School of Computing and Artificial Intelligence

Faculty of Engineering and Technology

Sunway University

Final Report

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

DATE OF SUBMISSION	09/08/2025
ACTUAL DATE OF SUBMISSION	09/08/2025
NO OF DAYS LATE	0

Table of Content

Contents

Smart Waitlist System for Enhancing iZone Subject Enrolment

Keertana A/P Subramaniam
23109614

Ayu Wen Li
22017867

Azaliya Turlubayeva
22046015

Liu Yong Le
22064141

Siow Qi Yung
22053037

Wong Hui San
22034540

ABSTRACT

The iZone Subject Enrolment is a portal that is used by Sunway University students to register for their subjects. However, students often missed their preferred class due to the limited class availability especially during a new semester. Therefore, this problem will cause poor registration experience. Having an efficient enrolment system is important in academic to ensure fairness and reduce stress for both students and administrators. Hence, the university should implement a system that is more efficient and fairer to allow students to have a smoother registration process. The existing system does not have a proper solution to the problem of student dissatisfaction due to full classes. It lacks queueing system and causes students to miss their preferred classes due to limited class availability. Students have to log in to iZone manually and check for class availability on numerous occasions. This problem negatively affects both students and administrators from the university. Therefore, the iZone Waitlist System was developed as an upgrade to the existing platform to solve this problem. There are some key features such as automated waitlist management, a live dashboard, and alert notifications. These features improve the fairness of the enrolment process and provide real-time information. The system was built using the Waterfall Model, which involves multiple stages such as requirement analysis, system design, implementation, testing, deployment, and maintenance. In this project, a usability test was carried out with ten users, including Sunway students, non-Sunway students, and a lecturer. The results show that more than 90% of respondents thought the system was well-integrated and intuitive. The live dashboard and alert notifications helped to reduce the need for manual checking. By implementing the system, it can increase the registration efficiency. Besides, students can also reduce their stress especially during subject enrolment time. The implementation of the system will also enhance the school's digital image in information systems. The system not only solves the subject enrolment issues but also reflects the user-centered system development concept. Lastly, the system's scalability allows future integration into more campus digital infrastructure. Hence, the proposed system has combined the technology with user experience effectively. As a summary, the Smart iZone Waitlist System represents a meaningful advancement in Sunway University's subject enrolment process.

Key Words: automation, dashboard, notifications, waterfall model, usability testing

1. INTRODUCTION

Student enrolment systems are crucial for guiding course registrations at universities - they help students pick subjects, secure time slots, and build their schedules. In today's higher education, especially when the registration rush hits, systems need to be both quick and dependable. They must support a surge of simultaneous users while ensuring that limited places are allocated fairly.

At Sunway University, the iZone portal is the gateway for subject enrolment. Here, students can check class offerings, sign up for subjects, and adjust their schedules. The portal handles these tasks well

enough for routine use, but it misses some of the intelligent automation now seen in peer institutions, such as automated waitlists, updating slot availability in real time, and applying First-Come First-Serve (FCFS) rules. The result is that students often must compete hard for their first-choice classes, especially when it comes to sought-after electives or full tutorials.

Integrating smarter automation throughout the enrolment workflow could sharpen the overall experience, lessen the load on staff, and make slot distribution more equitable. It is that gap, and its promise, that motivated us to develop the *Subject Enrolment Waitlist Management System*.

iZone registration currently lacks a waitlist that updates automatically. Once a course reaches capacity, students must log in repeatedly to refresh the course page, risking a missed slot whenever a delay or server overload occurs. Without a queue that reserves spots in the order requests arrive, the system wastes time, raises anxiety, and unevenly distributes open slots.

The fallout reaches the entire education ecosystem. Lecturers handle more requests to shift schedules, administrators receive a heavier complaints load, and the IT team must scale the server during heightened registration spikes.

This project will create, test, and iterate a Smart Waitlist System to streamline class sign-up by accomplishing the following goals which are deliver automated waitlist duties so slots in full sections are managed automatically, issue live notifications and slot availability updates, cut down on time students spend monitoring class availability and boost transparency and satisfaction among students, faculty, and administration.

This project zeroes in on refining the student-facing class enrolment pathway in iZone by integrating a streamlined waitlist mechanism. The prototype was crafted in Figma and underwent both pilot testing and usability sessions with a selection of Sunway University students. The defined deliverables are allowing students to join and cancel waitlist entries, automating slot offers when vacancies are confirmed and displaying real-time updates on dashboards and sending alert notifications.

The Smart Waitlist System is poised to reshape the enrolment experience at Sunway University by lowering student anxiety, curtailing the need for administrative intervention, and stabilising backend performance. Primary advantages for students are to gain a clear, objective, and less anxiety-inducing enrolment journey. Next, lecturers face fewer re-scheduling requests and can manage groups with less friction. Then, administrators deal with fewer queries and can work more efficiently. Finally, IT departments experience milder loads during peak enrolment.

The lessons and architecture can inform similar initiatives at other universities looking to update and refine their digital enrolment ecosystems.

2. RELATED STUDIES

The systems development life cycle (SDLC) is a structured process that provides a methodical framework for producing software that is dependable, affordable, and high quality (Gupta et al., 2021). It consists of clearly defined phases where each stage is essential in software engineering process. Those phases are requirement analysis, planning, designing, development, testing, and maintenance. All these phases serve act as a roadmap for software teams, ensuring effective and goal-oriented development. Without this kind of planning, the software development process can become disorganized and prone to mistakes particularly for complex systems. With the help of SDLC models, which offer structure and clarity, it can help teams to manage tasks, resources, and risk throughout the project lifecycle. Different types of SDLC models such as Agile, Spiral, Waterfall, V-Model, and RAD can each offer various strengths and weaknesses, and their selection depends on the nature and requirements of a given project.

