

# Milestone 1: Software Development Plan & Specification

for

## Airline Pilots

Prepared by

Team's Name: FLYSG AIRLINES

Ivan Ng Say Mun	2003108
Carmen Wong Jiawen	2201680
Kenan Puay-Tee Tan	2100851
Casanas Joseph Christopher	2202186
Cortes	

**Lab Group:** P9 - 6

**Github handle:** <https://github.com/users/denmire/projects/1/views/1>

**Date:** 23/09/2023

## Table of Contents

<b>1 Introduction</b>	<b>3</b>
1.1 Product Scope	3
1.2 Related Background Literature	3
1.3 Intended Audience and Document Overview	3
1.4 References and Acknowledgments	4
<b>2 Overall Description</b>	<b>4</b>
2.1 Product Overview	4
2.2 Product Functionality	5
2.3 Assumptions and Dependencies	6
<b>3 Specific Requirements</b>	<b>7</b>
3.1 User Interface Requirements	7
3.2 Functional Requirements	16
3.3 Use Case Model	17
3.4 Non-functional Requirements	19
3.4.1 Performance Requirements	21
3.4.2 Safety and Security Requirements	21
3.5 Other Requirements	23
<b>4 Project Estimation and Plan</b>	<b>24</b>
4.1 Software Estimation	24
4.2 Project Management	26
4.3 Gantt Chart	27
<b>5 Appendix A – Use Case Descriptions</b>	<b>31</b>
<b>6 Appendix B – Data Dictionary</b>	<b>39</b>

# 1 Introduction

Scheduling is vital for businesses and manual scheduling can be time-consuming and expensive. Airline Pilot aims to optimize and provide a time efficient solution for scheduling a pilots job assignment. Airline Pilot is a work allocation management system that will be used by FLYSG AIRLINE company. The intended purpose is to enhance the work life balance among the pilots, while providing an efficient work allocation process that reduces the workload of FLYSG AIRLINE's management team. In this section, the project scope, related background literature, intended audience and document overview will be discussed.

## 1.1 Product Scope

The software system includes features to collate pilot data on availability, rank and experience. With the collated schedule, the airline management can allocate the flight schedule manually on a centralised platform with the help of a recommendation feature that will ease the allocation process. The software is not designed to generate flight schedules automatically but to assist the airline management in allocating flights and provide ease of access for pilots to view their job allocations on the go.

The benefits of this software will enhance the efficiency, fairness and compliance with regulations of the current flight scheduling method in addition to easing the airline management's workload. In addition to the benefits mentioned, we hope that the project also helps to enhance the work-life balance of the pilots by allowing them to indicate their preferred working days.

The deliverable of this project is a web application that can be accessed by both pilots and airline management that is user-friendly and visually aesthetic.

## 1.2 Related Background Literature

Leon software's aircraft management software [1] for schedule operators to allocate rosters was used as a reference for this project. This software allows the schedule on a large scale in an efficient and comprehensive manner. It also provides various features that support the daily operations. This software allows us to have a rough idea on the features we can include in our project. On the other hand, the airline crew scheduling terminology from the article "Airline crew scheduling: models, algorithms, and data sets" written by A. Kasirzadeh 1, M. Saddoune, and F. Soumis [2] has allowed us to understand certain terminology of the flight scheduling process.

## **1.3 Intended Audience and Document Overview**

This document is intended for our client, David Cheng Tim Leck, Prof Alex Q. Chen. Furthermore, our internal team members, consisting of developers and designers, rely on this document for technical specifications and information to guide the development process while staying informed about the project's scope, objectives, and constraints.

The document is organized to provide a clear and structured overview of the project, making it easy for the reader to understand the project's goal, scope, requirements and execution plan.

The document is divided into four parts:

1. Introduction: Provides a concise project overview, introducing the product's scope, relevant background literature, intended audience, and references
2. Overall Description: Offers a high-level perspective of the project, including its scope, functionality, and underlying assumptions and dependencies.
3. Specific Requirements: Explores the intricate project requirements in depth, covering user interfaces, functionalities, use cases, and non-functional aspects like performance and security.
4. Project Estimation and Plan: Outlines the project's execution strategy, including software estimation, project management details, timelines, budgets, and risk assessment.

## **1.4 References and Acknowledgments**

[1] A. Kasirzadeh 1, M. Saddoune, and F. Soumis, “Airline Crew Scheduling: Models, algorithms, and data sets,” *EURO Journal on Transportation and Logistics*, <https://www.sciencedirect.com/science/article/pii/S2192437620300820#sec2> (accessed Sep. 23, 2023).

[2] L. Software, “Leon for scheduled operators,” *Aircraft Management Software - Scheduled Operators*, <https://www.leonsoftware.com/leon/scheduled-operators.html> (accessed Aug. 4, 2023).

## **2 Overall Description**

### **2.1 Product Overview**

Our product is a standalone system accessible via the internet. Users will need internet connectivity to access the system. Once connected to the internet, users can visit the website and access its features by logging into their accounts. The web application interacts with a MySQL database, where information is stored and managed by the company's IT administrators. Data management responsibilities are assigned to the IT administrators, granting them direct access to the application's functionalities. Please refer to Figure 1 for a top level overview of the system's architecture.

## Software Development Plan & Specification

---

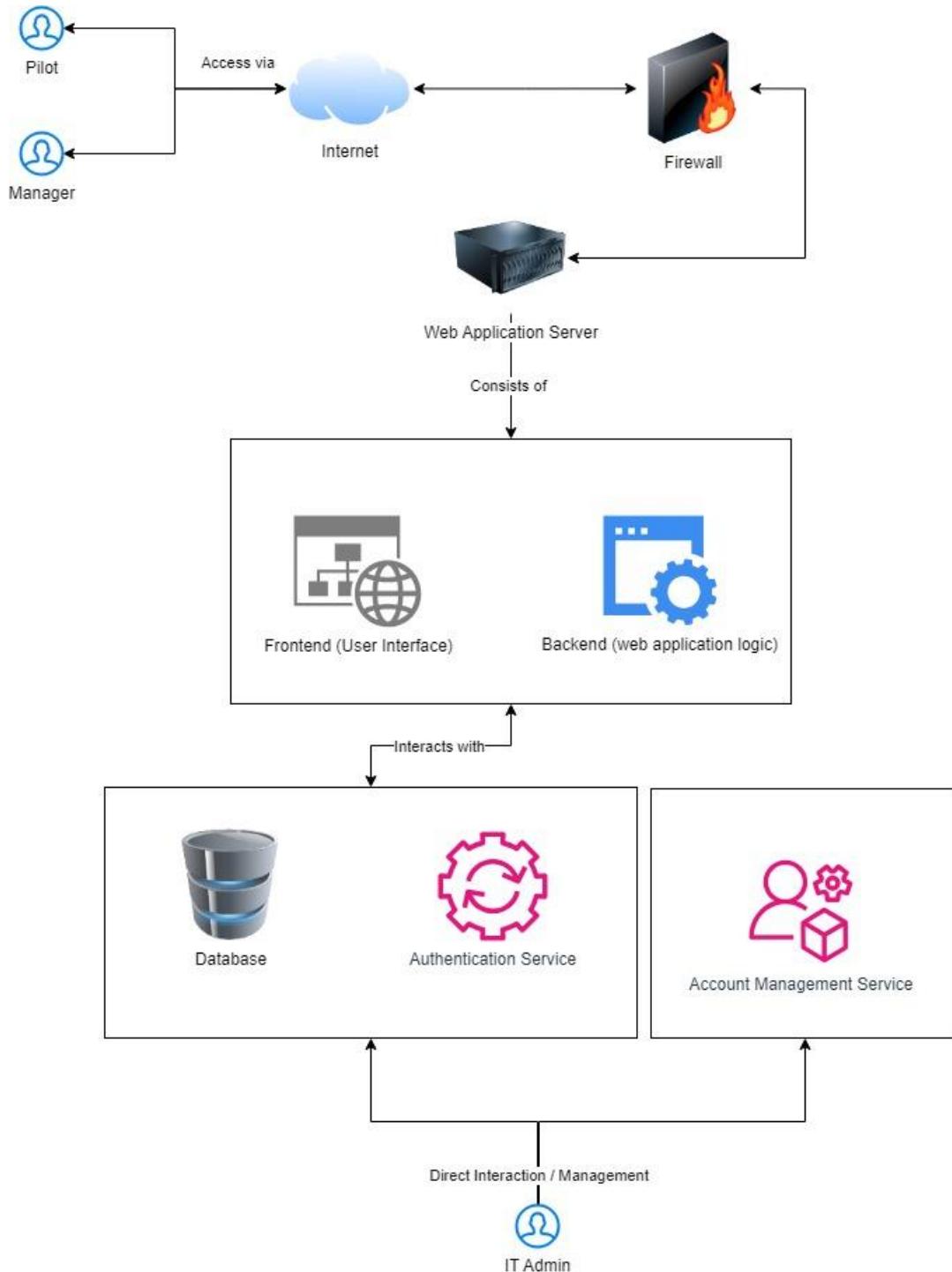


Figure 1: System Architecture Diagram

## 2.2 Product Functionality

- PF1: The system shall be able to display a dashboard as the landing page.
- PF2: The system shall allow users to search and filter for pilots.
- PF3: The system shall be able to display flight allocations.
- PF4: The system shall be able to display flight details.
- PF5: The system shall be able to display the pilot's workload.
- PF6: The system shall allow pilots to add and edit their schedules one month in advance.
- PF7: The system shall allow pilots to reject allocated jobs.
- PF8: The system shall include a reminder to indicate the date of schedule confirmation.
- PF9: The system shall indicate the warning of demerit points if the deadline is missed.
- PF10: The system shall be able to display the training details/history of each respective pilot.
- PF11: The system shall be able to allow the management to fill up the flight details for flight allocation purposes.
- PF12: The system shall be able to display recommended pilots when allocating flights.
- PF13: The system shall update the database after registering of information.
- remove the space.

Please also refer to the activity diagram mentioned in Figure 2 to visualise the product's main activity flow.

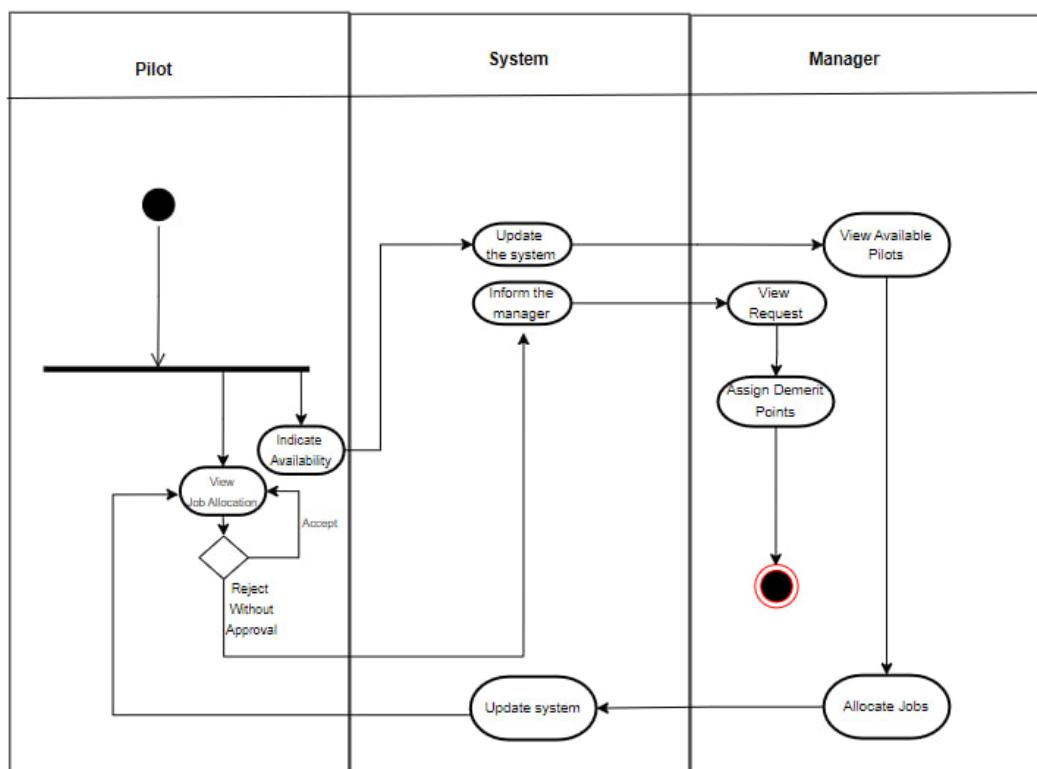


Figure 2 : Activity Diagram

## **2.3 Assumptions and Dependencies**

- The MySQL database stores data related to the web application. It continues to receive support and remains functional.
- Users have access to devices (e.g., computers, tablets, smartphones) that meet the minimum system requirements for using the web application effectively.
- The web application will be developed and rigorously tested for compatibility with widely used web browsers (e.g., Chrome, Firefox, Safari) to deliver a consistent user experience.
- Stable internet connectivity to access and use the web application without significant interruptions.
- The client's IT department possesses the necessary expertise to manage and maintain MySQL database.
- Continuous operation of the web application is contingent on the accessibility and reliability of this infrastructure.
- The client's IT department plays a pivotal role in infrastructure maintenance, including software updates, security configurations, and providing support to end-users.
- Adequate disaster recovery mechanisms, such as automated backups and redundancy, are expected to be in place with the hosting provider to mitigate potential disruptions.
- The performance and functionality of the application rely on various libraries, frameworks, and third-party components. Changes or updates to these components may impact the application if certain components become obsolete or have been updated to be used differently.
- Users possess basic internet browsing skills and can navigate web interfaces comfortably.
- The Django framework that is used for our backend code will remain actively supported and maintained by its developers.

