Project Report

Team Members:
CHAN, Kin Fai
CHOI, Long Hay Martin
CHUKWUMA, Michelle Kamdilichukwu
LAU, Knoon Leong
YUEN, Pak Kiu

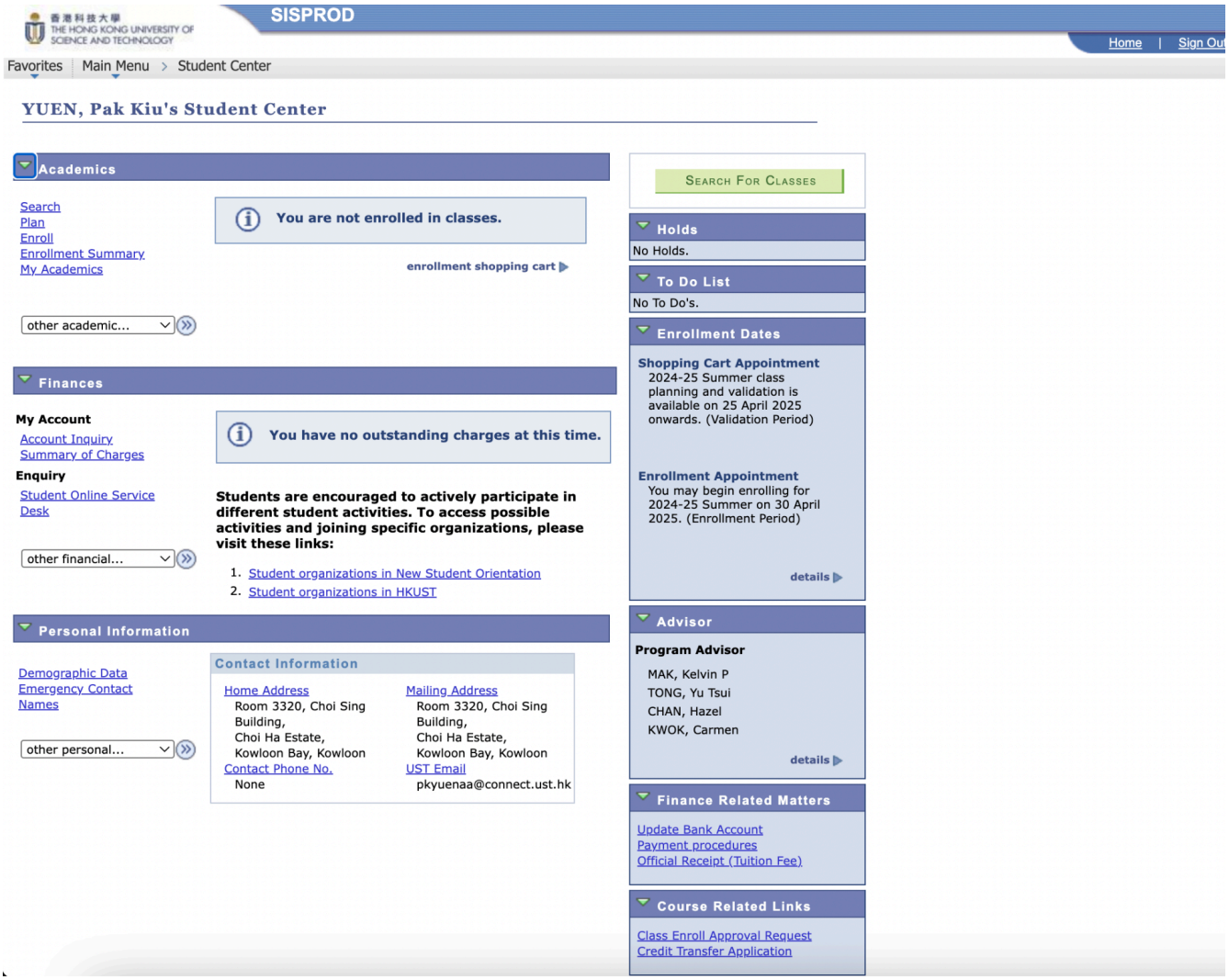
Table of Contents

Introduction **3**
Phase 1 — Develop Business Case & Project Charter **4**
1.1 Business case 4
1.1.1 Poor Usability 4
1.1.2 Scalability Gaps 4
1.2 Project Charter 5
1.3 Stakeholder Assessment Matrix 6
Phase 2 — Collect Requirements **7**
2.1 Key Stakeholder Interviews 7
2.2 Survey Distribution 7
2.3 Scope Statement Creation 7
2.3.1 Define Project Boundaries 7
2.3.2 Manage Stakeholder Expectations 7
2.3.3 Facilitate Decision-Making 7
2.3.4 Risk Reduction 7
2.4 Scope Statement 8
Phase 3 — Create Work Breakdown Structure (WBS) **9**
3.1 WBS Diagram 10
3.2 WBS Dictionary 11
Phase 4 — Develop a Detailed Project Schedule **14**
4.1 Duration Estimate 14
4.1.1 PERT estimate with Planning Poker 14
4.1.2 Delphi Technique 14
4.2 Project Schedule 15
4.2.1 Resource Sheet 15
4.2.2 Summarised Gantt Chart 15
4.2.3 Project Schedule & Detailed Gantt Chart 16
4.2.4 Network Diagram & Critical Path 18
Phase 5 — Create a Risk Management Plan and other Relevant Deliverables **20**
5.1 Risk Identification & Prioritization 20
5.2 Risk Monitoring & Reporting 21
5.2.1 Stage-Gate Meeting 21
5.3 Post-launch support 22
5.3.1 Communication channel 22
Phase 6 — Create Prototypes **23**
6.1 Prototype Features 23
6.2 Prototype of Mobile App 24
Phase 7 — Record Progress **26**
Phase 8 — Change Management/Usability Testing **27**
8.1 Change Management 27
8.2 Usability Testing 28
Phase 9- Closing **30**
Appendix **31**

Introduction

The Hong Kong University of Science and Technology (HKUST) is a world-renowned institution committed to advancing innovation and excellence in education. Central to its operations is the Student Information System (SIS), a critical platform that supports academic and administrative workflows for students, faculty, and staff. SIS enables essential functions such as course enrollment, academic record management, financial transactions, and equipment reservations. However, despite its foundational role, the current system faces significant challenges that hinder user experience and operational efficiency.

The existing SIS struggles with poor usability and scalability limitations. Its outdated interface and fragmented navigation create confusion, particularly during high-stakes tasks like course enrollment. Additionally, the system lacks the elasticity to handle peak traffic periods, leading to latency, crashes, and inequitable access during critical registration windows. These issues not only frustrate users but also strain IT resources and undermine HKUST's reputation as a leader in technological advancement.



Phase 1 — Develop Business Case & Project Charter

1.1 Business case

The current SIS system's limitations threaten operational efficiency and user satisfaction. To address these challenges, this proposal outlines a comprehensive modernization initiative to redesign and redeploy HKUST's SIS. Our project — focuses on 3 core objectives:

1.1.1 Poor Usability

The SIS interface, designed over a decade ago, lacks intuitive navigation and responsive design. Users—particularly students—report confusion during high-stakes tasks like course enrollment, often requiring external guidance or repeated attempts. Additionally, insufficient onboarding resources (e.g., tutorials, tooltips) exacerbate the learning curve, leading to frustration and errors.

1.1.2 Scalability Gaps

During peak periods such as enrollment or exam scheduling, the system struggles with high concurrent user traffic. Frequent crashes, latency, and inequitable access disproportionately affect students with limited device availability, undermining HKUST's commitment to fairness and technological excellence. These bottlenecks also strain IT resources, diverting staff time to troubleshooting rather than strategic improvements.

The proposed solution includes a phased development approach, beginning with a Figma-based UI/UX prototype and progressing to backend optimization and API integrations. By prioritizing stakeholder feedback—through ethnographic research, focus groups, and usability testing—we aim to deliver a system that aligns with the needs of students, faculty, administrators, and the IT department. For instance, students will benefit from a centralized dashboard with quick-access widgets, while administrators gain robust data management tools and communication channels.

This initiative also emphasizes change management to ensure smooth adoption. Training programs, interactive tutorials, and phased rollouts will minimize disruption, while biweekly risk reviews and contingency planning will safeguard against delays or security vulnerabilities. By combining waterfall development practices with stakeholder-centric design, this project seeks to transform SIS into a modern, equitable, and future-ready platform that upholds HKUST's commitment to excellence.

1.2 Project Charter

The project charter formally authorizes the SIS modernization initiative and establishes its governance structure.

Project Charter: HKUST SIS Modernization Initiative

Background

HKUST's Student Information System (SIS) is critical for academic and administrative tasks but faces two key issues: a clunky, outdated interface that confuses users during tasks like course enrollment, and frequent crashes during peak periods due to poor scalability. This project aims to modernize the SIS to improve user experience, ensure reliable access during high demand, and align the system with modern technological standards.

Goals

- Deliver an intuitive, mobile-responsive interface reducing student/staff support tickets by 50% post-launch.
- Achieve 99% system uptime during peak traffic (e.g., enrollment) via cloud optimization.
- Integrate real-time notifications (WhatsApp/SMS) and third-party tools (Canvas, payment gateways) to streamline workflows.

Scope

The project will redesign the SIS interface to work seamlessly on mobile and desktop, with a clear dashboard and step-by-step guides for new users. It will move the system to a cloud platform to handle traffic spikes and add features like waitlists for high-demand courses. The SIS will also connect to tools like Canvas for course materials and send alerts via WhatsApp/SMS.

Excluded: Building new third-party systems (e.g., Canvas) or updating non-SIS university systems.

Key Stakeholders

Client	ITSO
Project manager	Alan
Project team members	Asensio, Alan, Kamdi, Martin, Rovic

Project Milestones

Work begins on 30 May 2025 with a kickoff meeting. The redesigned interface will be finalized and tested by 2 June 2025. Cloud upgrades to manage traffic surges will finish by 1 July 2025, and the full system, including real-time alerts, will launch on 15 August 2025. Payments are split into three parts: 30% upfront, 40% after cloud upgrades, and 30% at launch.

Duration Estimates

3 months

Project Budget

Total cost: HKD 2.5 million, covering design (HKD 800,000), cloud upgrades (HKD 1,000,000), and training/testing (HKD 300,000). After launch, monthly costs for cloud services and updates will total HKD 70,000.

Constraints, Assumptions, Risks and Dependencies

Constraints	• Limited access to legacy SIS codebase for security reasons. • Strict 12-week timeline to avoid disrupting academic cycles.
Assumptions	• Stakeholders (students, faculty) will provide timely feedback during testing. • Third-party APIs (Canvas, WhatsApp) will remain stable and accessible.
Risks and Dependencies	Risks • User Resistance: Mitigated via phased training and onboarding tutorials. • Integration Delays: Contingency buffers added to Phase 3 timeline. Dependencies • Approval from HKUST IT Security for cloud migration. • Availability of third-party API documentation

Approval Signatures

Project Client	Alan, Project Manager

1.3 Stakeholder Assessment Matrix

Stakeholder	nterests / Needs	Influence	Unique information	Role	Approach strategy
Students	Intuitive UI, mobile access, fair course enrollment, real-time notifications.	Moderate	High reliance on mobile devices; time-sensitive needs (e.g., enrollment).	Primary End User	Involve in usability testing; prioritize mobile-first design; gamified onboarding.
Faculty/Professors	Streamlined course management, communication tools, integration with Canvas/email.	High	High Prefer minimal workflow disruption; need grading/attendance tools	Key User & Influencer	Co-design course management features; provide early access to prototypes for feedback.
Administrators	Data accuracy, analytics tools, internal communication channels, audit compliance.	High	Require bulk data processing; value automated reporting.	Decision-maker	Deliver customizable dashboards; align with HKUST's administrative policies.
IT Department	Scalability, security (2FA/encryption), maintainability, API/documentation clarity.	Very high	Concerned about legacy system compatibility and cloud migration risks.	Technical Sponso	Regular technical reviews; involve in penetration testing and load-testing phases.
HKUST Leadership Members	ROI, alignment with institutional reputation, minimal operational disruption.	Very high	Strategic focus on student satisfaction and technological innovation.	Sponsor/ Approver	Highlight metrics (e.g., reduced enrollment time, uptime); provide high-level updates.

Phase 2 — Collect Requirements

To enhance the SIS's functionalities, a comprehensive requirements gathering plan is crucial. This phase involves collecting requirements to understand the needs and expectations of our stakeholders, most important of which are the students, professors, administration and the IT department.

2.1 Key Stakeholder Interviews

The project manager first meets with key stakeholders, including the Head of the IT department, the Principal, student representatives and the Head (Academic Affairs) and Executive Assistant to the Provost. This is to get an idea of what they're hoping to get out of the project so that we can create our list of requirements and scope for the project.

2.2 Survey Distribution

Following the stakeholder interviews, we will distribute a survey (**see appendix 2.1**) designed to gather opinions and feedback through the HKUST email network, allowing all students, professors and administrative staff to voice their opinions on the current state and future of the SIS. This will allow us to gain feedback from a wider audience, as well as personal insight and perspectives from users of the SIS.

2.3 Scope Statement Creation

The key deliverable for this phase is the creation of the scope statement. The scope statement not only outlines the project's purpose, objectives with measurable goals, constraints, assumptions, deliverables, cost estimate and scope, but also serves as the fundamental reference point for all future project activities and decisions. The statement serves the following important purposes:

2.3.1 Define Project Boundaries

The scope statement clearly outlines what is within the project's scope and what is not. By clearly outlining the project scope, it helps prevent ambiguity and ensures a shared understanding of the project's scope and objectives, helping stakeholders understand their involvement.

2.3.2 Manage Stakeholder Expectations

The scope statement also serves to manage stakeholder expectations by providing them with a clear outline of the project's goals, objectives, costs and constraints. This will minimize future misunderstandings between stakeholders, and serve to align stakeholder expectations and goals within the project, reducing possibility of stakeholder conflicts or dissatisfaction.

2.3.3 Facilitate Decision-Making

The scope statement helps facilitate future decision-making, as the clearly defined project scope and constraints serve as an important reference point for all decisions, as it allows the project manager to easily evaluate proposed changes' alignment with project goals before making their decision.

2.3.4 Risk Reduction

The creation of a scope statement also reduces project risks such as overbudgeting or scope creep, or other challenges such as technology, time constraints and vulnerabilities. This is because the statement clearly identifies project scope, budget, constraints and assumptions, allowing stakeholders to plan accordingly and mitigating aforementioned risks.