One of the SDLC approach is the Agile methodology that emerged due to limitations of traditional, rigid software development processes (Gonzalez Moyano et al., 2022). Agile refers to a group of frameworks and methods based on common principles that prioritize flexibility, teamwork and iterative

development. The Manifesto for Agile Software Development, which lies at the heart of Agile, emphasizes on four key values which are individuals and their interactions over processes and tools, working software over extensive documentation, customer collaboration over contract negotiations, and adapting to change rather than following fixed plan. Together, these principles support a flexible development environment, where changing needs and stakeholders input are consistently incorporated into the software development process. Agile encourages self-organizing, cross-functional teams that quicky adapt to change and produce functional software in brief, iterative cycles unlike traditional models that frequently impose strict processes.

The Agile methodology's strength lies in its flexibility, speed, and continuous improvement (Trivedi, 2021). By minimizing strict pre-project planning, agile facilitates creativity and adaptability and enables development teams to swiftly change course in response to new information of opportunities. This adaptability enables teams to reorganize priorities in the backlog and redefine goals mid-project thanks to this flexibility, which encourages creativity and produces outputs of a higher quality. Agile development's iterative process ensures that testing and feedback are embedded within each cycle, which enables the early detection and resolution of defects and problems and ultimately improving the quality of software. A culture of open communication, a strong work ethic, and regular client engagement are also fostered by Agile's emphasis on stakeholder collaboration, which results in products that are more aligned with user expectations. However, there are disadvantages in using Agile methodology. For stakeholders who prefer set deadlines and clearly defined deliverables, its lack of planning may hinder a clear view of long-term objectives. In addition to delaying projects, frequent direction changes can make project manager's job more difficult. Furthermore, Agile requires highly qualified, cross-functional team members, and it can be logistically straining to keep them focused on a specific task throughout several iterations. Although Agile offers an effective solution to rapidly evolving development, its success heavily relies on the maturity and discipline of the team and stakeholders.

The next SDLC model is Spiral Model which was introduced as a response to the limitations of traditional linear approaches like that Waterfall Model (Risener, 2022). It follows an iterative process that is visually represented as a spiral, where each loop represents a stage of development. Planning, risk analysis and mitigation, development, and review and evaluation are the four main phases of the model. Developers work with the client to establish initial requirement and lay out the overall development plan during the planning phase. The next step is risk analysis and mitigation, where the team finds potential risks and develops a prototype to deal with them beforehand. In the development phase, actual coding occurs, integrating prototype feedback to better fulfil the needs of the client.

The Spiral Model is appealing due to its many advantages. Its adaptability to changes during the development cycle guarantees that changing client needs can be successfully met (Alazzawi et al., 2023). The developers and clients can interact with iterative system versions in the early development process thanks to the model's ability to handle many prototype. This improves the overall quality and usability of the completed product by increasing requirement accuracy and offering chances for early user feedback. Furthermore, the model is excellent at-risk management and is a good fit for high-risk projects due to is specific phase for risk identification and mitigation. However, there are disadvantages in suing the Spiral Model. The complexity of managing the model is one of its primary disadvantages, especially due to its iterative nature and the requirement for repeated planning, risk analysis, and evaluation stages. Furthermore, the indefinite looping may cause ambiguity regarding the project's completion date, particularly if there are n clear stopping criteria. The steps involved in each phase can be complex and difficult to coordinate, which could lengthen the development period. Additionally, the model requires extensive documentation of intermediate stages, which can be time-consuming and burdensome for development teams, especially those with tight deadlines or limited resources.

The next SDLC is Waterfall Model, which is also known as the traditional model. The Waterfall Model is a linear and sequential structure where each phase must be fully completed before the next begins (Guminski, 2023). Usually, the model progresses through phases like gathering stakeholder requirement, analysis, program design, testing, final deployment and maintenance. It emphasizes a clear and structured approach that leaves minimal opportunity for phase for overlap. Although the strict sequencing reduces misunderstandings during development, it provides little room for changes after a phase is completed. Since verification usually happens later in the process, errors made early in the process are often carried forward.

The Waterfall model is a reliable option due to its many benefits. Clear project planning, milestone definition, and time estimation are made possible by its linear and sequential nature, which promotes a highly structured and predictable development process (Pargaonkar, 2023). Every stage of

development results in comprehensive documentation that helps with future maintenance and system updates in addition to support clarity during development. Fixed requirements and clearly defined deliverables also simplify project management by making it easier to assign resources and monitor progress. Because of these characteristics, the Waterfall Model is particularly suitable for smaller projects with stable requirements that are unlikely to change. Nevertheless, this model does have significant limitation despite its advantages. Due to its strict phase structure, it can be difficult to go back and modify earlier work once a stage is finished, which could disrupt the project's timeline as a whole. Additionally, after the initial requirements gathering, there is typically little customer involvement, which limits the opportunity for ongoing feedback and iteration. This lack of adaptability raises risk of delays and costly rework since important problems might not be identified until much later in the testing process. The Waterfall Model is less effective in dynamic development environments because it frequently fails to adapt to larger or more complex projects with changing requirements.

