**CSIT214 IT Project Management**

**Group Project**

**MEMBERS And Contributions:**

He Shu Tsai(8348984) - Contributed:

Signature: HST

Ruiting Liu(8696548) - Contributed:

Signature: RL

WeiLin He(8443658) - Contributed:

Signature: WLH

Brody Key(9094209) - Contributed:

Signature: BK

Dehao Wu(5614740) - Contributed:

Signature: DW

**Table of Contents**

[Project Justification 2](#_Toc211029865)

[Business Case 4](#_Toc211029866)

[Project Charter 8](#_Toc211029867)

[Project Scope Statement 10](#_Toc211029868)

[Work Breakdown Structure and Schedule 12](#_Toc211029869)

[Risk Management 13](#_Toc211029870)

[Effort and Cost Estimation 14](#_Toc211029871)

[Demonstration of Project Execution 17](#_Toc211029872)

[Version Control Evidence 18](#_Toc211029873)

[Project Closing and Lessons Learned 19](#_Toc211029874)

[Product Artefacts 21](#_Toc211029875)

[User Interface Prototype 22](#_Toc211029876)

[Functionality Implementation 23](#_Toc211029877)

[References 24](#_Toc211029878)

[Appendices 25](#_Toc211029879)

Project Justification

Method and Criteria:

Our team uses the weighted scoring model as a project screening method because it allows us to evaluate each project based on multiple criteria, assigning weights to each criterion based on its importance. It provides us with a systematic project screening process based on multiple criteria.

The criteria defined by our group include:

Strategic alignment with core business goals (30%)

Customer value and coverage (20%)

Feasibility and technical complexity (15%)

Time to implement (5%)

Risk level (10%)

Expected financial value (20%)

Evaluation Results:

Each project is scored based on each criterion, with a maximum score of 100 and a minimum score of 0. A higher score indicates that the project more closely meets that criterion. The score is then multiplied by the criterion weight to create a weighted total score. The results showed that Project 1 achieved the highest weighted score (86), which was higher than Project 2 (75.75) and Project 3 (65.75).

Conclusion:

Based on the Weighted Scoring Model analysis, our group has selected Project 1. It is highly aligned with the organization's strategy and has the greatest impact on customers, so it received the highest score when evaluating the weighted criteria.

一張含有 文字, 螢幕擷取畫面, 數字, 軟體 的圖片

AI 產生的內容可能不正確。

**Figure 1:** Weighted Scoring Model

Business Case

Summary of the Case:

FlyDreamAir is a major airline operating across domestic and international routes. To stay competitive and meet modern travellers’ expectations, the company seeks to digitalise its customer management and booking processes. The proposed project, Customer Management and Booking System, will integrate key functionalities including flight booking, reservation management, seat selection, and in-flight service purchases.

Among all evaluated options, Project 1 was selected through a structured weighted scoring analysis due to its ability to directly enhance customer experience, increase revenue potential, and reduce operational inefficiencies. This business case further assesses the project’s justification and viability through a situation analysis, cost–benefit evaluation, and feasibility study.

It is critical to note that this initial project phase, defined by academic and time constraints, focuses on delivering a core functional prototype (Minimum Viable Product - MVP) for internal testing and validation. Subsequent phases will address real-world integration, mobile development, and cloud deployment necessary for FlyDreamAir's full digital transformation.

Situation Analysis:

Current Business Activities:

* Manual Processes: Many customer-facing activities such as booking, check-in and onboard purchases are handled manually or through incomplete systems, leading to inefficiencies, increased labor costs and a significant increase in the potential for human error.
* Insufficient Customer Experience: The existing system lacks integration and customer experience-oriented design, which leads to a very poor user experience for customers. This indirectly causes customers to frequently call customer service to resolve problems, thereby reducing overall customer satisfaction.
* Limited data utilization: Outdated systems cannot provide complete data analysis to company management, causing the company to miss many opportunities to use data to drive decision-making improvements.
* Internal coordination barriers: Different departments within the company use different systems (ticketing, customer service, catering), which leads to asynchronous coordination and frequent information gaps.

Market Environment:

* Competitive pressure: Many competitors have integrated their systems and launched multi-functional mobile platforms, which has led to a gradual decline in the company's market share.
* Increased customer expectations: With the advancement and development of technology, passengers have higher and higher requirements for airlines. They not only want a comfortable and convenient riding experience, but also want to have a good experience when using the company's systems.

Cost vs. benefit analysis

To better understand if this project is worth doing, we looked at both the cost and what we might gain from it. Based on the scoring result mentioned earlier, Project 1 received 86 out of 100, which is higher than Project 2 and Project 3. That’s why we chose this one for further analysis.

