## 

|  |
| --- |
| **International School**  Đồ Án CDIO  **CMU-CS 447**    **PROJECT PLAN**  Version 1.0  Date: 05- April - 2025  Airline Reservation System  Submitted by  Nguyen Pham Anh Huong  Cao Minh  Le Minh Hieu  Nguyen Thi Thanh Huong  **Approved by**  **Capstone Project 1 - Mentor:**  Name Signature Date  Tinh, Le Van \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_04 - April- 2025 |

**TABLE CONTENTS**

**[PROJECT INFORMATION 1](#_Toc17140)**

**[DOCUMENT INFORMATION 2](#_Toc22502)**

**[REVISION HISTORY 3](#_Toc3516)**

**[1. INTRODUCTION 4](#_Toc21928)**

[1.1. PROJECT NAME 4](#_Toc11081)

[1.2. PROJECT DURATION 4](#_Toc9514)

[1.3. GOAL OF PROJECT 4](#_Toc32362)

[1.4. PROJECT SCOPE 4](#_Toc14447)

**[2. TEAM ORGANIZE 5](#_Toc10692)**

[2.1. WATERFALL TEAM INFORMATION 5](#_Toc25997)

[2.2. ROLES AND RESPONSIBILITY 5](#_Toc30986)

**[3. SCHEDULES 6](#_Toc18297)**

[3.1. OVERALL SCHEDULES 6](#_Toc26920)

[3.2. DETAIL SCHEDULES 6](#_Toc9749)

**[4. COST 7](#_Toc28311)**

[4.1. RESOURCES 7](#_Toc10149)

[4.2. TOTAL COST ESTIMATE 8](#_Toc21358)

**[5. DEVELOPMENT PROCESS 8](#_Toc24358)**

[PRINCIPLE AND DIFFERENT STAGES 8](#_Toc4853)

[RISKS AND SOLUTIONS 10](#_Toc1723)

**[6. DEVELOPMENT ENVIRONMENT 10](#_Toc26021)**

**[7. COMMUNICATION & REPORTING 11](#_Toc6533)**

[7.1. REPORTING METHODOLOGY 11](#_Toc21114)

[7.2. COMMUNICATION METHODOLOGY 11](#_Toc5776)

**[8. RISKS 12](#_Toc30558)**

**[9. DELIVERABLES 13](#_Toc8183)**

**[10. CONFIGURATION MANAGEMENT 14](#_Toc3730)**

**[11. REFERENCE 14](#_Toc27968)**

# PROJECT INFORMATION

|  |  |  |  |
| --- | --- | --- | --- |
| **PROJECT INFORMATION** | | | |
| **Project Acronym** | ARS | | |
| **Project Title** | Airline Reservation System | | |
| **Project Web URL** |  | | |
| **Start Date** | 5-Apr - 2025 | | |
| **End Date:** | 24 - May - 2025 | | |
| **Lead Institution** | International School, Duy Tan University | | |
| **Project Mentor** | M.Sc Tinh, Le Van | | |
| **Scrum Master** | Minh, Cao | Minhcao05092004@gmail.com | 0905575080 |
| **Team Members** | Huong, Nguyen Thi Thanh | nguyennguyenkhanhquynh@gmail.com | 0358692336 |
| Huong, Nguyen Pham Anh | ahhuong312@gmail.com | 0774442236 |
| Hieu, Le Minh | lhieu20231@gmail.com | 0901942400 |

Table 1 - Project Information

# DOCUMENT INFORMATION

|  |  |  |  |
| --- | --- | --- | --- |
| **DOCUMENT INFORMATION** | | | |
| **Document Title** | Project Plan | | |
| **Author(s)** | Group 3 | | |
| **Role** | ProjectPlan\_v1.0 | | |
| **Date** | 5-Apr - 2025 | File name | ProjectPlan\_v1.0 |
| **URL** | https://github.com/nnkq/myproject.git | | |
| **Access** | Project and CMU Program | | |

Table 2 - Document Information

# REVISION HISTORY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Person(s)** | **Date** | **Description** | **Approval** |
| Draft |  | 5-Apr - 2025 | Initiate proposal | x |
| 1.0 | All members | 12-Apr - 2025 | Finish content of proposal | x |
| 1.1 | All members | 19-Apr - 2025 | Update content & format | x |
| 1.2 | All members | 26-Apr - 2025 | Update Tasks schedule | x |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 3 - Revision History

# INTRODUCTION

This document outlines the project plan for the Airline Reservation System (ARS), detailing the scope, schedule, resources, and development process to achieve a scalable, user-friendly flight booking system.