3. METHODOLOGY
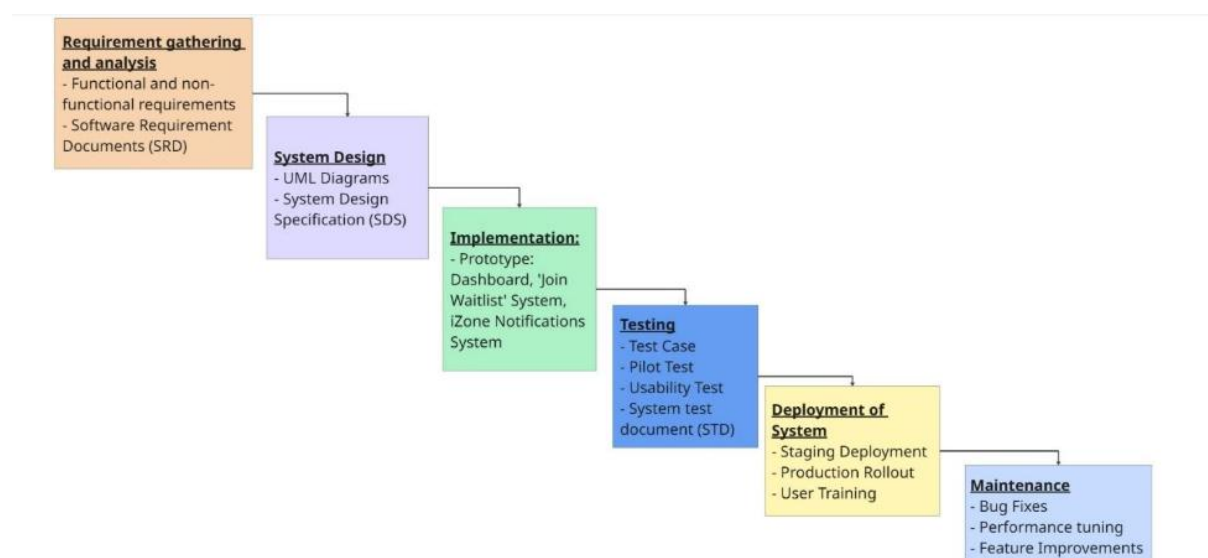
The SDLC method that was applied was the Waterfall Methodology which uses a sequential manner to develop software. This method is particularly suitable for projects that are typically complex as it is broken down into different stages with their own set of tasks and deliverables. Waterfall Methodology is typically utilised due to its systematic and organised nature (Mishra & Alzoubi, 2023). This makes it the ideal SDLC method for this project which involves extensive deadlines and distinct checkpoints as well as the lack of need for client interaction (GeeksforGeeks, 2018). The Waterfall Methodology is defined by the following stages which are Requirement Gathering and Analysis, System Design, Implementation, Testing, Deployment of System and finally, Maintenance.

During the Requirement Gathering and Analysis phase, our objective involved defining the functional and non-functional requirements. The Functional requirements included Automated Waitlist Management, Real-time monitoring, Notification System and the Live Dashboard. The key non-functional requirements that the Automated Waitlist System should have includes performance, scalability, reliability, usability, availability, security, maintainability and portability. The outcome of this phase is the Software Requirement Documents (SRD).

The next phase is System Design which will produce the System Design Specification (SDS) document containing the detailed and high-level and low-level architectural design of the project. The SDS contained the following UML diagrams: Data Dictionary, Use Case diagram, Sequence Diagram, and State Transition Diagram. Data dictionary refers to a compilation of information related to the information system or database, also known as the metadata. This contains the attributes, names, structure related to the data (UC Merced Library, 2023). Meanwhile, use case diagrams are used to outline the intended outcome of the system, thus making sure that the system is designed from the user's viewpoint (Visual Paradigm, 2019). Sequence diagrams are used to demonstrate the interplay

between different portions of the system interacts and the order of sequence of the interactions (Athuraliya, 2022). Finally, the last UML diagram to be created is the State Transition Diagram which illustrated the different interfaces a system has as well as the changes in between the interfaces (Topic 4.2.3 State-Transition Diagrams, n.d.).

In the third stage of the Waterfall Methodology, the prototype of the system was implemented and designed using Figma. Each of the elements such as the Waitlist System, Live Dashboard and Alert Notifications were developed and tested to ensure the reliability and well integration of the system. By the end of this stage, a fully functioning waitlist system was created and ready to be tested.

After the waitlist system was developed, the fourth phase was to conduct the testing in order to ensure that the system was working as expected. The first test that was conducted was the Use Case test to confirm that the system would meet functional requirements based on real-world situations. The next test was the Usability test to verify and review the ease of use of the system. The final test is the pilot test which involves selecting a small group of users to test the system in realistic scenarios in order to review the performance and identify any further issues with the system.

The last two phases are deployment and maintenance which we will not be conducting as we have only developed a prototype of the system, in line with the assignment-based nature of this project. In real life scenarios, deployment involves deploying the system to be used in a real-life environment. The system will first be deployed in a staging environment which is a setting that is closely similar to the actual production setting. This helps identify any remaining issues before the system undergoes complete rollout. Meanwhile, maintenance involves addressing any issues reported by the user. The system performance is maintained to ensure faster system responses and introducing new features that are needed. Ongoing updates and consistent monitoring will ensure the system remains secure, efficient, and meets the user expectations. The primary goal of this is to maintain a reliable and continuously improving waitlist management solution.

4.   RESULTS

A questionnaire consisting of two main sections was created using Google Form. The first section includes questions about the output users received after testing the use cases, this is to make sure the actual output matches the expected output. The second section is the System Usability Scale (SUS), a 10-item questionnaire. We used the SUS questions by UW Alacrity Center (UW ALACRITY CENTER, n.d.)

A total of 10 users participated in the usability testing of the prototype. The users came from different backgrounds including current Sunway students, non-Sunway students and a lecturer. This is to ensure feedback could be collected from various perspectives. The following are selected key findings that are most relevant for analysis in this report.

For questions regarding the output of the use cases, most users obtained actual outputs that matched the expected results. However, there was one test case about system error where many users' outputs did not align with the expected output.
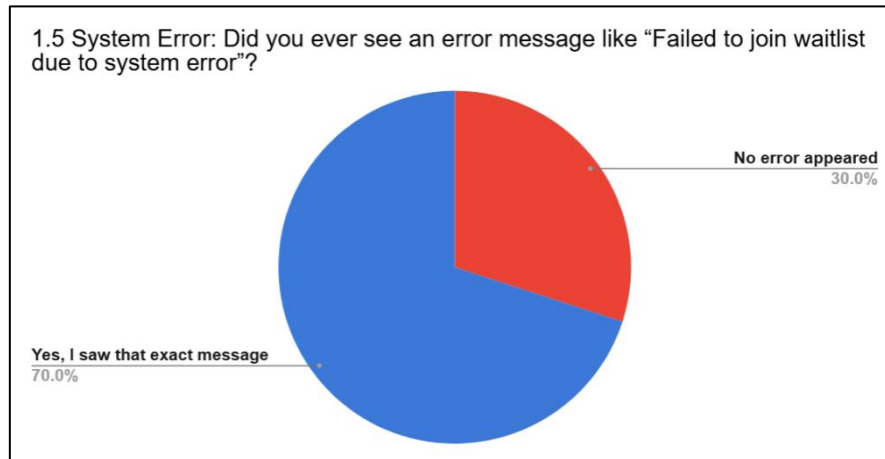
Figure 1: Respondents' Feedback on System Error Message