2.4 Scope Statement

Scope Statement

Project	Project Manager	Date
HKUST SIS Modernization Initiative	Group 12	01 May 2025

Purpose
To improve the HKUST SIS system's accessibility, reliability and user experience.

Business Objectives
Reduce support tickets by 50%, improve user satisfaction ratings by 25%, shorten average loading time by 30% for all users

Scope Description
In Scope
Mobile-responsive interface Reduce technical support tickets Integrating real-time notifications from Whatsapp & SMS Integrating common third-party tool Canvas and FPS vendor Cloud-based load balancing
Out of Scope
Caching & Pre-loading data Breadcrumb navigation Onboarding Tutorial

Project Deliverables
WBS, Project Schedule, Risk Management Plan, Prototype, Usability Report

Constraints
Budget of 2.5 million HKD, maximum project length of 3 months, limited access to SIS codebase,

Assumptions
Timely feedback from stakeholders during testing Third-party tools & apis for Canvas and Whatsapp remain stable and accessible

Cost Estimate				
Item	**Estimated Cost**	**Actual Cost**	**Cost Until Completion**	**Variance**
[Name of resource]	[Dollar figure for line item cost]	[Actual cost of line item]	[Estimated cost of line item for remaining project]	[Discrepancy between estimated and actual}
Interface design	HKD 800,000			
Cloud system	HKD 1,000,000			
Training/Testing	HKD 300,000			

Phase 3 — Create Work Breakdown Structure (WBS)

Phase 1: Project Initiation
- 1.1 Develop Project Charter
- 1.2 Develop Stakeholder Assessment Matrix
- 1.3 Sponsor Sign-off
- 1.4 Kick-off Meeting with Key Stakeholders

Phase 2: Project Planning
- 2.1 Scope Planning
  - 2.1.1 Define Client Requirements
    - 2.1.1.1 Interview with ITSO — the Head of Student & Teaching
    - 2.1.1.2 Define Client Requirements
  - 2.1.2 Define User Requirements
    - 2.1.2.1 Conduct Ethnography Research during summer registration
    - 2.1.2.2 Organise Focus Group Research
    - 2.1.2.3 Define User Requirements
  - 2.1.3 Define Scope Statement
  - 2.1.4 Create Work Breakdown Structure (WBS)
- 2.2 Project Scheduling
  - 2.2.1 Develop Network Diagram
    - 2.2.1.1 Define and Sequencing Activities
    - 2.2.1.2 Estimate Activity Duration
  - 2.2.2 Develop Gantt Chart
    - 2.2.2.1 Define Critical Path
- 2.3 Cost Planning
  - 2.3.1 Estimate Initial Budget
  - 2.3.2 Determine Budget Allocation
- 2.4 Risk Planning
  - 2.4.1 Risk Identification & Prioritization

Phase 3: System Design
- 3.1 Plot Architecture Diagram
  - 3.1.1 Plot Network Architecture Diagram
  - 3.1.2 Plot API Architecture Diagram
- 3.2 Design User Interface
  - 3.2.1 Design Mobile-Responsive Interface
  - 3.2.2 Design Web-Responsive Interface
- 3.3 Outline Database Schema
- 3.4 Design Third-Party API Integrations
- 3.5 Finalize Prototype

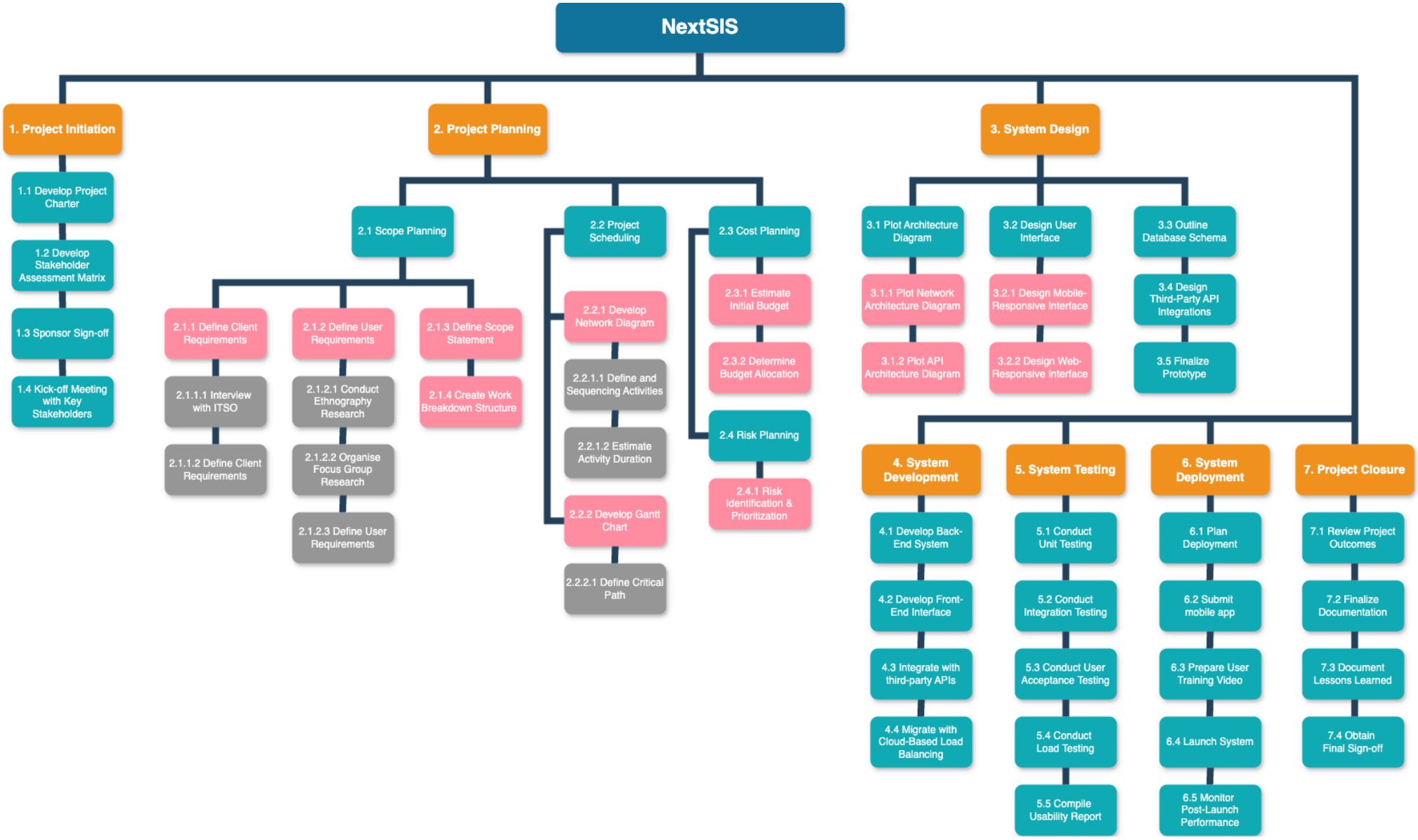
Phase 4: System Development
- 4.1 Develop Back-End System
- 4.2 Develop Front-End Interface
- 4.3 Integrate with third-party APIs
- 4.4 Migrate with Cloud-Based Load Balancing

Phase 5: System Testing
- 5.1 Conduct Unit Testing
- 5.2 Conduct Integration Testing
- 5.3 Conduct User Acceptance Testing with students, faculty, admins, and ITSO
- 5.4 Conduct Load Testing
- 5.5 Compile Usability Report

Phase 6: System Deployment
- 6.1 Plan Deployment
- 6.2 Submit mobile app to App Stores & Google Play
- 6.3 Prepare User Training Video
- 6.4 Launch System
- 6.5 Monitor Post-Launch Performance

Phase 7: Project Closure
- 7.1 Review Project Outcomes
- 7.2 Finalize Documentation (System Manual, Maintenance Guide)
- 7.3 Document Lessons Learned
- 7.4 Obtain Final Sign-off

3.1 WBS Diagram



3.2 WBS Dictionary

WBS Code	WBS Name	WBS Description
1.1	Develop Project Charter	Create a formal document that outlines the project's purpose, objectives, high-level scope, stakeholders, duration estimates, resource needs, constraints, assumptions, risks and dependencies. This document authorizes the project and serves as a reference throughout its lifecycle.
1.2	Develop Stakeholder Assessment Matrix	Identify all stakeholders and assess their level of influence, interest, and impact on the project. Categorize and document communication strategies to manage their expectations effectively.
1.3	Sponsor Sign-off	Obtain formal approval from the project sponsor to proceed. This includes endorsement of the project charter and agreement on initial scope, budget, and timeline.
1.4	Kick-off Meeting with Key Stakeholders	Organize and facilitate an initial meeting with key stakeholders to align on goals, roles, timelines, and communication. The session sets a shared understanding and lays the groundwork for collaboration.
2.1	Scope Planning	Identify and define all deliverables, boundaries, and requirements of the project. This phase ensures that both client and user expectations are captured and translated into actionable scope elements.
2.1.1	Define Client Requirements	Collect, interpret, and document specific expectations from the client side to guide the project direction and outcomes.
2.1.1.1	Interview with ITSO — the Head of Student & Teaching	Conduct a structured interview to capture institutional goals, pain points, and functional expectations of the revamped SIS from the ITSO representative.
2.1.1.2	Define Client Requirements	Translate insights from interviews into concrete requirements, categorized by functionality, performance, and integration needs.
2.1.2	Define User Requirements	Understand what users need from the new SIS system in terms of usability, features, and pain points to be addressed.
2.1.2.1	Conduct Ethnography Research during summer registration	Observe users during actual system interactions during the summer semester to identify behavioral patterns and operational bottlenecks.
2.1.2.2	Organise Focus Group Research	Facilitate sessions with representative students to collect qualitative feedback and expectations.
2.1.2.3	Define User Requirements	Summarize research findings into specific user stories and system feature requirements.
2.1.3	Define Scope Statement	Draft a formal document outlining project objectives, deliverables, inclusions/exclusions, assumptions, and constraints.
2.1.4	Create Work Breakdown Structure (WBS)	Break the overall project scope into manageable and hierarchical components to support planning and tracking.
2.2	Project Scheduling	Plan the sequence and timing of tasks to ensure the project stays on track, within timeline constraints.
2.2.1	Develop Network Diagram	Establish task dependencies and visualize workflow relationships among activities.
2.2.1.1	Define and Sequencing	Identify tasks and arrange them in logical execution order to understand

	Activities	dependencies (Start-Start, Finish-Start, Start-Finish, Finish-Finish).
2.2.1.2	Estimate Activity Duration	Provide realistic time estimates for each activity by 3-point estimate & Delphi Technique.
2.2.2	Develop Gantt Chart	Convert the network diagram into a visual timeline using Gantt charts to track progress and deadlines.
2.2.2.1	Define Critical Path	Analyze task dependencies to identify the longest sequence of essential tasks that determines project duration.
2.3	Cost Planning	Estimate and allocate financial resources to ensure sufficient budgeting and resource planning for the project.
2.3.1	Estimate Initial Budget	Calculate a preliminary cost projection including resources, tools, and contingency margins with MS Project.
2.3.2	Determine Budget Allocation	Assign budget values to specific project areas and tasks based on priority and scope with MS Project.
2.4	Risk Planning	Proactively identify and prepare for potential project risks that could affect schedule, scope, or budget.
2.4.1	Risk Identification & Prioritization	List all plausible risks and rank them based on likelihood and impact to enable developing mitigation strategy.
3.1	Plot Architecture Diagram	Visualize the structural layout of the system, showing key components and how they interact within the proposed infrastructure.
3.1.1	Plot Network Architecture Diagram	Illustrate the network components, including servers, gateways, and load balancers, ensuring secure and scalable data flow.
3.1.2	Plot API Architecture Diagram	Map out the structure of internal and external APIs, defining data flow and communication endpoints between system modules.
3.2	Design User Interface	Establish the look of the system interface, focusing on enhancement of usability and accessibility across devices.
3.2.1	Design Mobile-Responsive Interface	Create an integrated layout to HKUST Student app that adapts to mobile devices, ensuring smooth navigation, readability, and feature accessibility on smaller screens.
3.2.2	Design Web-Responsive Interface	Design a scalable and intuitive interface for desktop browsers on HKUST SIS, emphasizing clarity, responsiveness, and modern aesthetics.
3.3	Outline Database Schema	Define the existing database structure and newly added elements, including tables, relationships, and data types, to support efficient data storage and retrieval, especially preparing for migrating third-party API.
3.4	Design Third-Party API Integrations	Plan the integration points for services like payment gateways with PayMe, or communication channels like WhatsApp, and external SIS data sources.
3.5	Finalize Prototype	Consolidate design elements into an interactive prototype that reflects user flows and major system functionalities.
4.1	Develop Back-End System	Implement the server-side logic, database interaction, and API services to support core functionalities.
4.2	Develop Front-End Interface	Build the client-facing interface using modern frameworks, ensuring

		responsive and dynamic user interaction.
4.3	Integrate with Third-Party APIs	Connect the system with external services and data providers, enabling seamless cross-system operations.
4.4	Migrate with Cloud-Based Load Balancing	Deploy system components to a cloud infrastructure with load balancing to ensure high availability and scalability.
5.1	Conduct Unit Testing	Test individual components or modules of the system in isolation to ensure each functions correctly.
5.2	Conduct Integration Testing	Validate that different system modules work together as intended through controlled, combined testing scenarios.
5.3	Conduct User Acceptance Testing	Facilitate testing sessions with HKUST students, faculty, admins, and ITSO to verify the system meets user needs and expectations.
5.4	Conduct Load Testing	Simulate peak usage conditions to evaluate system performance, stability, and scalability under stress.
5.5	Compile Usability Report	Analyze testing feedback and identify usability issues, strengths, and areas for refinement based on test results.
6.1	Plan Deployment	Develop a structured deployment plan outlining steps, timing, rollback options, and stakeholder communication.
6.2	Submit Mobile App to App Stores & Google Play	Prepare, test, and publish the mobile application to official platforms, adhering to store guidelines and compliance.
6.3	Prepare User Training Video	Create a tutorial video that demonstrates system features and guides users through key tasks.
6.4	Launch System	Release the system to all users, ensuring deployment tasks are completed and initial support is in place.
6.5	Monitor Post-Launch Performance	Track system usage, response times, and error rates to ensure smooth operations and resolve issues proactively.
7.1	Review Project Outcomes	Evaluate whether project objectives were met, comparing results against the initial project charter and success metrics.
7.2	Finalize Documentation	Compile and complete all essential documents, including the system manual and maintenance guide for ongoing support.
7.3	Document Lessons Learned	Reflect on project execution, documenting challenges, successful strategies, and improvement areas for future projects.
7.4	Obtain Final Sign-off	Secure formal approval from key stakeholders to confirm the project is completed and deliverables are accepted.

