



School of  
**ENGINEERING &  
TECHNOLOGY**

School of Computing and Artificial Intelligence  
Faculty of Engineering and Technology  
Sunway University

### SOFTWARE REQUIREMENTS DOCUMENT

<b>SEMESTER</b>	: MAY 2025
<b>COURSE NAME</b>	: BIS2102 INFORMATION SYSTEM ANALYSIS AND DESIGN
<b>LECTURER</b>	: ASSOC. PROF. TS. DR. ASLINA BAHARUM
<b>SYSTEM NAME</b>	: Smart Waitlist System for Enhancing iZone Subject Enrolment
<b>PROGRAMME NAME</b>	: Bachelor of Information Systems (Honours) (Data Analytics)
<b>CONTACT PERSON</b>	: <i>Keertana, 23109614@imail.sunway.edu.my, 011-28353640</i>

<b>STUDENT ID</b>	<b>STUDENT NAME</b>	<b>GROUP</b>	<b>ROLE</b>
23109614	Keertana A/P Subramaniam	BIS 2-4	Team Leader, Project Planner, Use Case Writer, Reviewer

22034540	Wong Hui San	BIS 2-4	Use Case Writer, Scheduler, Project Planner
22053037	Siow Qi Yung	BIS 2-4	Use Case Writer and Designer, Requirement Analysts
22064141	Liu Yong Le	BIS 2-4	System Analysts, Scheduler, Requirement Analyst
22046015	Azaliya Turlubayeva	BIS 2-4	Researcher
22017867	Ayu Wen Li	BIS 2-4	Researcher, System Analyst, Use Case Designer

<b>DATE OF SUBMISSION</b>	27/06/2025
<b>ACTUAL DATE OF SUBMISSION</b>	27/06/2025
<b>NO OF DAYS LATE</b>	0

## Table of Contents

1.0 Introduction .....	5
1.1 Background Study.....	5
1.1.1 What kind of system .....	5
1.1.2 Who needs the system .....	6
1.1.3 Why they need the system.....	7
1.1.4 How the proposed system can improve their activities .....	8
1.1.5 An existing or similar system developed for other organizations.....	9
1.2 Development Methodology .....	11
1.3 Project Milestones and Deliverables .....	17
1.4 Project Schedules .....	18
1.4.1 Task Breakdown .....	18
1.4.2 Job Assignments.....	21
1.4.3 Gantt Chart .....	22
1.4.4 PERT Chart .....	23
2.0 Functional Requirements .....	27
2.1 Overall UseCase Diagram .....	27
2.2 Details of Each Use Case .....	31
2.2.1 Use Case 1: Join Subject Waitlist .....	31
2.2.2 Use Case 2: Auto Enrolment with Notification.....	35
2.2.3 Use Case 3: Live Dashboard for Waitlist .....	40
2.2.4 Complete Activity Diagram.....	43
3.0 Nonfunctional Requirements .....	46
4.0 Requirements Reviews.....	52

## List of Figures

Figure 1: Waterfall Model Development Methodology .....	11
Figure 2: Development Methodology.....	12
Figure 3: Gantt Chart .....	22
Figure 4: PERT Chart.....	26
Figure 5: Overall UseCase Diagram.....	28
Figure 6: Use Case 1 Diagram.....	34
Figure 7: Use Case 2 Diagram.....	39
Figure 8: Use Case 3 Diagram.....	42
Figure 9: Complete Activity Diagram .....	45

## **1.0 Introduction**

### **1.1 Background Study**

#### **1.1.1 What kind of system**

The proposed solution is a Subject Enrolment Waitlist Management System, which was deployed as an enhancement module for the current iZone student portal used in Sunway University. The system has its unique focus laid on optimizing the course registration process with the aim to address a universal issue for students. The issue that students are facing currently is the difficulty in enrolling in their preferred classes due to limited slots available. Instead of replacing the whole enrolment infrastructure, the system is integrated into the current infrastructure to offer critical features of automated waitlist management, real-time alerts, and a live status dashboard.

When an elective subject or specific time slot for any lecture, practical, or tutorial class reaches its maximum capacity, students are given the option to pin a waitlist for one preferred time slot if they are not satisfied with their currently selected time slot. To do this, students just need to click the 'Join Waitlist' button next to the time slot they wish to switch to. The system will continuously monitor class enrolments in real time. When a seat becomes available in the desired time slot selected, the system, automatically allocates it to the next eligible student on the waitlist. This allocation follows a first-come, first-served basis. Students who join the waitlist will not be enrolled in classes during that time slot until a seat becomes available. However, if they still do not get enrolled into the preferred time slot they want although they joined the waitlist, they will be automatically allocated to other time slots that has free seats.

In addition, the integration with Outlook ensures that students are notified instantly of any changes in their enrolment status. Before the Outlook email is sent, an alert message will also appear on the iZone home page with updates such as student confirmation, waitlist position, or registration status. This smart automation reduces the need for manual checking. Therefore, it can reduce students' stress during registration and creates a more efficient and fairer enrolment process for all students.

### **1.1.2 Who needs the system**

There are several groups who need the system as they can solve the issues they are facing in current state without the system. Firstly, the primary users of this system are Sunway University students, who face significant challenges during subject enrolment. These students often miss their preferred timetable due to limited class slots, which leads to frustration and repeated manual checking of the iZone platform. The new waitlist system is designed to address these problems by offering a more efficient and responsive enrolment process. With the automated enrolment and real-time status tracking system, students can gain greater control and visibility over their registration outcomes. Hence, students can reduce their stress and have a fairer subject enrolment process.

Besides, lecturers at Sunway University also stand to benefit from this system. Since students will receive clear and timely updates, the number of emails and requests regarding enrolment issues will decrease. This allows lecturers to focus more on academic planning instead of handling manual class adjustments requests.

Furthermore, the university administrators will find this system useful in terms of operational efficiency and data-driven decision-making. They can utilise resources more effectively by reducing the number of manual interventions and complaints during enrolment times.

Lastly, the Sunway IT Department also needs the system as they will experience fewer technical strain during peak enrolment periods. The current system often faces website crash as students repeatedly log in and refresh sites to secure a spot in a full class. This will make the portal more stable and reliable, thereby improving the university's technological infrastructure overall.

### **1.1.3 Why they need the system**

The Subject Enrolment Waitlist Management System was developed to solve problems that stem from the inefficiencies and manual skill required in the subject registration process on the Sunway University's iZone portal. Students face the ongoing and stressful challenge of refreshing the classes on the portal in a hope to grab a spot in their self-selected subject. This imbalance not only increases stress and anxiety but also leads to imbalance in enrolment experience as different students have unequal internet speeds or different timings which works to their advantage.

Furthermore, the lack of an intelligent waitlist mechanism leads to inefficient allocation where a student dropping a class does not reallocate the slot to other potential students. The absence of automation leads to unscheduled registration cycles, increased dissatisfaction among students, and high administrative workload.

Lecturers and administrative personnel have their share of problems as well since they can be easily overwhelmed with emails and manual requests for change of classes. This shift focus away from most important responsibilities and results in communication bottlenecks. During peak enrolment periods, the IT Department bear the brunt of the stress as the current system is unable to cope with high traffic which results in significant delay and even crashing of the portal.

In conclusion, each stakeholder group – students, lecturers, administrators, and IT personnel – encounters distinct yet interrelated issues due to the limitations of the existing system. The proposed solution is crucial for resolving these systemic difficulties and establishing a more structured, transparent, and student-focused enrolment process.

#### **1.1.4 How the proposed system can improve their activities**

The automated functionalities and real-time features of the Subject Enrolment Waitlist Management System, to be proposed, has the potential to change the operational workflow as well as the enrolment experience across the university for the better.

Stress and uncertainty regarding subject registration vanish for students with the incorporation of this new system. Tracking class vacancies automatically and single-click waitlist join makes class access monitoring easier. The students can be fairly granted access to the classes that they desire without tedious portal monitoring. Outlook Integration ensures students are timelessly updated in real-time about changes to their enrolment status, enabling them to act quickly if needed.

Reduced class change related questions and interruptions allow lecturers to concentrate on teaching and curriculum planning. Manual last-minute changes and constant student request repetition do not need to be accommodated anymore.

Having access to real-time data and class enrolment analytics provides information essential for administrative staff to make well-informed decisions regarding resource allocation, scheduling, and class capacities. The decrease in manual interventions, especially at high-traffic registration periods, improves streamlined operational workflows. The system alleviates the technical burden on the current infrastructure for the IT Department. Minimising repeated log-ins and refreshes on the portal significantly reduces the risk of crashes or slowdowns. This enhances the overall stability, scalability, and robustness of the platform. The system promotes a more equitable and efficient academic environment, wherein technology enhances both the student experience and institutional effectiveness.