# **3 Specific Requirements**

## **3.1 User Interface Requirements**

The user interface requirement consists of the first version of wireframe the software product. The software consists of two different sets of accounts, mainly the management level account and the pilot account. Both accounts differ in the information provided in a dashboard format as shown in Figure 5 to 5.5 for the management and Figure 11 to 11.2 for the pilot. Meanwhile, the rest of the functions such as the login page, 2FA authentication, account and password management are similar. In addition, Figure 8 and 12 shows two basic flow charts of the interaction between system and the management and pilot respectively. This wireframe version of the user interface is to provide the client with a rough idea of the appearance and functionality of the software.

Admin/ Management

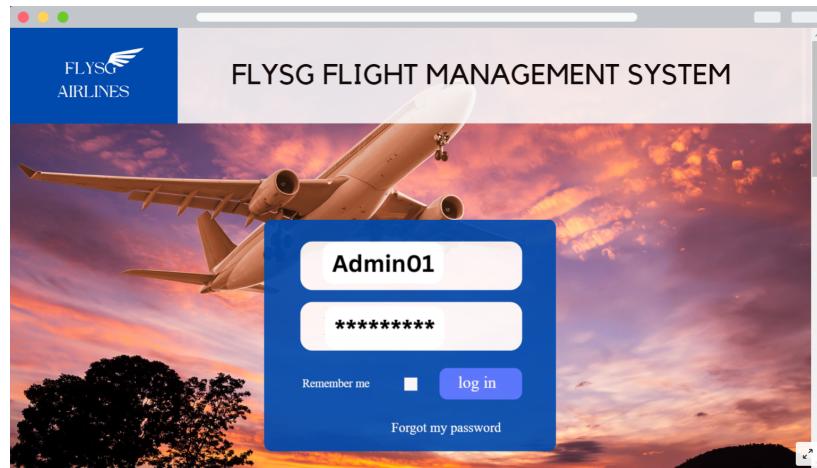


Figure 3: Admin POV login page

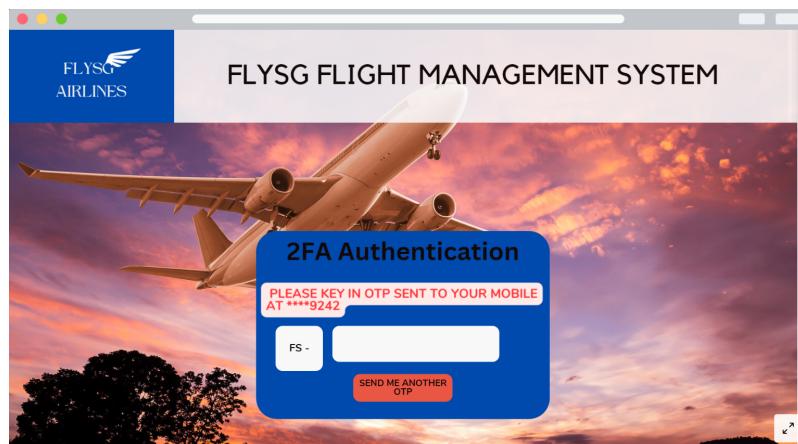