Phase 4 — Develop a Detailed Project Schedule

We adopted the Waterfall SDLC approach instead of Agile due to the nature of the project. This project focuses on revamping the SIS, where the technologies involved and the project scope are largely predictable with guidance from ITSO and the experience of the team members. Additionally, given the strict time constraint, any delay in this project would affect the entire HKUST institution. As such, project phases must be well-defined and executed in a linear manner. In other words, there is limited flexibility, and deliverables and tasks must follow a structured sequence aligned with fixed deadlines. Therefore, the Waterfall approach was selected over Agile. However, certain techniques inspired by Agile can still benefit the project and will be discussed in Section 4.1.

4.1 Duration Estimate

To estimate the project duration, we developed our schedule by utilising the mixture of Program Evaluation and Review Technique ("PERT") and Delphi Technique.

4.1.1 PERT estimate with Planning Poker

Firstly, the PERT estimate is given by the consolidated inputs from 5 team members. Members have sophisticated experience in their business roles. For instance, Kamdi has years of working experience in programming, and has her own side projects in Full stack developing on Github; Rovic has working experience in DevOps and maintains professional connections with third-party API service providers. With a well-experienced and high-skilled team, the estimation process is grounded in practical insights rather than speculative guesswork

Next, despite the Waterfall approach of the project, the estimate is given by the planning poker game in Agile development. This means the team would have a meeting held by the moderator / product owner — the project manager, with planning poker cards. Members give an initial estimate based on the selected WBS item as the base WBS item. Hence, members could give their estimate in the first round anonymously without the influence of others. This is essential since members could have been afraid of senior colleagues if they know senior's choices. They could be afraid of being teased by their inexperience. Therefore, the results from the first round would be an unbiased estimate. Additionally, uncertainty to the WBS item's requirement can be resolved within the team immediately and ask for clarification from the Project Manager. Members give estimates by 3 scenarios for each item — Optimistic, Most Likely and Pessimistic. Next, members should discuss with others if discrepancies are found among estimates, give their opinion and prepare for the next round until the estimate is equal or very similar.

4.1.2 Delphi Technique

The result from PERT might vary from the actual duration since the team is external. Imperfect information introduces uncertainty and higher variance from actual duration. Hence, Delphi technique is also used. This implies results from PERT will be sent to experts from ITSO, for instance, senior programmers and system analysts under the student and teaching department of ITSO. Their feedback would be collected anonymously to prevent peer influence. This acts as a validation to ensure the project team's estimate is grounded and identify blindspots if any.

This project team is highly-skilled with members from different expertise, as mentioned in 4.1.1., functioning as a professional external team. As a result, the above duration estimation approaches help eliminate the overestimation or the Parkinson's Law — "Work expands to fill the time available for its completion.". Yet, despite team members'

experience with using the SIS, the launch of SIS must not be delayed due to its criticality. Delays in completion of the system will affect all HKUST members, including students, faculty, and so on. Therefore, Project Buffer and Feeder Buffer is added in this project. For instance, there is 1.5 days free slack for WBS as the Feeder Buffer, 0.3 weeks free slack for submitting app to App Stores & Google Play, etc. This helps prevent the delay of the project completion day.

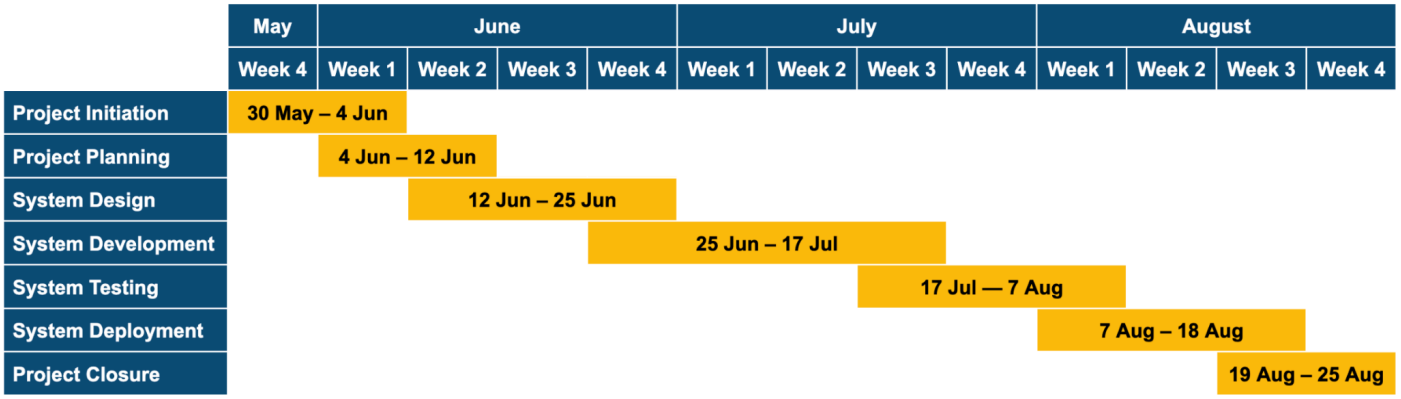
4.2 Project Schedule

4.2.1 Resource Sheet

Recourse Name	Initials	Group	Max.	Std. Wage (HK$)	Ovt. Wage (HK$)
Alan	A	PM (Project Manager)	100%	130	180
Kamdi	K	SP (Senior Programmer)	100%	130	180
Asensio	AS	JP (Junior Programmer)	100%	100	150
Martin	M	BA (Business Analyst)	100%	120	170
Rovic	R	SA (System Analyst)	100%	120	170

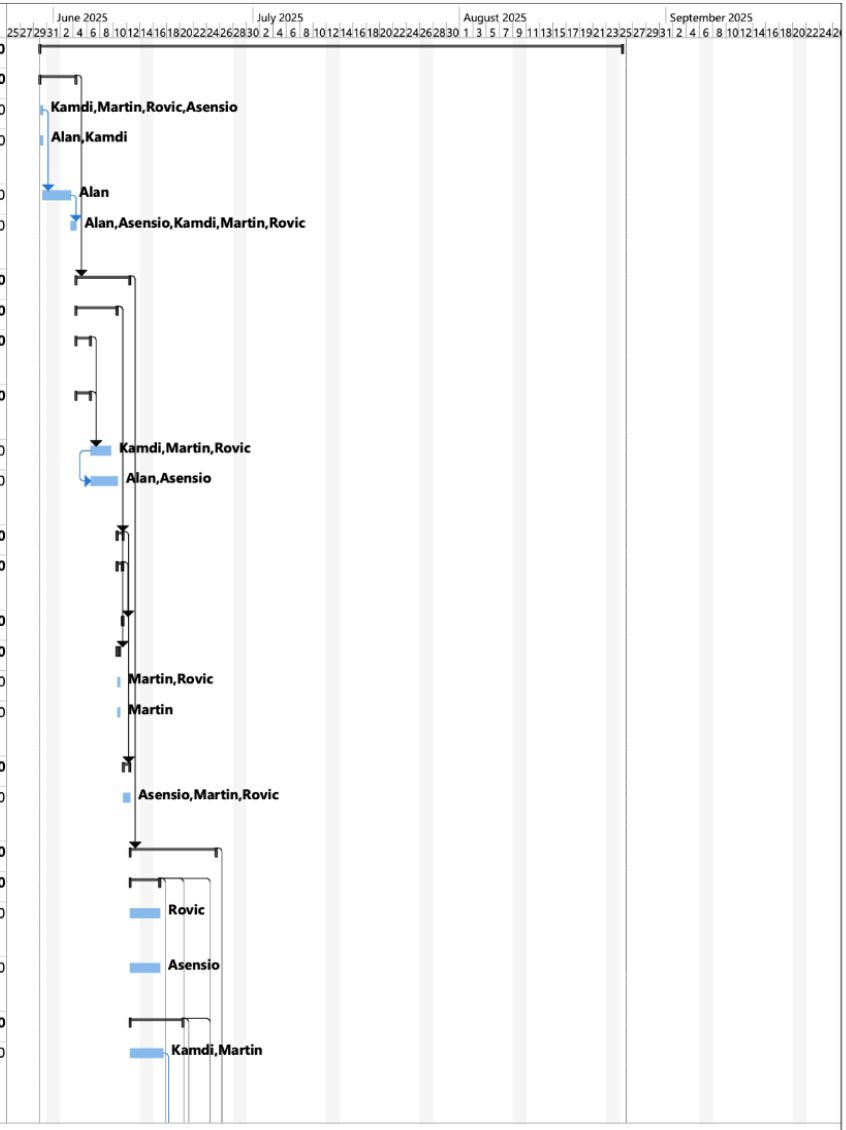
In this project team, everyone works as a full-time employee on weekdays from 9 a.m. to 6 p.m. Public holidays — Dragon Boat Festival & HKSAR Establishment Day are excluded in the project working days.

4.2.2 Summarised Gantt Chart



4.2.3 Project Schedule & Detailed Gantt Chart

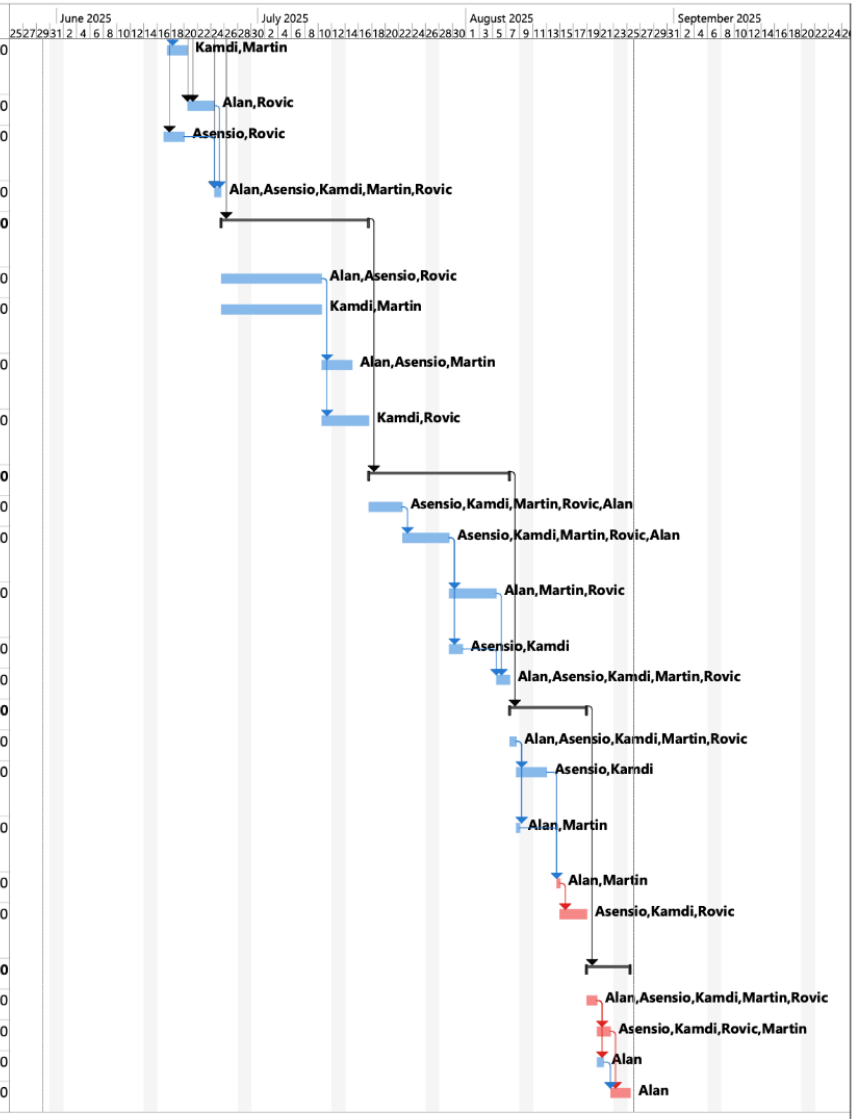
ID	Task Mode	Task Name	Duration	Start	Finish	Predec	Resource Names	Total Slack	Work	Cost
1		**HKUST SIS Modernization**	**60.5 days?**	**Fri 30/5/25**	**Mon 25/8/25**			**0 days?**	**1,891 hrs**	**$226,830.00**
2		**Phase 1: Project Initiation**	**3.25 days**	**Fri 30/5/25**	**Wed 4/6/25**			**1.5 days**	**51 hrs**	**$6,330.00**
3		Develop Project Charter	2 hrs	Fri 30/5/25	Fri 30/5/25		Kamdi,Martin,Rovic,A	12 hrs	8 hrs	$940.00
4		Develop Stakeholder Assessment Matrix	1.5 hrs	Fri 30/5/25	Fri 30/5/25		Alan,Kamdi	36.5 hrs	3 hrs	$390.00
5		Sponsor Sign-off	0.5 wks	Fri 30/5/25	Tue 3/6/25	3	Alan	0.3 wks	20 hrs	$2,600.00
6		Kick-off Meeting with Key Stakeholders	4 hrs	Tue 3/6/25	Wed 4/6/25	5	Alan,Asensio,Kamdi,M	12 hrs	20 hrs	$2,400.00
7		**Phase 2: Project Planning**	**6.25 days?**	**Wed 4/6/25**	**Thu 12/6/25**	**2**		**1.5 days?**	**208 hrs**	**$24,740.00**
8		**Scope Planning**	**4.5 days**	**Wed 4/6/25**	**Tue 10/6/25**			**1.5 days**	**156 hrs**	**$18,640.00**
9		**Define Client Requirements**	**2.5 days**	**Wed 4/6/25**	**Fri 6/6/25**			**1.5 days**	**40 hrs**	**$5,000.00**
12		**Define User Requirements**	**2.5 days**	**Wed 4/6/25**	**Fri 6/6/25**			**1.5 days**	**60 hrs**	**$7,000.00**
16		Define Scope Statement	1 day	Fri 6/6/25	Mon 9/6/25	9,12	Kamdi,Martin,Rovic	1.5 days	24 hrs	$2,960.00
17		Create Work Breakdown Structure (WBS)	2 days	Fri 6/6/25	Tue 10/6/25	16SS	Alan,Asensio	1.5 days	32 hrs	$3,680.00
18		**Project Scheduling**	**0.75 days**	**Tue 10/6/25**	**Wed 11/6/25**	**8**		**1.5 days**	**22 hrs**	**$2,660.00**
19		**Develop Network Diagram**	**0.5 days**	**Tue 10/6/25**	**Wed 11/6/25**			**1.5 days**	**20 hrs**	**$2,400.00**
22		**Develop Gantt Chart**	**0.25 days**	**Wed 11/6/25**	**Wed 11/6/25**	**19**		**1.5 days**	**2 hrs**	**$260.00**
24		**Cost Planning**	**0.25 days**	**Tue 10/6/25**	**Tue 10/6/25**	**8**		**3 days**	**6 hrs**	**$720.00**
25		Estimate Initial Budget	0.25 days	Tue 10/6/25	Tue 10/6/25		Martin,Rovic	3 days	4 hrs	$480.00
26		Determine Budget Allocation	0.25 days	Tue 10/6/25	Tue 10/6/25		Martin	3 days	2 hrs	$240.00
27		**Risk Planning**	**1 day?**	**Wed 11/6/25**	**Thu 12/6/25**	**18**		**1.5 days?**	**24 hrs**	**$2,720.00**
28		Risk Identification & Prioritization	1 day?	Wed 11/6/25	Thu 12/6/25		Asensio,Martin,Rovic	1.5 days?	24 hrs	$2,720.00
29		**Phase 3: System Design**	**9 days**	**Thu 12/6/25**	**Wed 25/6/25**	**7**		**1.5 days**	**256 hrs**	**$30,480.00**
30		**Plot Architecture Diagram**	**2.5 days**	**Thu 12/6/25**	**Mon 16/6/25**			**4 days**	**40 hrs**	**$4,400.00**
31		Plot Network Architecture Diagram	2.5 days	Thu 12/6/25	Mon 16/6/25		Rovic	4 days	20 hrs	$2,400.00
32		Plot API Architecture Diagram	2.5 days	Thu 12/6/25	Mon 16/6/25		Asensio	4 days	20 hrs	$2,000.00
33		**Design User Interface**	**6 days**	**Thu 12/6/25**	**Fri 20/6/25**			**1.5 days**	**96 hrs**	**$12,000.00**
34		Design Mobile-Responsive Interface	3 days	Thu 12/6/25	Tue 17/6/25		Kamdi,Martin	1.5 days	48 hrs	$6,000.00