## PROJECT NAME

The project’s name is: “Airline Reservation System”

Team members:

|  |  |
| --- | --- |
| 1. Cao Minh | Scrum Master/ Leader |
| 1. Nguyen Thi Thanh Huong Team | Member |
| 1. Nguyen Pham Anh Huong Team | Member |
| 1. Le Minh Hieu Team | Member |

## PROJECT DURATION

* Project will be started on: 05-Apr-2025
* Project will be finished on: 24-Apr-2025

## GOAL OF PROJECT

The Airline Reservation System (ARS) aims to streamline flight booking, ticket management, and customer service for airlines and travel agencies. Key objectives include:

* Develop a scalable, user-friendly system for flight reservations.
* Enable real-time seat availability checks, online payments, and automated notifications.
* Improve operational efficiency and customer satisfaction through a modern, cloud-based platform.

## PROJECT SCOPE

The ARS will implement the following features:

* Flight search and filtering by date, price, and destination.
* Multi-payment gateways (credit card, PayPal, crypto).
* Automated email/SMS confirmations for bookings and updates.
* Admin dashboard for analytics, flight scheduling, and inventory control.
* Support for customer booking, admin management, and travel agent bulk bookings.
* Language: English, Vietnamese
* Duration: 40 days (April 5 – May 24, 2025)

# TEAM ORGANIZE

## WATERFALL TEAM INFORMATION

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Email** | **Phone number** | **Role** |
| Cao Minh | Minhcao05092004@gmail.com | 0905575080 | Project manager |
| Nguyen Thi Thanh Huong | nguyennguyenkhanhquynh@gmail.com | 0358692336 | Team member |
| Nguyen Pham Anh Huong | ahhuong312@gmail.com | 0774442236 | Team member |
| Le Minh Hieu | lhieu20231@gmail.com | 0901942400 | Team member |

## ROLES AND RESPONSIBILITY

|  |  |  |
| --- | --- | --- |
| **Role** | **Responsibility** | **Name** |
| **Product Owner** | Defines the vision, prioritizes backlog, ensures alignment with business goals. | Le Minh Hieu |
| **Project manager** | Plans, coordinates, manages risks, ensures timely delivery within scope and budget. | Cao Minh |
| **Business analyst** | Gathers requirements, analyzes business needs, ensures feasibility of solutions. | Cao Minh, Le Minh Hieu, Nguyen Pham Anh Huong,  Nguyen Thi Thanh Huong |
| **Developer** | Writes code, implements web functionalities, ensures performance and security. | Nguyen Thi Thanh Huong |
| **Tester** | Validates website quality, finds bugs, ensures compatibility across devices. | Nguyen Pham Anh Huong |
| **Team member** | Collaborates, contributes expertise, supports project success. | Cao Minh, Le Minh Hieu, Nguyen Pham Anh Huong,  Nguyen Thi Thanh Huong |

# SCHEDULES

## OVERALL SCHEDULES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Phase** | **Iteration** | **Start Day** | **End Day** |
| 1 | Development | Requirements | 12-Apr-2025 | 17-Apr-2025 |
| Design | 18-Apr-2025 | 01-May-2025 |
| Implementation | 02-May-2025 | 08-May-2025 |
| System testing | 09-May-2025 | 12-May-2025 |
| Maintenance | 12-May-2025 | 22-May-2025 |