Figure 4: Admin POV 2FA Authentication

## Software Development Plan & Specification

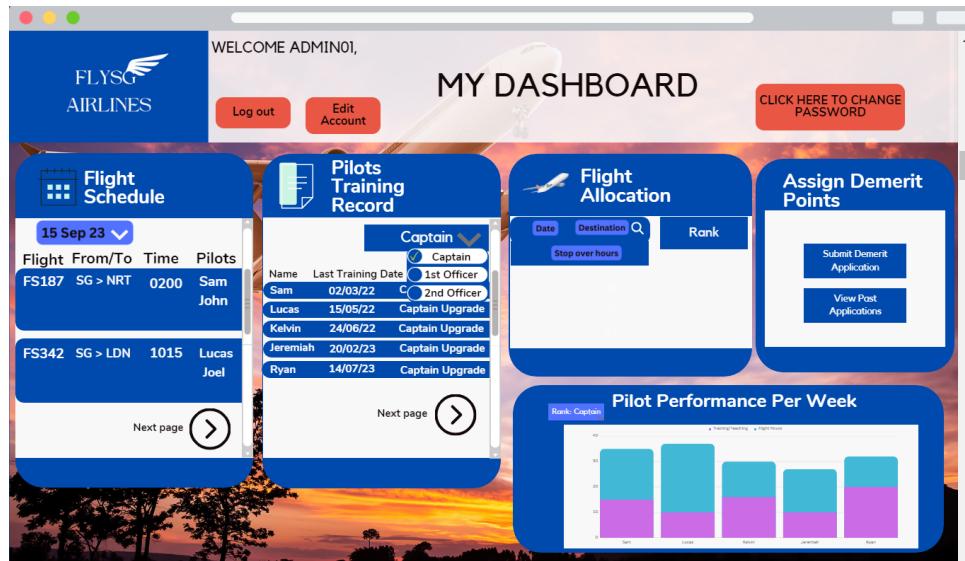


Figure 5: Admin POV Dashboard



Figure 5.1: Admin POV Dashboard

## Software Development Plan & Specification

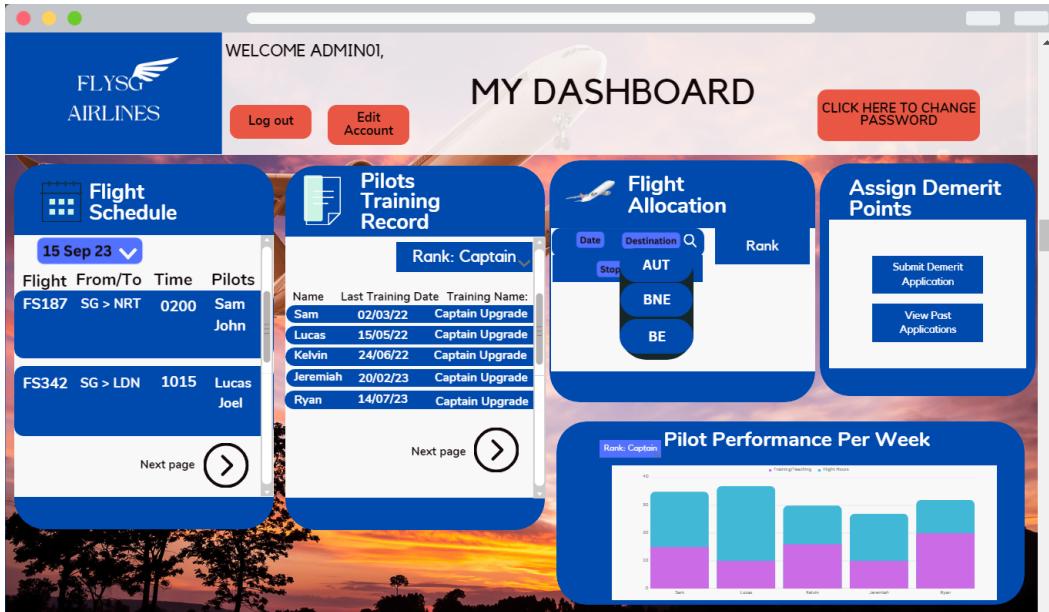


Figure 5.2: Admin POV Dashboard

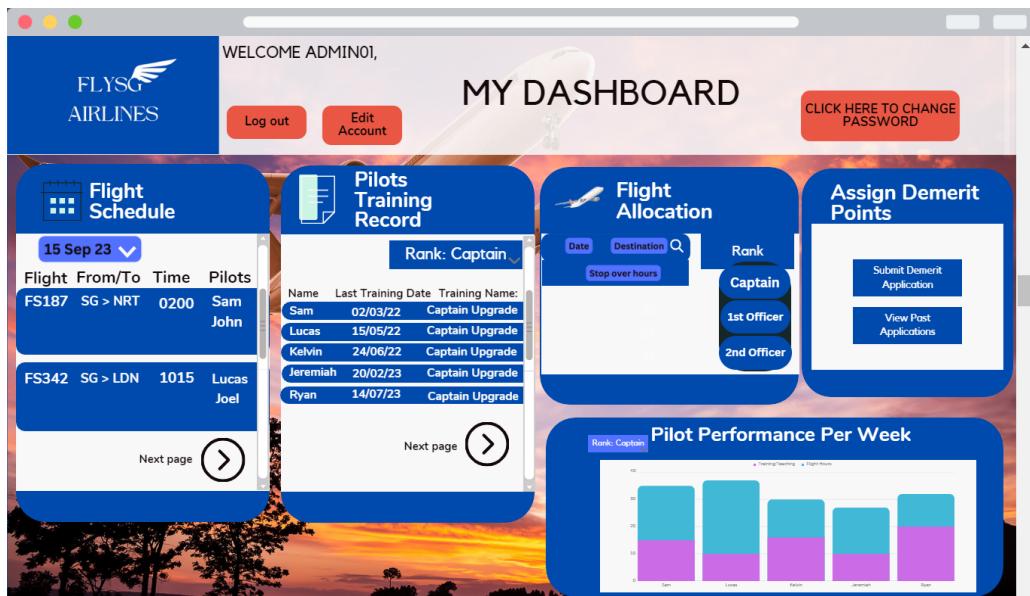


Figure 5.3: Admin POV Dashboard

## Software Development Plan & Specification

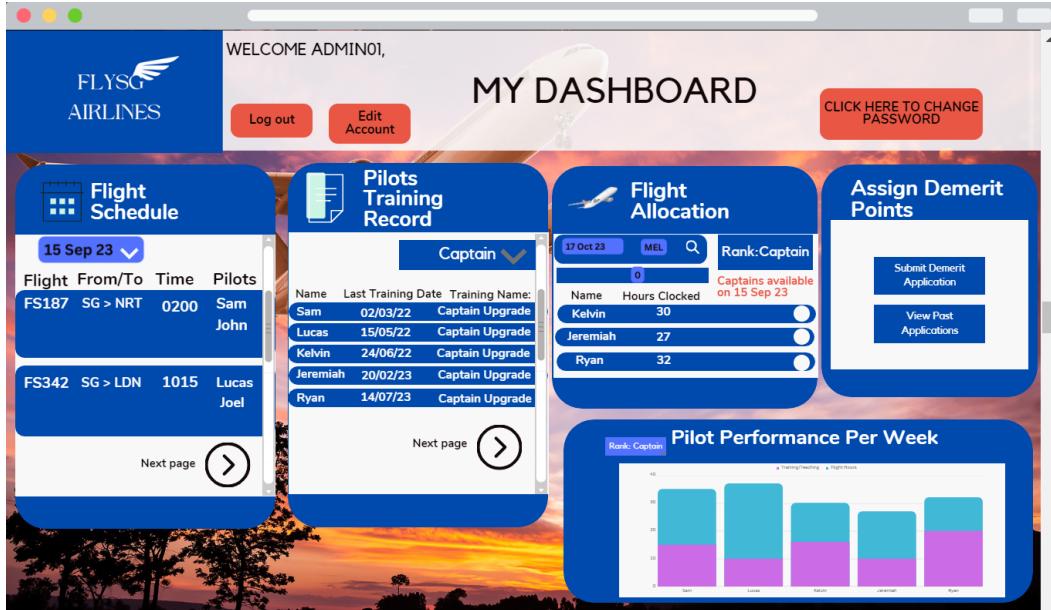


Figure 5.4: Admin POV Dashboard

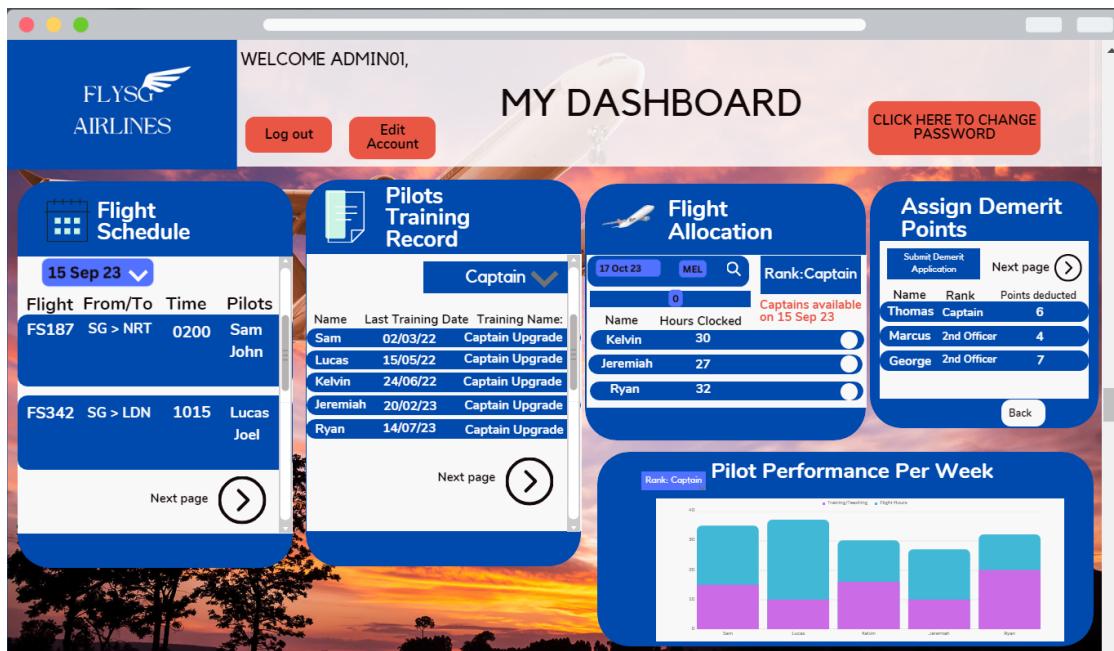


Figure 5.5: Admin POV Dashboard

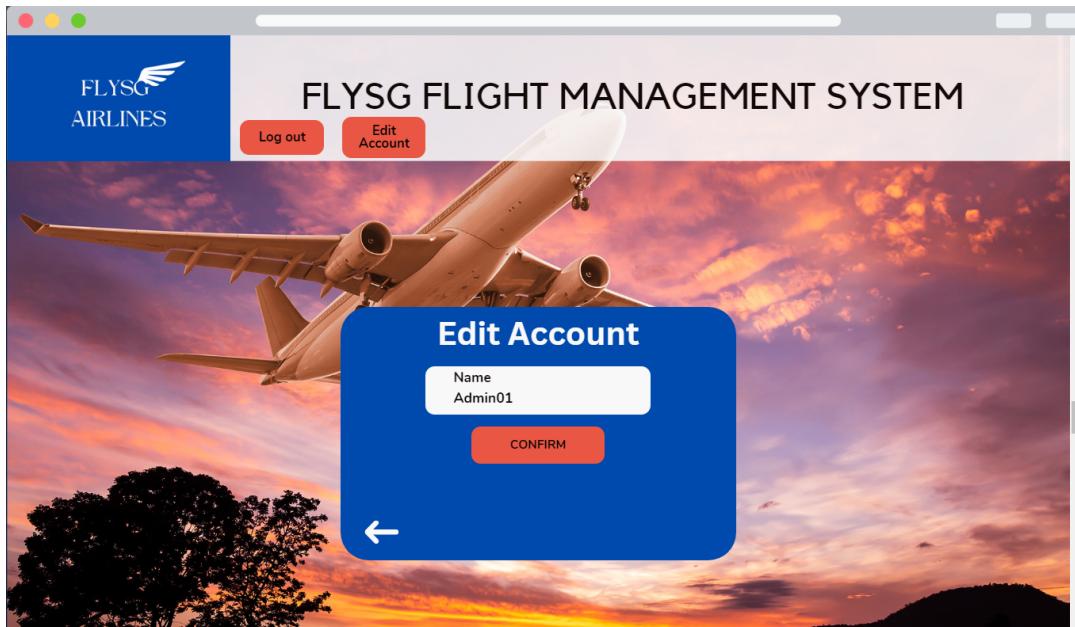


Figure 6: Admin account management page



Figure 7: Admin change of password page

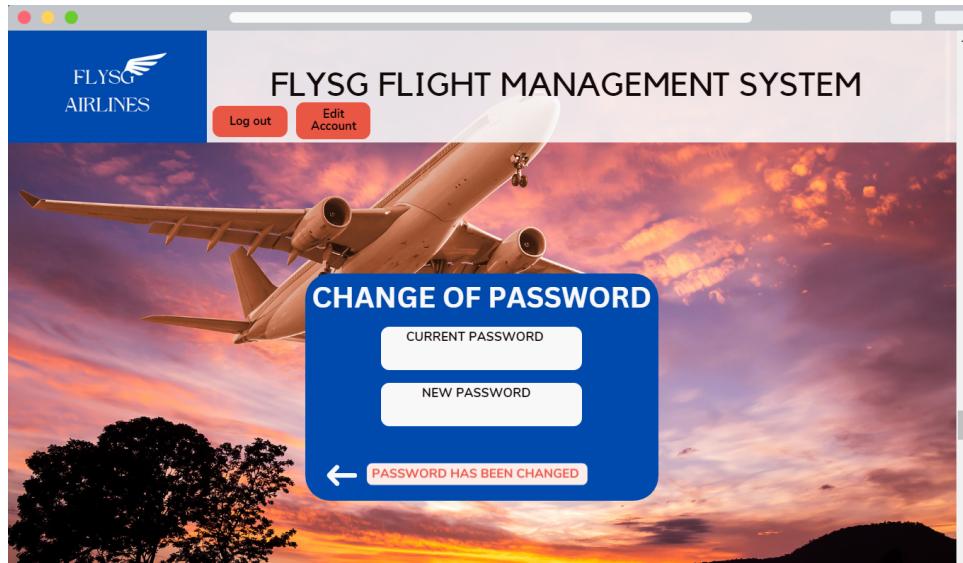


Figure 7.1: Admin change of password page

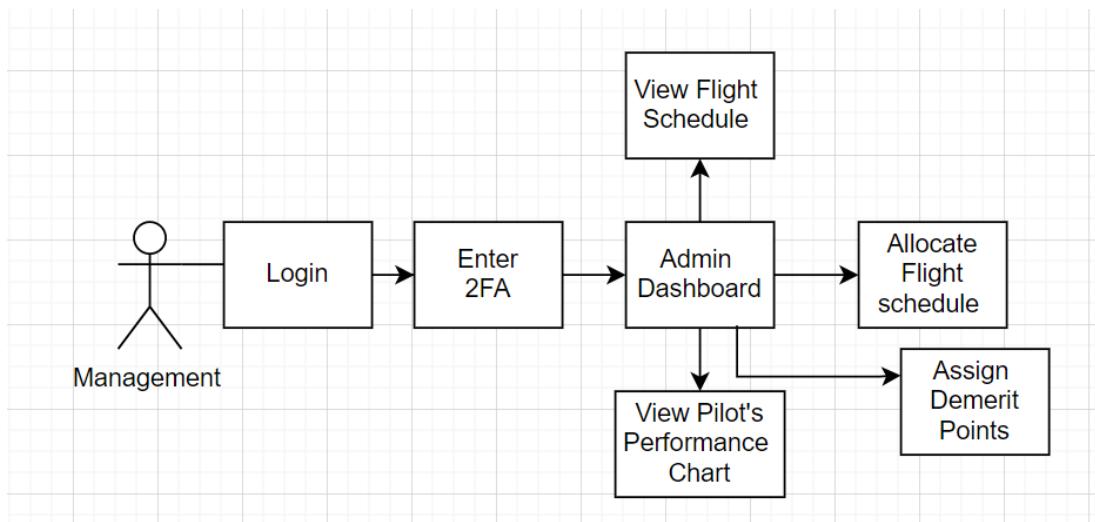


Figure 8: Flow chart of the interaction between management and system

Staff / Pilot POV:

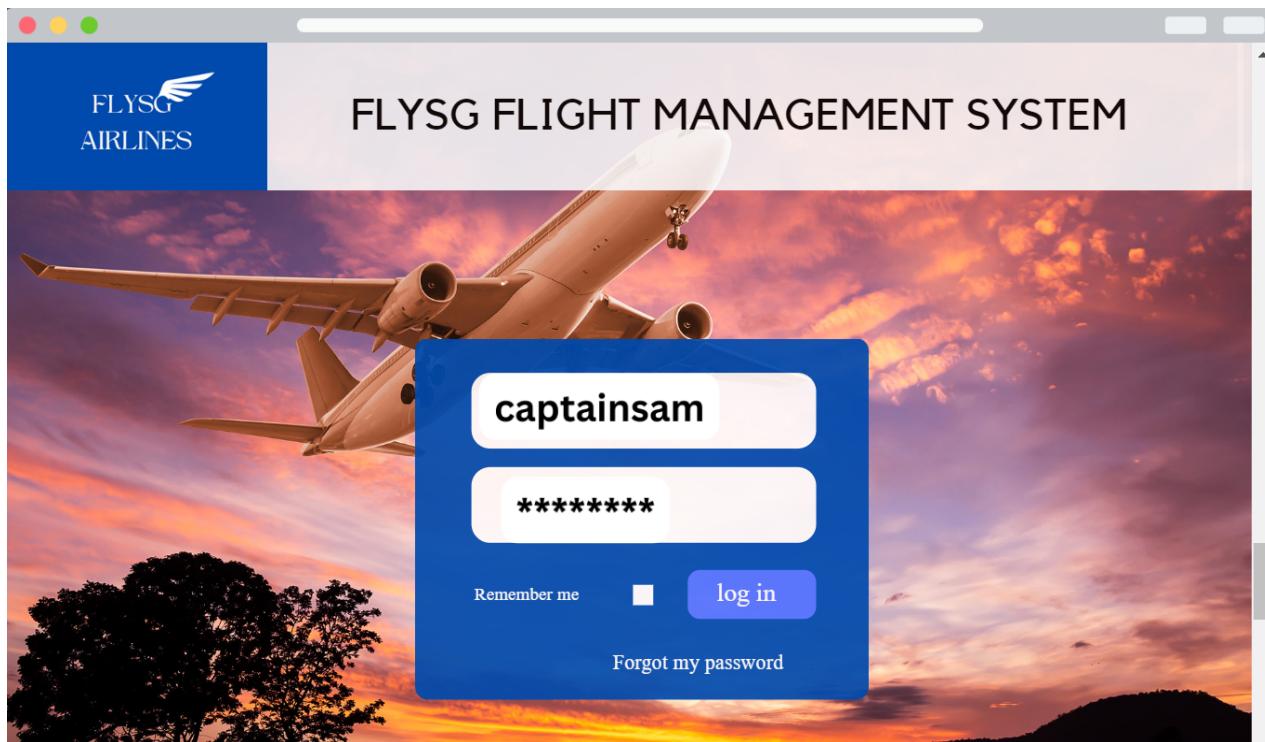


Figure 9: Pilot login page



Figure 10: Pilot 2FA authentication page

## Software Development Plan & Specification

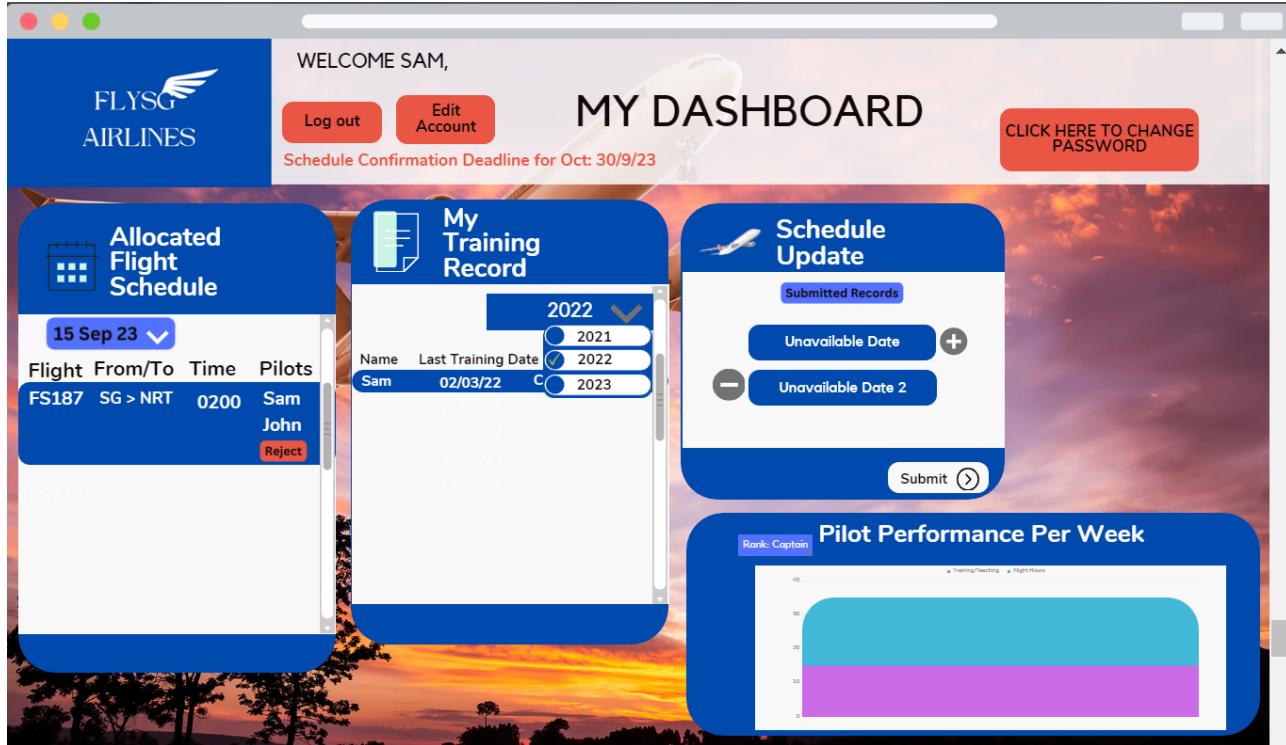


Figure 11: Pilot dashboard

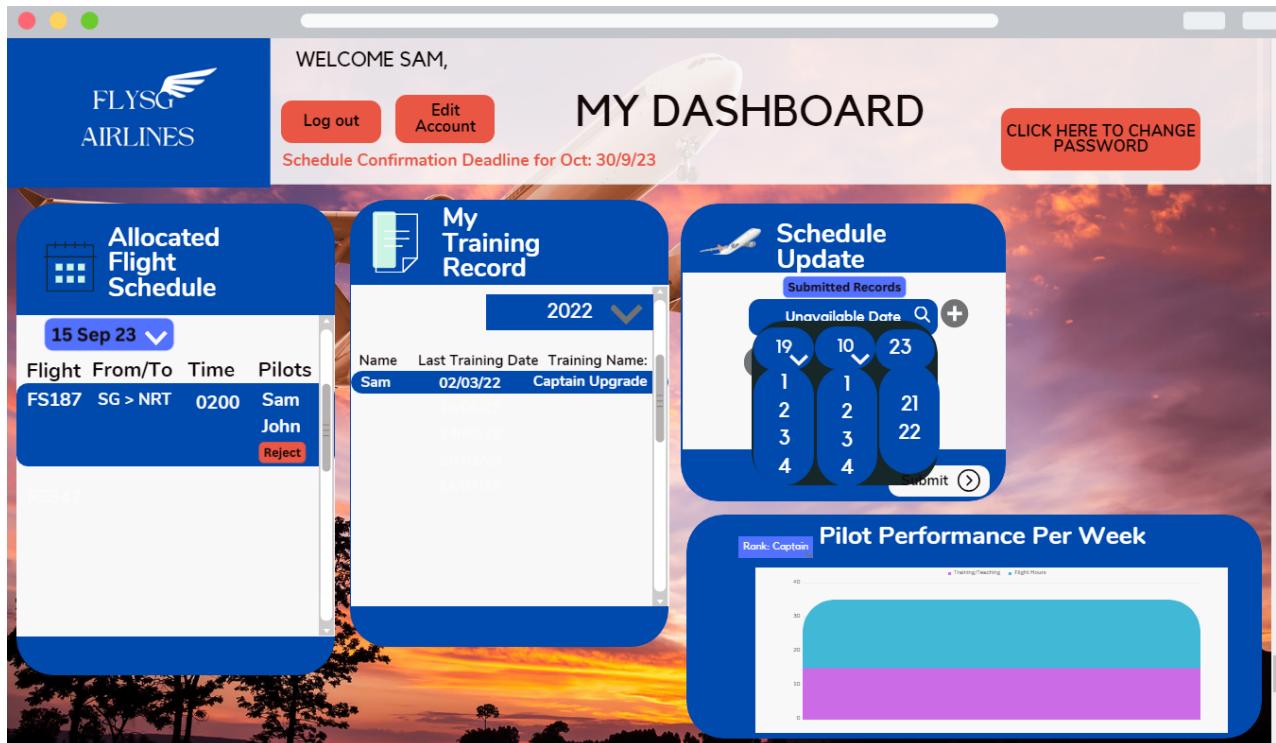


Figure 11.1: Pilot dashboard

## Software Development Plan & Specification

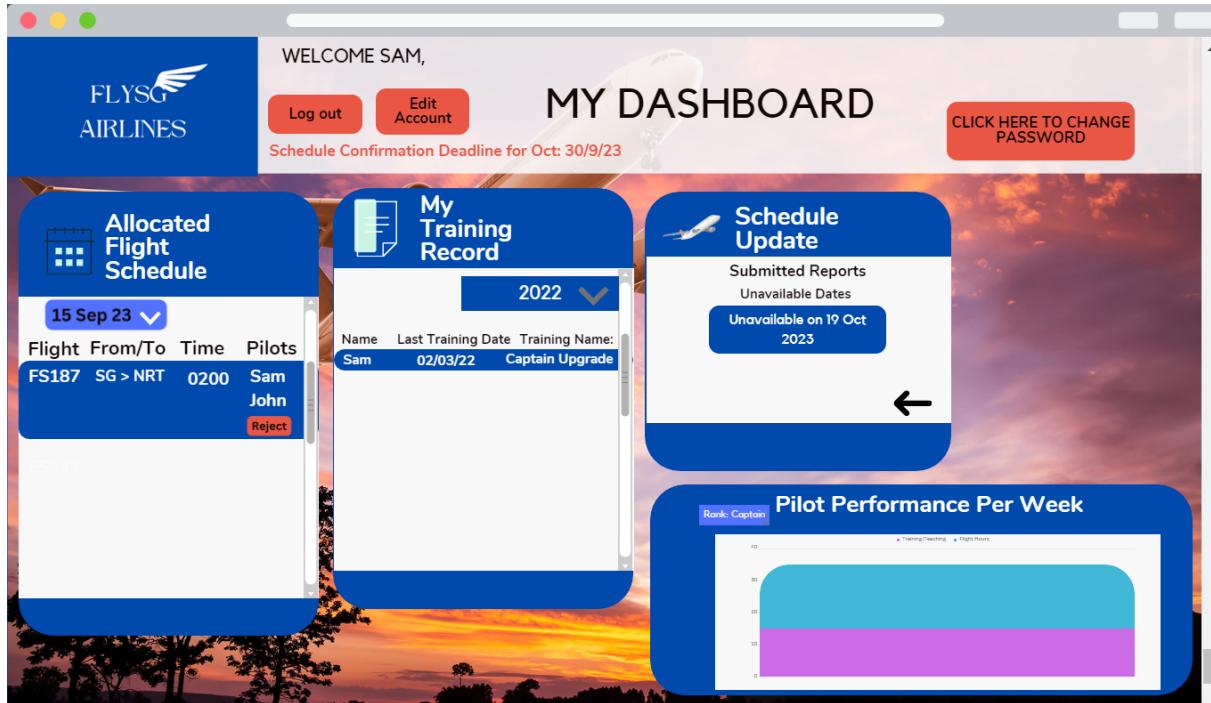


Figure 11.2: Pilot dashboard

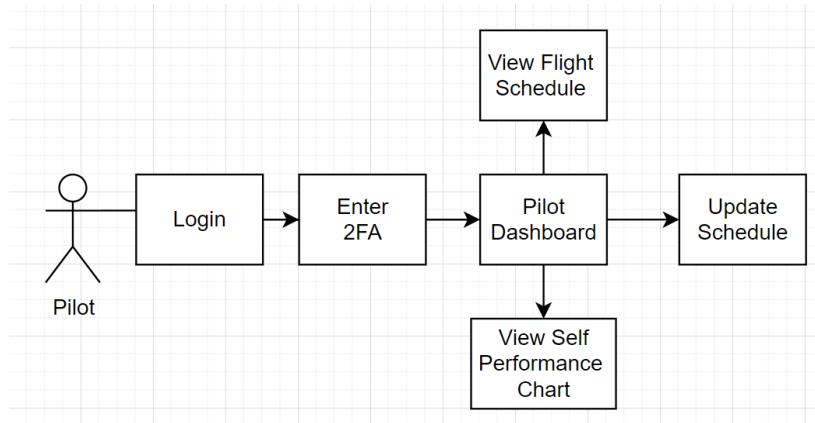


Figure 12: Flow chart of the interaction between pilot and system

## **3.2 Functional Requirements**

- FR1: The system shall uniquely identify the credentials of each user.
- FR2: The system shall be able to generate the one time password for the 2FA authentication.
- FR3: The system shall be able to send the one time password to the designated mobile number of the user.
- FR4: The system shall automatically block out dates that have passed the deadline of confirmation.
- FR5: The system shall display the deadline of schedule confirmation.
- FR6: The system shall be able to generate notifications when information has been registered.
- FR7: The system shall be able to display notifications received.
- FR8: The system shall be able to generate and send emails to relevant users when information has been registered.
- FR9: The system shall update the database when information has been registered.
- FR10: The system shall be able to display job allocations to the user.
- FR11: The system shall be able to display job details to the user.
- FR12: The system shall be able to display a dashboard to the user.
- FR13: The system shall allow the user to search and filter for other pilots.
- FR14: The system shall allow the user to edit his personal details.
- FR15: The management shall be able to view top 3 pilots with less than 40 working hours within the week on his dashboard.
- FR16: The management shall be able to view all pilots with more than 40 working hours within the week on his dashboard.
- FR17: The management shall be allowed to pick the date for flight allocation.
- FR18: The management shall be allowed to fill in flight details during the flight allocation process.
- FR19: The management shall be able to view recommended pilots during the flight allocation process.
- FR20: The management shall be able to search for pilots during the flight allocation process.
- FR21: The management shall be able to assign pilots during the flight allocation process.
- FR22: The management shall be allowed to view pilot training records.
- FR23: The management shall be allowed to assign demerit points to pilots.
- FR24: The management shall be allowed to remove rejected job allocations.
- FR25: The pilot shall be allowed to view his training records.
- FR26: The pilot shall be allowed to select dates that are one month ahead during the schedule update process.
- FR27: The pilot shall be allowed to include more than one unavailable date during the schedule update process.