Project: ISOM3010_Step3&4
Date: Tue 13/5/25

Task, Split, Milestone, Summary, Project Summary, Inactive Task, Inactive Milestone, Inactive Summary, Manual Task, Duration-only, Manual Summary Rollup, Manual Summary, Start-only, Finish-only, External Tasks, External Milestone, Deadline, Critical, Critical Split, Progress, Manual Progress

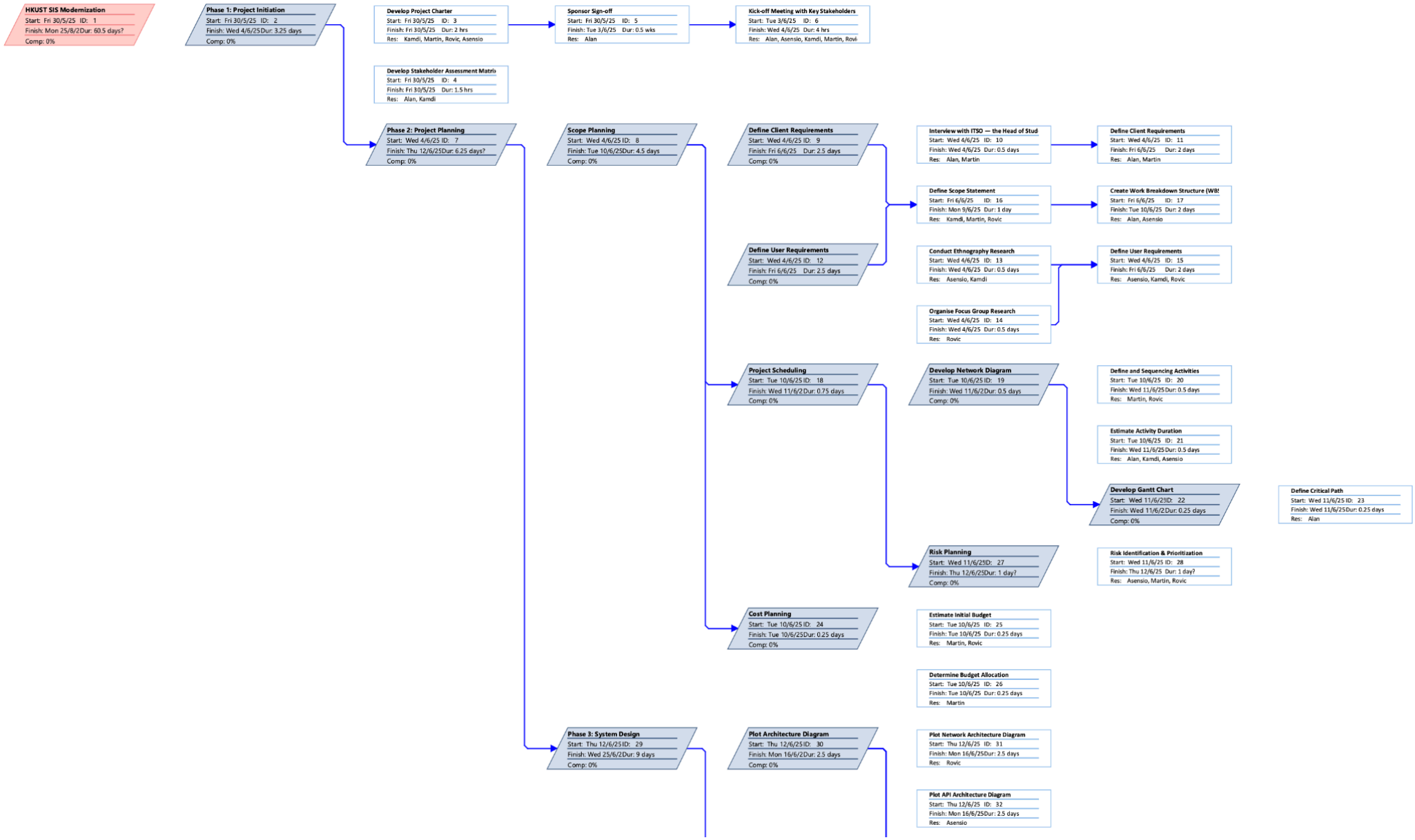
ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Total Slack	Work	Cost
35		Design Web-Responsive Interface	3 days	Tue 17/6/25	Fri 20/6/25	34	Kamdi,Martin	1.5 days	48 hrs	$6,000.00
36		Outline Database Schema	2 days	Fri 20/6/25	Tue 24/6/25	30,33	Alan,Rovic	1.5 days	32 hrs	$4,000.00
37		Design Third-Party API Integrations	3 days	Tue 17/6/25	Thu 19/6/25	30	Asensio,Rovic	4 days	48 hrs	$5,280.00
38		Finalize Prototype	1 day	Tue 24/6/25	Wed 25/6/25	30,33,	Alan,Asensio,Kamdi,M	1.5 days	40 hrs	$4,800.00
39		**Phase 4: System Development**	**15 days**	**Wed 25/6/25**	**Thu 17/7/25**	**29**		**1.5 days**	**540 hrs**	**$65,000.00**
40		Develop Back-End System	2 wks	Wed 25/6/25	Thu 10/7/25		Alan,Asensio,Rovic	0.3 wks	240 hrs	$28,000.00
41		Develop Front-End Interface	2 wks	Wed 25/6/25	Thu 10/7/25		Kamdi,Martin	1.3 wks	160 hrs	$20,000.00
42		Integrate with third-party APIs	0.5 wks	Thu 10/7/25	Mon 14/7/25	40	Alan,Asensio,Martin	0.8 wks	60 hrs	$7,000.00
43		Migrate with Cloud-Based Load Balancing	1 wk	Thu 10/7/25	Thu 17/7/25	40	Kamdi,Rovic	0.3 wks	80 hrs	$10,000.00
44		**Phase 5: System Testing**	**15 days**	**Thu 17/7/25**	**Thu 7/8/25**	**39**		**1.5 days**	**552 hrs**	**$66,480.00**
45		Conduct Unit Testing	3 days	Thu 17/7/25	Tue 22/7/25		Asensio,Kamdi,Martin	1.5 days	120 hrs	$14,400.00
46		Conduct Integration Testing	5 days	Tue 22/7/25	Tue 29/7/25	45	Asensio,Kamdi,Martin	1.5 days	200 hrs	$24,000.00
47		Conduct User Acceptance Testing	1 wk	Tue 29/7/25	Tue 5/8/25	46	Alan,Martin,Rovic	0.3 wks	120 hrs	$14,800.00
48		Conduct Load Testing	2 days	Tue 29/7/25	Thu 31/7/25	46	Asensio,Kamdi	4.5 days	32 hrs	$3,680.00
49		Compile Usability Report	2 days	Tue 5/8/25	Thu 7/8/25	47,48	Alan,Asensio,Kamdi,M	1.5 days	80 hrs	$9,600.00
50		**Phase 6: System Deployment**	**7.5 days**	**Thu 7/8/25**	**Mon 18/8/25**	**44**		**0 days**	**144 hrs**	**$17,000.00**
51		Plan Deployment	1 day	Thu 7/8/25	Fri 8/8/25		Alan,Asensio,Kamdi,M	1.5 days	40 hrs	$4,800.00
52		Submit mobile app to App Stores & Google Play	0.5 wks	Fri 8/8/25	Tue 12/8/25	51	Asensio,Kamdi	0.3 wks	40 hrs	$4,600.00
53		Prepare User Training Video	0.5 days	Fri 8/8/25	Fri 8/8/25	51	Alan,Martin	3.5 days	8 hrs	$1,000.00
54		Launch System	0.5 days	Thu 14/8/25	Thu 14/8/25	52,53	Alan,Martin	0 days	8 hrs	$1,000.00
55		Monitor Post-Launch Performance	2 days	Fri 15/8/25	Mon 18/8/25	54	Asensio,Kamdi,Rovic	0 days	48 hrs	$5,600.00
56		**Phase 7: Project Closure**	**4.5 days**	**Tue 19/8/25**	**Mon 25/8/25**	**50**		**0 days**	**140 hrs**	**$16,800.00**
57		Review Project Outcomes	1.5 days	Tue 19/8/25	Wed 20/8/25		Alan,Asensio,Kamdi,M	0 days	60 hrs	$7,200.00
58		Finalize Documentation	2 days	Wed 20/8/25	Fri 22/8/25	57	Asensio,Kamdi,Rovic,M	0 days	64 hrs	$7,520.00
59		Document Lessons Learned	1 day	Wed 20/8/25	Thu 21/8/25	57	Alan	1 day	8 hrs	$1,040.00
60		Obtain Final Sign-off	1 day	Fri 22/8/25	Mon 25/8/25	58,59	Alan	0 days	8 hrs	$1,040.00



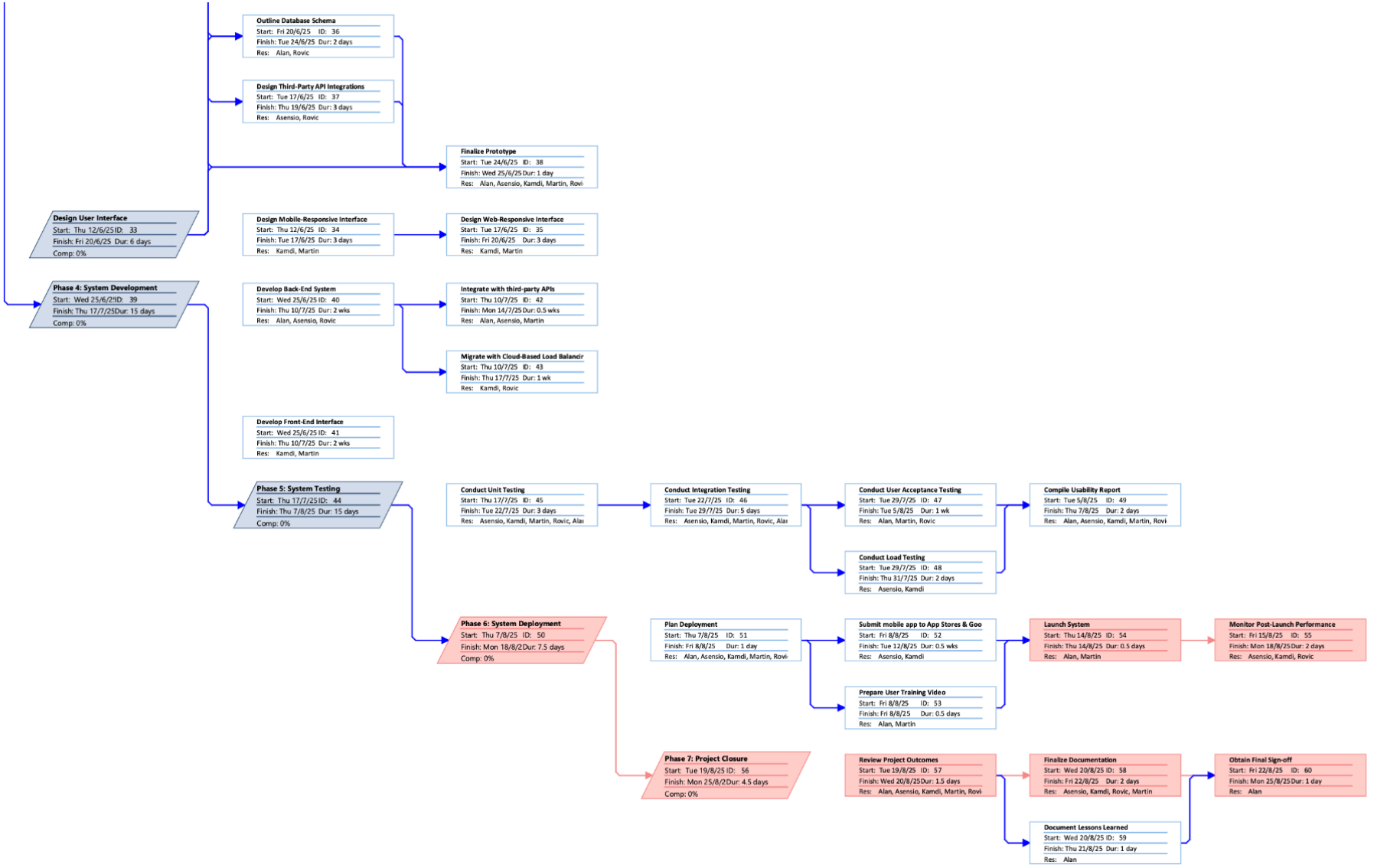
Project: ISOM3010_Step3&4
Date: Tue 13/5/25

Task	Project Summary	Manual Task	Start-only	Deadline	Manual Progress		
Split	Inactive Task	Duration-only	Finish-only	Critical			
Milestone	Inactive Milestone	Manual Summary Rollup	External Tasks	Critical Split			
Summary	Inactive Summary	Manual Summary	External Milestone	Progress			

4.2.4 Network Diagram & Critical Path



4.2.4 Network Diagram & Critical Path (Con't)



Phase 5 — Create a Risk Management Plan and other Relevant Deliverables

Risks were identified from a brainstorming session with the UST lying Team Members because of their ability price with implementing similar projects. They discussed the risks that arose from previous projects when implementing functions on the mobile platform and in hindsight, how they could have mitigated those risks.
Below are the types of risks that they have identified and prioritized.