One of our criteria in the scoring model was "expected financial value", and it counted for 20% of the total score. So we believe it’s important to look into the financial side, such as development cost and long-term benefits.

The total estimated cost of the system is about AUD 125,000 over five years. This includes development, maintenance, staff training, and support. In return, the system is expected to reduce staff workload, improve booking efficiency, and create new income opportunities (like seat upgrades or extra services).

We also did a simple NPV (Net Present Value) calculation. With a 10% discount rate, the NPV came out as positive AUD 29,620. This means the project is likely to pay for itself and even bring in more value in the long run.

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

AI 產生的內容可能不正確。

**Figure 2:** Expected Financial Value

Feasibility study:

Technical Feasibility:

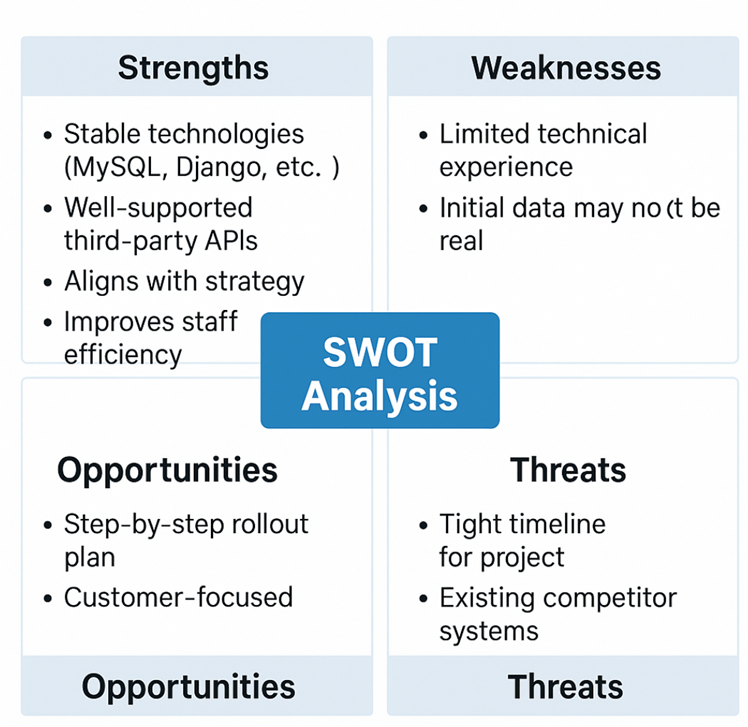
From the technical side, the system is possible to build. We can use common tools like MySQL, Django, and third-party APIs for things like payment and notifications. These technologies are stable and well-supported, so it shouldn’t be a problem for the team.

Operational Feasibility

The system can help staff reduce manual work and also give customers a smoother experience. Since the project is also aligned with the company’s strategy and focuses on customer needs (as mentioned in the justification part), we think it will work well in real use. We can also launch it step by step to avoid sudden changes.

Economic Feasibility

Looking at the cost and benefit, the system is not very expensive, and the NPV is positive. Also, we expect the payback period to be Year 4. So overall, it seems to be a safe and valuable investment

****

**Figure 3:** SWOT Analysis

Project Charter

Project Title: FlyDream System

Date of Authorization: September 1, 2025

Project start date: September 1, 2025

Project finish date: September 1, 2030

Project Manager: He Shu Tsai (hst511@uowmail.edu.au)

Key Schedule Milestones:

* Complete system design and prototype development within the first three months (by December 1, 2025).
* Complete full software production and submission within the first year (by September 1, 2026).
* Operate and maintain the system for the remaining four years (September 2026 – September 2030).

Budget Information:

The total estimated cost of this project is AUD 125,000 over five years, which includes development, maintenance, staff training, and support. The project is expected to reduce staff workload, improve booking efficiency, and create new income opportunities (such as seat upgrades and extra services). With a 10% discount rate, the calculated Net Present Value (NPV) is positive AUD 29,620 , and the payback period is expected to be in Year 4 , indicating that the project is financially viable

Project Objectives:

The objective is to complete the first version of the system software by December 2025 and deliver the production version by September 2026, followed by system operation and maintenance until 2030.

Main Project Success Criteria:

The system must meet all written specifications, undergo thorough testing, and be completed on schedule. The CEO will formally approve the project with advice from other key stakeholders.

Approach：

Recruit technical staff and part-time support to establish a dedicated development and maintenance team for the five-year project.

Within the first three months, complete system design, prototype, and detailed planning documents (WBS, Scope, Gantt Chart).