## ***Software Development Plan & Specification***

---

- FR28: The pilot shall be allowed to reject any allocated flight.
- FR29: The pilot shall be allowed to view his weekly workload.
- FR30: The pilot shall be allowed to view his monthly workload.
- FR31: IT Admins shall be allowed full access to the database.
- FR32: IT Admins shall be allowed to manage the database.
- FR33: IT Admins shall be allowed to manage accounts.
- FR34: IT Admins shall be allowed to search and filter for all personnel.
- FR35: IT Admins shall be allowed to edit personal details.

### 3.3 Use Case Model

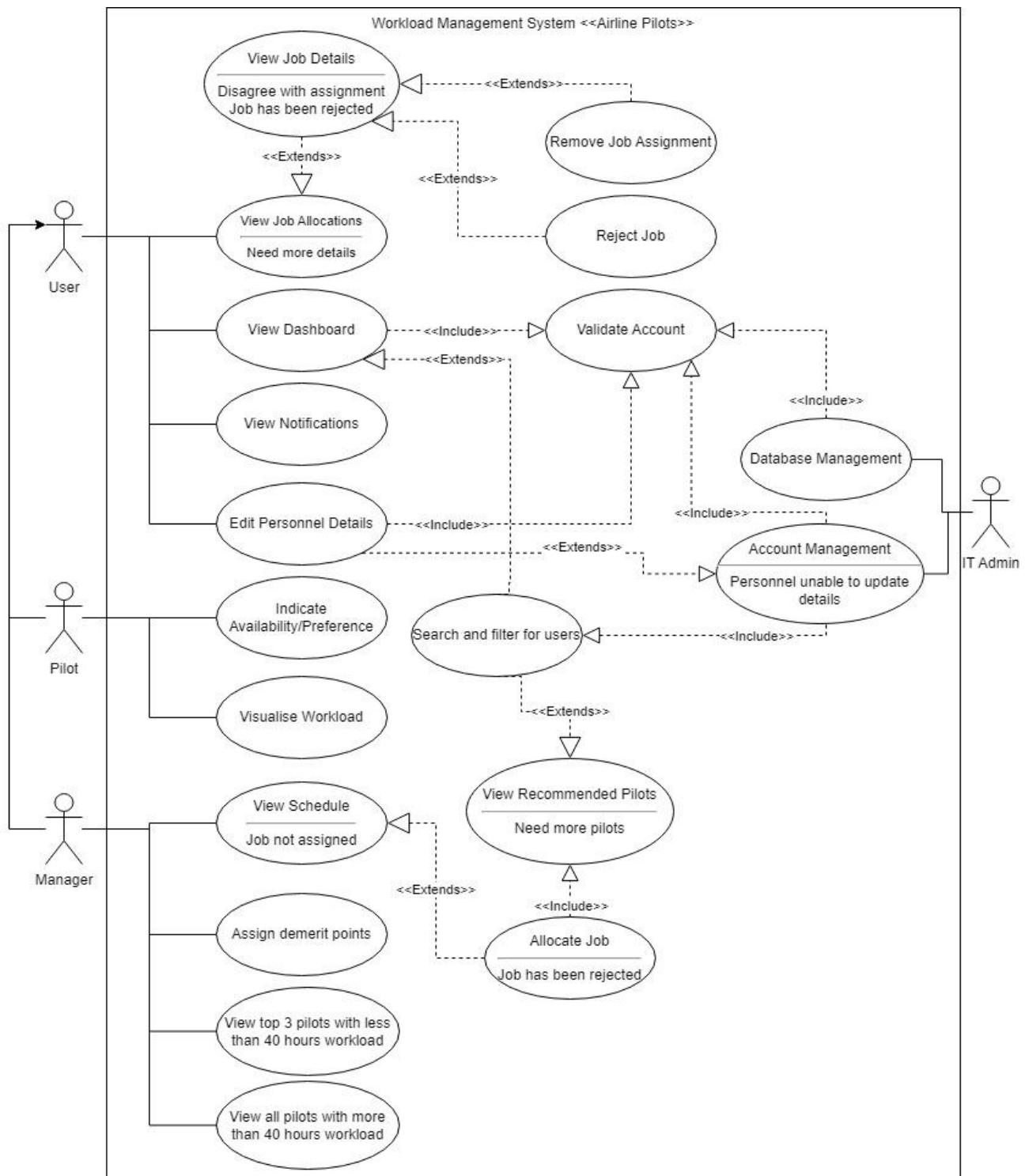


Figure 13: Use Case Diagram

## **Explanation**

Use case description can be found in **Appendix A**.

For both pilots and managers, they will have access to their dashboard after logging in using their account. A search and filtering functionality on the dashboard will be helpful for focusing on particular pilots. Both users should be able to view jobs that have been allocated, and clicking on the assignment should bring up more details of the job. For managers, if the job has been rejected, they should have the option to remove the job allocation to prevent unnecessary clutter on the dashboard. For pilots, they should have the option to reject jobs, given that they are aware of company policies and terms and conditions before doing so. Both users should be able to view notifications to avoid missing important information. Additionally, both users should be able to edit their personal details when needed.

Pilots should be able to visualize their workload on the dashboard and indicate their working availability on the schedule.

Managers should be able to assign jobs on the schedule after viewing it and confirming that a job has not been assigned. While assigning jobs, pilots will be recommended to the manager to streamline the job allocation process. If the manager prefers to assign a pilot who is not recommended, they will be able to use the search functionality to find a specific pilot for the job. When allocating the job, the manager has to input the necessary flight details as well. On the dashboard, the manager should be able to see highlighted pilots who have less than 40 working hours and those who have more than 40 working hours for the week. Finally, managers have the authority to assign demerit points based on feedback received about a pilot.

## **3.4 Non-functional Requirements**

- NFR1: The user shall authenticate themselves with the one time password sent to their device.
- NFR2: The user shall not reveal their credentials under any circumstances.
- NFR3: The pilot shall be responsible for meeting the deadline requirement of indicating their availability.
- NFR4: The pilot shall be responsible for providing a valid reason to reject a job allocation.
- NFR5: The pilot shall be responsible for reporting to work on time.
- NFR6: The pilot shall be responsible for updating their information when necessary.
- NFR7: The manager shall be responsible for allocating flights to pilots on time.
- NFR8: The manager shall be responsible for reminding pilots to indicate their availability on time.
- NFR9: The manager shall be responsible for reminding the pilots to update their information when necessary.
- NFR10: The system shall be available to users at all times.
- NFR11: The system and its functionalities shall work seamlessly.
- NFR12: The system shall comply with necessary privacy protection of data which includes but is not limited to the Personal Data Protection Act (PDPA).

- NFR13: The system shall comply with best practices as best they can when dealing with matters of security.
- NFR14: The system shall comply with the airline regulations where necessary.
- NFR15: The system shall have no limits to the number of accounts.
- NFR16: The system shall only have accounts for employees who use the system.
- NFR17: The system shall be in the English language.
- NFR18: The system shall be accessible on both the desktop and mobile browsers, ensuring a consistent experience across all devices.
- NFR19: The system shall provide responsive web pages to ensure optimal user experience on various devices.
- NFR20: The system's default timezone shall be in Singapore Time (SGT), providing synchronized timing for all users.
- NFR21: The system shall provide a visually aesthetic website to provide optimal user experience.
- NFR22: The system shall provide timezone conversion where necessary.
- NFR23: The system will clearly state which timezone is being used when time is told.
- NFR24: The system shall be maintained by the IT department.

### **3.4.1 Performance Requirements**

- NFR25: Following ICA's guidelines, profile pictures should be of passport photo size. The dimension is 400x514 and should not be more than 2MB. Under circumstances where management is required to send an employee's picture to relevant authorities, it can be easily extracted from the database. Having a size limitation prevents the database from running out of space too fast.
- NFR26: With 200 simultaneous users or more, the system should maintain a page load time of under 5 seconds for commonly accessed pages. We expect that the system will only be used when scheduling is involved and do not anticipate more than 200 simultaneous users at the same time.
- NFR27: Real-time notifications (e.g., alerts, messages) should be delivered to users within 2 seconds upon setting off an event. This is also subject to the connectivity to the internet from where the user is using. Ensuring notifications arrive promptly ensures that all users are made aware of informational updates.
- NFR28: Under normal load conditions, database queries should be executed within 3 seconds. This ensures data is reflected quickly so as to provide optimal user experience.
- NFR29: The system shall load web pages and perform actions within 5 seconds to ensure a responsive experience.
- NFR30: User sessions shall remain active for a minimum of 30 minutes before prompting the user. If the user does not respond within 10 seconds of the prompt appearing, he will be

forced out of the session. This ensures that unauthorised personnel do not have access to the system while using the original user's device.

- NFR31: Load balancing mechanisms will evenly distribute traffic among multiple servers to ensure optimal performance and prevent overloads.
- NFR32: Error messages shall be displayed to users within 3 seconds of encountering errors in order to help users understand the error encountered.
- NFR33: Frequently accessed data shall be cached to improve overall system performance. Data that is cached will be subject to the Personal Data Protection Act (PDPA).

### **3.4.2 Safety and Security Requirements**

- NFR34: HTTPS is used when serving the web pages. This encrypts data sent to and from the web server.
- NFR35: All accounts are secured with 2 Factor-Authentication(2FA). Dual security measure to ensure only the authorised personnel has access to the system.
- NFR36: During login, all users must enter a One-Time Password (OTP) sent to their registered mobile phone to enhance authentication security.
- NFR37: The account will be suspended after 10 unsuccessful logins. This prevents malicious actors from brute-forcing their way into the system. Only admins will have the permissions to reverse its suspended state.
- NFR38: Data related to passwords, keys, etc stored on the system will be encrypted with the SHA-256 hashing algorithm. This enhances data security and confidentiality.
- NFR39: Passwords must meet a specific complexity criteria to increase security. This includes a minimum password length of 10 characters, at least one uppercase letter, one digit, and one special character.
- NFR40: The system's firewall must be active and operational at all times. This ensures continuous protection against known cyber threats.
- NFR41: If the primary firewall becomes unavailable due to various reasons, a backup firewall will be activated automatically.
- NFR42: Users are required to change their passwords every 3 months. This ensures the account is continuously secured with new passwords.
- NFR43: Session timeout mechanisms will be implemented, automatically logging out users after 30 minutes of inactivity if the user does not respond to a timeout prompt from the system.
- NFR44: Logging of data shall be implemented in order to track and monitor user activity, providing a record for security analysis and compliance.
- NFR45: Access controls shall be enforced to ensure that users can only access functionalities and data relevant to their roles and responsibilities.
- NFR46: A load-balancing mechanism will be activated which redistributes traffic amongst servers to maintain as much availability as possible in the event of DDOS attacks.
- NFR47: All data will be encrypted to ensure secure data transmission.