5.1 Risk Identification & Prioritization

Risk	Probability	Impact	Owner	Mitigation Strategy
User Resistance	High	High	Alan (Project Manager)	• Conduct phased training workshops. • Develop embed interactive tutorials into new interface
Integration Delays	Medium	High	Asensio (Junior Programmer)	• Assign contingency time in Phase 3. • Weekly check-ins with third-party vendors.
Legacy Code Issues	Medium	Medium	Kamdi (Senior Programmer)	• Conduct code audits early. • Document workflows to avoid bottlenecks
Budget Overruns	Low	High	Martin (Business Analyst)	• Track expenses biweekly. • Allocate 15% of budget to contingency reserves.
Security Vulnerabilities	Medium	Critical	Rovic (System Analyst)	• Implement 2FA and encryption. • Conduct penetration testing.

Key risks include user resistance due to unfamiliar workflows, integration delays from third-party API dependencies, legacy code compatibility issues, budget overruns, and security vulnerabilities during cloud migration.

To mitigate these risks, tailored strategies were developed. For instance, user resistance—a high-probability, high-impact risk—is addressed through phased training programs, including gamified tutorials and “change champions” among students and faculty to advocate for the new system. Integration delays, often caused by third-party vendor dependencies, are managed by formalizing service-level agreements (SLAs) and allocating contingency time in the development schedule. Legacy code challenges are tackled through biweekly code audits with HKUST’s IT team and modular development to reduce dependencies. Financial risks are controlled by dedicating 15% of the budget ($180,000) to contingency reserves and conducting biweekly financial reviews. Security concerns, deemed critical due to the sensitivity of student data, are mitigated through mandatory two-factor authentication, penetration testing, and encryption protocols

5.2 Risk Monitoring & Reporting

Risk monitoring is conducted via a real-time RAID log (Risks, Assumptions, Issues, Dependencies) tracked in Jira, ensuring visibility for all stakeholders. Biweekly team meetings review risk statuses, while critical threats, such as security breaches, trigger immediate escalation to the IT Directorate. A structured communication plan complements these efforts, tailoring updates to stakeholder needs: students and faculty receive biweekly progress summaries via email and the HKUST mobile app, while the IT Directorate reviews technical milestones in weekly Zoom briefings. Third-party vendors are held accountable through weekly check-ins on Microsoft Teams to monitor API delivery timelines.

Training and support materials further reduce adoption risks. Short video tutorials, such as “How to Swap Courses,” are hosted on HKUST’s learning management system (LMS), while printable quick-start guides provide step-by-step instructions for common tasks. IT staff receive detailed technical manuals covering cloud infrastructure maintenance and API troubleshooting, ensuring long-term system sustainability.

5.2.1 Stage-Gate Meeting

Stage-gate meetings act as formal checkpoints to evaluate progress, validate deliverables, and authorize transitions between phases. These meetings ensure risks are addressed proactively and align the project with institutional priorities:

Meeting Type	Timing	Purpose	Attendees	Format
Gate 1: Initiation Review	June 4 (end of Project Initiation)	Validate project charter, scope, and stakeholder alignment.	All team members	In-person
Gate 2: Planning Review	June 12 (end of Project Planning)	Review WBS, PERT estimates, and Delphi feedback; approve or refine timeline and task ownership	All team members	Zoom + Slides
Gate 3: Design Approval	June 25 (end of System Design)	Review UI mockups, architecture diagrams, and schema; approve for development	All team members, ITSO designers	In-person / Zoom
Gate 4: Dev Readiness	July 7 (mid Development Phase)	Assess backend/frontend readiness, API stubs; decide if system is ready for integration	All team members	In-person
Gate 5: Pre-UAT Review	July 28 (end of Integration Testing)	Final testing status check, bug triage, and UAT scheduling; Go/No-Go for deployment	All team members	In-person

Gate 6: Launch Readiness	August 12 (before System Launch)	Final checklist for system launch training materials, App version submission	All team members	Zoom
Gate 7: Closure Review	August 25	Review project outcomes, lessons learned, and gather sign-off	All team members + ITSO reviewers	In-person / Zoom

While the stage-gate framework is established, the following elements remain under finalization:

- **Approval Criteria**: Specific metrics for "Go/No-Go" decisions at each gate (e.g., maximum bug count for UAT approval) require alignment with HKUST's IT governance team.
- **Post-Meeting Documentation**: Templates for gate review reports (e.g., risk summaries, action items) are pending approval from the Provost's Office.
- **Stakeholder Roles**: Participation requirements for non-core attendees (e.g., faculty representatives at Design Approval) are being negotiated.

These details will be finalized during the Project Initiation phase, with provisional guidelines in place to ensure continuity.

5.3 Post-launch support

Post-launch support includes a 30-day hypercare period and real-time performance dashboards to monitor latency and error rates. Long-term strategies involve monthly system health checks and quarterly user satisfaction surveys to identify emerging issues. Stage-gate meetings at critical milestones, such as design approval, pre-UAT review, and post-launch evaluation ensure alignment with institutional goals. For example, the Pre-UAT Review Gate assesses testing outcomes and deployment readiness, determining whether the system meets HKUST's standards for a full launch.

This risk management approach balances technical rigor with stakeholder engagement, inspired by PMI's PMBOK framework. By prioritizing transparency, proactive mitigation, and continuous feedback, the plan safeguards both the project's deliverables and HKUST's reputation as a leader in educational innovation.

5.3.1 Communication channel

Stakeholder	Information Needed	Frequency	Channel
Students/Faculty	Progress updates, testing schedules	Biweekly	Email & WhatsApp
HKUST IT Directorate	Technical milestones, risks	Weekly	Formal report + Zoom
Project Team	Task assignments, blockers	Daily	Slack stand-ups

Phase 6 — Create Prototypes

It is critical to have an overview on the project's scope and stakeholder requirement before creating prototypes, and we had held a number of meetings to double confirm some details of the prototype. Since the system design was on the critical path, a backlog of progress would exist once delay happened. The final duration for creating prototypes was given 6 days in total, including both web & app versions.. The team agreed that the duration was reasonable for the prototype and system design.

Three Phases were included. For the first phase, we focus on redesigning the dashboard and navigation by using Figma as the visual prototype. After the visual is done, we implement cloud scalability & caching by AWS/Azure test environment to ensure the functionality of the prototype. Lastly, we integrate our prototype to allow real-time notifications and third-party API connections.

6.1 Prototype Features

Many distinct important features were identified in the interview and survey sections with different stakeholders, 6 features were prioritized by the product backlog, listing the first priority features at the top, the features after 3 can be considered as “low priority”:

1. Ease of use / integration - Intuitive navigation & useful redirection links or integration with other systems such as courses information, enrollment/scheduling, facilities booking, academic progress etc
    a. Raised by: Student, Teacher
    b. Area to be kept: The overall design of the existing UI, e.g: places of different functions located in the apps
    c. Area to improve: Include more functions without changing most of the outfit of UI
2. Mobile friendly - Allow students to access system from all devices and browsers
    a. Raised by: Student, Teacher
    b. Area to be kept: The existing website and functionality
    c. Area to improve: Allowing users to use functions that only can be used on website currently by mobile phone
3. Timely updates - Instant notifications for important announcements and updates
    a. Raised by: Student, Administrator
    b. Area to be kept: After analyzing, we figured out the current delayed response is due to the limited speed and capacity of cloud services.
    c. Area to be improved: Data Structuring, Cloud services
4. Effective communication - Tools for communicating with students, from announcements to direct messaging
    a. Raised by: Teacher, Administrator
    b. Area to be kept: *Nun*
    c. Area to be improved: Deliver real-time notifications by WhatsApp / SMS for course registration status, payment due
5. Communication Tools - Channels for internal messaging between staff & broadcasting tools
    a. Raised by: Administrator

b. Area to be kept: Current broadcast functions by Canvas and Outlook
c. Area to be improved: Showing student status on the announcement, i.e.: “read” and “unread”

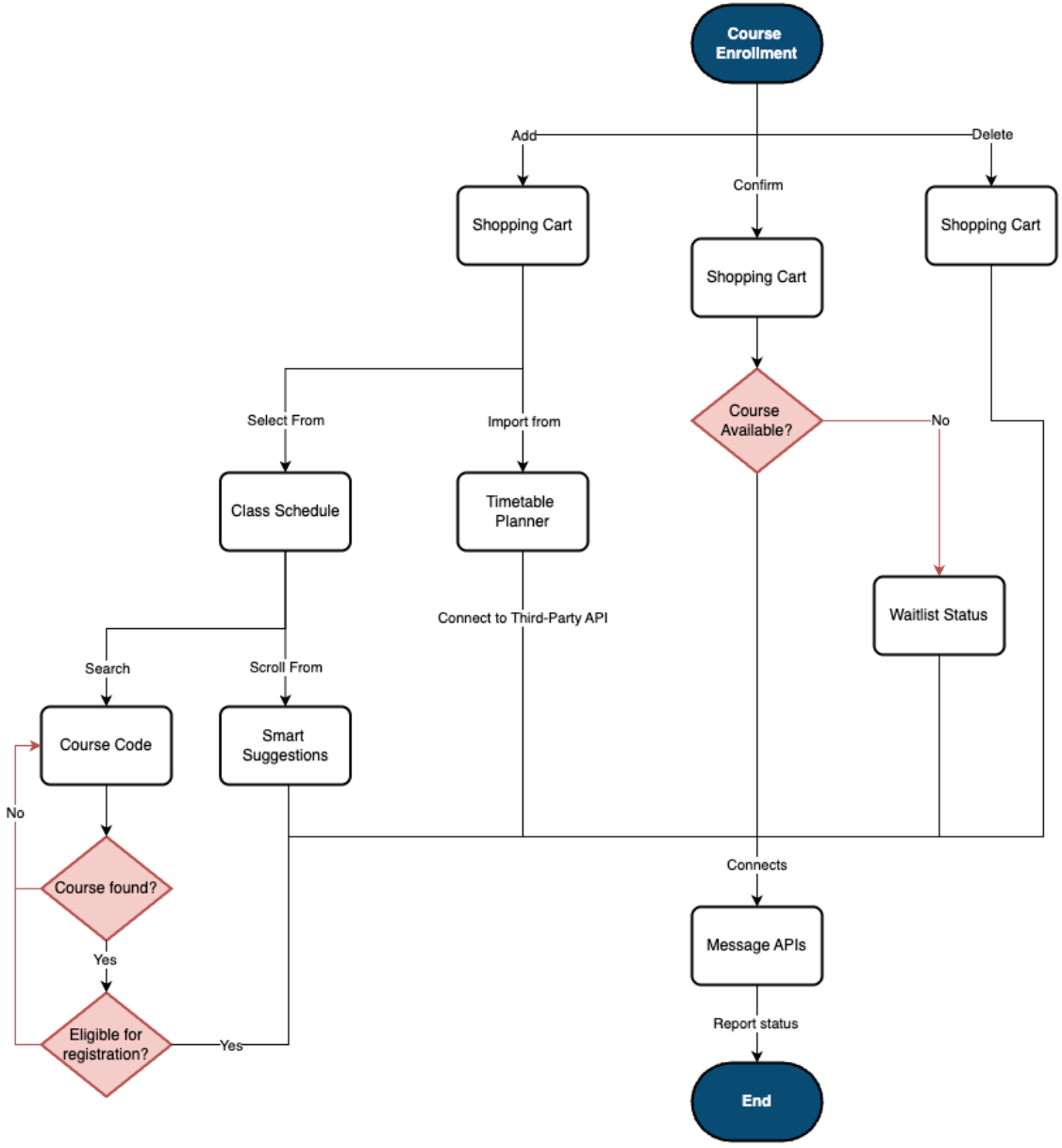
6. Security - Secure data with 2-factor authentication and encryption
    a. Raised by: IT Department
    b. Area to be kept: current google authenticator, duo mobile authentication method
    c. Area to be improved: *Nan*

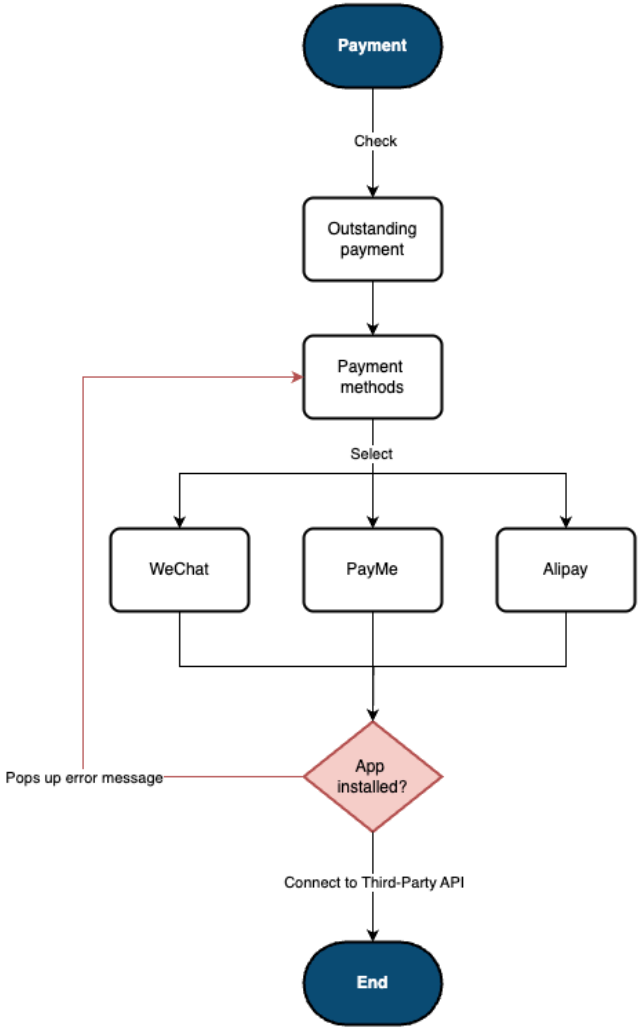
6.2 Prototype of Mobile App

After analyzing and prioritizing features, we determined that developing a mobile UI prototype first would most effectively address the product backlog. The mobile UI incorporates new functionalities, such as class enrollment and payment systems, previously available only on the web platform. To ensure an intuitive design and minimize the learning curve, we drew inspiration from existing applications. We introduced a Single Dashboard Access design, featuring a centralized dashboard with quick-access widgets for course enrollment, advisement reports, financials, and equipment booking. The design of the web would be similar to the mobile UI - precise and intuitive.

