

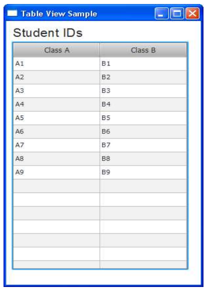


Table View Sample

Student IDs

Class A	Class B
A1	B1
A2	B2
A3	B3
A4	B4
A5	B5
A6	B6
A7	B7
A8	B8
A9	B9

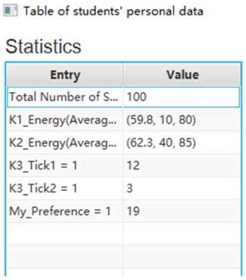


Table of students' personal data

Statistics

Entry	Value
Total Number of S...	100
K1_Energy(Averag...	(59.8, 10, 80)
K2_Energy(Averag...	(62.3, 40, 85)
K3_Tick1 = 1	12
K3_Tick2 = 1	3
My_Preference = 1	19

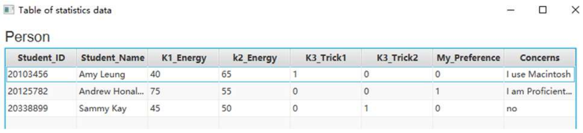


Table of statistics data

Person

Student_ID	Student_Name	K1_Energy	k2_Energy	K3_Trick1	K3_Trick2	My_Preference	Concerns
20103456	Amy Leung	40	65	1	0	0	I use Macintosh
20125782	Andrew Honal...	75	55	0	0	1	I am Proficient...
20338899	Sammy Kay	45	50	0	1	0	no

2) PROCESS (TASK ID: 231B)

i) The project team has to design a program to distribute those 100 students into 33 project teams. The task name of program in this PROCESS we called "ATU Engine".

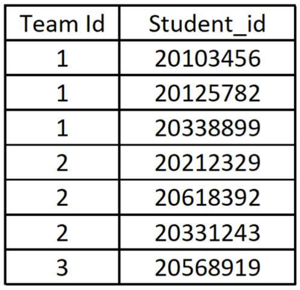
ii) Constrains to construct the ATU Engine are listed below:

- Each Team must have at least 3 at most 4 students;
- Each Team should have at least one student whose K1_energy >= average_k1_energy (Top Priority);
- Design an algorithm to maintain the balance of K1_Energy & K2_Energy for each team (2nd Priority) in order to minimize the standard deviation of between teams;

iii) Sample algorithm of program logics in Pseudo Code to implement the 1st three items of constrains.

```
{
Team_Size = integer(total_record(StuPi))
Function Select_K1_member()
    Select top(Team_Size) student_id from StuPi order by K1_Energy in descending order
    Into K1_List
Function Select_K2_member()
    Select top(Team_Size) student_id from Stupi where student_id not in K1_List order by
    K2_Energy in ascending order into K2_List
Function Select_K3_member()
    Select student_id from Stupi where student_id not in (K2_List & K2_List) into K3_List without
    Sorting sequency
Function Create_Team
    for i = 1 to Team_Size
        append record in ATU_Team( i, “T-“+string(i, 4, ‘0’), “”, 0)
Create dataset ATU_Team(team_id, team_name, student_id, is_team_Leader)
for j = 1 to Team_Size
    select j-th records from K1_List
        append record in ATU_Team( i, “T-“+string(i, 4, ‘0’), K1_List.student_id, 0)
    select j-th records from K2_List
        append record in ATU_Team( i, “T-“+string(i, 4, ‘0’), K2_List.student_id, 0)
    select j-th records from K3_List
        append record in ATU_Team( i, “T-“+string(i, 4, ‘0’), K3_List.student_id, 0)
Select the remainder 1 student record, append into any team, how about if the remainder is 2?
}
```

After the ATU Engine is executed, you’ll got a new dataset with 33 records like this:

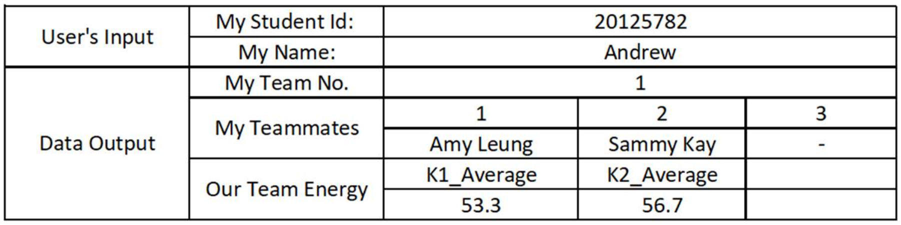


Team Id	Student_id
1	20103456
1	20125782
1	20338899
2	20212329
2	20618392
2	20331243
3	20568919

3) **OUTPUT (Task ID: 231C)**

When PROCESS is done, the Project Team is required to provide below outputs

i) An online student inquiry service, by giving a student id or name:



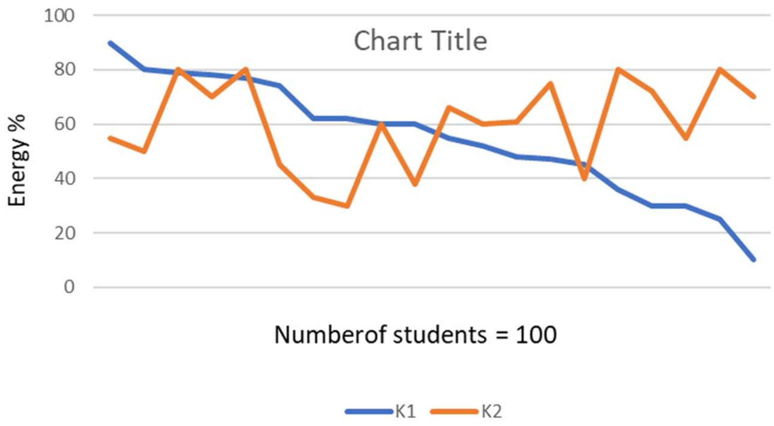
User's Input	My Student Id:	20125782		
	My Name:	Andrew		
Data Output	My Team No.	1		
	My Teammates	1	2	3
		Amy Leung	Sammy Kay	-
	Our Team Energy	K1_Average	K2_Average	
		53.3	56.7	

ii) 2 basic statistics 2D-Line Chart (Student who takes this task should choose only one of the following 2D-Line Chart whether to plot the chart with all individual students’ comparison before running the ATU engine or the team average in class after running the ATU engine)

a) Student Key Energies Zoom Out View:

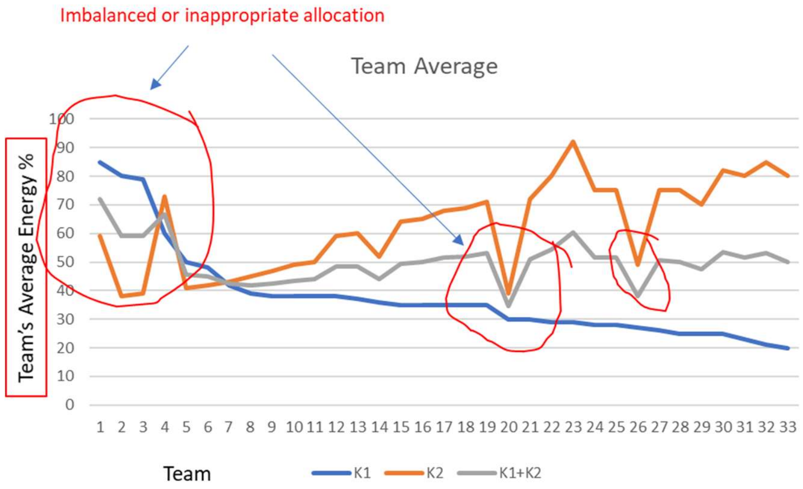
- Select all records (100) from StuPi, sorted by K1_Energy in descending order;

- Presenting from left top (the highest energy) down to right bottom (the lowest energy) of the line chart.



b) Teaming Up performance by Team Average

- Select team average of K1 & K2 from all 33 teams;
- Also sorted by descending order of K1 to check the result;
- If the algorithm of the program logics is not ideal enough, take below chart as example, the global team average of (K1 is 37.9%%, K2 is 62.5%, K1+K2 is 50.2%), however, the curve marked with a red circle are imbalanced or inappropriate allocation.