Figure 1 shows that some users failed to get the expected output for the System Error test case under the Join Waitlist use case. This test case was designed to demonstrate the system's ability to handle error by displaying a system error message when users attempt to join the waitlist for a time slot. 30% of the users stated that they did not see any error message being displayed. This could be due to the System Error test case was set up in a separate flow in Figma. Although instructions were provided in the description to guide users to the correct flow for testing the error scenario, some users failed to notice this information or guidance, as a result, they did not experience the intended outcome.

Moving on to the SUS 10-item questionnaire. The questionnaire covers three key aspects of usability which are effectiveness, efficiency and satisfaction. The questions are a mix of positively and negatively worded statements. Below are the positively worded questions in the SUS questionnaire.
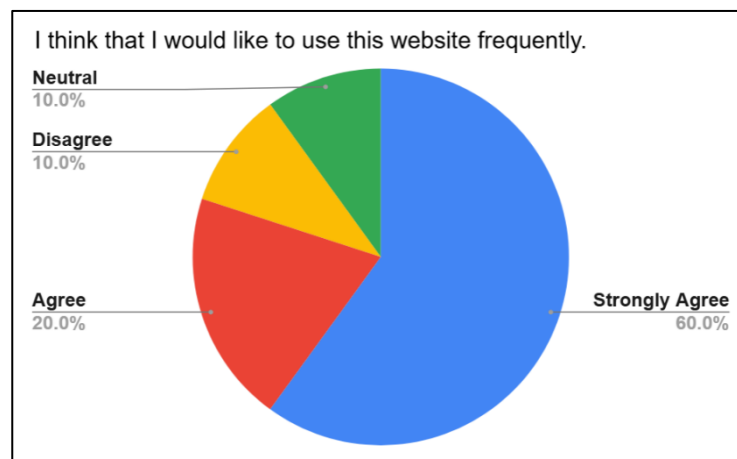

Figure 2: Respondents' Feedback on Willingness to Use the Website Frequently

Figure 2 shows that 80% of the users stated they would likely use the iZone subject enrolment portal with the smart waitlist system frequently. This indicates that users who have tested out all the features found the system helpful in solving the subject enrolment issues. 10% of the users remained neutral while another 10% disagreed with the statement. These responses suggest that there is still room for improvement in terms of the feature concept and enhancing the overall system design.
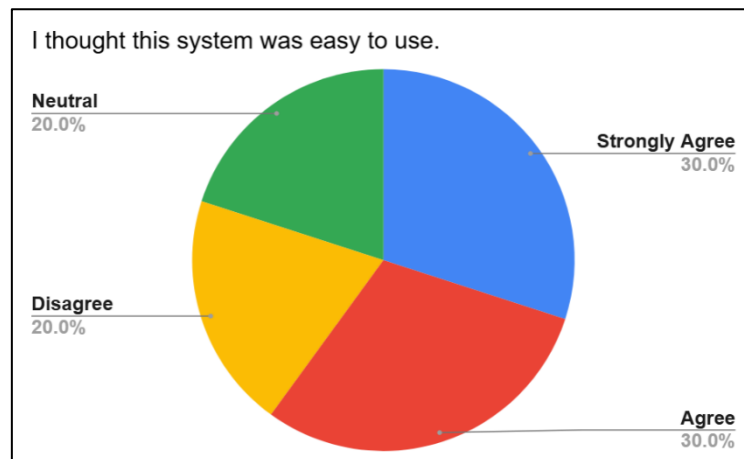
Figure 3: Respondents' Feedback on Ease of Use of the System

Figure 3 shows that 60% of the users strongly agreed or agreed that the system was easy to use, 20% were neutral and 20% disagreed with the statement. This indicates that some parts of the system may be difficult for users to navigate, possibly due to the newly added features and changes in the flow. For example, the live waitlist dashboard is a new element. More guidance may be needed to help users navigate the system more effectively.
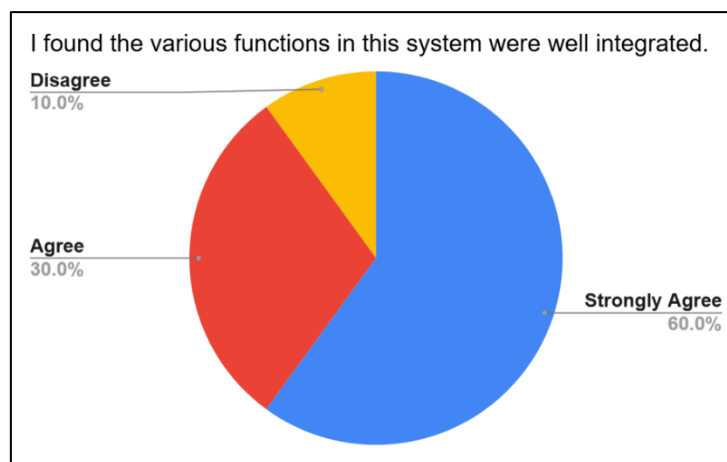