### **1.1.5 An existing or similar system developed for other organizations**

An example of a similar student enrolment system is the PeopleSoft Campus Solutions developed by Oracle. PeopleSoft Campus Solutions is a student information system (SIS) that specialises in managing students' entire lifecycle in higher education institutions, improving students' experience through incorporating advance tools into features such as student recruitment, enrolment and student records. The key feature of PeopleSoft that is similar to our proposed system is their student enrolment system, providing functions such as:

#### **Course Registration**

Students are able to register for courses through web-based service portals. The system allows students to validate their courses in real-time, such as confirming if they had successfully fulfilled prerequisites as well as checking for conflicts in schedules and the course availability. This is to ensure that the enrolment abides by the academic regulations and avoids any potential errors (Oracle, n.d.).

#### **Waitlist Management**

When a student's preferred subject slot reaches full capacity, students may join a waitlist. PeopleSoft manages this queue by holding the students spot in their currently enrolled course as they wait for available slots in their preferred class schedule. The waitlist is based on priority rules set by the educational institution such as the time of request and student course. Students will also receive real-time notifications when a slot opens up and they are moved from the waitlist into the course.

#### **Course Add, Drop and Swap**

PeopleSoft Campus solution also allows students to add classes, which is the process of adding enrolment into a student's record during their enrolment and add-drop period. Additionally, students may utilise the drop function to opt out of a class from their timetable in the case whereby they decide they would like to modify their schedule and no longer wants to take it after enrolling (Rochester Institute of Technology, 2019). The Swap

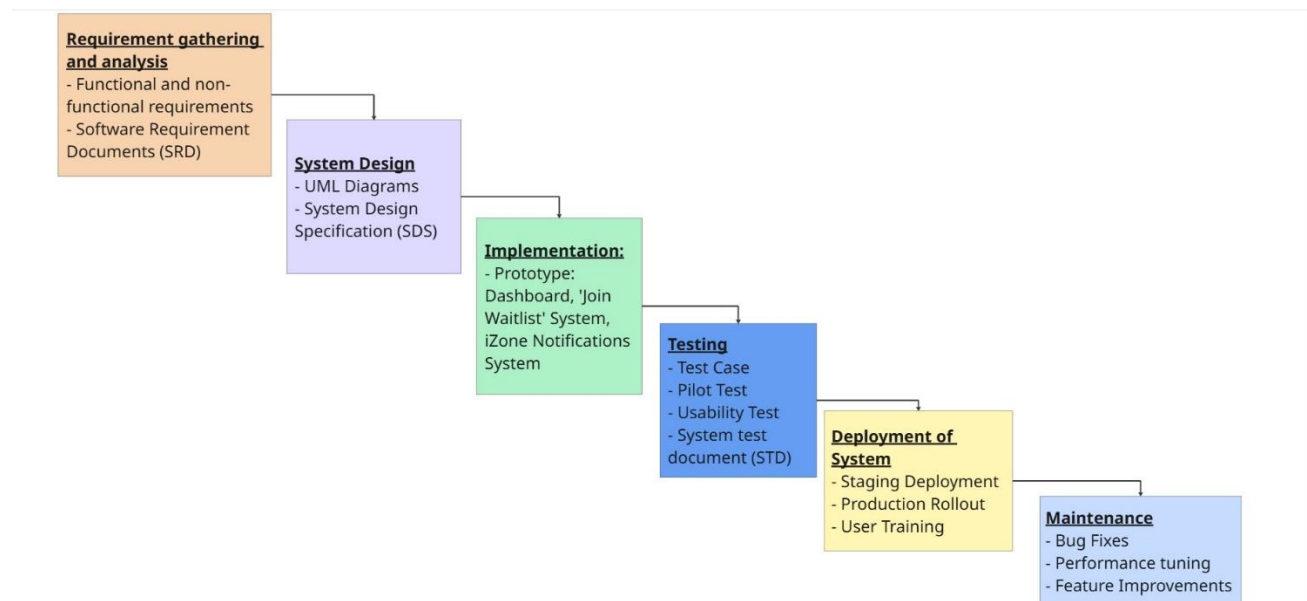
function is similar to the waitlist function in which students may join a waitlist for a fully occupied subject slot whilst still holding their place of their enrolled course and only “swapping” when a spot opens up in the waitlisted class.

### **Administrative Overrides and Reporting Tools**

Administrative staff are able to override enrolment blocks by manually enrolling students and generating detailed student reports. The enrolment overrides can be classified into class overrides and general overrides. Class overrides are used for specific course-related issues, such as letting a student join a full class, changing the number of credits for a class, or adjusting the grading method. Meanwhile, general overrides include broader enrolment issues such as allowing a student to register outside their assigned time, taking more or fewer credits than usual, enrolling in classes with time conflicts, or overlooking missing prerequisites. These options allow administrative staffs to ensure that enrolment can proceed when necessary.

## 1.2 Development Methodology

The development methodology used is the Waterfall Model development methodology. The Waterfall Model is a form of Software Development Model typically used for complicated and elaborate software development or project management projects. The projects that use the Waterfall Model tend to have extensive timelines with distinct project objectives and deliverables. This model divides and follows each of its phases in a chronological and successive order (GeeksforGeeks, 2025).



*Figure 1: Waterfall Model Development Methodology*

Phase	Objective	key Activities	Deliverables
1. Requirements Gathering & Analysis	Define the functional and non-functional requirements of the waitlist system	<ul style="list-style-type: none"> <li>Identify <b>functional requirements</b> (automated waitlist management, real-time monitoring, notification system, live status dashboard)</li> <li>Identify <b>non-functional requirements</b> (performance, security, scalability, usability)</li> </ul>	Software Requirements Document (SRD)
2. System Design	Create a high-level and detailed design of the system	Develop <b>UML Diagrams</b> <ul style="list-style-type: none"> <li>Data Dictionary</li> <li>Use Case Diagrams</li> <li>sequence diagrams</li> <li>State Transition Diagrams</li> </ul>	System Design Specification (SDS)
3. Implementation	Develop the system based on the design	<b>Prototype development</b> <ul style="list-style-type: none"> <li>dashboard system</li> <li>Join Waitlist System</li> <li>iZone Notification System</li> </ul>	Fully functional prototype of waitlist system
4. Testing	Ensure the system works as expected	<ul style="list-style-type: none"> <li>Use Case Test</li> <li>Pilot Test</li> <li>Usability Test</li> <li>System test document (STD)</li> </ul>	Test Reports & Bug Fixes
5. Deployment	Release the system to production	<ul style="list-style-type: none"> <li>Staging deployment</li> <li>production rollout</li> <li>user training</li> </ul>	Live waitlist system on iZone
6. Maintenance	Ensure smooth operation post-deployment	<ul style="list-style-type: none"> <li>Bug fixes</li> <li>performance optimization</li> <li>feature enhancements</li> </ul>	Updated and optimized system

*Figure 2: Development Methodology*

## 1. Requirements

In this initial phase, we gathered all necessary functional and non-functional requirements from relevant stakeholders, including students, lecturers, the IT department, and university administrators. The main problem identified was that students had trouble securing their preferred time slots during enrolment. We

documented requirements such as automated waitlisting, real-time alerts, integration with Microsoft Outlook, and a live dashboard.

### **Key Requirements:**

#### **Functional Requirements:**

- **Automated Waitlist Management:** Students can join a waitlist for their preferred time slot if the class they want is full. Seats will be filled on a first-come, first-served basis when one becomes available.
- **Real-time Monitoring:** The system checks seat availability constantly. It will automatically assign seats to the next eligible student.
- **Notification System:** Students will receive email notifications through Outlook when a seat opens up or their waitlist position changes.
- **Live Status Dashboard:** The system shows the current waitlist position and the number of students ahead. This helps students decide whether to wait or choose a different class.

#### **Non-Functional Requirements:**

The frontend will be a web-based dashboard at the architectural level. It will be prototyped with Figma and will operate with the iZone platform. A relational database will keep track of information about waitlists, student preferences, and the status of class enrolment. The Outlook API will connect to the notification system to handle email alerts. The key non-functional requirements that the Automated Waitlist System should have includes performance, scalability, reliability, usability, availability, security, maintainability and portability.

## ***2. System and Software Design***

After the requirements have been defined and evaluated, the next step of the Waterfall Model is System Design which translates the states requirements into a functional format that can be designed and implemented. This typically include high-level and low-level

design architecture. During this stage, various Unified Modelling Language Diagrams were created including:

**Data Dictionary:** Data Dictionary defines the metadata used in the system's database. The metadata include names, data attributes, relationship, schema and definition.

**Use Case Diagram:** These diagrams are used to represent the functional part of the system together with the actors' interaction with the system. It is a diagram that depicts the various situations in which the system is used, giving developers a high-level perspective into performance capability of the system before defining the implementation specifics.

**Sequence Diagram:** This diagram showcases the visual interaction of various components in system. It visualises the sequential order of communication exchanged between the system objects, showing the timeline and orders in which messages are sent.

**State Transition Diagram:** This diagram is used to represent the different shifts in interface or states of the system, showing all the possible transitional states that a system can have.

The UML diagrams were documented in the System Design Specification (SDS) document.

### ***3. Implementation***

During this phase, developers built the system modules based on the design specifications. Each module, like the "Join Waitlist" button, enrollment monitor, and email notification engine will be developed individually and went through various testings to ensure it worked well and reliably.