By the ninth month, complete acquisition and integration of all required tools and APIs.

By December 2025, deliver the first version.

Hold quarterly project review meetings (instead of weekly) with the project team and sponsor to evaluate milestones and adjust plans.

By September 2026, deliver the production version and begin system operation and maintenance.

|  |  |  |
| --- | --- | --- |
| ROLES AND RESPONSIBILITIES | | |
| He Shu Tsai | Project Manager | hst511@uowmail.edu.au |
| Ruiting Liu | Team Member | lr263@uowmail.edu.au |
| WeiLin He | Team Member | 13917666468@163.com |
| Brody Key | Team Member | bk865@uowmail.edu.au |
| Dehao Wu | Team Member | dw328@uowmail.edu.au |

Project Scope Statement

Project Objective:

To design, develop, and test a full-stack software system that enables customers to register, search and book flights, manage seat reservations, and purchase in-flight services (e.g., food, drinks), while also providing administrative tools for managing flights and bookings.

Scope Description:

This project will deliver a locally hosted, web-based platform that supports both customer-facing and administrator-facing interfaces. Core components include account management, flight booking, reservation control, and service purchases. The solution will follow a modular architecture with back-end, front-end, and database layers, and will be developed using version-controlled collaborative tools.

This deliverable is constrained by the academic environment and is intended as a Proof-of-Concept (PoC) of the core business logic, utilizing mock data and a local testing environment to manage technical complexity and project timeline risks. The resulting local application serves as a foundational blueprint for future, production-ready development.

In Scope:

* + User registration, login, and profile management
  + Flight search with filters (origin, destination, date)
  + Flight booking and real-time seat selection
  + Interface for purchasing in-flight services
  + Admin dashboard to manage:
  + Flight schedules
  + Customer records
  + Booking data
* Fully responsive front-end (desktop only)
* Back-end API with mock flight data (no real integrations)
* Local test environment setup
* Documentation:
* Technical documentation
* User manual
* Testing (unit + system-level)

Out Of Scope:

* Payment processing (e.g., PayPal, Stripe)
* Integration with real airline APIs
* Mobile application development
* Cloud hosting or public server deployment
* Ongoing post-submission support or updates

Deliverables:

* A functional web application running on a local test environment
* Technical documentation and user guide
* GitHub project repository with commit logs
* Milestone tracking (Microsoft Project or Agile tool)
* Final group presentation/demo

Constraints:

* Team of 5 student developers
* Must use
* GitHub for version control
* Microsoft Project for schedule tracking
* Taiga/JIRA optionally for Agile boards
* All development must run on local-host/test machine

Final scope must be frozen by Week 4

Assumptions:

* All team members are available for weekly meetings and work
* Mock flight data will be generated or preloaded manually
* All assets (UI, icons, content) are created by the team
* Tools and frameworks used are free/open-source or provided by the university

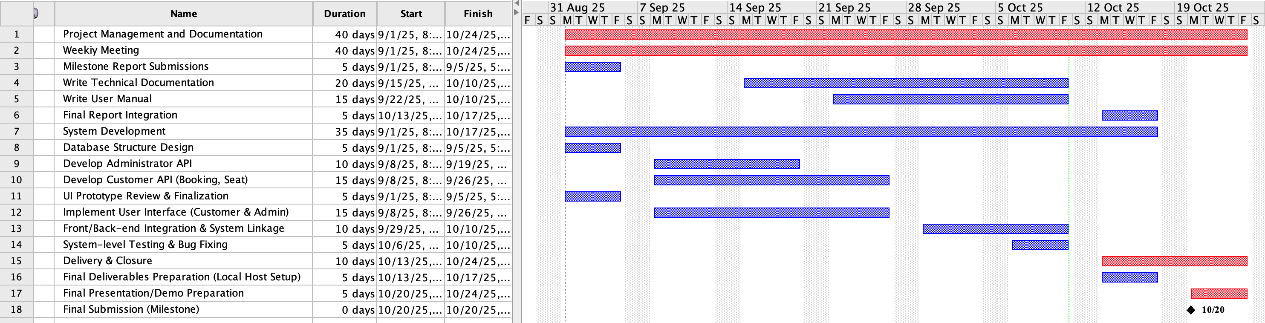
Work Breakdown Structure and Schedule

WBS:

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

AI 產生的內容可能不正確。

**Figure 4:** Work Breakdown Structure

Project Schedule:

**Figure 5:** Gantt Chart

Risk Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk ID** | **Description** | **Likelihood** | **Impact** | **Mitigation Strategy** |
| R1 | Group member drops out or becomes unresponsive | Medium | High | Define backup roles early; redistribute workload dynamically |
| R2 | Communication breakdown (e.g., unclear tasks or misalignment) | Medium | Medium | Use Agile tools like Trello/Taiga, have weekly sync calls |
| R3 | Git merge conflicts or overwrites | Low | High | Strict branch-per-feature workflow, use pull requests only |
| R4 | Missed deadlines on milestones | Medium | High | Set internal buffer deadlines; weekly status meetings |
| R5 | Technical/tooling issues (e.g., GitHub downtime, MS Project crashes) | Low | Medium | Keep local backups, export task plans regularly |
| R6 | Scope creep due to enthusiasm or user feedback | Medium | Medium | Lock features by Week 4; require team vote for new features |
| R7 | Team workload spikes from overlapping uni assignments | High | Medium | Plan around assessment calendar; allow flexible task assignments |
| R8 | Incomplete or insufficient testing | Medium | Medium | Schedule test sessions; assign dedicated test ownership |
| R9 | Front-end/back-end integration bugs | Medium | Medium | Define API contracts early; conduct integration testing weekly |
| R10 | Misestimation of development effort | Medium | High | Use function point estimation; re-evaluate task loads regularly |

Effort and Cost Estimation

Function Point Analysis:

Identify System Functions:

The following table summarizes the system’s functional modules, their corresponding FP types, complexity levels, and assigned weights (based on Albrecht's complexity multipliers):

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Module** | **FP Type** | **Complexity** | **Weight** |
| User Registration / Login | EI | Low | 3 |
| Flight Search | EQ | Medium | 4 |
| Flight Info Display | EO | Medium | 5 |
| Seat Selection | EQ | Medium | 4 |
| Ticket Info Input | EI | Medium | 4 |
| Ticket Confirmation / Payment | EO | Medium | 5 |
| Order History Viewing | EQ | Medium | 4 |
| Admin Flight Data Management | EI | High | 6 |
| Flight Database | ILF | High | 15 |
| User Data | ILF | Medium | 10 |

Total Unadjusted Function Point (UFP) = 3 + 4 + 5 + 4 + 4 + 5 + 4 + 6 + 15 + 10 = 60

General System Characteristics (GSCs):

We evaluated the 14 General System Characteristics (GSCs) to calculate the Value Adjustment Factor (VAF). The total Degree of Influence (TDI) was determined to be 43, based on the following criteria:

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **GSC Name** | **Score** | **Justification** |
| 1 | Data Communications | 3 | Requires online interaction between customer-facing front-end and back-end server/database for real-time seat selection and booking updates. |
| 2 | Distributed Data Processing | 1 | System is mainly web-based with limited distributed logic. |
| 3 | Performance | 4 | High-performance expected during peak booking hours. |
| 4 | Heavily Used Configuration | 2 | System supports multiple users simultaneously but under moderate load. |
| 5 | Transaction Rate | 3 | Transactions are moderately frequent. |
| 6 | Online Data Entry | 4 | Users input booking and payment data online. |
| 7 | End-user Efficiency | 3 | Designed with usability in mind. |
| 8 | Online Update | 4 | Supports real-time seat selection and updates. |
| 9 | Complex Processing | 3 | Involves data validation, rule checks, and payment confirmation. |
| 10 | Reusability | 2 | Backend services may be reused in other modules. |
| 11 | Installation Ease | 2 | Relatively easy to deploy. |
| 12 | Operational Ease | 3 | Basic logging and error handling present. |
| 13 | Multiple Sites | 2 | System accessed by users from multiple locations. |
| 14 | Facilitate Change | 3 | Modular architecture allows for future enhancements. |

Total Degree of Influence (TDI) = 43

Value Adjustment Factor (VAF) is calculated as:

VAF=0.65+(0.01×TDI)=0.65+0.43=1.08

Final FP Count:

Final Function Point=UFP×VAF=60×1.08=64.8 ≈ 65

COCOMO Analysis:

To further estimate the development effort, we apply the COCOMO II model using the Function Point–derived size. The adjusted final Function Point is 65, which we convert to estimated Lines of Code (LOC) using a general industry average of 80 LOC per FP:

Estimated LOC = 65 × 80 = 5,200 LOC → 5.2 KLOC

Since our project is an organic system (developed by a small team in a familiar in-house environment), we use the constants:

* A = 2.94
* B = 0.91

Scale Factor Calculation:

We assign the following ratings based on the system characteristics:

|  |  |  |
| --- | --- | --- |
| **Factor** | **Value** | **Reason** |
| **PREC** | 3.72 (Nominal) | Similar past systems |
| **FLEX** | 3.04 (Nominal) | Some flexibility allowed |
| **RESL** | 2.83 (High) | Risks partially resolved |
| **TEAM** | 2.19 (High) | Tightly coordinated team |
| **PMAT** | 4.68 (Nominal) | Standard processes applied |

sf = 0.91 + 0.01 × (3.72 + 3.04 + 2.83 + 2.19 + 4.68) = 0.91 + 0.1646 = 1.0746

Effort Calculation:

We use the COCOMO II formula:

Effort = A × (Size)^sf = 2.94 × (5.2) ^1.0746 ≈ 8.23 person-months

Cost Estimation and Comparison:

Assuming an average cost of AUD 10,000 per person-month, the development cost based on COCOMO would be:

8.23 × 10,000 = AUD 82,300

This estimate fits reasonably within the broader AUD 125,000 total project cost discussed in the cost-benefit analysis. That analysis includes additional long-term costs such as staff training, maintenance, and system support.

This consistency between technical estimation and financial planning reinforces the feasibility and justification of the project.

Demonstration of Project Execution

Milestone reports:

1. Project Approval (Week 7)

Achieve goals: Complete and lock in the Project Charter, Scope Statement, WBS, and initial schedule.

Challenge: The team spent extra time prioritizing requirements.

Control: The project manager formally signed off on the project scope baseline, and all subsequent changes were subject to the change control process.

2. Design Sign-off (Week 10)

Achieve goals: The system's database structure and all user interface UI prototype designs.

Challenge: Slow confirmation of backend API specifications may cause delays.

Control: The team decided to increase meeting time to speed up communication and signing of front-end and back-end interfaces.

3. Core MVP delivery (Week 12)

Achieve goals: All core functions (booking, customer records) are written and have passed system integration testing.

Challenge: Many minor errors were discovered during the testing phase.

Control: Adjust schedule to allocate more time from documentation work to bug fixing to ensure the system is stable before the final presentation.

Project Tracking:

We implemented schedule tracking using ProjectLibre to compare the differences between plan and actual execution.

When the project was finalized, we set the entire plan schedule. After that, the PM would check every week to see if the set goals were achieved. We would organize meetings every week and each team member would report the project progress to the PM. Most of the time, these meetings were held in the library, and only a few times were held online through Discord.

Version Control Evidence

Project Closing and Lessons Learned

Project Performance Evaluation: Triple Constraints

We evaluated the project's success based on its three constraints: scope, time, and cost.

1. Scope Goal

* Evaluation: The project successfully achieved all core functionality defined in the Project Scope Statement. We successfully avoided scope creep, and all additional requirements were rejected or deferred to future phases through formal change control.
* Conclusion: Success.

2. Time Goal

* Evaluation: By using ProjectLibre to set goals and implement milestone tracking, we delivered all deliverables two days ahead of the project's original deadline. Although we experienced a slight delay during the database design phase, we were able to get the project back on track by compressing the page design schedule.
* Conclusion: Success (or nearly so).

3. Cost Goal

* Evaluation: The project's actual hours and resource utilization slightly exceeded the budget, resulting in an actual cost of AUD 84,000, exceeding the COCOMO baseline estimate of AUD 82,300. This overrun primarily stemmed from addressing the R9 risk (front-end/back-end integration error), which resulted in an additional 40 hours of work. However, the total cost remained within our budget.
* Conclusion: Acceptable (well-controlled but with a slight overrun).

What Went Right and Wrong:

What Went Right:

* Strong Planning Foundation: A detailed WBS provided a clear blueprint for development, ensuring team members had a consistent understanding of the scope of work and significantly reducing communication errors.
* Proactive Risk Management: We identified and planned for R4 (Missed Deadlines) and R7 (Team Workload Spikes) early in the project. When these issues occurred, we were able to quickly apply pre-planned mitigation measures and avoid significant delays.

What Went Wrong:

* Optimistic time estimates: The System Level Testing duration estimate was overly optimistic, causing the actual duration of this phase to exceed the ProjectLibre baseline estimate by 20%.
* Meeting efficiency: Although we held regular meetings, early meeting minutes were sometimes inaccurate, and the allocation of action items was unclear.

Lessons Learned for the Next Project:

Based on the experience of this project, we have summarized the areas that will be improved in the next project.

* Add buffer time when scheduling important tasks to reduce the time squeezed out of other processes due to lack of time in important stages such as testing.
* This time we did not use MS Project for tracking, but later found that using this software could track the progress of the project more effectively, saving a lot of time and also reducing meeting time.

Product Artefacts

User Interface Prototype

Functionality Implementation

References

Appendices