## DETAIL SCHEDULES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WBS** | **Task** | **Task owner** | **Start** | **End** | **Days** |
|  | **Sprint 1** |  | **Apr 12, 2025** | **Apr 21, 20250** | **10** |
| 1 | Preparation | Thanh Huong | Apr 12, 2025 | Apr 12, 2025 | 1 |
| 2 | Data Modeling | Cao Minh | Apr 13, 2025 | Apr 15, 2025 | 3 |
| 3 | Physical Database Design | Minh Hieu | Apr 16, 2025 | Apr 17, 2025 | 2 |
| 4 | Set up database | Anh Huong | Apr 18, 2025 | Apr 19, 2025 | 2 |
| 5 | Project Plan | Thanh Huong | Apr 12, 2025 | Apr 14, 2025 | 2 |
| 6 | Product Backlog | Minh Hieu | Apr 15, 2025 | Apr 16, 2025 | 2 |
|  | **Sprint 2** |  | Apr 22, 2025 | Apr 01, 2025 | **10** |
| 1 | Finding Data | Anh Huong | Apr 22, 2025 | Apr 23, 2025 | 2 |
| 2 | Data processing | Cao Minh | Apr 24, 2025 | Apr 25, 2025 | 2 |
| 3 | Data Warehouse Designing | Minh Hieu | Apr 26, 2025 | Apr 27, 2025 | 2 |
| 4 | ETL Process Validating | Anh Huong | Apr 28, 2025 | Apr 29, 2025 | 2 |
| 5 | Expanding database | Cao Minh | Apr 30, 2025 | May 01, 2025 | 2 |
|  | **Sprint 3** |  | May 02, 2025 | May 11, 2025 | **10** |
| 1 | RDF Data Cubes Designing | Thanh Huong | May 02, 2025 | May 03, 2025 | 2 |
| 2 | Setting up DW2RDF Process Tool | Cao Minh | May 04, 2025 | May 05, 2025 | 2 |
| 3 | Implementing DW2RDF Process | Minh Hieu | May 06, 2025 | May 07, 2025 | 2 |
| 4 | Setting up RDF Data Cubes & SPARQL | Anh Huong | May 08, 2025 | May 09, 2025 | 2 |
| 5 | Building a history data source | Thanh Huong | May 10, 2025 | May 11, 2025 | 2 |
|  | **Sprint 4** |  | May 12 2025 | May 22, 2025 | **10** |
| 1 | Validating RDF Data Cubes | Minh Hieu | May 12, 2025 | May 13, 2025 | 2 |
| 2 | Building SPARQL-REST API | Anh Huong | May 14, 2025 | May 15, 2025 | 2 |
| 3 | Building UI | Thanh Huong | May 16, 2025 | May 18, 2025 | 2 |
| 4 | Integrate | Cao Minh | May 19, 2025 | May 20, 2025 | 2 |
| 5 | Testing | Minh Hieu | May 21, 2025 | May 21, 2025 | 1 |
| 6 | Deploy | Anh Huong | May 22, 2025 | May 22, 2025 | 1 |
| 7 | Release | Cao Minh | May 22, 2025 | May 22, 2025 | 1 |

# COST

## RESOURCES

|  |  |  |
| --- | --- | --- |
| **Full Name** | **Role** | **Salary Rate (USD / Hour)** |
| Cao Minh | Scrum Master | 2.0 |
| Nguyen Thi Thanh Huong | Team Member | 2.0 |
| Nguyen Pham Anh Huong | Team Member | 2.0 |
| Le Minh Hieu | Team member | 2.0 |

## TOTAL COST ESTIMATE

|  |  |  |
| --- | --- | --- |
| **Stage** | **Duration(hours)** | **Cost ($)** |
| 1 | 376 | $752 |
| 2 | 356 | $712 |
| 3 | 387 | $774 |
| 4 | 498 | $996 |
| **Total** | **1617** | **$3234** |

# DEVELOPMENT PROCESS

**PRINCIPLE AND DIFFERENT STAGES**

The ARS project adopts the Scrum methodology, emphasizing incremental development with a transparent product backlog. The system is delivered in four sprints, each adding features like flight search, payment integration, and admin dashboards. Scrum involves:

* Daily Meetings: 15-minute stand-ups to discuss progress, plans, and impediments.
* Planning Meetings: Team defines sprint goals and tasks.
* Work Review Meetings: 2-hour demos of new features (e.g., booking UI, API functionality).
* Retrospective Meetings: 15–30-minute sessions to reflect on successes and improvements. This approach minimizes documentation, focusing on essential records for decision history and maintenance.

**AGILE - SCRUM ORGANISATION**

The SCRUM methodology involves the following three main players:

* Product Owner: Represents stakeholders, prioritizes features (e.g., booking, payment gateways).
* Scrum Master: Ensures Scrum adherence, removes blockers.
* Team: Four members with roles in development, testing, and documentation, self-organizing to deliver sprints.