A2.3 Requirement Specification for Enhanced Functions (Mandatory for COMP3111H students)

Aggressive Project Team may choose one or as many below enhanced tasks as they can.

1) INPUT – A creation of simulation

Instead of providing a dataset "StuPi.csv", Project team may have to write their own program to generate testing data, or in other words: Client requests the ATU System to provide "Data Auto Generation" function for the proof of concepts with the following requirement (Input Parameters to be keyed in by the client)

Generating a set of Sample Data for testing			
Definition		Input	
1	Number of students = (Sample Data Size)	999	<= 900
2	Average_K1_Energy per class	99	%
3	Average_K2_Energy per class	99	%
4	Probability of K3_Tick1 is true	99	%
5	probability of K3_Tick2 is true	99	%
6	probability of My_Preference is true	99	%

e.g.

- If user keys in line 1 Number of students = 100, the system should able to generate 100 students records with unique Student_ID;
- If user keys in line 2 Average_K1_Energy = 67%, the final result of those 100 records should maintain an average_k1_Energy per class within the range of 5% +/- 67% (i.e. eventually result in the range of 62%-72%), and so as line 3 ;
- If user keys in line 4 Probability of K3_Tick1 = 12%, then there're eventually 12 (round up to integer) students tick on the check box of K3_Tick1, and so as line 5 & line 6
- Hint: The Data Generation may use **RANDOM** algorithm under the user's target boundary.
- So, what techniques we can use to generate student name? Here are two matrices:

First Name Matrix	Last Name Matrix
Sunflower	Lemongrass
Rose	Rosemary
Cat	Lavender
Tiger	Thyme
Lion	Basil
Wolf	Dill

The two matrices have record size of 6, the combination of two matrices mixed could generate 36 student names. if you need a sample size of 625 student records, you have to maintain 25 records per each matrix.

2) PROCESS

Aggressive Learner may have to think more about the given constrains and Programming Logics, whether or not there are areas need to be improved such as

i) K3_Tick1 and K3_Tick2 are weak force signal that we need to take care with them, but you need to think: how?

ii) Consider if a teammate who select his/her preference want to be the team leader, while other teammates don't intend to do so, he/she will then be assigned as "Team Leader" of the team?

iii) Each team will have their team average in Average(K1_Energy), Average(K2_Energy), and Average(K1_Energy+K2_Energy). How to minimize the standard Deviation between teams.

iv) What else of data source can help to enhance or improve the ATU Engine.

v) If the project team can derive more ideas that make sense, and/or more creative, more fairness in arrangement, feel free to add or modify the constraints our scenario used.

3) OUTPUT

i) Is it a Perfect Match Engine? When the ATU engine is executed, the team up data is released, what else of data or graph can help you to monitor the programming logics if it is correct or not.

ii) From client's point of view, any more outputs they are interested in? Think more and design for it.

iii) Use data visualization tools to perform data reporting with relevant graphs, charts, and visual elements.

THINK MORE AND DESIGN FOR IT.

A3.0 ACTIVITY 2 – SUBMISSION

Submission & Grading (Please refer to [1.1.5 Components of Assessment] P.3)

1) Below documents are required to submit the softcopy (in one pdf file) in Canvas on/or before the submission deadline – 20th November 2022.
    - Full set documentation to be sorted with index page in ONE pdf file, covering all documents listed in 1.1.5.

2) A 5-10 minutes Zoom online live demo for project team to show their final works in operating the ATU System will be arranged after submission deadline of 20th November 2022. Detail schedule for project live demo will be released later in October.