### **3.5 Other Requirements**

- OR1: The data retention will be set at 2 years. The time given here is what we recommend to be the minimum amount and should change according to the company's policy.
- OR2: The data will conform to the requirements of the PDPA.
- OR3: In the case of a data breach, the PDPC and affected persons would be notified according to the PDPA.
- OR4: Adhere to all relevant legal and regulatory requirements.
- OR5: Maintain logs and reporting capabilities to track user activity, system events, and security-related incidents. This ensures system compliance and accountability purposes.
- OR6: Users will have clearly defined permissions to data they contribute.
- OR7: Conform to the Web Content Accessibility Guidelines (WCAG) 2.1 AA standards to ensure accessibility for users with disabilities. This guideline is more focused on the managers as pilots are expected to be physically fit without disabilities to perform their duties.
- OR8: Develop user training resources to educate them on security best practices, data protection, and responsible system usage.
- OR9: Assign a Data Protection Officer (DPO) from the client's side to manage data security policies as mentioned in the PDPA.
- OR10: Establish a user manual on proper system usage. Documentations on system and architecture will ensure that information can be found when necessary.
- OR11: Create and implement a comprehensive disaster recovery plan to ensure system availability and data recovery in unforeseen disaster scenarios or system failures.
- OR12: Specify third-party software or libraries used in the project, including licensing agreements, version control, and compliance with open-source software guidelines.
- OR13: Implement continuous system monitoring.
- OR14: Implement performance optimization practices to ensure the system's reliability, scalability, and optimal performance are maintained or even improved over time.
- OR15: Data will be securely removed when it is not required or when the retention date is reached.
- OR16: Store and collect data in accordance with policies and already obtained consent.

## 4 Project Estimation and Plan

### 4.1 Software Estimation

Unadjusted Use Case Weight (UUCW)	$  \begin{aligned}  & (\text{Total number of simple use cases} * 5) + (\text{Total number of average use cases} * 10) + (\text{Total number of complex use cases} * 15) \\  & = (9 * 5) + (9 * 10) + (1 * 15) \\  & = 45 + 90 + 15 \\  & = 150  \end{aligned}  $																																																											
Unadjusted Actors' Weight (UAW)	$  \begin{aligned}  & (\text{Total number of simple actors} * 1) + (\text{Total number of average actors} * 2) + (\text{Total number of complex actors} * 3) \\  & = (0 * 1) + (0 * 2) + (4 * 3) \\  & = 12  \end{aligned}  $																																																											
Technical Factors (TF)	<table border="1"> <thead> <tr> <th>Factor</th><th>Description</th><th>Weight</th><th>Assessment</th><th>Product</th></tr> </thead> <tbody> <tr> <td>T1</td><td>Distributed system</td><td>2</td><td>1</td><td>2</td></tr> <tr> <td>T2</td><td>Response time/performance objectives</td><td>1</td><td>3</td><td>3</td></tr> <tr> <td>T3</td><td>End-user efficiency</td><td>1</td><td>3</td><td>3</td></tr> <tr> <td>T4</td><td>Internal processing complexity</td><td>1</td><td>2</td><td>2</td></tr> <tr> <td>T5</td><td>Code reusability</td><td>1</td><td>4</td><td>4</td></tr> <tr> <td>T6</td><td>Easy to install</td><td>0.5</td><td>3</td><td>1.5</td></tr> <tr> <td>T7</td><td>Easy to use</td><td>0.5</td><td>4</td><td>2</td></tr> <tr> <td>T8</td><td>Portability to other platforms</td><td>2</td><td>3</td><td>6</td></tr> <tr> <td>T9</td><td>System maintenance</td><td>1</td><td>4</td><td>4</td></tr> <tr> <td>T10</td><td>Concurrent/parallel</td><td>1</td><td>3</td><td>3</td></tr> </tbody> </table>					Factor	Description	Weight	Assessment	Product	T1	Distributed system	2	1	2	T2	Response time/performance objectives	1	3	3	T3	End-user efficiency	1	3	3	T4	Internal processing complexity	1	2	2	T5	Code reusability	1	4	4	T6	Easy to install	0.5	3	1.5	T7	Easy to use	0.5	4	2	T8	Portability to other platforms	2	3	6	T9	System maintenance	1	4	4	T10	Concurrent/parallel	1	3	3
Factor	Description	Weight	Assessment	Product																																																								
T1	Distributed system	2	1	2																																																								
T2	Response time/performance objectives	1	3	3																																																								
T3	End-user efficiency	1	3	3																																																								
T4	Internal processing complexity	1	2	2																																																								
T5	Code reusability	1	4	4																																																								
T6	Easy to install	0.5	3	1.5																																																								
T7	Easy to use	0.5	4	2																																																								
T8	Portability to other platforms	2	3	6																																																								
T9	System maintenance	1	4	4																																																								
T10	Concurrent/parallel	1	3	3																																																								

**Software Development Plan & Specification**

	processing																																											
T11	Security features	1	4	4																																								
T12	Access for third parties	1	1	1																																								
T13	End user training	1	3	3																																								
Total				38																																								
Technical Complexity Factor (TCF)	$0.6 + (0.01 * \text{TF}) = 0.6 + (0.01 * 38) = 0.98$																																											
Environmental Factors (EF)	<table border="1"> <thead> <tr> <th>Environment</th><th>Weight</th><th>Assessment</th><th>Product</th></tr> </thead> <tbody> <tr> <td>Familiar with development process</td><td>1.5</td><td>3</td><td>4.5</td></tr> <tr> <td>Part time workers</td><td>-1</td><td>0</td><td>0</td></tr> <tr> <td>Analyst capability</td><td>0.5</td><td>3</td><td>1.5</td></tr> <tr> <td>Application experience</td><td>0.5</td><td>3</td><td>1.5</td></tr> <tr> <td>Object oriented experience</td><td>1</td><td>3</td><td>3</td></tr> <tr> <td>Motivation</td><td>1</td><td>4</td><td>4</td></tr> <tr> <td>Difficult programming language</td><td>-1</td><td>3</td><td>-3</td></tr> <tr> <td>Stable requirements</td><td>2</td><td>4</td><td>8</td></tr> <tr> <td>Total</td><td></td><td></td><td>19.5</td></tr> </tbody> </table>				Environment	Weight	Assessment	Product	Familiar with development process	1.5	3	4.5	Part time workers	-1	0	0	Analyst capability	0.5	3	1.5	Application experience	0.5	3	1.5	Object oriented experience	1	3	3	Motivation	1	4	4	Difficult programming language	-1	3	-3	Stable requirements	2	4	8	Total			19.5
Environment	Weight	Assessment	Product																																									
Familiar with development process	1.5	3	4.5																																									
Part time workers	-1	0	0																																									
Analyst capability	0.5	3	1.5																																									
Application experience	0.5	3	1.5																																									
Object oriented experience	1	3	3																																									
Motivation	1	4	4																																									
Difficult programming language	-1	3	-3																																									
Stable requirements	2	4	8																																									
Total			19.5																																									
Environmental Complexity Factor (ECF)	$1.4 + (-0.03 * \text{EF}) = 1.4 + (-0.03 * 19.5) = 0.815$																																											
Use Case Points (UCP)	$(\text{UUCW} + \text{UAW}) * \text{TCF} * \text{EF} = (150 + 12) * 0.98 * 0.815 \approx 129.4$																																											

Estimated Effort in	Estimate of hours for each UCP = 15 to 30 hours for each UCP
---------------------	--

Developer-Hours	<table border="1"> <thead> <tr> <th>Minimum Effort</th><th>Maximum Effort</th><th>Average Effort</th></tr> </thead> <tbody> <tr> <td> <math>OD = UCP * 15 \text{ hours}</math>  <math>= 129.4 * 15 = 1,941</math> </td><td> <math>PD = UCP * 30 \text{ hours}</math>  <math>= 129.4 * 30 = 3,882</math> </td><td> <math>ED = (\text{Min} + \text{Max}) / 2</math>  <math>= (1,941 + 3,882) / 2</math>  <math>= 2,911.5</math> </td></tr> </tbody> </table>			Minimum Effort	Maximum Effort	Average Effort	$OD = UCP * 15 \text{ hours}$ $= 129.4 * 15 = 1,941$	$PD = UCP * 30 \text{ hours}$ $= 129.4 * 30 = 3,882$	$ED = (\text{Min} + \text{Max}) / 2$ $= (1,941 + 3,882) / 2$ $= 2,911.5$
Minimum Effort	Maximum Effort	Average Effort							
$OD = UCP * 15 \text{ hours}$ $= 129.4 * 15 = 1,941$	$PD = UCP * 30 \text{ hours}$ $= 129.4 * 30 = 3,882$	$ED = (\text{Min} + \text{Max}) / 2$ $= (1,941 + 3,882) / 2$ $= 2,911.5$							
<p>There are 4 developers working on this project. To calculate the developer-hours:</p> <p>Minimum = 1,941 hours / 4 developers = 485.25 hours per developer      Maximum = 3,882 hours / 4 developers = 970.5 hours per developer      Average = 2,911.5 hours / 4 developers ≈ 727.9 hours per developer</p>									
Estimated Duration	<p>In the upcoming project phase, we are looking at a timeline of 9 weeks starting from week 5 onwards.</p> <p>Minimum = 485.25 developer-hours / 9 weeks ≈ 53.9 developer-hours per week      Maximum = 970.5 developer-hours / 9 weeks ≈ 107.8 developer-hours per week      Average = 727.9 developer-hours / 9 weeks ≈ 80.9 developer-hours per week</p>								

## 4.2 Project Management

Our team, consisting of four members, faces the challenge of completing a project within a tight 12-week timeframe. It's imperative that we select the most appropriate Software Development Life Cycle (SDLC) methodology for our project. After careful consideration, we have decided to adopt the Waterfall Methodology. Several key factors influenced our decision, including project estimation, streamlined documentation, and the need to work within a limited timeframe. The main factor that contributed to our decision was due to how the project has been outlined for us as a project whereby we are to gather all required information first before moving on to the next phase. This process eliminated the use of Agile Methodology which required continuous interaction with the client and prototyping of the product. This meant that the product may be introduced with new changes every sprint which could result in a bigger workload every week. With the Waterfall Methodology, the workload for a team consisting of 4 members will be more manageable and consistent as we will be able to plan out the work in advance and subsequently deliver the project on a strict timeline.

## 4.3 Gantt Chart

Id	Level	Name	Start	Due
1	1	Planning Phase:	06/09/2023	19/09/2023
1.1	2	Requirements Gathering	06/09/2023	06/09/2023
1.1.1	3	Understanding User Requirements	06/09/2023	06/09/2023
1.1.2	3	Determine Functional Requirements	06/09/2023	06/09/2023
1.1.3	3	Determine Non-Functional Requirements	06/09/2023	06/09/2023
1.2	2	Requirements Engineering	07/09/2023	10/09/2023
1.2.1	3	Requirements Elicitation	07/09/2023	08/09/2023
1.2.1.1	4	Client Meeting	07/09/2023	07/09/2023
1.2.1.2	4	Prototype Testing	07/09/2023	07/09/2023
1.2.1.3	4	Observation	07/09/2023	08/09/2023
1.2.2	3	Requirements Analysis	09/09/2023	10/09/2023
1.2.2.1	4	Software Architecture Development	09/09/2023	10/09/2023
1.2.2.2	4	Requirement Validation	10/09/2023	10/09/2023
1.3	2	Use Case Analysis	11/09/2023	11/09/2023
1.3.1	3	Define each Use Case Diagram	11/09/2023	11/09/2023
1.3.2	3	Creation of Use Case Diagram	11/09/2023	11/09/2023
1.3.3	3	Calculation of Use Case Points	11/09/2023	11/09/2023
1.3.4	3	Estimation of Project Size using Use Case Points	11/09/2023	11/09/2023
1.4	2	Project Scheduling and Tracking	12/09/2023	12/09/2023
1.4.1	3	Task Assignment	12/09/2023	12/09/2023
1.4.2	3	Creation of Gantt Chart	12/09/2023	12/09/2023
1.4.3	3	Creation of Kanban Board	12/09/2023	12/09/2023
1.5	2	Technical Specifications	13/09/2023	13/09/2023

Figure 14: Gantt Chart(1)

