ICT2101/2201
Introduction to Software Engineering

Milestone 1: Software Development Plan & Specification

for

Airline Pilots

Prepared by

Team's Name: ***Hands over Head***

Ng Chin Guan Adrian	**<2100584>**
Adriel Koh Guang Hui	**<2101296>**
He Haiqi	**<2102948>**
Shathiya Sulthana D/O Shajahan	**<2102605>**

Lab Group: *P6 - 2*

GitHub handle: *https://github.com/adriel723/ICT2201-Team-Project*

Date: *23/09/2022*

Table of Contents

1 Introduction

The project aims to provide Airline Pilots for The Company an interactive and informative way of viewing their work allocations using a Web-Based Workload Management System. The system hopes to provide a streamline experience for employees to perform tasks such as informing job availability and viewing jobs assigned. Managers using the system will be able to visualize manpower availability of all employees and assign jobs accordingly.

1.1 Product Scope

The initial wireframe design and functionalities will be completed using PowerPoint and Photoshop. The product will be a Web application that employees in The Company can access to keep track of their individual workload and tasks assigned. Employees and managers will be able to access the application using a web browser on their desktop and their mobile phones.

Employees using the system will be able to view their job assignments for the week, including details such as working hours assigned for each job and the location of each flight. In the event where employees are unable to fulfill their job assignments, the system will allow employees to inform their manager about their unavailability. Overall, the product is designed to enable employees to easily keep track of their workload and achieve a healthy work-life balance.

The product will allow the managers to easily assign workload to their employees. Managers using the system will be able to assign employees their job assignments, as well as review job availability and preferences made by employees. In the event where the employee rejects an assignment, the employee must inform their manager by Wednesday 2359. If their request is approved, the manager will reallocate the employee's workload by Thursday 2359.

1.2 Related Background Literature

Managing the workload of employees is challenging, especially when the goal is to achieve work-life balance to not only retain employees, but also attract new potential employees. Despite the challenges, it is not impossible with adequate strategies put in place [1].

Once such strategy is adopting the use of a Workload Management System. Workload Management Systems have been around for decades; hence it is nothing new. They are commonly known as batch systems in the world of computing [2].

A study conducted has shown that companies are constantly looking for tools to ease workload for their employees. The study explores the use of workload management tools and how workload management creates a work-life balance cycle that is highly sustainable and rewarding [3].

1.3 Intended Audience and Document Overview

This document is intended for the client, project managers, system developers, IT administrators and testers who will be using the system.

This document outlines the overall description of the product, the functional and non-functional requirements gathered, and the project estimation plan.

For clients and IT administrators, refer to page 5 for the overall description of the product. The section covers the product overview, product functionality, and assumptions and dependencies of the product.

For system developers, refer to page 7 for specific requirements. This section outlines the user interface requirements, functional and non-functional requirements, and the use-case model. Refer to Appendix A for use case diagrams.

Project managers should refer to page 10 for the project estimation and planning. This section covers the software estimation cost and provides the project management Gantt Chart.

1.4 References and Acknowledgments

[1] Hieu, Vu & Tai, Tran. (2020). Strategies and Analyzing Workload Management of Employee Achievement in an Organization. 4. 1-6.

[2] I Sfiligoi (2008) J. Phys.: Conf. Ser. 119 062044 An update on the scalability limits of the Condor batch system

[3] I. Lupu, M. Ruiz-Castro, and B. Leca, “Role Distancing and the Persistence of Long Work Hours in Professional Service Firms,” Organization Studies, p. 017084062093406, Jun. 2020, doi: 10.1177/0170840620934064.

2 Overall Description

2.1 Product Overview

The product will be a new, self-contained product that will be developed into a web application. The web application will be a workload management system that employees and staff can access using their web browser. The product aims to assist staff in achieving work-life balance by providing an interface that can be used to enter in their job availability ahead of time. Essentially, the product also serves as a leave management system since staff members can also indicate their job preferences.