Figure 4: Respondents' Feedback on System Function Integration

Figure 4 shows that 90% of the users stated the functions in the system were well integrated. This indicates they experienced smooth navigation and seamless transitions between different use cases and features with buttons functioning as intended. Most of them encountered no error or issue when using the system's features.
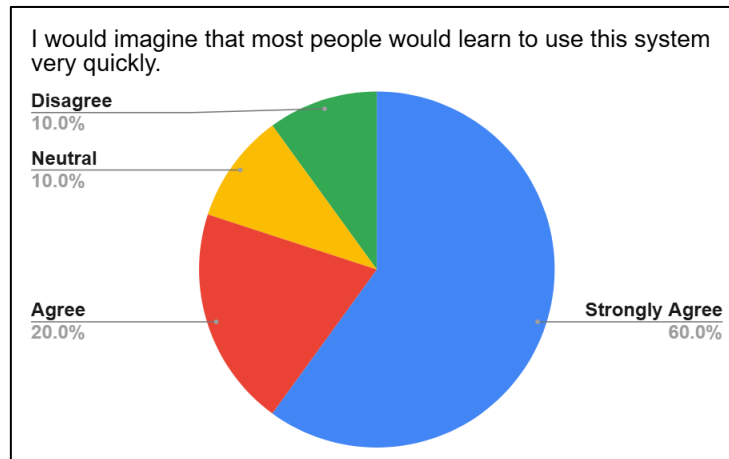
Figure 5: Respondents' Feedback on System Learnability

Figure 5 shows that most users agree it is quick to learn how to use the system. This means that users perceived the system as generally easy to learn, not only for themselves but also others. Some features or navigation flow may seem unfamiliar at first, but with minimal effort and guidance, people would be able to pick up how to use the system quickly.
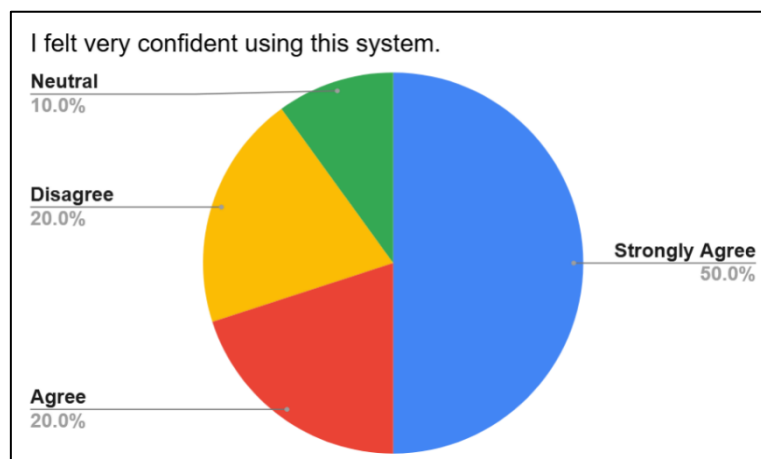

Figure 6: Respondents' Feedback on Confidence in Using the System

Figure 6 shows that 70% of the users stated they felt confident using the system. This indicates that the system flow is smooth and intuitive with minimal errors that could cause confusion. As a result, users feel confident in navigating to the correct flows and accessing the intended features.

Below are the negatively worded questions included in the SUS questionnaire.
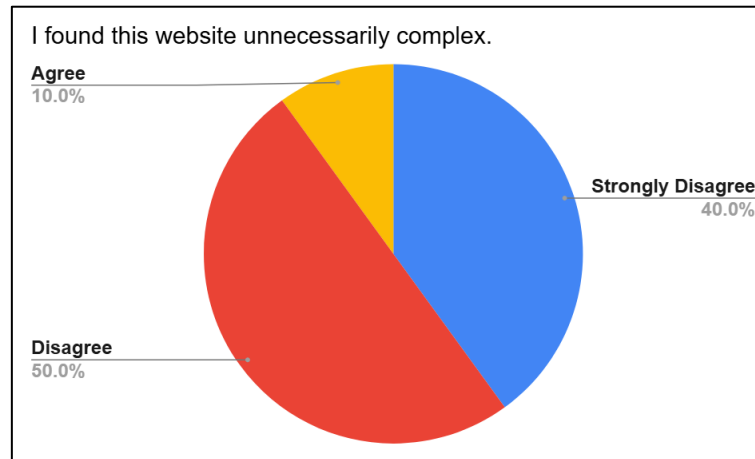
Figure 7: Respondents' Feedback on System Complexity

Figure 7 shows that most of the users strongly disagreed with the statement saying that the system is complex. This indicates that users found the system straightforward and easy to understand, Users are able to use the features without unnecessary complications.
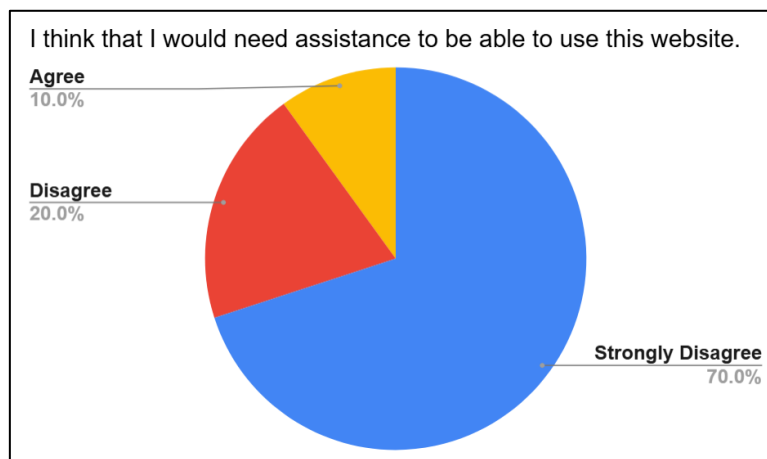