The implementation phase means turning the system design into working software. This involves creating both frontend and backend parts. The frontend will have a dashboard

that shows the current waitlist status and includes a responsive "Join Waitlist" button. Backend development will focus on creating a real-time seat availability checker and the logic for automatic seat allocation based on the first-come, first-served principle.

Integrating the database is a key part of this phase, allowing the system to store, retrieve, and manage waitlist data effectively. The notification service will be set up to send automated alerts through the Outlook API to inform students about any changes in their status. The result of this stage is a fully functional and integrated waitlist system, ready for testing.

#### **4. Testing**

Once the prototype has been developed and implemented, the systems are tested in order to ensure that the systems are functional, reliable and meets the functional and non-functional requirements. The prototype will go through a series of tests which will be documented in the System Test Document (STD):

- Use case testing: Use case testing is a form of functional testing with the objective of assisting developers in finding and checking different situations on the whole system. This helps in identifying errors in the system and that the system satisfies the requirements and expected outcomes.
- Pilot testing: Pilot testing involves having selected group of different users testing the entire system and its components to gather their feedback for improvements. This test is done before the system can be deployed so that system can be improved to satisfy user's needs, feasibility and costs.
- Usability Testing: This is another method for developers to assess the design and functionality of the user interface. This helps better understand the end users experience when using the system. The Usability Testing will be conducted for more cohesive user feedback which will be used to amend the final prototype before deployment.

## **5. Deployment**

During the deployment phase, the system is put into a real-world setting. The first step is a staging deployment, in which the system is tested in an environment that is similar to the production setup. This process helps uncover any last-minute problems before the complete rollout. Once stability is established, the iZone platform will gradually make the system available to all users.

Students will get short training or instructional materials that show them how to use the new waitlist system so that it goes smoothly. Once this phase is over, the system will be officially live and ready for usage throughout enrolment periods.

## **6. Maintenance**

After deployment, the system moves into the maintenance phase to keep up its performance and relevance. This phase involves fixing bugs or errors reported by users, improving system performance for quicker response times, and adding new features when necessary. One possible improvement could be creating priority waitlists for final-year students or specific academic situations.

Regular updates and monitoring will help the system stay efficient, secure, and in line with user needs as time goes on. The goal of this phase is to have a stable and steadily improved waitlist management system.



### 1.3 Project Milestones and Deliverables

No.	Date/Week	Milestones	Deliverables
1.	4	Submission of project proposal	Project Proposal
2.	8	Submission of Software Requirements Document (SRD)	Software Requirements Document
3.	9	Submission of System Design Specification (SDS)	System Design Specification
4.	11	Submission of System Test Document (STD)	System Test Document
5.	13	Submission of Final Report and Presentation	Final Report and Presentation

## 1.4 Project Schedules

### 1.4.1 Task Breakdown

No.	Task	Duration
<b>Phase 1: Project Initiation</b>		
1.1	Brainstorm system idea, type of application, and target company	1 day
1.2	Decide between Business Case or System Request	4 days
1.3	Finalize target system (iZone enrolment system)	3 days
<b>Phase 2: Requirements Gathering</b>		
2.1	Prepare survey form (Google Form)	1 day
2.2	Distribute survey to Sunway students	2 days
2.3	Analyze responses and identify problem + proposed solution	2 days
<b>Phase 3: Proposal Development</b>		
3.1	Start business proposal + system request form	4 days
3.2	Carry out a quick proposal presentation	3 days
3.3	Complete business proposal + system request form (revise with feedback)	3 days
3.4	Submit Business Proposal & System Request on eLearn and wait for feedback	4 days
3.5	Receive lecturer feedback (to be used for improvements)	1 day
<b>Phase 4: Software Requirements Document (SRD)</b>		
4.1	Discuss & decide on use cases	2 days
4.2	Draw activity diagram	2 days
4.3	Create Gantt Chart and PERT chart	2 days
4.4	Finalise SRD report	8 days
4.5	Prepare slides for SRD presentation	2 days
4.6	Carry out presentation for SRD	1 day
4.7	Email the ITS department staff to request an interview and schedule a date	1 day

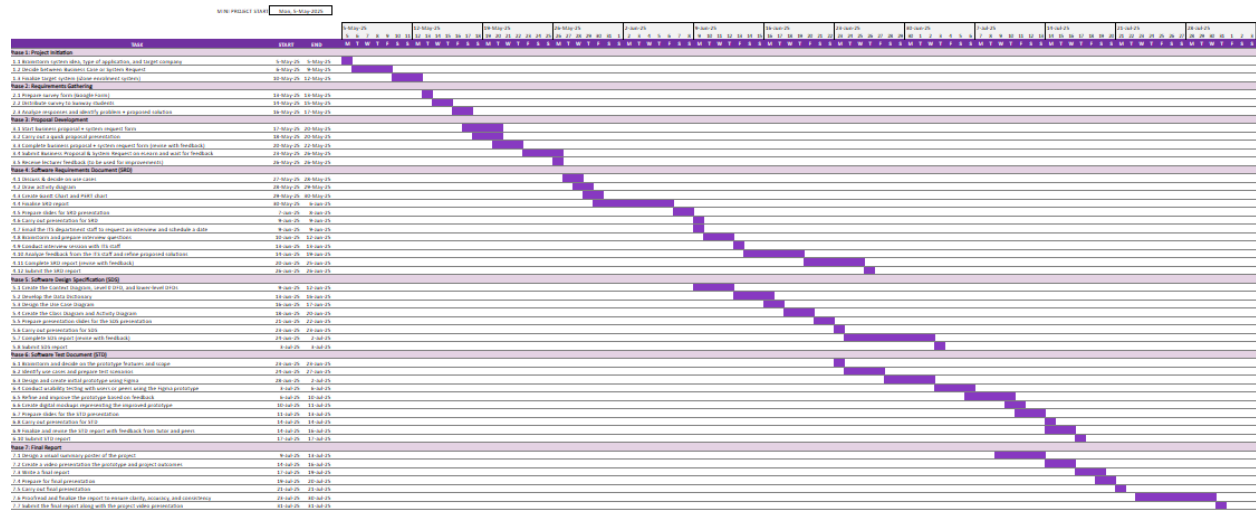
4.8	Brainstorm and prepare interview questions	3 days
4.9	Conduct interview session with ITS staff	1 day
4.10	Analyze feedback from the ITS staff and refine proposed solutions	6 days
4.11	Complete SRD report (revise with feedback)	6 days
4.12	Submit the SRD report	1 day
<b>Phase 5: Software Design Specification (SDS)</b>		
5.1	Create the Context Diagram, Level 0 DFD, and lower-level DFDs	4 days
5.2	Develop the Data Dictionary	4 days
5.3	Design the Use Case Diagram	2 days
5.4	Create the Class Diagram and Activity Diagram	3 days
5.5	Prepare presentation slides for the SDS presentation	2 days
5.6	Carry out presentation for SDS	1 day
5.7	Complete SDS report (revise with feedback)	10 days
5.8	Submit SDS report	1 day
<b>Phase 6: Software Test Document (STD)</b>		
6.1	Brainstorm and decide on the prototype features and scope	1 day
6.2	Identify use cases and prepare test scenarios	4 days
6.3	Design and create initial prototype using Figma	5 days
6.4	Conduct usability testing with users or peers using the Figma prototype	4 days
6.5	Refine and improve the prototype based on feedback	5 days
6.6	Create digital mockups representing the improved prototype	2 days
6.7	Prepare slides for the STD presentation	3 days
6.8	Carry out presentation for STD	1 day
6.9	Finalize and revise the STD report with feedback from tutor and peers	3 days
6.10	Submit STD report	1 day
<b>Phase 7: Final Report</b>		

7.1	Design a visual summary poster of the project	5 days
7.2	Create a video showcasing the prototype and project outcomes	3 days
7.3	Write a final report	3 days
7.4	Prepare for final presentation	2 days
7.5	Carry out final presentation	1 day
7.6	Proofread and finalize the report to ensure clarity, accuracy, and consistency	8 days
7.7	Submit the final report along with the project video presentation	1 day

### 1.4.2 Job Assignments

No.	Name	Job	Description
1.	Keertana	Project Leader/Developer	<ul style="list-style-type: none"><li>• Lead team</li><li>• Assign job to team members</li><li>• Support Design Planning</li><li>• Lead system logic and backend development</li></ul>
2.	Wong Hui San	System Analyst	<ul style="list-style-type: none"><li>• Analyse from Data Gathering and System Requirement</li><li>• Contribute to wireframes and design flow</li></ul>
3.	Ayu Wen Li	UI Coordinator	<ul style="list-style-type: none"><li>• Lead UI layout structure and component design</li><li>• Create and manage Figma designs</li></ul>
4.	Siow Qi Yung	UX Coordinator	<ul style="list-style-type: none"><li>• Plan user journey and interactions</li><li>• Support Figma prototyping</li></ul>
5.	Liu Yong Le	Backend Developer / Data Lead	<ul style="list-style-type: none"><li>• Design database schema (waitlist, users, enrolment)</li><li>• Handle backend logic</li><li>• Support integration with UI</li></ul>
6.	Azaliya	Tester / QA Lead	<ul style="list-style-type: none"><li>• Test system functionality and usability</li><li>• Log and report bugs</li><li>• Assist with UI feedback</li></ul>