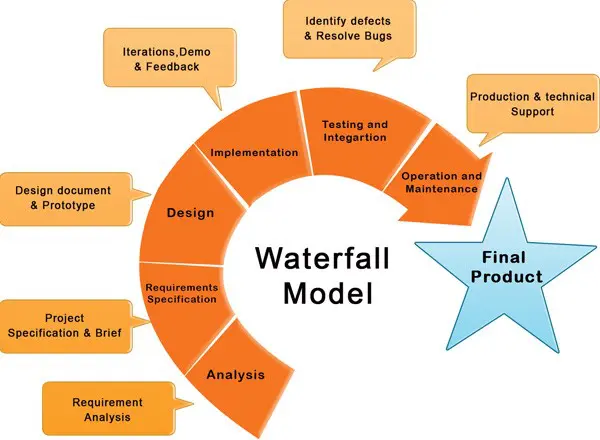


Figure 4.2: team members

**AGILE - ADVANTAGES**

Scrum differs from other development methods through its advantages which turn it into a pragmatic response to product owners' current needs:

Iterative and incremental method: this allows to avoid the "tunnel effect", i.e. the fact of seeing the result only at the final delivery, and nothing or almost nothing during the entire development phase, which is so frequent with V-cycle developments.

Maximum adaptability for product and application development: the sequential composition of the Stage content allows to add a modification or a feature which was not initially planned. This is precisely what renders this method "agile".

* Iterative delivery avoids tunnel effect, showcasing features like flight search early.
* Adaptable to changes, such as new payment methods or UI tweaks.
* Participatory, with all members contributing to decisions.
* Enhances communication via daily stand-ups and tools like Zoom.
* Maximizes cooperation with stakeholders through sprint reviews.
* Increases productivity by reducing formal documentation.

**RISKS AND SOLUTIONS**

The SCRUM methodology does not offer a universal answer to all the problems that are inherent to software development. Teams need to pay attention to the risks below, which, nevertheless, offer a systematic answer based on extrapolating the method:

* Risk: Payment gateway integration failures.
* Probability: Medium
* Severity: High
* Mitigation: Use mock APIs for testing, validate PCI-DSS compliance early.
* Risk: Flight data API downtime or unreliability.
* Probability: Medium
* Severity: Medium
* Mitigation: Implement fallback mock data, cache responses.
* Risk: Scalability issues with 10,000+ concurrent users.
* Probability: Low
* Severity: High
* Mitigation: Conduct load testing, optimize Node.js backend, use cloud load balancing.
* Risk: Team coordination challenges (4 members, tight timeline).
* Probability: Low
* Severity: Medium
* Mitigation: Daily stand-ups, clear task assignments via GitHub issues.

# DEVELOPMENT ENVIRONMENT

|  |  |
| --- | --- |
| **Component** | **Development Environment** |
| **Operating system** | * Windows |
| **Development Tools** | * Visual Studio Code * Cursor * Github * MySql workbench |
| **Data Warehouses** | * PostgreSQL |
| **Frameworks** | * Express.js * React |
| **Libraries** | * mysql2 * Sequelize * bcryptjs * Jsonwebtoken * Bootstrap * Axios |
| **Browsers** | * Chrome * Firefox * Edge |
| **Language** | * JavaScript (Node js/ React) * Html5 * Css3 * Php * ….. |

# COMMUNICATION & REPORTING

## REPORTING METHODOLOGY

|  |  |  |  |
| --- | --- | --- | --- |
| **Audience/ Attendees** | **Topic/ Deliverable** | **Frequency** | **Method** |
| - Product Owner  - Scrum Master  - Team Members | Project Progress Review | Weekly | Email, Skype or Zoom |
| - Product Owner  - Scrum Master  - Team Members | Explicit Requirement | When needed | Email, Skype or Zoom |
| - Mentor  - Scrum master  - Team members | Milestone review | End of each Milestone | Skype, Zoom |
| - Scrum master  - Team members | Daily tasks | Each day | Slack, Discord, Messenger |