A high-level diagram of the System Architecture is shown below

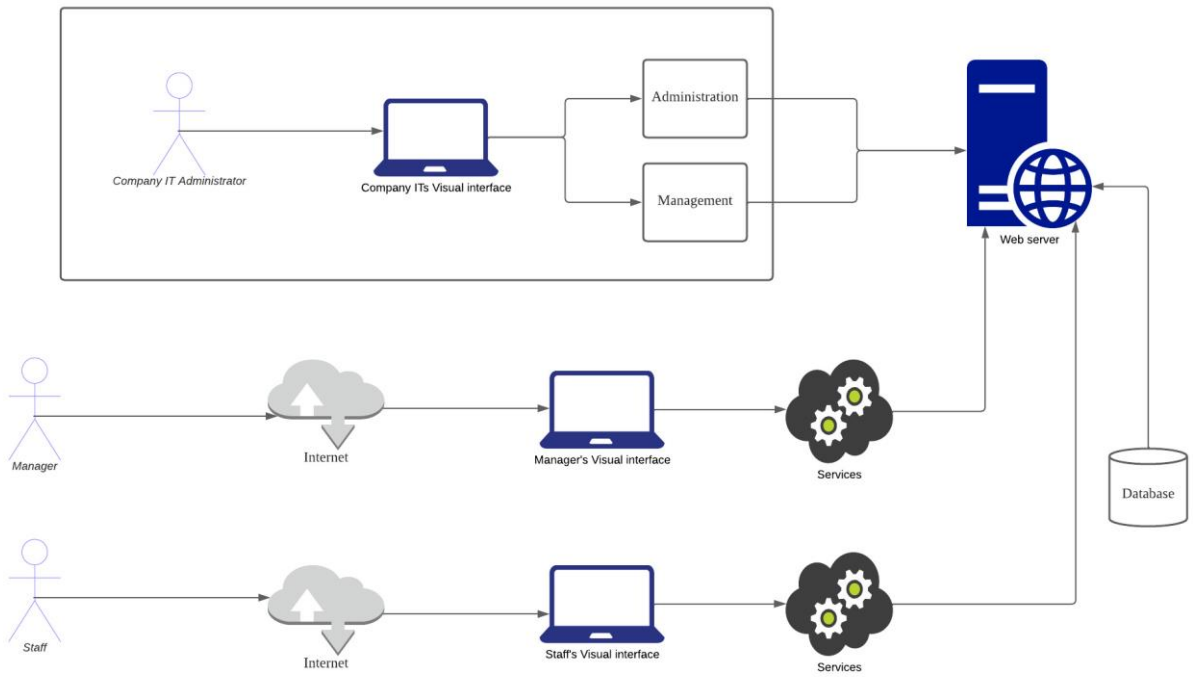


Figure 1: System Architecture Design

As the product is a web application, staff and managers must be connected to the internet. Once connected, they will be required to login using their credentials before accessing the application and viewing their individual interface. As each employee is uniquely identified using their employee ID, no two employees will have the same details displayed in their interface. When staff performs an operation, such as editing their job availability, the request will be stored in the database. Company IT will be able to directly access the web server and perform administration and management actions.

2.2 Product Functionality

PF1: The product shall allow employees to view his/her own job assignments, working hours, and inform the system of their job availability.
PF2: The product shall allow employees to indicate any assigned job they cannot fulfill ahead of time to the manager.
PF3: The product shall allow the manager to visualize all manpower availability at any time up to one month earlier.
PF4: The product shall allow the manager to allocate jobs for staff based on their job availability.

2.3 Assumptions and Dependencies

It is assumed that the client wants a secure system. Hence, security measures such as firewalls, encryption of passwords, and input validation will be implemented. The server will also reside on client premises, and data will be stored out of cloud premises. The system will also be designed with scalability in mind, as it is assumed that the number of staff using the system can potentially grow over time.

Since the product is a web-based application, an internet browser and internet connection will be required. As such, it will be assumed that the user possesses stable internet connectivity to use the product. All users are assumed to be familiar with using the internet, using a keyboard and mouse to navigate through the product. All users are also assumed to have working proficiency in English.