### 1.4.3 Gantt Chart



*Figure 3: Gantt Chart*

Gantt Chart Excel link:

<https://imailsunwayedu->

[my.sharepoint.com/:x/g/personal/22034540\\_imal\\_sunway\\_edu\\_my/EdRP5pHTVP5Lilq7LF2\\_KjkBr2N75jAyEaTX3Pby33950A?e=tqbkqY](https://my.sharepoint.com/:x/g/personal/22034540_imal_sunway_edu_my/EdRP5pHTVP5Lilq7LF2_KjkBr2N75jAyEaTX3Pby33950A?e=tqbkqY)

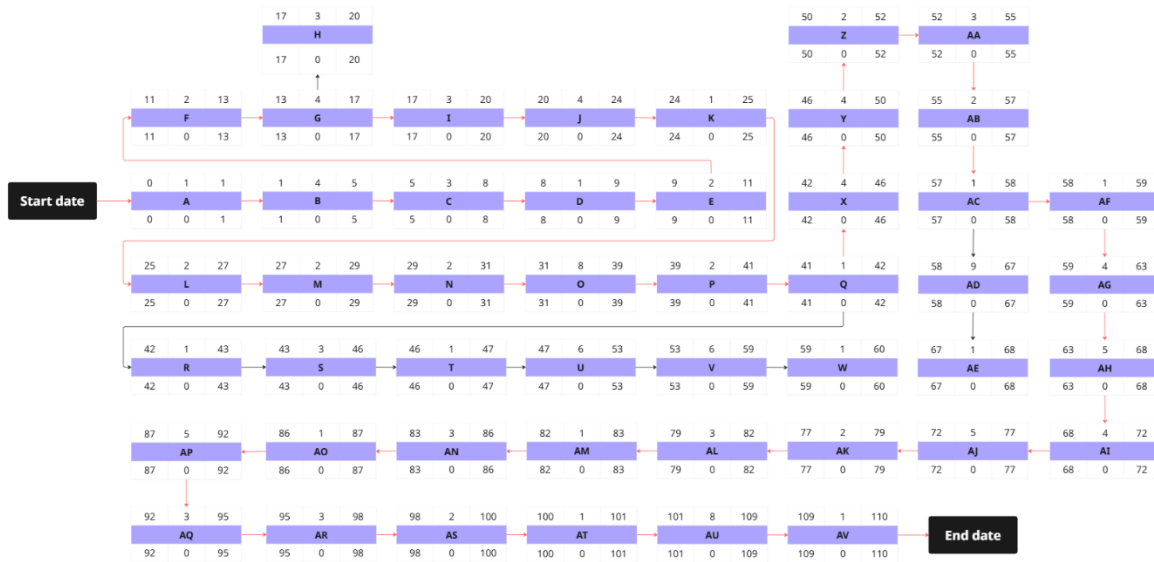
#### 1.4.4 PERT Chart

Activity	Predecessors	Optimistic (O)	Most Likely (M)	Pessimistic (P)	Expected Time (TE)
A (Brainstorm system idea)	None	1	1	1	1
B (Decide case/request)	A	2	4	4	4
C (Finalize target system)	B	2	3	3	3
D (Prepare survey form)	C	1	1	1	1
E (Distribute survey)	D	1	2	2	2
F (Analyse responses)	E	2	2	2	2
G (Start proposal)	F	2	4	4	4
H (Prepare slides)	G	2	3	3	3
I (Revise proposal)	G	2	3	3	3
J (Submit & wait feedback)	I	1	4	4	4
K (Receive feedback)	J	1	1	1	1
L (Discuss use cases)	K	2	2	2	2
M (Draw activity diagram)	L	1	2	2	2
N (Create Gantt/PERT)	M	1	2	2	2
O (Finalize SRD)	N	5	8	10	8
P (Prepare SRD slides)	O	2	2	2	2
Q (SRD Presentation)	P	1	1	1	1
R (Email ITS Department)	Q	1	1	1	1

S (Brainstorm interview questions)	R	3	3	3	3
T (Conduct interview with ITS)	S	1	1	1	1
U (Analyse interview feedback & refine solution)	T	6	6	6	6
V (Revise and complete SRD report)	U	6	6	6	6
W (Submit SRD report)	V	1	1	1	1
X (Create Context / Level 0 / DFDs)	Q	4	4	4	4
Y (Develop Data Dictionary)	X	4	4	4	4
Z (Design Use Case Diagram)	Y	2	2	2	2
AA (Create Class & Activity Diagrams)	Z	3	3	3	3
AB (Prepare SDS Presentation Slides)	AA	2	2	2	2
AC (Present SDS)	AB	1	1	1	1
AD (Revise & complete SDS report)	AC	9	9	9	9
AE (Submit SDS report)	AD	1	1	1	1
AF (Brainstorm prototype features)	AC	1	1	1	1
AG (Identify use cases & test scenarios)	AF	4	4	4	4



AH (Design Figma prototype)	AG	5	5	5	5
AI (Usability testing using prototype)	AH	4	4	4	4
AJ (Refine prototype based on feedback)	AI	5	5	5	5
AK (Create final mockups)	AJ	2	2	2	2
AL (Prepare STD presentation slides)	AK	3	3	3	3
AM (Present STD)	AL	1	1	1	1
AN (Revise STD report)	AM	3	3	3	3
AO (Submit STD report)	AN	1	1	1	1
AP (Design visual summary poster)	AO	5	5	5	5
AQ (Create final video presentation)	AP	3	3	3	3
AR (Write final report)	AQ	3	3	3	3
AS (Prepare for final presentation)	AR	2	2	2	2
AT (Deliver final presentation)	AS	1	1	1	1
AU (Proofread + revise report)	AT	8	8	8	8
AV (Submit final report and video)	AU	1	1	1	1



miro

Figure 4: PERT Chart

**Critical Path:** A + B + C + D + E + F + G + H + I + J + K + L + M + N + O + P + Q + X + Y + Z + AA + AB + AC + AF + AG + AH + AI + AJ + AK + AL + AM + AN + AO + AP + AQ + AR + AS + AT + AU + AV = 110 days

**PERT Chart Link:**

[https://miro.com/app/board/uXjVlmaU22g/?share\\_link\\_id=285759257641](https://miro.com/app/board/uXjVlmaU22g/?share_link_id=285759257641)

## **2.0 Functional Requirements**

### **2.1 Overall UseCase Diagram**

#### **Short Description of Overall Use Case Diagram:**

##### **1. Actor: Students**

###### **Use Cases:**

Join Subject Waitlist - Students can join a waitlist for a preferred time slot if the time slot is currently full.

View Waitlist Position - Students can check their current position and status on the waitlist and see how many students are ahead of them.

Cancel Waitlist – Students have the option to cancel their waitlist request after joining the waitlist for their preferred time slot.

##### **2. Actor: Waitlist Automation System**

###### **Use Cases:**

Monitor Class Capacity – System continuously tracks class enrolment levels to detect when a seat becomes available.

Auto-allocate Waitlist Seats - When a seat becomes available, the system automatically assigns it to the next eligible student in the waitlist.

Update Waitlist - Adjusts the waitlist by removing enrolled students or skipped students (e.g., due to schedule conflict) and reordering the list.

##### **3. Actor: iZone Notification System**

###### **Use Case:**

Send Alert Message - Displays real-time alerts on the iZone homepage before sending email notification to students.

##### **4. Actor: Live Dashboard System**

###### **Use case:**

Display Waitlist Position - Shows live waitlist information (position, status, number of students ahead) on the Live Dashboard on iZone homepage.