## **Software Development Plan & Specification**

1.5.1	3	Django, Chart.js Familiarisation	13/09/2023	13/09/2023
1.5.2	3	Hosting Cloud Infrastructure and Database Creation	13/09/2023	13/09/2023
1.5.3	3	Obtain TLS certificate lease for Encryption	13/09/2023	13/09/2023
1.6	2	Web User Interfaces	14/09/2023	19/09/2023
1.6.1	3	Creation of Prototypes for each Use Case	14/09/2023	18/09/2023
1.6.2	3	Client to Review and Sign Off on each Prototype	19/09/2023	19/09/2023
2	1	Analysis and Design Phase:	20/09/2023	27/10/2023
2.1	2	Object-Oriented Analysis	20/09/2023	27/10/2023
2.1.1	3	Identifying Classes and Objects	20/09/2023	27/10/2023
2.1.1.1	4	Entity-Relationship Model	20/09/2023	20/09/2023
2.1.1.2	4	Categorizing Concepts into Real-World Objects	20/09/2023	20/09/2023
2.1.1.3	4	Noun Extraction	21/09/2023	21/09/2023
2.1.1.3.1	5	Noun Identification	21/09/2023	21/09/2023
2.1.1.3.2	5	Removing Duplicates and Converting to Singular	21/09/2023	21/09/2023
2.1.1.3.3	5	Identify Candidate Classes	21/09/2023	21/09/2023
2.1.1.3.4	5	Identify Sub Classes	21/09/2023	21/09/2023
2.1.1.4	4	Creation of Class Diagrams	22/09/2023	22/09/2023
2.1.1.4.1	5	Iterations of Different Diagrams	22/09/2023	22/09/2023
2.1.1.5	4	Modelling Requirements	23/09/2023	23/09/2023
2.1.1.5.1	5	Analytical Model	23/09/2023	23/09/2023
2.1.1.5.2	5	Constructive Model	23/09/2023	23/09/2023
2.1.1.5.3	5	UML Diagram	23/09/2023	23/09/2023
2.1.1.6	4	Structured System Analysis	27/10/2023	27/10/2023
2.2	2	Software Project Development Plan	24/09/2023	24/09/2023
2.2.1	3	Cost Estimation	24/09/2023	24/09/2023
2.2.2	3	Duration Estimation	24/09/2023	24/09/2023
2.2.3	3	Coming up with Deliverables	24/09/2023	24/09/2023
2.2.4	3	Setting up Milestones	24/09/2023	24/09/2023
2.2.5	3	Budgetting	24/09/2023	24/09/2023
3	1	Development Phase:	25/09/2023	28/10/2023
3.1	2	Code Development	25/09/2023	17/10/2023
3.1.1	3	Code and Design Web Pages	25/09/2023	02/10/2023
3.1.2	3	Backend Code	03/10/2023	10/10/2023
3.1.3	3	Reuse of Code for OOP	11/10/2023	11/10/2023
3.1.4	3	Documentation of Code	13/10/2023	16/10/2023
3.1.5	3	Validate Web Pages according to Requirements and UML Diagram	16/10/2023	17/10/2023
3.2	2	Database Development	17/10/2023	18/10/2023
3.2.1	3	Setting up Test Database	17/10/2023	18/10/2023
3.2.2	3	Identification of Tables and Schemas	17/10/2023	18/10/2023
3.2.3	3	Identification of Keys Needed (Primary/Unique/Foreign)	17/10/2023	18/10/2023
3.3	2	Unit Testing	19/10/2023	28/10/2023
3.3.1	3	Unit Test Planning	19/10/2023	22/10/2023
3.3.2	3	Unit Test Case Preparation	23/10/2023	24/10/2023
3.3.3	3	Conducting of Unit Tests	25/10/2023	28/10/2023
3.3.4	3	Unit Test Recording	25/10/2023	28/10/2023

Figure 14.1: Gantt Chart(2)

### ***Software Development Plan & Specification***

<b>4</b>	<b>1</b>	<b>Deployment Phase:</b>	<b>29/10/2023</b>	<b>13/11/2023</b>
4.1	2	Launch of Hosting Site	29/10/2023	13/11/2023
4.1.1	3	Test Configuration of Servers, Domains and Database	29/10/2023	05/11/2023
4.1.2	3	Release of Live Site	06/11/2023	07/11/2023
4.1.3	3	Monitor Site for Anomalies	08/11/2023	13/11/2023
<b>5</b>	<b>1</b>	<b>Handover Phase:</b>	<b>14/11/2023</b>	<b>25/11/2023</b>
5.1	2	Training	14/11/2023	23/11/2023
5.1.1	3	Handover Documentation to the Client's Team	14/11/2023	14/11/2023
5.1.2	3	Handover Access Keys and Credentials	15/11/2023	15/11/2023
5.1.3	3	Prepare Training Materials for Maintenance and Use of Web Applications	19/11/2023	19/11/2023
5.1.4	3	Training of Client's Team	23/11/2023	23/11/2023
5.2	2	Handover	24/11/2023	25/11/2023
5.2.1	3	Checking of Requirement Fulfilled by Client	24/11/2023	25/11/2023
5.2.2	3	Sign Off	25/11/2023	25/11/2023

Figure 14.2: Gantt Chart(3)

## Software Development Plan & Specification

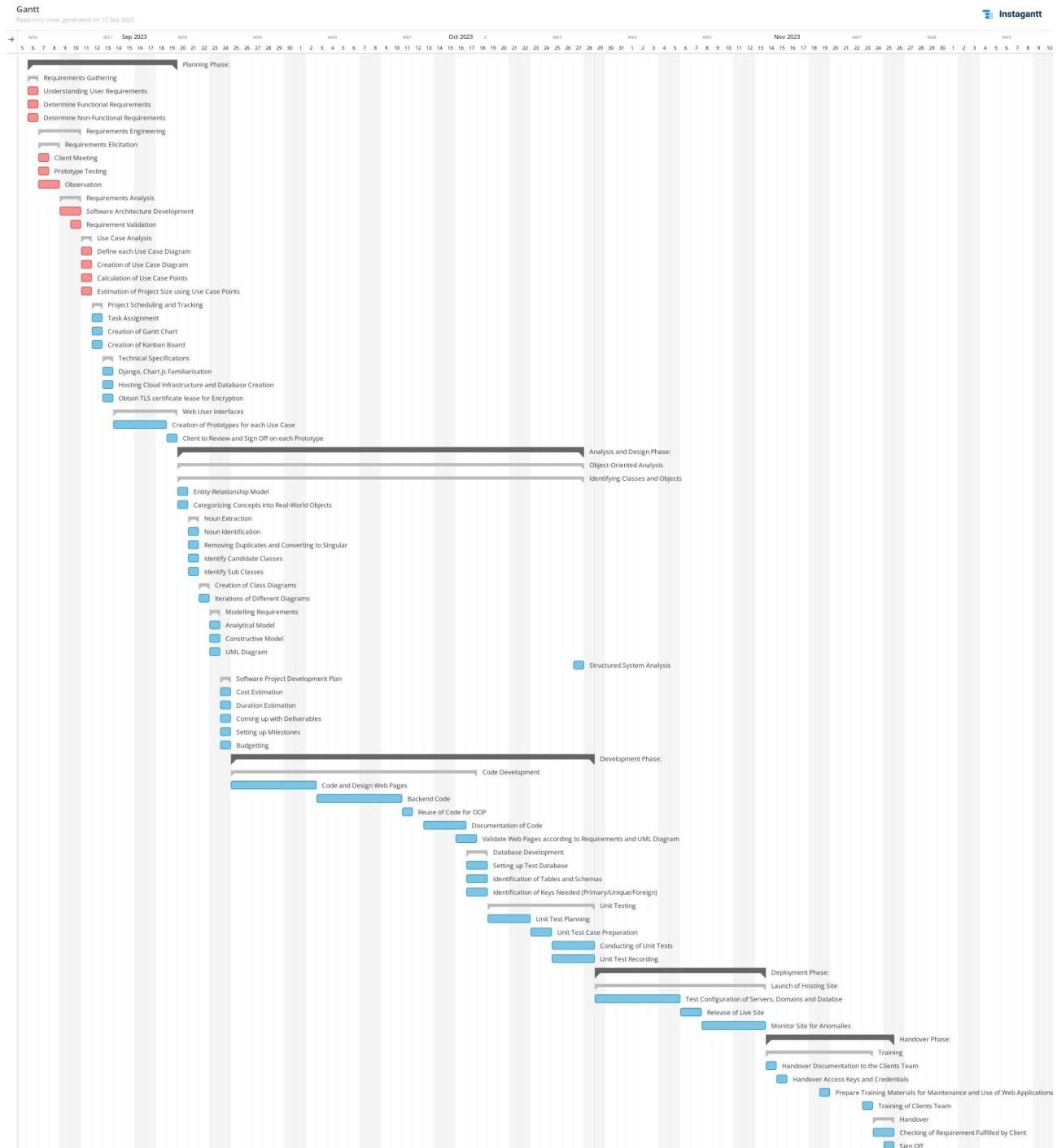


Figure 14.4: Gantt Chart(4)

## 5 Appendix A – Use Case Descriptions

Use Case ID:	<b>UC-1</b>
Use Case Name:	<b>Validate Account</b>
Description:	Users will only have access to features after validating their accounts via a logging in and logging out system with two-factor authentication.
Primary Actor:	Pilot, Manager, IT Admin
Preconditions:	They must have an account set up. They must have two-factor authentication set up.
Postconditions:	Dashboards related to the account type of the user will be shown.
Main Success Scenarios:	<ol style="list-style-type: none"><li>1. User visits website</li><li>2. User enters his username</li><li>3. User enters his password</li><li>4. User clicks on “login” button</li><li>5. User receives two-factor authentication code</li><li>6. User enters two-factor authentication code into website</li><li>7. A dashboard will be displayed and accessible by the user.</li></ol>
Alternative Scenarios:	4a User unable to login 4a1 An error message will be displayed to explain what went wrong 4a2 A retry button will be offered 4a3 A reset password button will be offered 6a Authentication code error 6a1 User re-enters authentication code 6a2 Generate new code button will be offered 6a3 User is forced back into login page after time runs out. An email and notification will be sent to IT Admin 6a4 User is forced back into login page after 3 times of trying. An email and notification will be sent to IT Admin

## **Software Development Plan & Specification**

---

Use Case ID:	<b>UC-2</b>
Use Case Name:	<b>View Dashboard</b>
Description:	For both pilots and managers, they will have access to their dashboard after logging in using their account.
Primary Actor:	User
Preconditions:	They must be logged into the website.
Postconditions:	Dashboards related to the account type of the user will be shown.
Main Success Scenarios:	Dashboard displays a summary of workload assigned
Alternative Scenarios:	

Use Case ID:	<b>UC-3</b>
Use Case Name:	<b>Search and filter for users</b>
Description:	A search and filtering functionality on the website. Note that search results will only show pilots since they are the focus of the system. IT Admins will have this functionality so that he will be able to manage accounts.
Primary Actor:	Pilot, Manager, IT Admin
Preconditions:	User must have access to his dashboard.
Postconditions:	Dashboard is updated according to search and filtering results.
Main Success Scenarios:	User types in relevant keywords and/or use filters in the search bar Relevant results will be displayed to the user Managers will be able to assign jobs to displayed pilots IT Admin will be able to search for all users
Alternative Scenarios:	1a. Keywords fail to produce results 1a1. An error message shows up saying there were no results from the search 1a2. Repeat 1 to 2

Use Case ID:	<b>UC-4</b>
Use Case Name:	<b>View Job Allocations</b>
Description:	Job assignments can be viewed on the dashboard.
Primary Actor:	User
Preconditions:	User must have access to his dashboard.
Postconditions:	
Main Success Scenarios:	User is able to view job assignment and limited but relevant information on the dashboard The job assignment can be clicked to view more details

**Software Development Plan & Specification**

Alternative Scenarios:	1a. System does not display job assignment 1a1. System will display an error telling the user what went wrong
------------------------	--

Use Case ID:	<b>UC-5</b>
Use Case Name:	<b>View Job Details</b>
Description:	Job details can be viewed on the website after clicking on the summarised job allocation on the dashboard.
Primary Actor:	User
Preconditions:	User must have access to his dashboard. Job allocation must be displayed on the dashboard.
Postconditions:	
Main Success Scenarios:	The job assignment can be clicked to view more details Details of job assignment will be displayed to the user
Alternative Scenarios:	1a. System does not display job assignment 1a1. System will display an error telling the user what went wrong 1a2. System offers a reload button