Figure 8: Respondents' Feedback on Using the System Without Assistance

Figure 8 shows that most of the respondents stated they can use the system without needing assistance. This shows that the system has a clean and intuitive interface, for example in terms of the buttons placement, layout of the design, and the instructions or guidance provided. This made it easier and more convenient for users to navigate the system even if they had never used the iZone subject enrolment portal before.
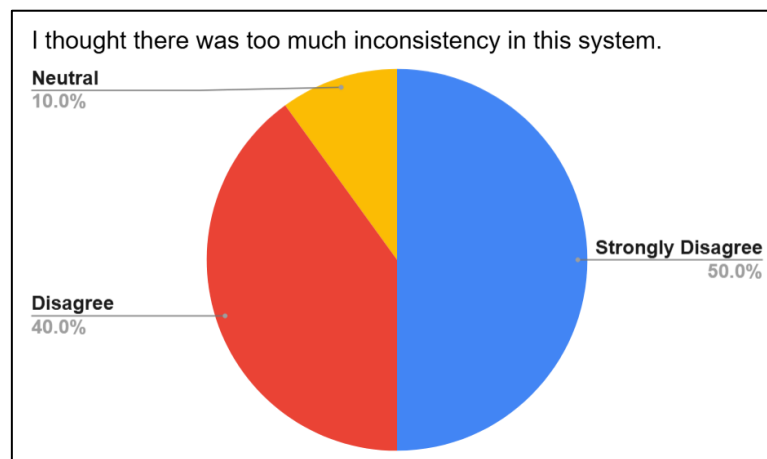
Figure 9: Respondents' Feedback on the Consistency of the System

Figure 9 shows that most users disagreed with the statement saying that there is too much inconsistency in the system. This means that the system has maintained its consistency in aspects like button placement, screen layout, flow of information and the use of colours. The consistency provided a smoother user experience by making navigation more predictable.
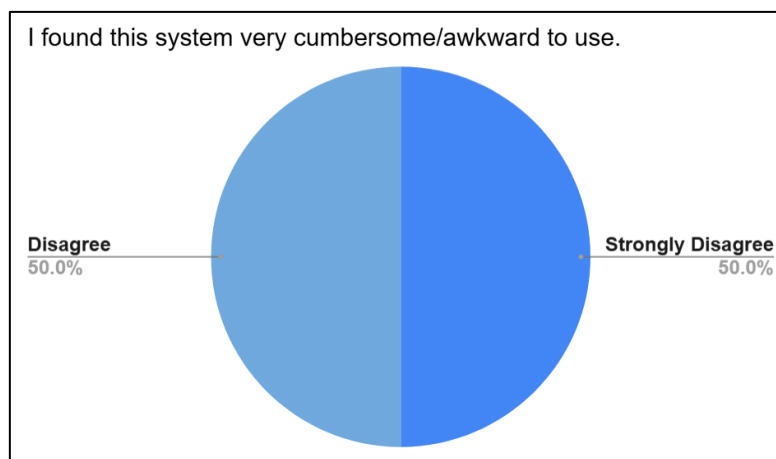

Figure 10: Respondents' Feedback on How Awkward the System Was to Use

Figure 10 shows that all users disagreed with the statement saying the website was awkward to use. This shows that the system is free from unnecessary or confusing elements that could disrupt the user experience. The navigation is simple and clear, allowing users to interact with the system at ease.
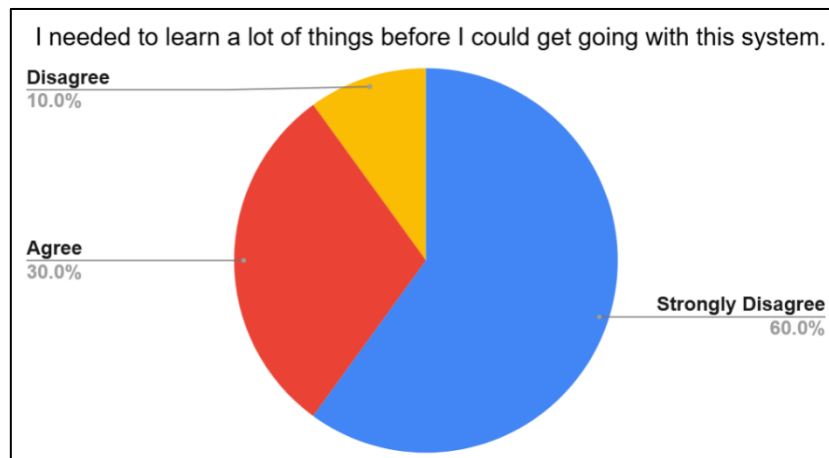
Figure 11: Respondents' Feedback on the Learning Effort Required to Use the System

Figure 11 shows that 30% of users agreed that they needed to learn a lot of things in order to use the system. The responses suggest that the system is highly user-driven, requiring users to manually interact with each feature. The system can be improved by automating and simplifying certain processes to reduce the learning curve.

In conclusion, by calculating the score of the SUS questionnaire, the system received a score of 80.75, indicating an excellent usability (Hadi Alathas, 2018). This proves that the system is effective in addressing the identified problems, efficient because of its user-intuitive design and provides a satisfying experience for users.

5. DISCUSSION

The existing iZone enrolment platform at Sunway University operates on a first-come, first-served basis. Once a class reaches full capacity, there is no structured queueing system in place. This causes frequent manual checking, confusion, and missed opportunities. In response to these challenges, the iZone Waitlist System was developed to introduce a more automated, transparent, and student-friendly enrolment process.

The Smart Waitlist System introduces several key features. Firstly, the automated queue management allows students to join a virtual waitlist for a full class. When a seat becomes available, it is automatically assigned to the next eligible student. This eliminates the need for constant manual monitoring and ensures a fairer and more transparent process. Secondly, the system includes a Live Dashboard that allows students to track their real-time position in the queue. This is further supported by automated in-app alerts via iZone and email notifications via Outlook for events such as joining the waitlist, seat assignment, schedule conflicts, cancellations, and position changes.

Based on the pilot test conducted with three students from Sunway University, we received feedback suggesting further improvements to the waitlist system. Firstly, the system improved the visibility of the notification alert bell icon. Due to the small size of the alert icon in the initial prototype, users would frequently miss notifications. The final version addressed this issue by increasing the icon's size and marking it with a red dot and number indicator. This helped improve responsiveness and ensured that students did not miss key notifications such as waitlist confirmations and auto-enrolment updates.