3 Specific Requirements

3.1 User Interface Requirements

UIR1: Users will be required to login before being able to access the application
UIR2: Users will be prompted to re-enter their details if invalid credentials were entered.
UIR3: A calendar view should be shown immediately on the landing page for staff members.
UIR4: The calendar view should show all relevant details, such as jobs allocated for the week and overall workload for the month
UIR5: Staff members will be able to inform their manager of their job availabilities through a request function.
UIR6: Staff members will be able to view their current pending requests made to their manager.
UIR7: Staff members will receive a notification pertaining to the outcome of the request made.
UIR8: Managers will be able to view the top three staff members with the lowest workload on the landing page.
UIR9: Managers will also be able to view staff members who have been assigned jobs over 40 hours for the week**.**
UIR10: Managers will be able to view requests from staff members and either approve or reject the request

3.2 Functional Requirements

FR1: The system shall uniquely identify staff members based on their six-digit employee ID.
FR2: The system shall allow allocation planning of employees every Thursday.
FR3: The system shall accept all employee's availabilities up till every Wednesday.
FR4: The system shall allow Managers to allocate jobs to staff members weekly.
FR5: The system shall allow Managers to view the workload of all staff
FR6: The system shall display a staff's availability for the week, workload assigned, job preference and the location at a particular date.
FR7: The system shall allow managers to reject or accept staff requests after every Wednesday of the week.
FR8: The system shall display the top three staff with the lowest workload, and all staffs over 40 hours of jobs allocated on the manager's landing page.
FR9: The system shall display the staff's own weekly job assignment and overall workload for the month.
FR10: The system shall allow staffs to add and edit their availabilities up to 5 weeks ahead of time.
FR11: The system shall allow staffs to reject jobs that are assigned to them in return the system will notify the staff to discuss with their manager before rejecting.
FR12: The system shall allow IT administrator to add new staffs and managers.
FR13: The system shall not allow managers to assign jobs to staff who have been working over 40 hours.
FR14: The system shall allow employees to view the status of their requests.
FR15: The system shall allow the manager to update the training status of all staff.

3.3 Use Case Model

The Use Case Diagram is as shown below. Refer to Appendix A for the Use Case Description.