Use Case ID:	<b>UC-6</b>
Use Case Name:	<b>Reject Job</b>
Description:	They are allowed to reject job assignments based on certain regulations such as reaching prior mutual agreement after discussion with manager. If they reject a job without first discussing it with his manager, the manager will approach the pilot and decide if they have a valid reason to do so.
Primary Actor:	Pilot
Preconditions:	User must have access to his dashboard. Job allocation must be displayed on the dashboard
Postconditions:	An email and notification will be sent to the user
Main Success Scenarios:	User clicks on reject job button A warning will popup explaining to the user about the regulation and policies surrounding the rejection of jobs User clicks on the checkbox agreeing to the terms and conditions User clicks on confirm button to register the job rejection An email is sent to the user A notification is sent to the user's dashboard'
Alternative Scenarios:	4a. System could not register information

**Software Development Plan & Specification**

	4a1. System shows error message and provides reason for not being able to register information 4a2. System offers a retry or quit option 5a. Email not sent 5a1. System offers a retry or quit option 6a. Notification not sent 6a1. System offers a retry or quit option
--	--

Use Case ID:	<b>UC-7</b>
Use Case Name:	<b>Remove Job Assignment</b>
Description:	If a job has been rejected, managers can remove the allocation to prevent cluttering the dashboard.
Primary Actor:	Manager
Preconditions:	User must have access to his dashboard User must view job allocation.
Postconditions:	Job allocation will be removed from both the pilot's and manager's dashboard.
Main Success Scenarios:	A 'remove job' button will be displayed User clicks on 'remove job' button User clicks on confirm to register information An email is sent to the user and the pilot A notification is sent to the user and the pilot
Alternative Scenarios:	3a. System could not register information 3a1. System shows error message and provides reason for not being able to register information 3a2. System offers a retry or quit option 4a. Email not sent 4a1. System offers a retry or quit option 5a. Notification not sent 5a1. System offers a retry or quit option

Use Case ID:	<b>UC-8</b>
Use Case Name:	<b>Edit Personal Details</b>
Description:	Updating personal information should also be a function for pilots if needed. Information like details of his last training, rank and personal information must be updated regularly by the pilots.
Primary Actor:	User, IT Admin

**Software Development Plan & Specification**

Preconditions:	User must have access to his dashboard
Postconditions:	An email and notification will be sent to the user
Main Success Scenarios:	User clicks on profile icon Information relating to user will be displayed User clicks on update information User edits information User clicks on submit button to register new information An email is sent to the user A notification is sent to the user
Alternative Scenarios:	5a. System could not register information 5a1. System shows error message and provides reason for not being able to register information 5a2. System offers a retry or quit option 6a. Email not sent 6a1. System offers a retry or quit option 7a. Notification not sent 7a1. System offers a retry or quit option

Use Case ID:	<b>UC-9</b>
Use Case Name:	<b>View Notifications</b>
Description:	They should also be able to receive notifications so that they do not miss out on important updates.
Primary Actor:	User
Preconditions:	User must have access to his dashboard
Postconditions:	Notifications can be viewed on the website
Main Success Scenarios:	User clicks on notification icon Notifications will be displayed to user Notification will be marked as read after user views it
Alternative Scenarios:	

Use Case ID:	<b>UC-10</b>
Use Case Name:	<b>Indicate Availability/Preference</b>
Description:	Indicate their availability to work
Primary Actor:	Pilot
Preconditions:	Pilot must have access to the website.

***Software Development Plan & Specification***

Postconditions:	Confirmation email will be sent to the pilot. Notification will be sent to the pilot's dashboard.
Main Success Scenarios:	User selects a date and indicate availability to work that day User confirms that he is able to work that day System registers information System sends email and notification of confirmation to the user
Alternative Scenarios:	3a. System could not register information 3a1. System shows error message and provides reason for not being able to register information 3a2. System offers a retry or quit option 4a. Email or notification not sent 4a1. System offers a retry or quit option

## **Software Development Plan & Specification**

---

Use Case ID:	<b>UC-11</b>
Use Case Name:	<b>Visualise Workload</b>
Description:	For pilots, they should be able to visualise their overall workload within the week and month.
Primary Actor:	Pilot
Preconditions:	They must be logged into the website. They must have access to their dashboard.
Postconditions:	Dashboards related to the account type of the user will be shown.
Main Success Scenarios:	Dashboard displays a summary of workload assigned
Alternative Scenarios:	

Use Case ID:	<b>UC-12</b>
Use Case Name:	<b>View Schedule</b>
Description:	View the schedule to see if there are unassigned jobs.
Primary Actor:	Manager
Preconditions:	User must have access to the website. User must have access to dashboard.
Postconditions:	Confirmation email will be sent to the user. Notification will be sent to the user's dashboard.
Main Success Scenarios:	A schedule is displayed. On the schedule, a summary of jobs assigned and not assigned will be displayed If job has been rejected, the date will be highlighted User can click on unassigned dates to allocate jobs
Alternative Scenarios:	

Use Case ID:	<b>UC-13</b>
Use Case Name:	<b>Allocate Job</b>
Description:	Allocate jobs to pilots.
Primary Actor:	Manager
Preconditions:	User must have access to his dashboard
Postconditions:	An email and notification is sent to the user and the pilot

**Software Development Plan & Specification**

Main Success Scenarios:	User clicks on a date A list of recommended pilots will be displayed User hovers his mouse over a pilot to reveal an ‘assign job’ button User clicks on ‘assign job’ button User fills out job details User clicks on submit button to register pilots An email is sent to the user and the pilot A notification is sent to the user and the pilot
Alternative Scenarios:	6a. System could not register information 6a1. System shows error message and provides reason for not being able to register information 6a2. System offers a retry or quit option 7a. Email not sent 7a1. System offers a retry or quit option 8a. Notification not sent 8a1. System offers a retry or quit option

Use Case ID:	<b>UC-14</b>
Use Case Name:	<b>View Recommended Pilots</b>
Description:	System should recommend pilots based on matching requirements to ease the assignment process. Up to 3 pilots should be recommended per job that matches the requirements. In the case where the manager chooses not to select recommended pilots, he has the option to search and filter for other pilots to allocate that job to him.
Primary Actor:	Manager
Preconditions:	User must have access to his dashboard. User is trying to allocate a job.
Postconditions:	Details of flight information will be accurately displayed
Main Success Scenarios:	A list of recommended pilots will be displayed User can search for other pilots to assign the job User assigns job to pilot by clicking ‘assign job’ button User registers information to the system by clicking ‘submit’ button
Alternative Scenarios:	2a. Search yields no result 2a1. A system message appears saying there are no matching results 2a2. The search bar offers a retry button 4a. System could not register information 4a1. System shows error message and provides reason for not being able to register information 4a2. System offers a retry or quit option

## **Software Development Plan & Specification**

---

Use Case ID:	<b>UC-15</b>
Use Case Name:	<b>View top 3 pilots with less than 40 hours workload</b>
Description:	On the dashboard, the top 3 pilots with less than 40 working hours in a week should be highlighted
Primary Actor:	Manager
Preconditions:	User must have access to his dashboard
Postconditions:	Pilots will be displayed
Main Success Scenarios:	Dashboard displays pilots Information on pilot shown will show which week he has less than 40 hours of workload User can click on pilot for more details
Alternative Scenarios:	

Use Case ID:	<b>UC-16</b>
Use Case Name:	<b>View all pilots with more than 40 hours of workload</b>
Description:	Highlight all pilots that has more than 40 working hours.
Primary Actor:	Manager
Preconditions:	User must have access to his dashboard
Postconditions:	Pilots will be displayed
Main Success Scenarios:	Dashboard displays pilots Information on pilot shown will show which week he has more than 40 hours of workload User can click on pilot for more details
Alternative Scenarios:	

Use Case ID:	<b>UC-17</b>
Use Case Name:	<b>Assign Demerit Points</b>
Description:	They should also be able to assign demerit points based on whether a pilot has valid reasons for rejecting a job upon discussion.
Primary Actor:	Manager
Preconditions:	User must have access to his dashboard
Postconditions:	Details of flight information will be accurately displayed
Main Success Scenarios:	A notification of the pilot rejecting a job will be displayed to the user Pilot's information will be displayed on the dashboard User can click on the pilot to view more details

**Software Development Plan & Specification**

	<p>User assign demerit points on the more details page  User can click on confirm to register job rejection information  An email is sent to the user and the pilot  A notification is sent to the user and the pilot</p>
Alternative Scenarios:	<p>5a. System could not register information  5a1. System shows error message and provides reason for not being able to register information  5a2. System offers a retry or quit option  6a. Email not sent  6a1. System offers a retry or quit option  7a. Notification not sent  7a1. System offers a retry or quit option</p>

Use Case ID:	<b>UC-18</b>
Use Case Name:	<b>Account Management</b>
Description:	Only IT administrators should be able to add and delete accounts to the system.
Primary Actor:	IT Admin
Preconditions:	User must have administrative access to the website and database.
Postconditions:	User can successfully add and delete accounts
Main Success Scenarios:	<p>On the website, user can search and filter for personnel  User clicks on ‘add personnel’ button to add new account  User clicks on ‘delete personnel’ button to delete existing account  User can click on personnel  User edits personnel details  User registers information by clicking on the ‘submit’ button.  An email is sent to the user involved.  A notification is sent to the user involved.</p>
Alternative Scenarios:	<p>6a. System could not register information  6a1. System shows error message and provides reason for not being able to register information  6a2. System offers a retry or quit option  7a. Email not sent  7a1. System offers a retry or quit option  8a. Notification not sent  8a1. System offers a retry or quit option</p>

***Software Development Plan & Specification***

Use Case ID:	<b>UC-19</b>
Use Case Name:	<b>Database Management</b>
Description:	IT Admins have access to database for administrative purposes.
Primary Actor:	IT Admin
Preconditions:	User must have administrative access to database.
Postconditions:	
Main Success Scenarios:	User can search for and edit databases User can search for and edit tables User can search for and edit columns User can search for and edit data
Alternative Scenarios:	1a Database does not show up 1a1 Error will be shown to tell user what is wrong 1a2 Repeat 1. 2a Table does not show up 2a1 Error will be shown to tell user what is wrong 2a2 Repeat 2. 3a Column does not show up 3a1 Error will be shown to tell user what is wrong 3a2 Repeat 3. 4a Data does not show up 4a1 Error will be shown to tell user what is wrong 4a2 Repeat 4.

## 6 Appendix B – Data Dictionary

### **Pilots/Users**

*It refers to staff that are either under training or are trained to control the various planes of FLYSG Airlines.*

### **Managers/Management/Users**

*It refers to any admin staff or manager and any staff within the human resource management level under FLYSG Airlines.*

### **IT Administrator**

*The staff that belongs to the IT department of FLYSG Airlines that manages any IT system or matters of the company.*

### **IT**

*Information Technology*

### **MYSQL**

*A database management system*

### **Wireframe**

*A visual representation to outline the basic layout of the work allocation webpage*

### **2FA**

*Two-Factor Authentication*

### **ICA**

*Immigration & Checkpoints Authority*

### **PDPA**

*Personal Data Protection Act, a Singapore Data Protection Law*

### **PDPC**

*Personal Data Protection Commission, a Singapore regulatory authority to enforce data protection laws*

### **HTTPS**

*Hypertext Transfer Protocol Secure, a secure communication protocol*

## ***Software Development Plan & Specification***

---

### **AES-256**

*Advanced Encryption Standard, 256-bit key, used to protect data*

### **DoS**

*Denial of Service, a cyberattack to disrupt a computer system or network's availability by flooding it*

### **SOC 2**

*Service Organization Control , a cybersecurity framework*

### **Firewall**

*A network security device or software that monitors and controls in and outgoing traffic of the network*

### **WCAG**

*Web Content Accessibility Guidelines, a set of internationally recognised standards for accessibility of digital contents*

### **TLS**

*Transport Layer Security, a security protocol that provides privacy and data integrity for Internet communications*

*Software Development Plan & Specification*

---