Another substantial usability improvement was the alert message for schedule conflicts. In the earlier version, the message lacked clarity because it did not indicate which specific subjects and groups were in conflict. The final version resolved this by clearly stating the subject and group involved, which significantly improved the user experience. Moreover, the Live Dashboard received a major enhancement. Previously, students had to refresh the dashboard manually to check for any changes to their waitlist status. This was improved with the introduction of an auto-refresh feature, which increased real-time responsiveness and eliminated the need for manual actions by users.

After that, usability test was conducted on the final prototype with 10 testers, including Sunway University students and a lecturer. The results shows that the system received a score of 80.75, indicating excellent usability. According to Hadi Alathas (2018), a score above 80.3 is considered excellent. This strong positive response shows that students are satisfied with the added convenience, clarity, and automation provided by the system. The high satisfaction rate can be attributed to the improvements made in the final version compared to the initial prototype.

These enhancements significantly improved the usability, responsiveness, and overall user experience of the system. Usability testing confirmed this, with participants reporting seamless navigation, clear notifications, and dependable enrolment functionality. The success of the final prototype reflects a user-cantered design approach, where continuous feedback was collected and integrated into system improvements. This collaborative strategy between designers, developers, and end-users ensured that real student needs were effectively addressed. By incorporating feedback into iterative design changes, the team streamlined the system and aligned it with Sunway University's digital transformation efforts.

In summary, the Smart iZone Waitlist System represents a meaningful advancement in Sunway University's enrolment process. It addresses core pain points such as the lack of structured queueing and real-time updates. The positive response from users not only validates the design choices made but also emphasizes the importance of technology-driven self-service tools in improving the academic experience.

6. CONCLUSION

The iZone Waitlist System has provided an effective solution to an issue that students often faced during subject enrolment period. The issue that students are facing currently is the difficulty in students enrolling in their preferred classes due to limited slots available. The aim of this system is to ensure that students can have a fairer and smoother enrolment process. In this project, we have identified the system requirements to understand user and business needs. In addition, we have successfully designed an effective system solution that align with the identified requirements. The system includes some relevant features such as real-time waitlist management, live dashboard, and alert notifications. By implementing the system, students can reduce their stress during the subject enrolment process.

Through this project, we have learned the significant procedures of developing a good system. For example, we have gained valuable insights into user-centred design, system requirements analysis, and the importance of iterative testing to enhance usability. We also discovered that solving user problems such as limited slots requires more than just technical functionality, but also an intuitive interface with transparent details. For instance, users may feel confused if they could not understand their waitlist position clearly although the system has an automated waitlist system. Therefore, having a user-friendly interface is very essential in developing a system. Besides, we have enhanced our technical skills, improved understanding of the topic, and gained insights during problem solving throughout the project. We have received a lot of feedback from supervisors and peers that help in improving our system design.

As a conclusion, we have conducted system testing to ensure functionality, accuracy, and reliability. The iZone Waitlist System has successfully solved the identified challenges by integrating the relevant features such as automated waitlist system, live dashboard, and alert notifications. This proposed solution can improve user satisfaction by allowing students to have a smoother enrolment experience.

It also improves user trust due to the reduction of users refreshing the website repeatedly. In addition, the system enhances the efficiency of management as the administrators can focus on more important tasks. This is because they do not have to worry about the subject enrolment issues anymore. By using our system, it enables students to join a waitlist and get auto enrolled with real-time updates as well as eliminates manual checking. Therefore, students can enjoy a fairer and more efficient subject enrolment process while reducing administrative pressure on staff.

7. REFERENCES

ALazzawi, A., M.Yas, Q., & Rahmatullah, B. (2023). Iraqi Journal for Computer Science and Mathematics. *4*, *4*, 173–190. https://iasj.rdd.edu.iq/journals/uploads/2024/12/09/d3c0e5df63ec3b0cab2c4ed5fedf3a6a.pdf

Albarzanji, A. K., & Alsabawy, A. Y. (2021). Causes of IT Project Failure: A Systematic Review. TANMIYAT AL-RAFIDAIN, 40(132), 135–168. https://doi.org/10.33899/tanra.2021.170354

Athuraliya, A. (2022, June 20). Sequence Diagram Tutorial – Complete Guide with Examples | Creately. Creately.com. https://creately.com/guides/sequence-diagram-tutorial/

Baghizadeh, Z., Cecez-Kecmanovic, D., & Schlagwein, D. (2019). Review and critique of the information systems development project failure literature: An argument for exploring information systems development project distress. Journal of Information Technology, 35(2), 026839621983201. https://doi.org/10.1177/0268396219832010

GeeksforGeeks. (2018, March 18). Waterfall Model Software Engineering. GeeksforGeeks. https://www.geeksforgeeks.org/software-engineering/waterfall-model/

González Moyano, C., Pufahl, L., Weber, I., & Mendling, J. (2022). Uses of business process modeling in agile software development projects. *Information and Software Technology*, *152*, 107028. https://doi.org/10.1016/j.infsof.2022.107028

Guminski, A., Dohn, K., & Oloyede, E. (2023). Advantages and disadvantages of traditional and agile methods in software development projects – case study. *Zeszyty Naukowe*, *2023*(188). https://doi.org/10.29119/1641-3466.2023.188.11

Gupta, A., Rawal, A., & Barge, Y. (2021). Comparative Study of Different SDLC Models. *International Journal for Research in Applied Science and Engineering Technology*, *9*(11), 73–80. https://doi.org/10.22214/ijraset.2021.38736

Hadi Alathas. (2018, November 19). How to Measure Product Usability with the System Usability Scale (SUS) Score. Medium; UX Planet. https://uxplanet.org/how-to-measure-product-usability-with-the-system-usability-scale-sus-score-69f3875b858f

Mishra, A., & Alzoubi, Y. I. (2023). Structured Software Development versus Agile Software development: a Comparative Analysis. International Journal of System Assurance Engineering and Management, 14(4). springer. https://doi.org/10.1007/s13198-023-01958-5

Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. *International Journal of Scientific and Research Publications*, *13*(8), 120–124. https://doi.org/10.29322/ijsrp.13.08.2023.p14015