## COMMUNICATION METHODOLOGY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of**  **Communication** | **Communication**  **Schedule** | **Communication**  **way** | **Who Initiates** | **Recipient** |
| Status Report  ( Daily meeting) | Daily | Slack, Discord | Scrum Master | Scrum Team |
| Schedule and  Effort Tracking | Daily | Face to face or GSheets | Scrum Master | Scrum Team |
| Work Review | Daily | Face to face | Scrum Master | Scrum Team |
| Work Report | Every Wednesday, Sunday | Face to face or Discord | Scrum Master | Scrum Team |
| Project Review,  ask problems | Every Wednesday, Sunday  (flexible) | Face to face, Discord or Zoom | Scrum Master | Scrum Team,  Mentor |
| Ask & Review  problems | Anytime | Face to face, Slack, Discord, Zoom | Scrum's Member | Mentor and  Scrum Team |

# RISKS

In this part of the document, it contains several risks that could happen to the development team in the future. It also includes probability, severity and mitigation strategy for each risk.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Definition** | **Probability** | **Severity** | **Mitigation Strategy** |
| Lack of coding experiences | No one in team member work with Python, React, Express, Data Warehouses, Data Cubes, Crawl data | H | M | Each team member has to learn and help the other to learn quickly. |
| Source Code conflict | Problems while merging code between members to master branch | H | H | Each team member must resolve conflicts by using git merge CLI before merging to master branch. |
| Member conflict | Team member maybe conflict with each other while discussing | H | M | Team building, playing board games to get everyone together. |
| Less equipment | No machine or hosting for deploying the server. | L | M | Try free hosting for deployment. |
| Time management | Every member has to go to work or school. | H | H | Overtime |
| Language barrier | Most of documents the are in English, sometimes it hard to understand clearly the articles and the informations | H | H | Improve the individual English skills in meanwhile doing the project. Asking the mentor technology for specific |

|  |  |  |  |
| --- | --- | --- | --- |
| **Probability** | | **Severity** | |
| L | Rarely happened. | L | Low damaged |
| M | Sometime happened | M | Medium damaged |
| H | Usually happened | H | Serious damaged |

# DELIVERABLES

|  |  |  |
| --- | --- | --- |
| **No** | **Activities** | **Deliverables** |
| 1 | Project Proposal | Project Proposal Document v1.0 |
| 2 | Project Plan | Project Plan Document v1.0 |
| 3 | Product Backlog | Product Backlog Document v1.0 |
| 4 | Architecture Document | Architecture Document v1.0 |
| 5 | Database Design | Database Design Document v1.0 |
| 6 | Interface Design | Interface Design Document v1.0 |
| 7 | Test Plan | Test Plan Document v1.0 |
| 8 | Test Case | Test Case Document v1.0 |
| 9 | Acceptance Criteria | Acceptance Criteria v1.0 |
| 10 | Stage Backlog & Burndown Chart | Stage Backlog & Burndown Chart v1.0 |
| 11 | Team Reflection | Team Reflection v1.0 |
| 12 | Technologies Stack | Technologies Stack Document v1.0 |

# CONFIGURATION MANAGEMENT

|  |  |  |
| --- | --- | --- |
| **No** | **Tool** | **Content** |
| 1 | Google Sheet | Track member activities. At the end of each day, team members will post on time log and scrum master will check. |
| 2 | Google Document | Track the changing of documents & manage versions of documents. |
| 3 | Github | Repositories for source code version management. |
| 4 | Zoom | Hold a meeting every week to assign tasks to each member.  If there are some emergencies but we cannot sit together then we can use Discord to discuss online. |
| 5 | Document | All meetings must be documented and pictured. |
| 7 | Slack | Store document resources and designed components |
| 8 | Discord | Discuss online, stream and share problems |
| 9 | CI/CD tool | Automated testing and deployment via Google Cloud pipelines. |

# REFERENCE

* PostgreSQL Global Development Group. (2023). PostgreSQL 15 Documentation. https://www.postgresql.org/docs/15/
* React Documentation. (2024). https://react.dev/learn
* Express.js Documentation. https://expressjs.com/
* Google Cloud. (2024). App Engine Deployment Guide. https://cloud.google.com/appengine/docs
* JWT Best Practices. https://jwt.io/introduction
* Schwaber, K. & Sutherland, J. (2020). The Scrum Guide. https://scrumguides.org/