## Overall Use case Diagram:

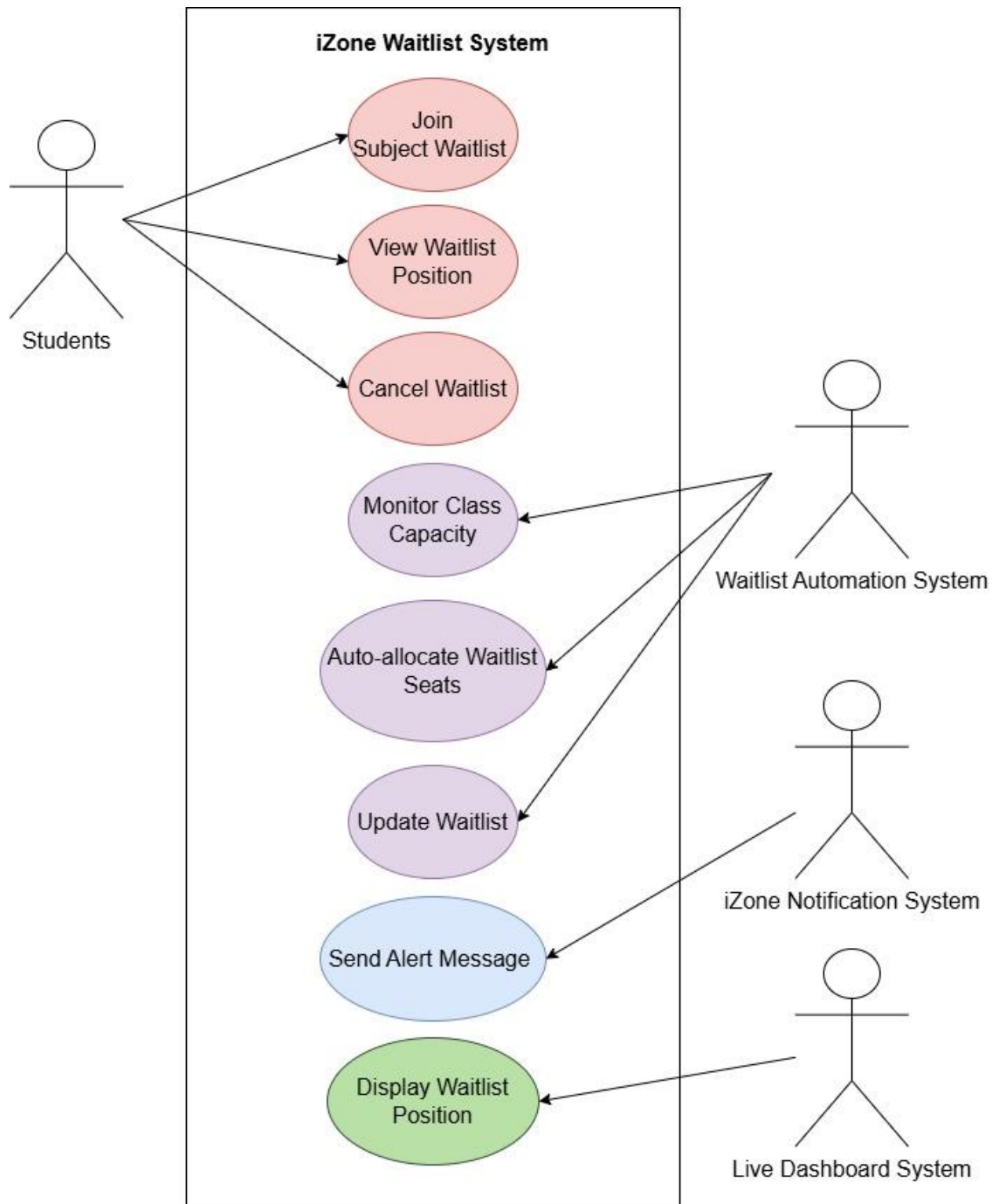


Figure 5: Overall UseCase Diagram

**Actors:**

**1. Students** – Sunway University students who use the iZone platform to enroll in subjects and classes (lecture, tutorial, practical).

- Join the waitlist through the subject enrolment page if the preferred class time slot is full. Click "Join Waitlist" next to the desired time slot.
- View their waitlist position, status and number of students ahead on Waitlist Live Dashboard in real time.
- Cancel a waitlist through the subject enrolment page and click on "Cancel Waitlist" next to the time slot they waitlisted for.
- Receive alert messages (a bell icon with a red dot) before they get any email notification on the iZone homepage when waitlist updates occur.
- Get email notification via Outlook when:
  - The students successfully joined a waitlist for a time slot.
  - They have schedule conflict.
  - They are removed from the waitlist if they cancel the waitlist request or enrol successfully in their desired time slot.
  - The waitlist and enrolment period are end.
  - They are not successfully enrolled in their preferred time slot and are enrolled in another available time slot.

**2. Waitlist Automation System** - An automated backend service that manages the waitlist functionality on the iZone platform.

- Monitor class capacities in real time.
- Automatically allocate available seats to the next eligible student on the waitlist.
- Remove students from the waitlist if they cancel their waitlist or are successfully enrolled in a class time slot.
- Update waitlist position and status for all students.
- Trigger and send email notifications via Outlook and update the iZone Notification System to alert students to ensure that students are promptly informed of any updates related to the waitlist.

**3. iZone Notification System** – A front-end alert system that visually notifies students of important updates within the iZone platform.

- Send alert messages (a bell icon with a red dot) on the iZone homepage when waitlist updates occur.

**4. Live Dashboard System** – A student-facing real-time interface on the iZone platform that allows Sunway University students to:

- Show current position and status on the waitlist.
- Display how many students are ahead of them for a specific class.
- Receive live updates when their position changes due to enrolment activity.

This can enhance decision-making such as whether to choose another class time slot or continue to stay on the waitlist.

## 2.2 Details of Each Use Case

### 2.2.1 Use Case 1: Join Subject Waitlist

#### Brief Description

A student can join a waitlist only once for a subject that is already full. The student is given an option to join the waitlist instead of being blocked from entering the subject. The student is added to the waitlist queue.

#### Precondition

Before this use case can be initiated, the student must be logged into the iZone portal already. Then, they can enter the enrolment section to enrol into a class. The student has already searched for the time availability that they're interested in. However, the selected time slot is currently full.

#### Postcondition

The student is then added to the waitlist queue system and their position the list is shown. The student will be enrolled automatically into their preferred time slot when a slot becomes available and their next in line in the queue.

#### Normal Flow

Steps	Student Action	Waitlist Automation System	iZone Notification System
1	Log into iZone		
2	Enter the enrolment portal to start enrolment		
3		Displays multiple time slots for a subject, including the availability status	

<b>4</b>	Searches for preferred time slot for a subject		
<b>5</b>	Attempts to enroll in the time slot which is full		
<b>6</b>		Displays “Join Waitlist” option button for time slots with full availability	
<b>7</b>	Clicks the “Join Waitlist” button		
		Checks if the student is already on waitlist	
<b>8</b>		Adds student to the waitlist	
<b>9</b>		Display waitlist registration confirmation message	
<b>10</b>	Confirm join waitlist		
<b>11</b>			Displays pop-up alert saying “You’ve been added to the waitlist. Please check your email for details.”
<b>12</b>	Notice the pop-up and are reminded to check their email		
<b>13</b>		Disable “Join Waitlist” option to avoid duplicate entries	



### **Exception/Alternative Flow**

Exception Flow 1: When student clicks on “Join Waitlist” button again

When the student clicks “Join Waitlist” button again for the same time slot after already joining, the system will disable the “Join Waitlist” button and display message “You have already joined the waitlist. You can view your position on the Live Dashboard in the homepage.”

Exception Flow 2: System error occurs during waitlist registration

System will display error message “Failed to join the waitlist. Please try again later.”, when an error occurs during the process if joining the waitlist.

Exception Flow 3: Schedule conflict during waitlist registration

When the student attempt to join waitlist for a time slot that has a schedule conflict with the current enrolled time slot, system will display error message “Join waitlisted failed for [Subject Code], [Group Number] due to schedule conflict with [Subject Code], [Group Number]” to indicate waitlist enrolment unsuccessful. Similar error message will also appear in iZone Homepage.