6.2 Prototype of Mobile App (Con't)

The Graph below shows the flowchart of the main functions in the mobile app. Please see **Appendix — Phase 6** for the detailed UI/UX design.





Phase 7 — Record Progress

Managing and recording our project progress is pivotal for ensuring timely completion and staying within the budget, and is also the prerequisite to making necessary adjustments when needed. After careful consideration, our team decided on using both the renowned project management software MS Project and the waterfall project management and bug tracking software Jira. This is because the two software have different useful functionalities such as report generation, real-time bug tracking and others that we believe are useful for managing this project. Using the previous gantt chart and WBS as the baseline, we have decided to update the project progress with MS Project weekly with a meeting, while stakeholders can use Jira to monitor risks through the RAID logs (refer to 5.2).

The project manager will host a weekly meeting where the team will discuss their progress and challenges, then he/she will input everyone's progress. Afterwards, to ensure accountability and visibility, the project manager will also publish several relevant reports generated through MS Project, such as cost overviews (see **Appendix 7.1**) or work overviews (see **Appendix 7.2)** so all team members understand their progress and adjust accordingly.

Phase 8 — Change Management/Usability Testing

8.1 Change Management

Upgrading HKUST's SIS will impact students and staff across the university. To minimise risks, a change management approach has been designed. It is built on 3 key pillars: clear communication, phased adoption, and iterative feedback. which will all ensure that the transition is successful.

Clear Communication Strategy

To ensure that the change is successful, communication throughout the project will be key. This communication must start early on, and be on-going, to prepare stakeholders for the upcoming transformation, and address their concerns as the project continues.

Information about the upgrade will be delivered through emails, and briefing sessions to explain the reasons for the upgrade, the benefits it offers, and what will be expected from users at each stage. The communication must be tailored for different groups, i.e. students, faculty, administrative staff, to ensure that it is relevant and clear to each group. There will be regular Q&A forums and feedback channels to allow concerns or suggestions to be heard and addressed early, and this will ultimately reduce uncertainty and manage expectations.

Phased Adoption

Rather than a single, large-scale implementation, the SIS upgrade will be introduced in manageable phases to minimise disruption, and maximise learning from each phase.

Pilot group deployment will be utilised, as the new system will first be tested with select departments or volunteer groups to refine the system before wider deployment. The legacy system will also operate in parallel with the newer system, to ensure that key functions like course registration and grade viewing will always be accessible, and by adopting this phased approach, risks will be contained, and feedback can be adapted to quickly.

Iterative Feedback and Continuous Improvement

As change isn't static and requires ongoing refinement, feedback mechanisms will be embedded to improve the system.

There will be multiple feedback channels such as surveys, usability testing sessions, focus groups, and in-system feedback tools to capture user experiences at every stage. Any common issues will then be prioritised for immediate fixes, with these updates then communicated back to the community. As these changes are implemented, system usage rates, task completion times, and help desk ticket trends will then be tracked to identify areas that need additional support or adjustments.

Complexity of the System: High The SIS upgrade will need both front-end and back-end development to improve the interface design, use cloud-based load balancing, and integrate third-party APIs for the different payment gateways and notifications. Although it is technically complex, it mainly impacts the IT infrastructure and the user's experience, rather than changes to the hierarchy of the organisation.	**Structure:** There will not be major changes to the current organisational structure, but there will need to be collaboration between the IT department, and also admin departments to ensure all the current features of SIS are covered, so clear communication channels have to be established so that the project is executed smoothly.
Process: Processes such as course enrollment, swapping courses, and paying fees will become easier to do, and experience less downtime. Real-time notifications and a more user-friendly interface will make all user interactions significantly smoother, and mean that there will be less manual administrative efforts.	**People:** The new SIS system will be more convenient for both students and staff as tasks will be completed faster because the wait times for tasks to be executed will be shorter. Initially, it may involve a learning curve, but this can be addressed through targeted training sessions and guides to ease the transition.
Enhanced Accuracy- Admin Staff & IT personnel: When students are not eligible for a course, the course will not appear on SIS, so this automation will minimise manual input errors, and mean that data accuracy of student records and financial transactions will be higher. This will mean that admin staff can spend less time troubleshooting errors, and focus on more critical tasks. IT personnel will need to be trained on the new systems, especially the cloud infrastructure that would increase their workloads short-term.	**Improved Monitoring- Management:** As there will be greater oversight of the system due to real-time analytics and real-time reporting, faster responses to any failures can be made. However, they may require briefings to become familiar with the analytical tools and actual reporting functionalities.

8.2 Usability Testing

Alpha-Beta Testing

Firstly, to make sure that the core functionalities of SIS meet the initial expectations, Alpha testing will be conducted internally by the development team. Any issues will be addressed before moving on to beta testing. Beta testing will involve a limited release to a small group of end users. This group will provide important feedback based on realistic scenarios, and allow us to identify any usability concerns.

Integration Testing

To ensure that all components of SIS work seamlessly together, integration testing will be conducted. This phase will focus on validating the interactions between integrated modules such as enrollment, payment, and notifications, and ensure that they operate as expected without any performance issues.

We will employ an incremental integration approach, by first combining the core components like the dashboard and enrollment system, and then progressively integrating additional modules. Any issues that then arise will be documented to be promptly addressed by the development team.

System Testing

After successful integration testing, system testing will be carried out to assess the overall functionality. This testing will check how the system works under stressful environments, for example, enrolling in multiple courses at a time, or when multiple enroll in courses at a time.

System development will verify performance under both typical and peak conditions to identify potential bottlenecks, responsiveness issues and just any unexpected behaviour. Results will be carefully analysed, and necessary adjustments will be made before moving onto User Acceptance Testing.

User-Acceptance Testing (UAT)

For UAT, we will deploy the refined prototype within a controlled live environment accessible to end users for a defined period. All user groups will be invited to check that the system meets their daily needs.

During UAT, participants will utilise a structured checklist to check each function. Issues will then be documented and categorised by their severity (in the comments section of the table), so that they can be rectified in priority order. If UAT outcomes are successful, that will indicate that the system is ready to be deployed, which will keep stakeholders confident in the final product.

ID	Test Description	Steps	Test Steps	Expected Results	Actual Results	Pass/Fail	Comments
1	Login and dashboard navigation	2	Dashboard and navigation clearly visible				
2	Course Enrollment	3	Course selected and enrollment confirmed clearly				
3	Course swap	3	Clear swapping process and confirmation				
4	Payment Transactions	3	Payment clearly selected and confirmed				
5	Navigation and Responsiveness	2	Smooth navigation and responsive design				
6	Clarity and visibility	2	Clear text, buttons, visuals, and colour scheme				
7	Error Handling	2	Clear error messages and easy recovery				
8	System Feedback	2	Prompt notifications with accurate redirection				

Phase 9- Closing

Close-out Meeting

Our team will conduct an internal close-out meeting at the end of the NextSIS project. The main goal of the meeting will be to review the project performance, document lessons learned, and identify any areas that can be improved later down the line. This will ensure that we have the best understanding of all the processes undergone, and which decision-making strategies worked best. As a result, the insights gained will encourage continuous improvement for future project management practices.

Project Progress Tracking

Throughout the project's lifecycle, we will use Microsoft Project and Jira to closely track our project progress. A detailed baseline of the WBS tasks, durations, and budget estimates were established when planning the project, so in weekly progress meetings, regular updates and monitoring can be assessed against this baseline.

Project Review and Performance Analysis

A detailed project review will be carried out to assess the outcomes against our original goals: improved usability, mobile responsiveness, reduced system crashes, and improved administrative efficiency.

The usability testing will allow us to assess the task success rate across major functions, to understand what refinements will need to be made. Performance testing under a simulated peak load will also be able to confirm whether the new SIS maintains responsiveness with zero downtime during enrollment simulations involving many concurrent users. Furthermore, any feedback collected from beta testers will focus mainly on improving the interface, instead of identifying system faults, as these will be detected in system and integration testing. Finally, our phased deployment and continuous communication will ensure that our system is highly adaptable, and that the risks of user resistance and system misconfigurations are mitigated.

Final Deliverables and Knowledge Transfer

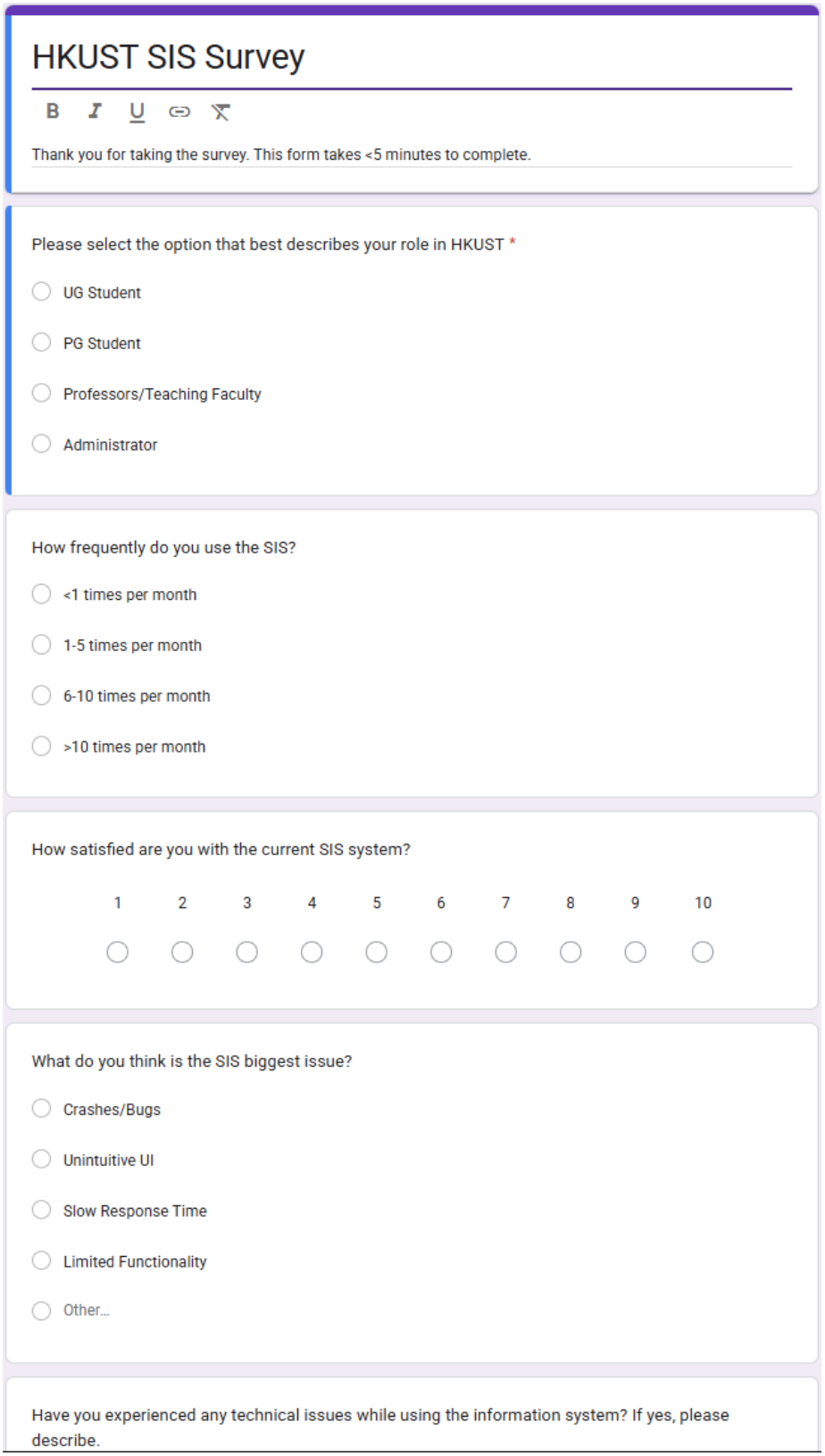
The project will conclude with a final report, including performance metrics, stakeholder feedback summaries, and backlogged items for future improvements for the project. We will also produce a knowledge transfer pack that will include technical documentation and resources to train users so that the upgraded SIS system can be sustained long-term.

Overall, the plans for the project have not only met, but exceeded its critical success criteria, aiming to achieve significant improvements to how the system is used and can be scaled in the future, whilst remaining within the allocated budget and timeline. This experience has reinforced our team's ability to manage complex system upgrades, and may possibly inform our approach to future IT transformation projects at HKUST.

Appendix

Phase 2

2.1 User survey for HKUST SIS



HKUST SIS Survey

Thank you for taking the survey. This form takes <5 minutes to complete.

Please select the option that best describes your role in HKUST *

- UG Student
- PG Student
- Professors/Teaching Faculty
- Administrator

How frequently do you use the SIS?

- <1 times per month
- 1-5 times per month
- 6-10 times per month
- >10 times per month

How satisfied are you with the current SIS system?