Risener, K. (2022). A Study of Software Development Methodologies. *Computer Science and Computer Engineering Undergraduate Honors Theses*. https://scholarworks.uark.edu/csceuht/103

Topic 4.2.3 State-transition diagrams. (n.d.). https://www.cambridgeinternational.org/images/285020-topic-4.2.3-state-transition-diagrams-9608-.pdf

Trivedi, D. (2021). Agile Methodologies. *International Journal of Computer Science & Communication*, *12*(2), 91–100. https://www.researchgate.net/profile/Devharsh-Trivedi/publication/356924683_Agile_Methodologies/links/61b389391d88475981ddb773/Agile-Methodologies.pdf

UC Merced Library. (2023). What is a data dictionary? Library.ucmerced.edu. https://library.ucmerced.edu/data-dictionaries

Ulven, J. B., & Wangen, G. (2021). A Systematic Review of Cybersecurity Risks in Higher Education. Future Internet, 13(2), 39. MDPI. https://doi.org/10.3390/fi13020039

UW ALACRITY CENTER. (n.d.). *System Usability Scale*. https://www.uwalacrity.org/wp-content/uploads/2020/01/SUS.pdf

Visual Paradigm. (2019). What is Use Case Diagram? Visual-Paradigm.com. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

Fact Findings: Interview

Proof of Interview:

Link to Interview Recording:

https://drive.google.com/file/d/17oDEUV2_AtNUTEiyRh7up8uSvHEDc9z-/view?usp=sharing

Feedback from Interview:

Positive insights we got regarding our system from this interview by Mr Tan Boon Kwan and Mr Chin Kok Joon who are the administrators in the IT Services department in Sunway University. They mentioned that the dashboard was a useful tool to help both the students and the admins as it can help manage the enrolment-related tasks even more effectively. The waitlist system was also seen as beneficial potentially but more useful during the add-drop period as students are automatically removed from the classes they enrolled in as they do not meet the criteria during this period. They acknowledged that even though implementing the waitlist system can be complex, its long term value could be worthwhile especially if the return of investment (ROI) is sufficient.

Additionally, since the IT Services department is currently working on improving the server performance and expanding classroom capacity, it aligns well with the system that was proposed which suggests a readiness for future scalability. Mr Tan and Mr Chin also highlighted that it was important to keep the students well-informed during the enrolment period, and the proposed system could support with clearer updates and features that are user-friendly. Finally, subject clashes and enrolment delays are recurring issues that are currently happening in the enrolment period. This suggests that a more intelligent system could ease the load for the administrative side and improve the overall experience for the students.

Contribution Statement

No	Name	Milestones	Contribution	Percentage
1.	Ayu Wen Li	Proposal	- Question (e)	17.57%
		SRD	- 1.1.5 An existing or similar system developed for other organizations - Activity Diagram for Use Case 2 - 1.2 Development Methodology - 4.0 Requirement Reviews - Conducted Interview	
		SDS	- State Transition Diagram	
		STD	- Test Case 1 to 4 for Use Case 2: Auto Enrolment with Notification - System/Prototype Tester - Help document pilot testing - 3.1.4 Colour and Layout Consistency	
		Final Report	- Methodology	
		Presentation	- Design Slides	
2.	Azaliya Turbulayeva	Proposal		10.81%
		SRD	- 1.1.3 Why they need the system - 1.1.4 How the proposed system can improve their activities - 4.0 Requirement Reviews	
		SDS	- Sequence Diagram Description	
		STD	- 1.3.2 Test Case 2	
		Final Report	- Discussion - Introduction	
		Presentation	- Design Slides	
3.	Keertana A/P Subramaniam	Proposal	- Question (d) - System Request	18.92%
		SRD	- 2.2.1 Use Case 1 & 2.2.3 Use Case 3 - 4.0 Requirement Reviews - Conducted Interview	
		SDS	- Context Diagrams - Data Flow Diagrams (Level 1 and 2) - Data Dictionary - Class Diagram	
		STD	- Test Case 1 & 3	

			- UI/UX Design for Live Dashboard - 3.1.1 – 3.1.2 - 3.2 Pilot Testing - 3.3 Usability Testing	
		Final Report	- Related Studies	
		Presentation	- Presenter	
4.	Liu Yong Le	Proposal	- Question (c)	
		SRD	- 1.1.1 What kind of system - 1.1.2 Who needs the system - 1.4.4 PERT Chart - 3.0 Nonfunctional Requirements - 4.0 Requirement Reviews	
		SDS	- State Transition Diagram	17.57%
		STD	- Test Case 1 to 5 for Use Case 1 - UI/UX Designer for Use Case 1 - 3.1.3 Interactive Elements	
		Final Report	- Abstract - Conclusion	
		Presentation	- Design slides	
5.	Siow Qi Yung	Proposal	- Question (b)	
		SRD	- 2.1 Overall Use Case Diagram - Activity Diagram for Use Case 1 and 3 - Nonfunctional Requirements - 2.2.4 Complete Activity Diagram - 4.0 Requirement Reviews	17.57%
		SDS	- Class Diagram - Use Case Diagram with Relationships	
		STD	- Test Case 1 to 4 for Use Case 2 - UI/UX Designer for Notification - Support Figma prototyping for all UI/UX design	
		Final Report	- Discussion	
		Presentation	- Video Designer	
6.	Wong Hui San	Proposal	- Question (a)	
		SRD	- Task Breakdown - Gantt Chart - 2.2.2 Use Case 2: Auto Enrolment with Notification	17.57%

			- 4.0 Requirement Reviews	
		SDS	- Overall Sequence Diagram - Sequence Diagram for Use Case 1, 2 and 3	
		STD	- Test Case 1 to 5 for Use Case 1 - UI/UX Designer for Use Case 1 - 3.1.5 Accessibility & Responsiveness	
		Final Report	- Results	
		Presentation	- Poster Designer - Presenter	

Total = 17.57% + 10.81% + 18.92% + 17.57% + 17.57% + 17.57% = 100%