## Activity Diagram

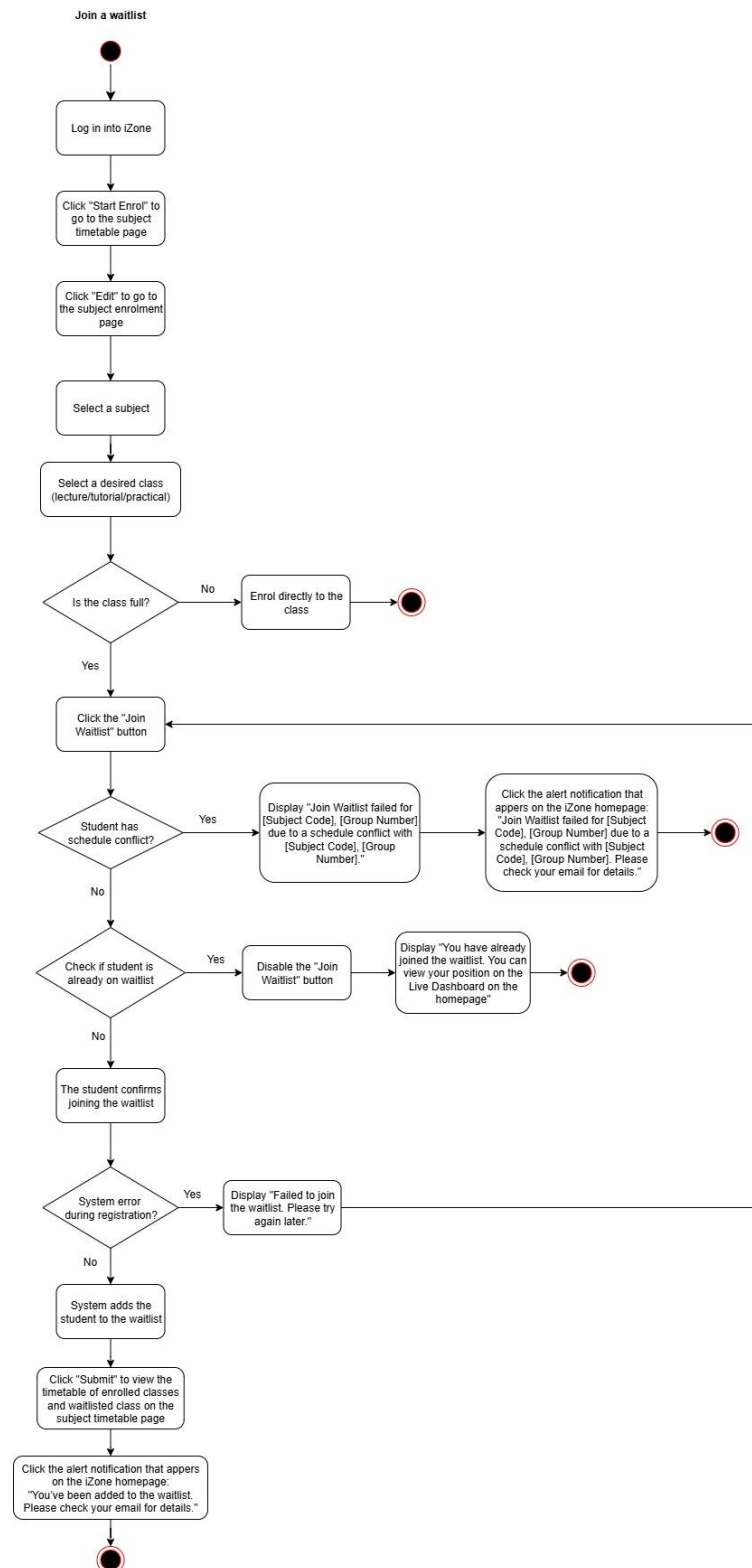


Figure 6: Use Case 1 Diagram

## **2.2.2 Use Case 2: Auto Enrolment with Notification**

### **Brief Description**

After students join the waitlist for a class time slot, they will receive a pop-up notification in iZone asking them to check their email for successful enrolment into a waitlist. The system then automatically tracks student's position in the queue. Once a seat becomes available, the system automatically enrolls the next student in line into the class. Then, the system removes the enrolled students from waitlist to indicate they are no longer waiting for enrolment, as they have already been auto enrolled into classes

Once a student is auto-enrolled into a class, the system sends a confirmation email to student and displays a second pop-up notification in iZone, reminding them to check their email for details of their successful enrolment.

When the subject enrolment period ends, students who remain on the waitlist will be automatically placed into any remaining available time slots for that subject. A pop-up notification will appear, prompting students to check their email for the updated class enrolment details.

### **Precondition**

Before this use case can be initiated, students must join the waitlist of a class and time slot.

### **Postcondition**

At the end of this use case, all students who joined the waitlist have been successfully enrolled into a class, either through automatic enrolment when seats became available or through automatic placement into whatever remaining slots after the enrolment period ends. The relevant pop-up messages/notifications have been displayed in iZone to inform students to check for enrolment status details in their email.

### Normal Flow

Steps	Students	Waitlist Automation System	iZone Notification System
1.	Join waitlist for a time slot.		
2.		Adds students to the waitlist queue for that time slot.	
3.			Displays a pop-up notification prompting students to check their email for confirmation of successful enrolment into waitlist.
4.	Notice the pop-up and is reminded to check their email.		
5.		Tracks the student's position in the waitlist queue and check for the seat availability.	
6.		When a seat becomes available, auto enrolls next student in the waitlist queue into the class.	

7.		Removes students from the waitlist once they have been successfully enrolled.	
8.			Display a pop-up notification prompting students to check their email for successful class enrolment confirmation.
9.	Notice the pop-up and is reminded to check their email.		

### **Exception/Alternative Flow**

#### Alternative Flow 1: Student cancels waitlist request

1. Student has joined a waitlist but decided to cancel the request before a seat becomes available.
2. Students will be removed from the waitlist.
3. A pop-up notification appears in iZone Homepage confirming the cancellation of waitlist request: "You have successfully been removed from the waitlist for [Subject Code], [Group Number]. Please check your email for details.

#### Alternative Flow 2: Auto-assignment after enrolment period ends

1. When the subject enrolment period is over, system checks and identifies the remaining students who are still on the waitlist.
2. The system automatically assigns all these students to any remaining available time slots randomly for the subject.

3. After assigning time slots to students, the system removes them from waitlist to indicate they are no longer waiting for enrolment, as they have already been auto enrolled into classes.
4. A pop-up notification is displayed in iZone, informing students that the enrolment period has ended and prompting them to check their email for the enrolment details.: “Enrolment period has ended. You have been auto-enrolled into an available class: [Subject Code], [Group Number]. Please check your email for class details.”

## Activity Diagram

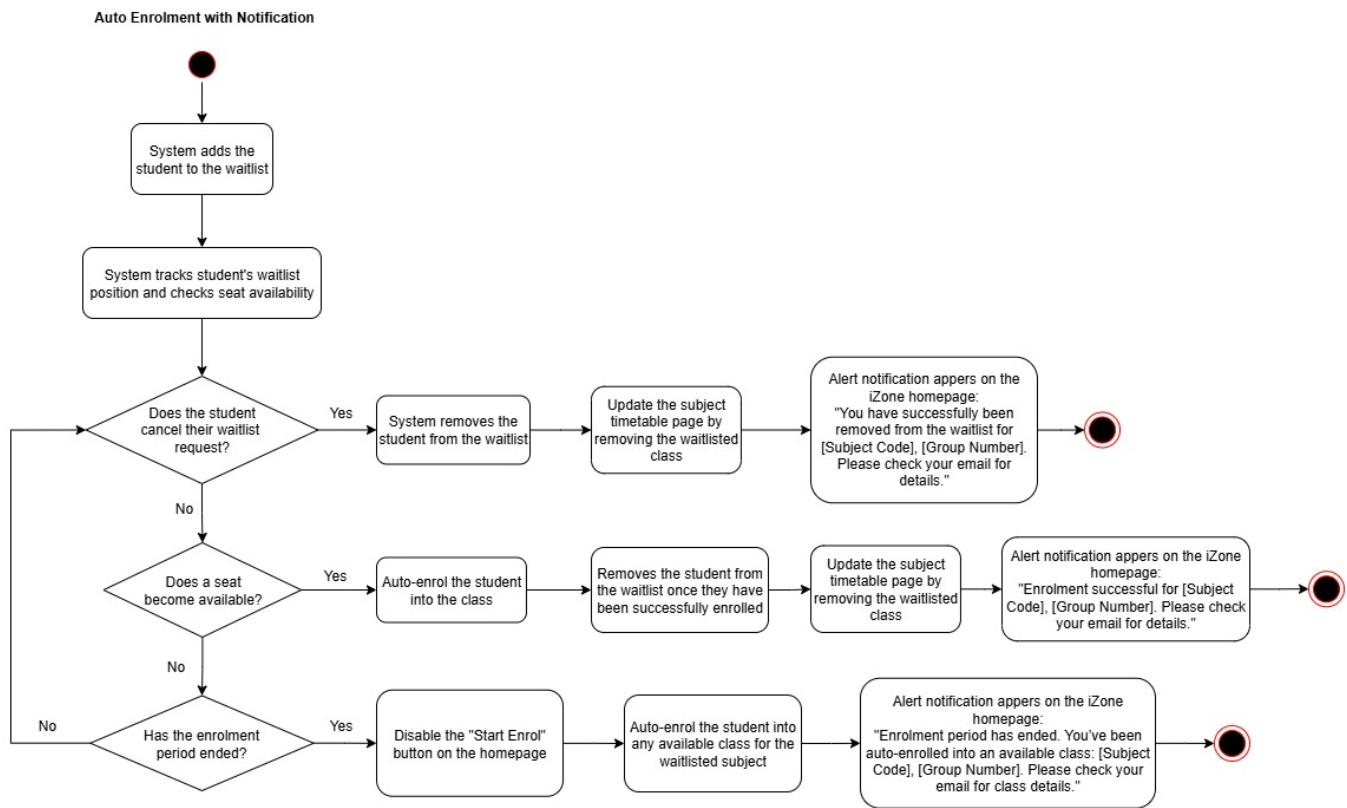


Figure 7: Use Case 2 Diagram

### 2.2.3 Use Case 3: Live Dashboard for Waitlist

#### Brief Description

A student can view their waitlist status and current position in the waitlist through a live dashboard in the main page of iZone portal during enrolment period. The system will update the dashboard based on real-time changes until the enrolment period ends.

#### Precondition

Student have already logged into their account in iZone portal. The student have already joined a waitlist previously.

#### Postcondition

The student is able to monitor their status and position based on the information displayed in the dashboard.

#### Normal Flow

Steps	Students	Live Dashboard System
1.	Log into iZone portal	
2.	Navigates to “Live Dashboard” on the iZone homepage	
3.		Loads the student dashboard
4.		Displays the time slot for the subject that the student is waitlisted for
5.		Shows the information for waitlist position, status and number of students ahead
6.	View details on status and position	
7.		Continuously updates in real-time



8.		Updates on live dashboard ends when enrolment period officially closes
----	--	------------------------------------------------------------------------

### **Exception/Alternative Flow**

#### Alternative Flow 1: Successful Enrolment

When a seat becomes available and the student meets the criteria for eligibility, the system will automatically enrol the student into their preferred class time slot. The dashboard will display a message saying, "Congratulations! You've enrolled successfully for [Subject Code], [Group Number]." The student is then removed from the waitlist for that subject.