1 2 3 4 5 6 7 8 9 10

What do you think is the SIS biggest issue?

- Crashes/Bugs
- Unintuitive UI
- Slow Response Time
- Limited Functionality
- Other...

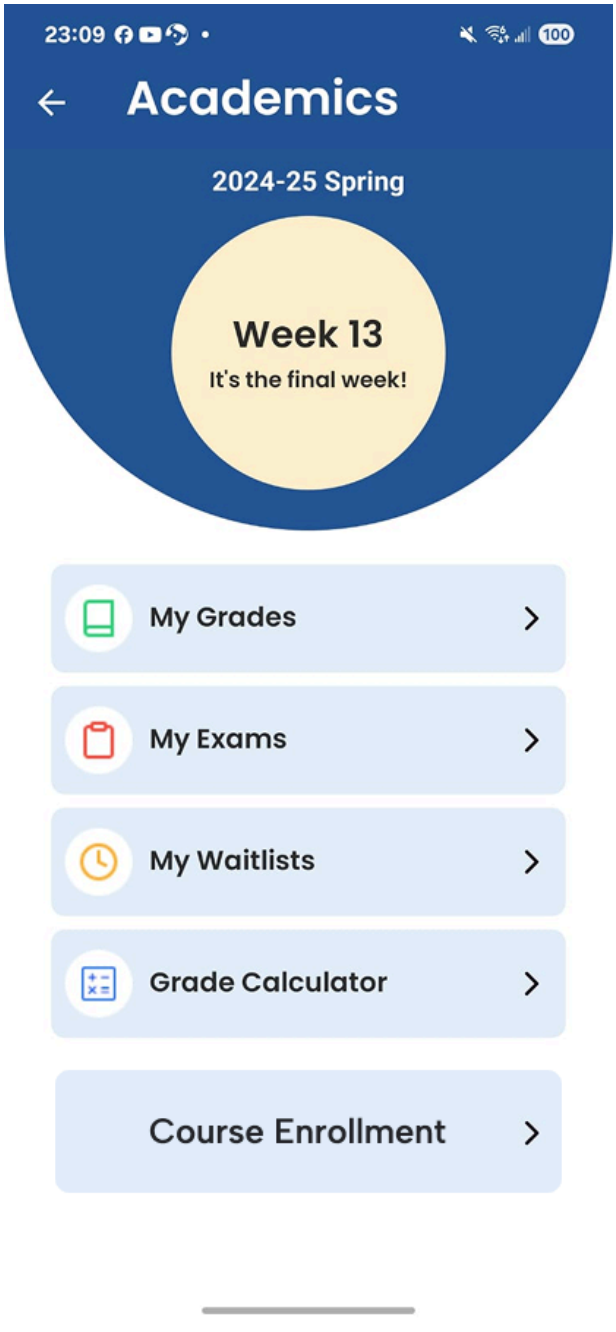
Have you experienced any technical issues while using the information system? If yes, please describe.

Questionnaire link:
https://docs.google.com/forms/d/e/1FAIpQLSc22cSkRGzEtHOLV84E_xuZLIfs5U7o5X9EvhT9QMjepM7qAA/viewform?usp=dialog

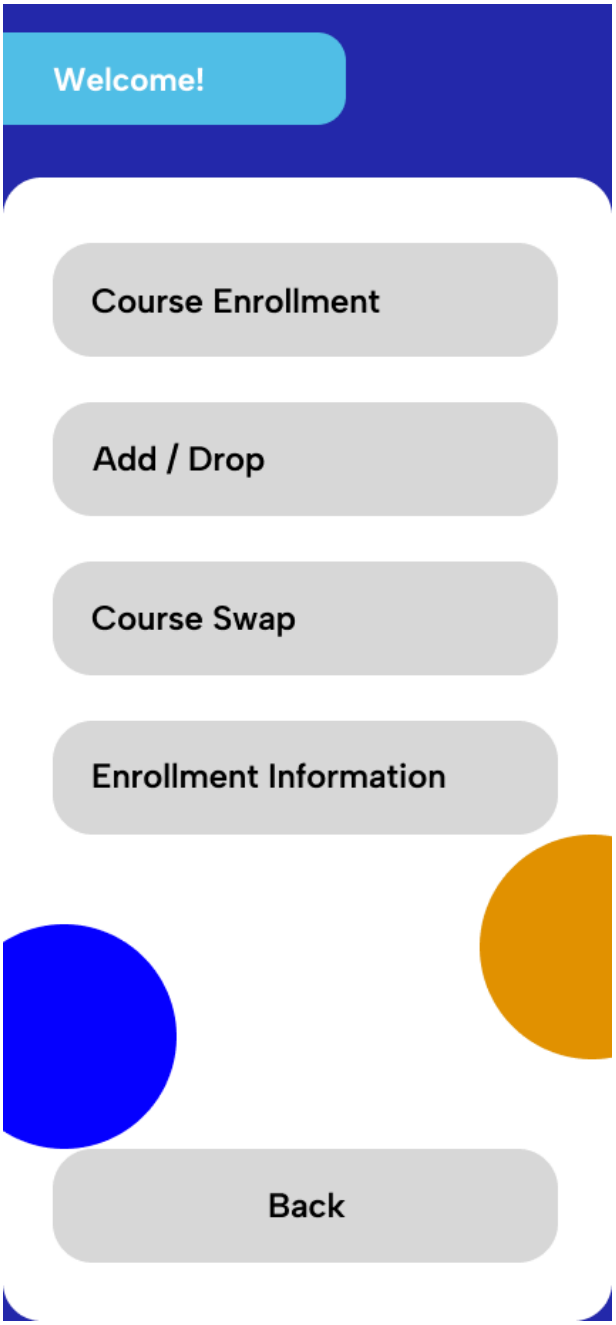
Phase 6

Mobile UI Prototype

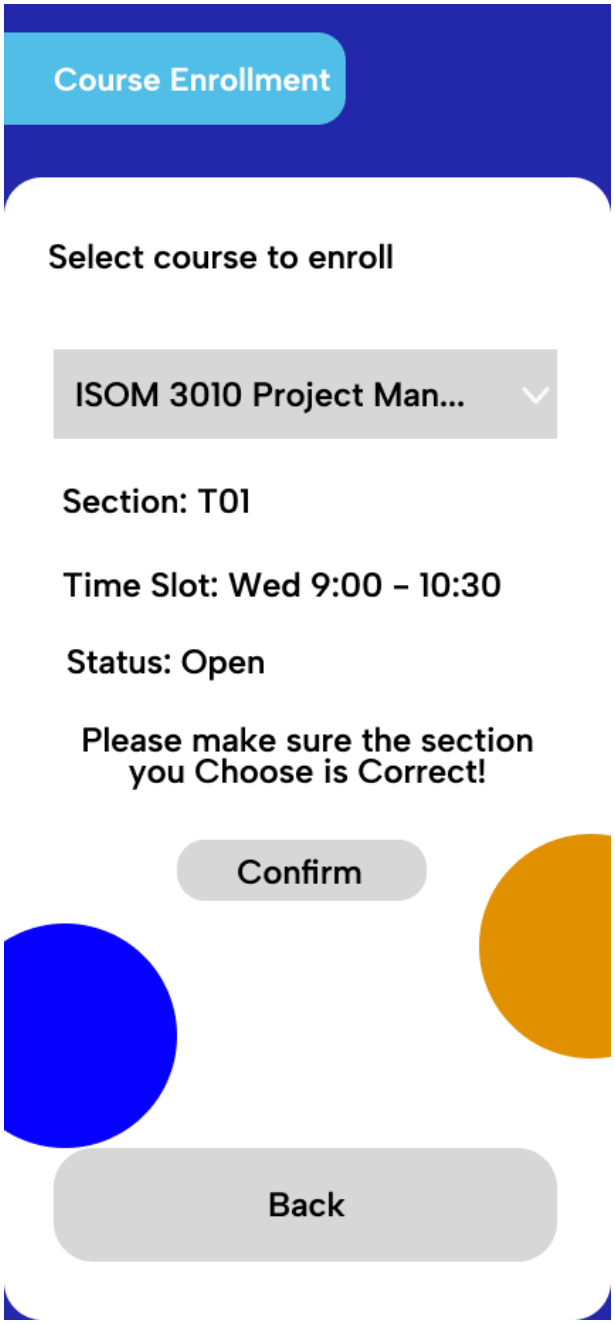
6.1 Academic page for mobile UI



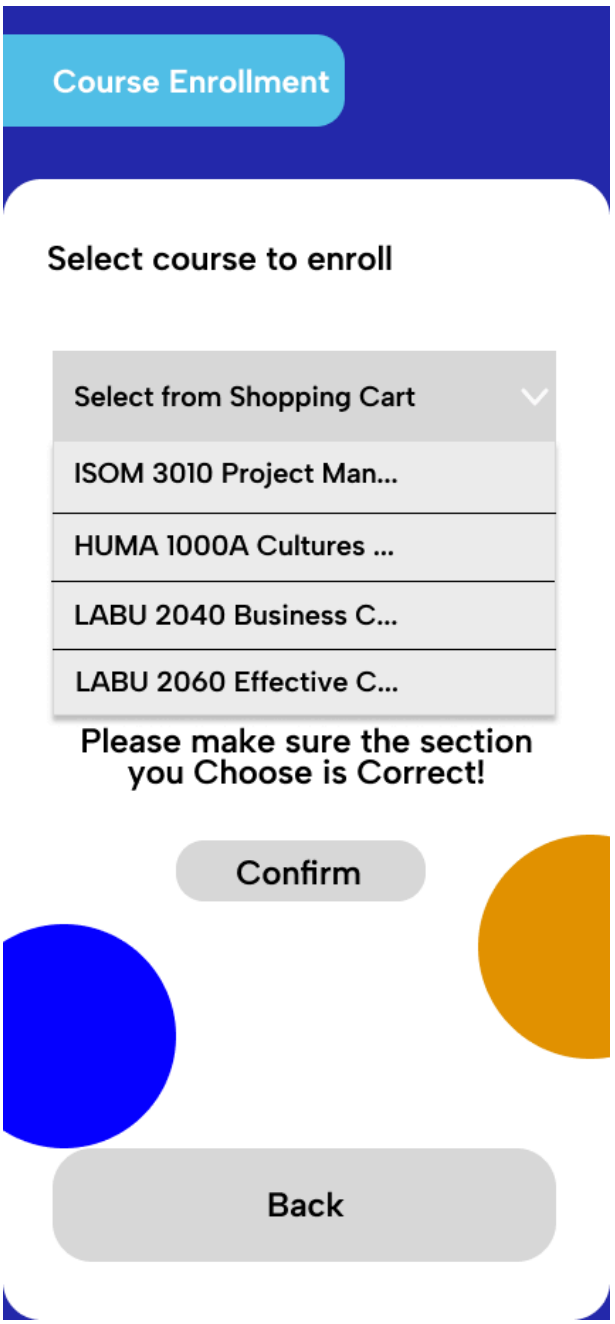
6.2 Selection of closure enrollment functions