#### Alternative Flow 2: Live Waitlist Update

When there are any changes in the student's position, status, or the number of students ahead in the waitlist, the dashboard will update in real-time to reflect the latest data. The information which is updated is shown immediately without the student requiring to refresh manually.

#### Exception Flow 1: No Waitlist Records

The system will display "You are not on the waitlist for any subjects." If the student has not joined any waitlist.

#### Exception Flow 2: Enrolment period ends

When the official enrolment period has ended, the system will display notice "Waitlist and enrolment period have ended. Dashboard is now closed." and stops all live updates. → disable live dashboard button

## Activity Diagram

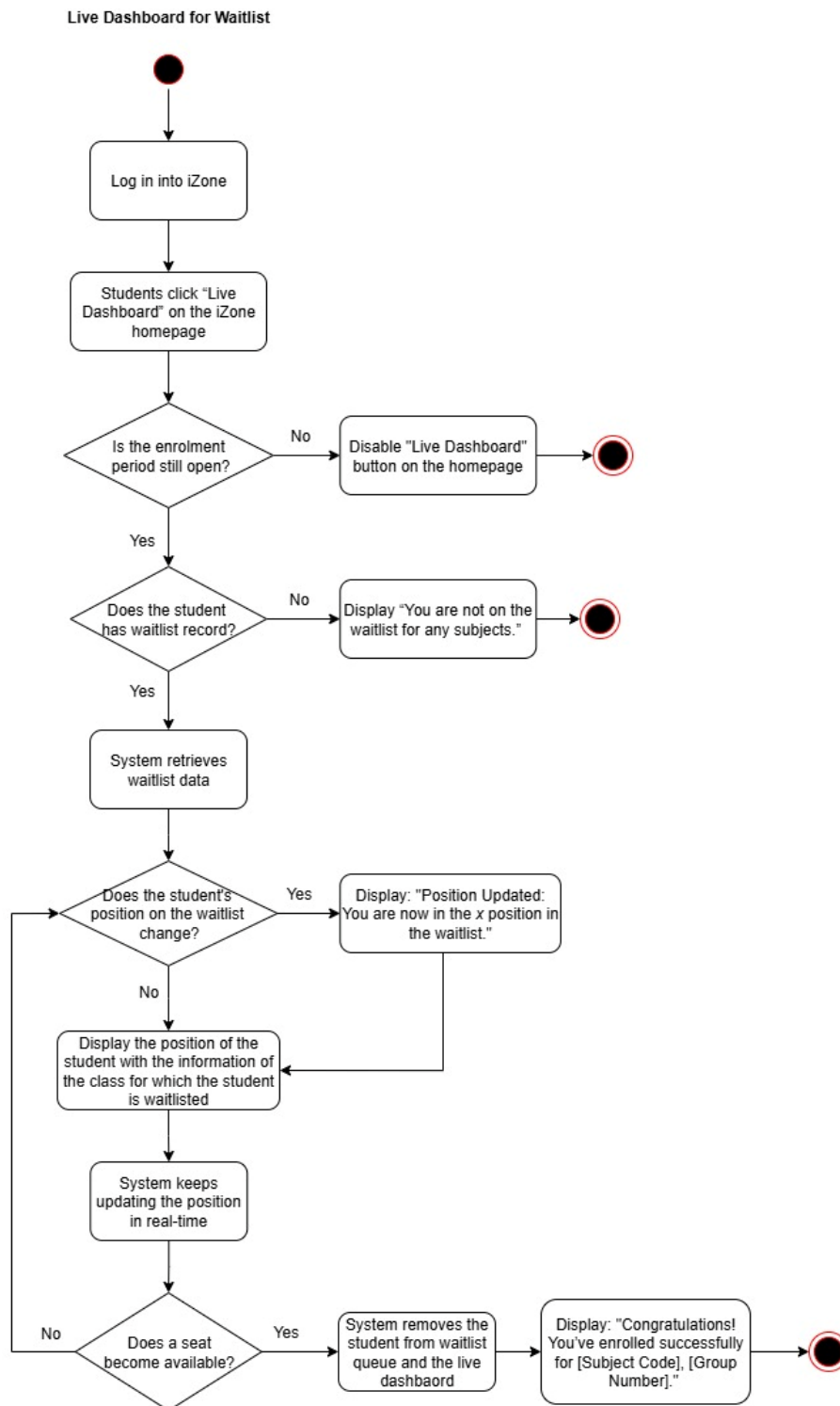
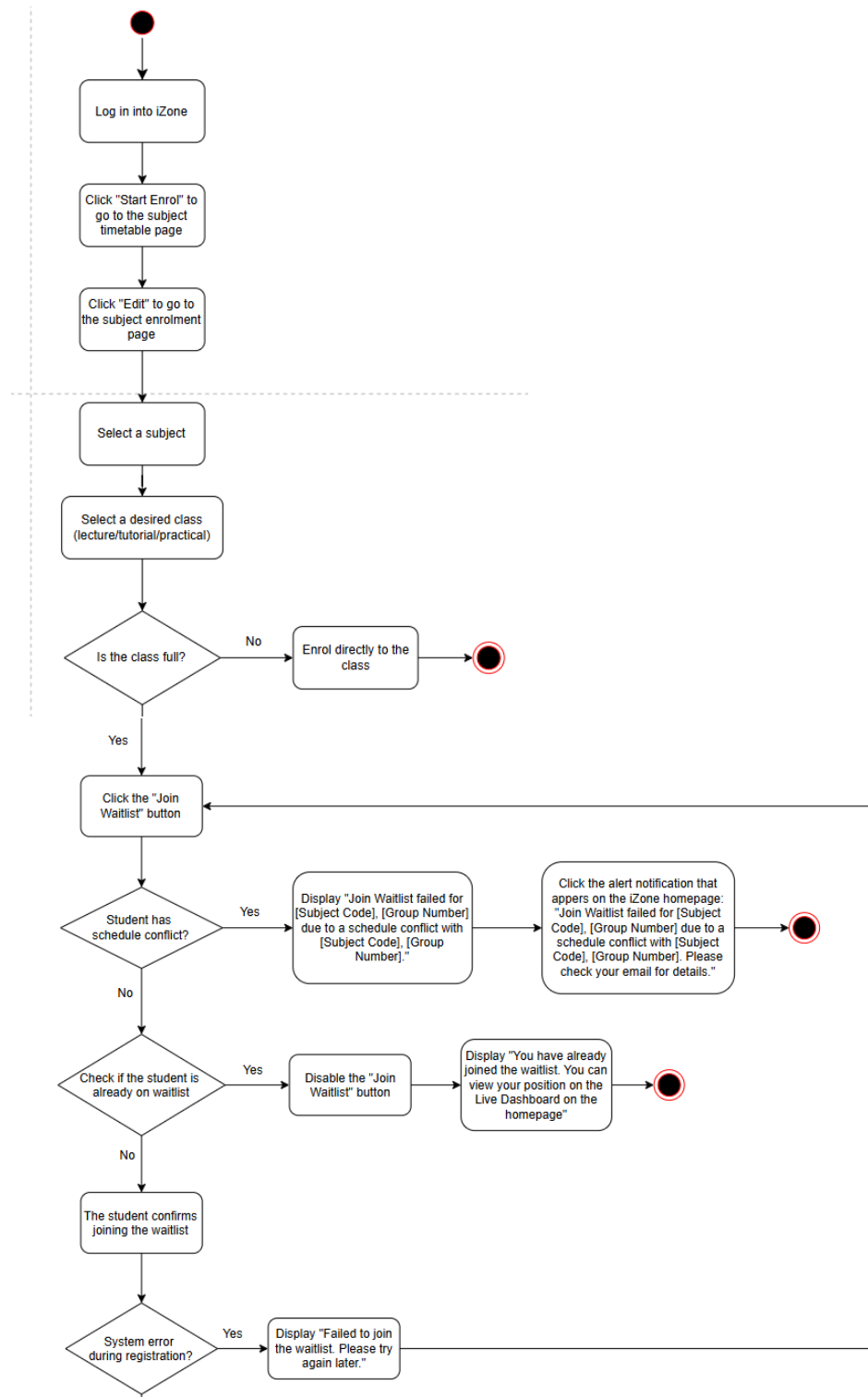
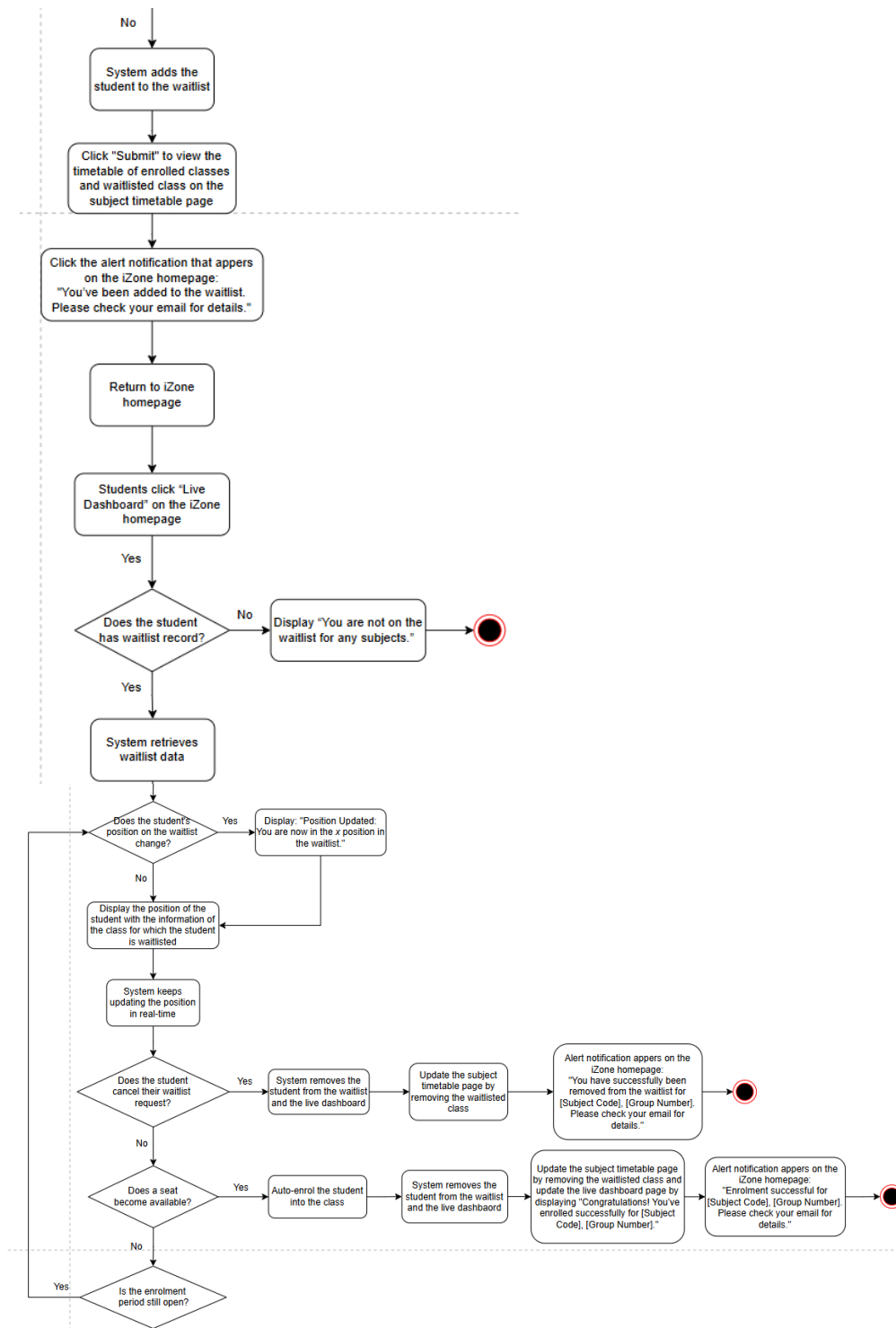
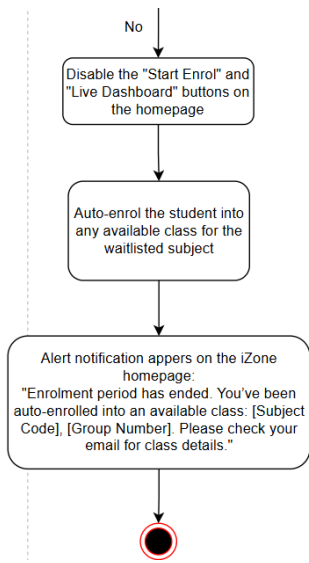


Figure 8: Use Case 3 Diagram

## 2.2.4 Complete Activity Diagram







*Figure 9: Complete Activity Diagram*

### 3.0 Nonfunctional Requirements

Type	Explanation	Example	Result
Performance	The system is expected to be highly responsive and capable of operating in real time. Automatic seat assignment and real-time status updates are a couple of the most critical capabilities. These features required the capability of the system to manage changes immediately and send out accurate data	If a student withdraws from a course and later a seat becomes available, the system will pick up the next eligible student on the waitlist and issue an Outlook e-mail notification. This quick response ensures that students do not miss important information and receive it on time.	This will significantly improve the effectiveness and fairness of the enrolment process. The result is a more efficient registration process, particularly during peak enrolment periods.
Scalability	It requires the system to be capable of handling a long-term rise in both user and data volume. This is because Sunway University's student population increases, and more classes are offered. Therefore, the	The system should remain stable and accessible when thousands of students are registering and modifying their schedules during the first week of the semester. The	A scalable system guarantees that future expansions such as offering cross-campus enrolment or new faculties, which can be supported without major

	system should continue to perform without lag or crashes.	architecture should support parallel processing and load balancing to ensure performance is not compromised.	rework or downtime.
Security	The system must protect student data and restrict unauthorised access. Sensitive information such as student enrolment preferences, personal emails, and academic records must be encrypted and accessible only to authorised users.	The use of institutional email which is Outlook ensures secure and verified communication. In addition, the integration with iZone must include role-based access control to ensure that only students can view and act on their own enrolment status, while administrative users can access reporting and management tools.	By implementing strong data protection measures and secure communication channels, the system maintains confidentiality and prevents potential misuse or data breaches.

Reliability	<p>It is important for ensuring that the system works consistently. The system should have minimal failure and be highly available, especially during peak registration periods when students' usage is the highest. As a result, automatic operations such as seat allocation and email reminders must perform as expected under any situation. Besides, the system must reliably detect the change and automatically assign a seat without any human intervention.</p>	<p>When a seat becomes available in the desired time slot selected, the system automatically allocates it to the next eligible student on the waitlist.</p>	<p>This dependable operation will boost user trust and ensure that all users may rely on the system to perform its intended functions without interruption or error.</p>
Availability	<p>The waitlist system must be highly available during the subject enrolment periods. Students rely on the system to join waitlists in real time,</p>	<p>- The waitlist system on iZone platform should achieve at least 99.9% uptime during the subject enrolment period.</p>	<p>This ensures students can always access the waitlist system and reduces the risk of missing important class time slots.</p>



	<p>often within short and high traffic windows. Any downtime could result in missing enrolment opportunities and creating frustration.</p>	<p>- The system should have redundant server infrastructure (backup systems and processes to ensure operations can continue even if a failure or disruption occurs) to ensure uninterrupted service even during unexpected issues.</p>	
Usability	<p>Most students interact with the system under time pressure during the subject enrolment periods, so the waitlist system must be intuitive and user-friendly to minimize confusion and errors.</p>	<p>- A clearly labelled "Join Waitlist" button placed next to the subject's preferred time slot.</p> <p>- A live dashboard for waitlist is prominently displayed on the iZone homepage, with clear headers such as 'Live Waitlist Dashboard' to allow students to quickly find their</p>	<p>This improves the user experience, reduces frustration, and increases satisfaction among students.</p>

		position on the waitlist."	
Maintainability	The system must be easy to maintain and update. Over time, university policies, class structures, or student needs may change. The system must be adaptable so that developers can easily modify or extend its features without disrupting current operations.	The codebase of the system should use a modular design with clear documentation. It means that the code should be split into clear sections and each section should be well explained. This makes it easier for developers to do any updates for the waitlist system.	Easier to update the system in the future, saves time and cost for maintenance.
Portability	Students may use different devices including laptop, tablet and phone or operating systems including Windows, macOS, Android and iOS to access the iZone platform. Therefore, the waitlist system must be accessible across	The waitlist system should use responsive web design and be compatible with all browsers such as Chrome, Firefox, Safari and Edge, and devices without requiring installation.	This enables more students to access and use the system regardless of their devices.

	various platforms and devices.		
--	--------------------------------	--	--

## 4.0 Requirements Reviews

Requirements	Completed by	Reviewed by	Review Results
Describe what kind of system we are proposing and who needs the system.	Yong Le	Azaliya	Approved
Justify the need for the system and explain how the proposed system can improve user's activities	Azaliya	Yong le	Approved
Research and review an existing or similar system developed for another organization	Wen li	Yong le	Approved
Research and identify a suitable development methodology	Wen li & Azaliya	Qi Yung	Approved
Develop a Gantt Chart using Excel and create task breakdown	Hui San	Yong le	Approved
Construct a PERT chart and identify critical path	Yong Le	Keertana	Approved
Design the overall use case diagram with justification	Qi Yung	Hui San	Approved
Describe Use Case 1: Join subject waitlist	Keertana	Qi Yung	Approved
Create Activity Diagram for Use Case 1	Qi Yung	Keertana	Approved
Describe Use Case 2: Auto enrolment with notification	Hui San	Wen Li	Approved

Create Activity diagram for Use Case 2	Wen li	Hui San	Approved
Describe Use Case 3: Live dashboard for waitlist	Keertana	Wen Li	Approved
Create Activity Diagram for Use Case 3	Qi Yung	Keertana	Approved
Create the overall, complete activity diagram	Qi Yung	Hui San	Approved
Define the non-functional requirements	Yong le & Qi Yung	Keertana	Approved