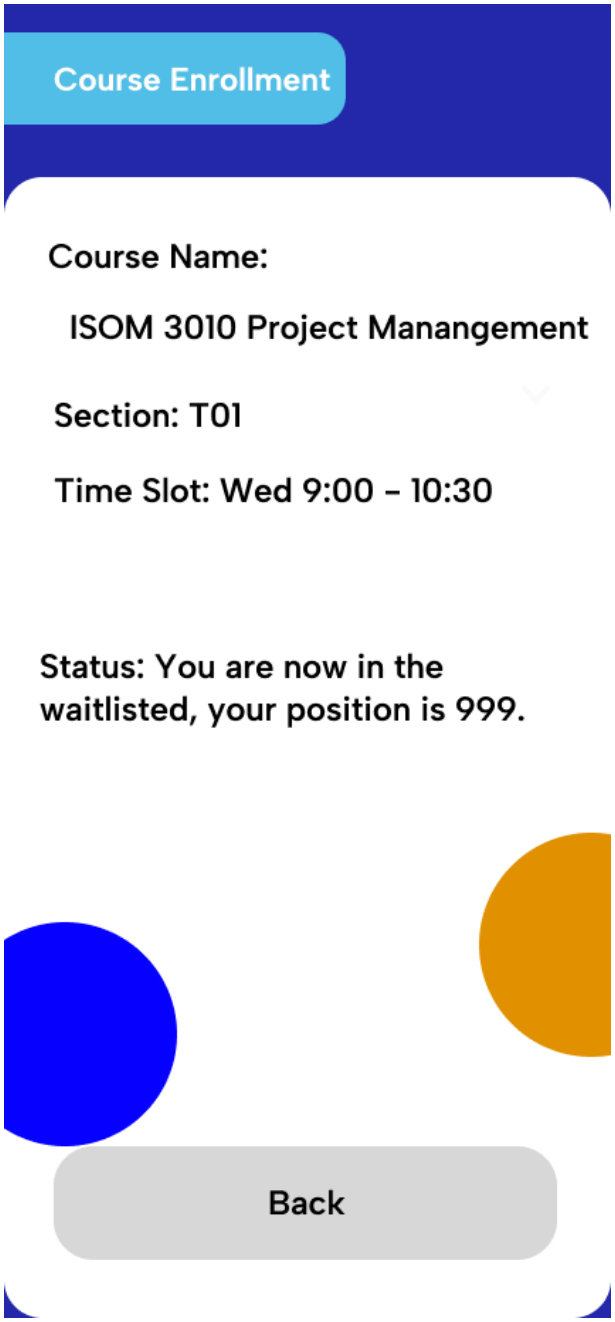
6.3 Selection of course to enroll



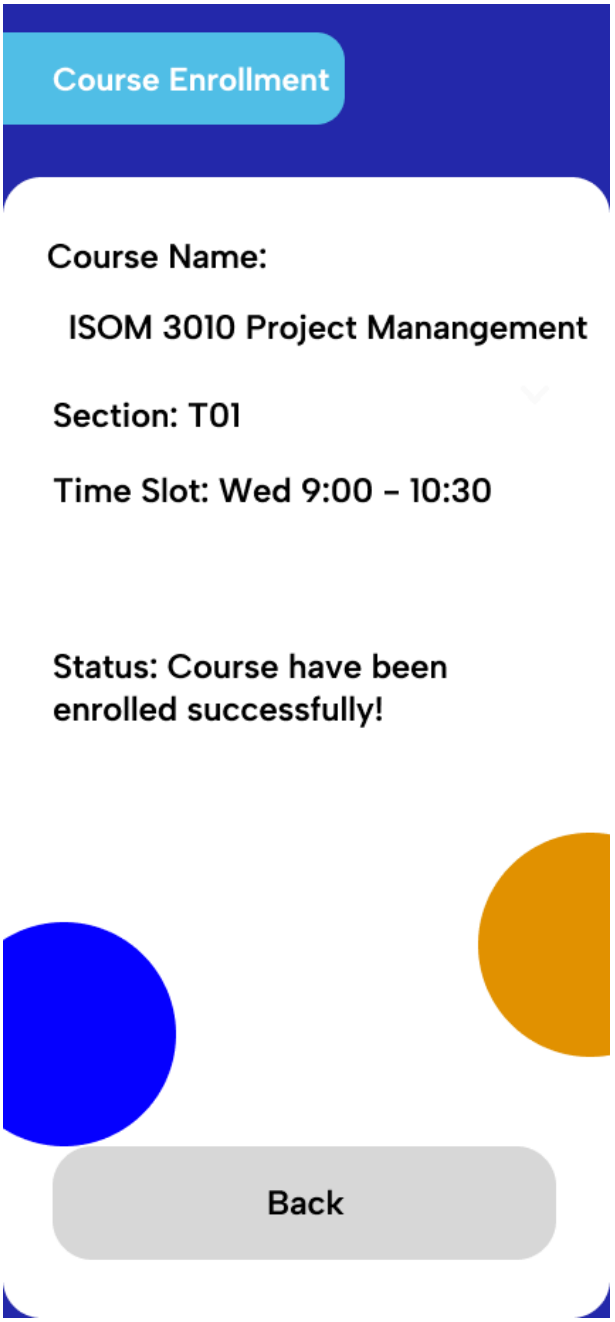
6.4 Scrollbar Shopping Cart



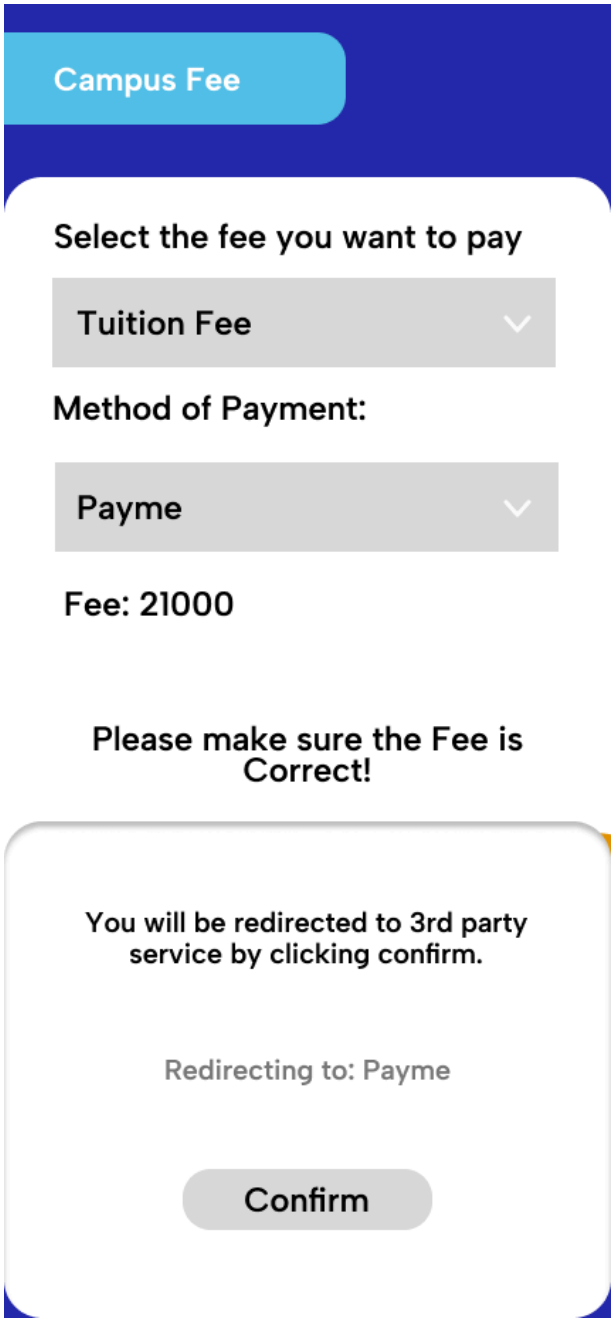
6.5 Status of Enrollment (1)



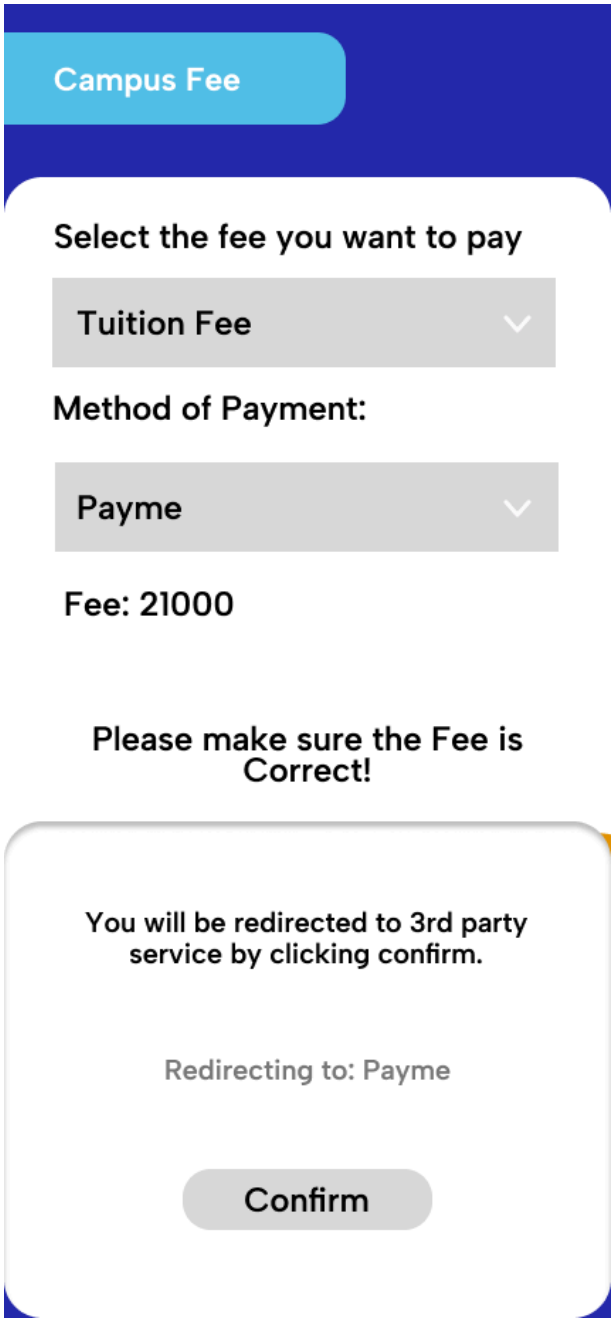
6.6 Status of Enrollment (2)



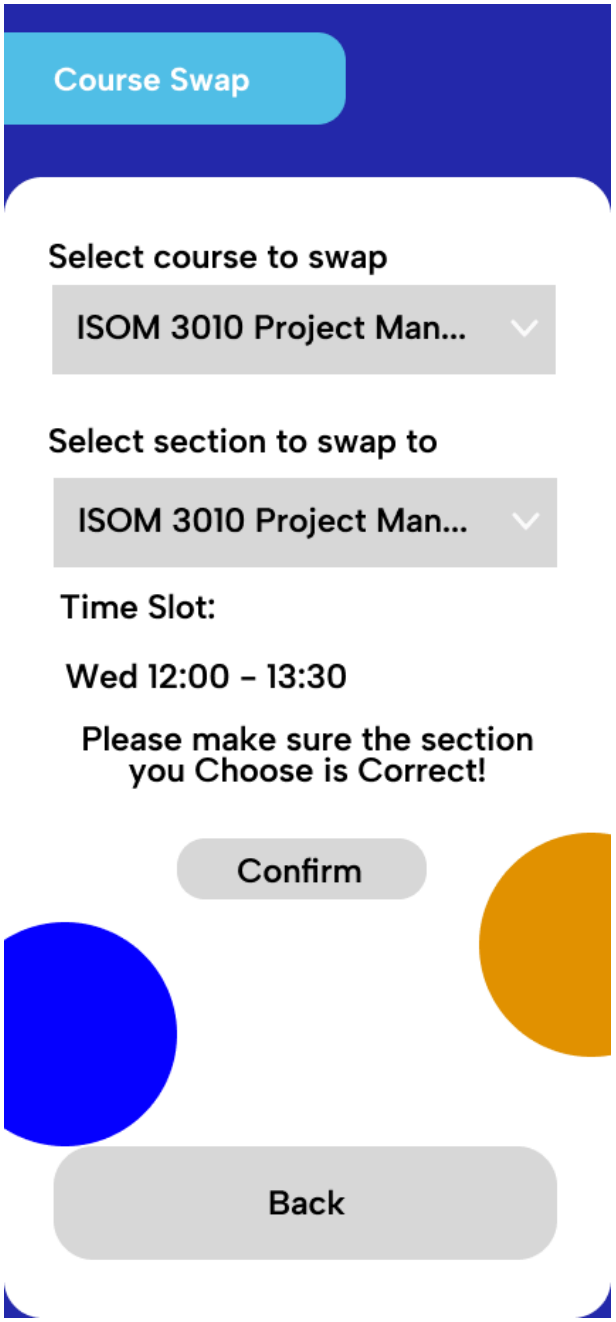
6.7 Tuition Fee Payment Function



6.8 Payment Confirmation



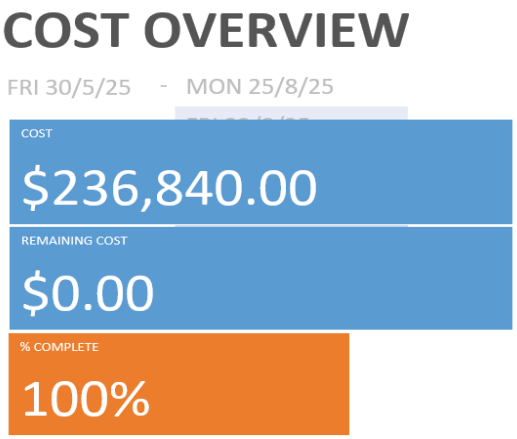
6.9 Course Swap

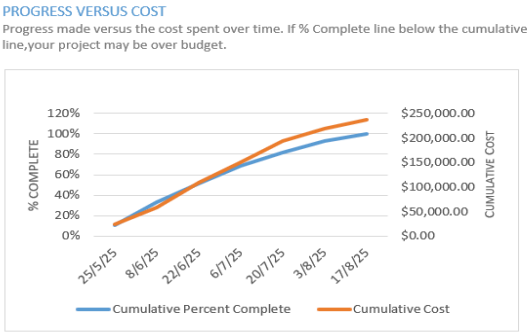


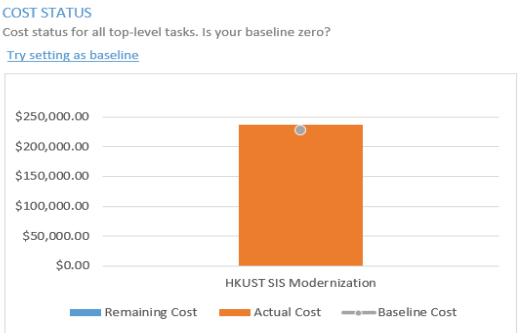
Phase 7

MS Project Reports

7.1 Final Cost Overview





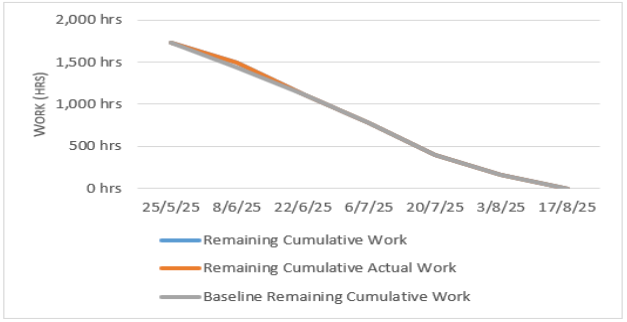


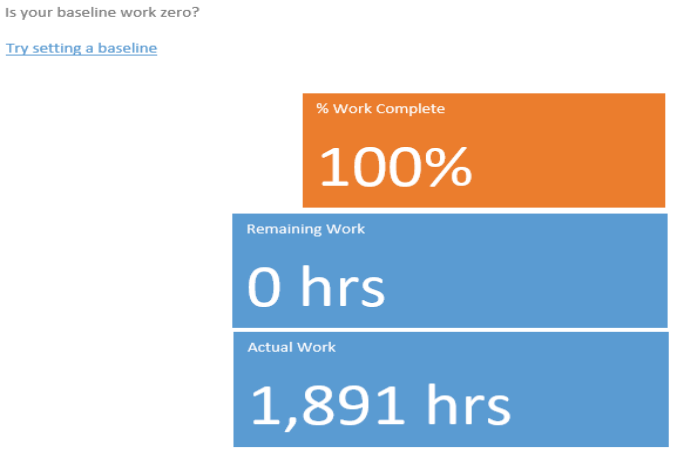
COST STATUS

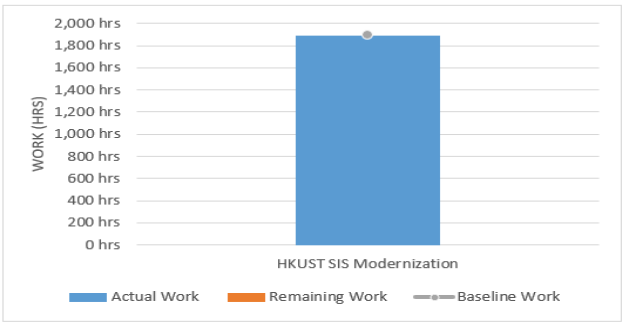
Cost status for top level tasks.

Name	Actual Cost	Remaining Cost	Baseline Cost	Cost	Cost Variance
HKUST SIS Modernization	$236,840.00	$0.00	$226,830.00	$236,840.00	$10,010.00

7.2 Final Work Overview







WORK STATS

Shows work stats for all top level tasks.