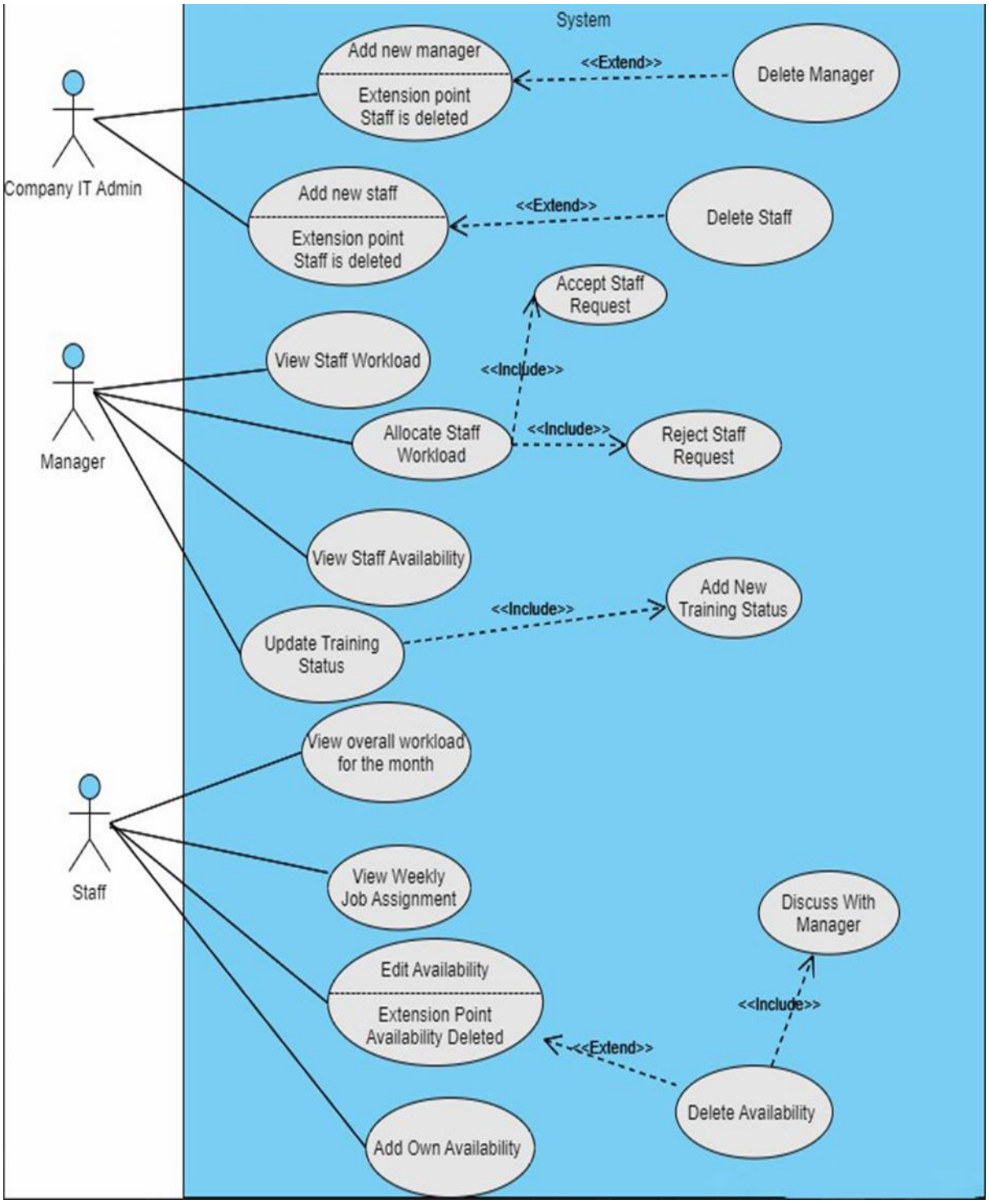


Figure 2: Use Case Diagram

3.4 Non-functional Requirements

NFR1: The system shall be hosted on client premises.
NFR2: The system should be accessible on popular mobile device web browsers such as Chrome, Safari, and Firefox.
NFR3: Staff members will not be allowed to log into other staff members accounts. If caught, such an attempt will be reported to the security administrator.
NFR4: The system should be migratable from one Operating System to another Operating System.

3.4.1 Performance Requirements

PR1: Every page should be loaded within 2 seconds.
PR2: The system should be able to handle 1000 requests simultaneously.
PR3: Requests raised by staff should be reflected in the pending within 10 seconds.
PR4: The system should be able to reflect the timings based on the timezone that the user is in.
PR5: The system should be able to perform without failure in 95 percent of use cases.

3.4.2 Safety and Security Requirements

SSR1: The system should be backed up monthly.
SSR2: Data transmitted from and received are always encrypted with AES-128.
SSR3: All approving roles must use a different factor of authentication from the user authentications (on top of 2FA) for approval processes.
SSR4: The system will require all staff to enter the username (Employee ID) and password, to be able to access the functions in this product. The system will only grant access when the users enter the correct password.
SSR5: The system will only allow the user to have access if the user sets a strong password. A strong password here refers to a password with a minimum of 8 characters including at least one capital letter and one special character.
SSR6: The system shall lock out a staff if they enter a wrong password 10 times in a row.
SSR7: The system shall maintain user logs for the past 30 days.
SSR8: The system shall be protected against OWASP top 10 web application vulnerabilities
SSR9: The system will prompt users to change their password after 3 months. The most recent 3 passwords may not be reused.

3.5 Other Requirements

OR1: The system shall be designed to take PDPA regulations into account.
OR2: The system shall adopt the Singapore date and timezone (GMT +8).
OR3: The Manager will not be able to access Staff's account to edit job availability.
OR4: Only the Company IT administrator is allowed to edit entries in the database.

4 Project Estimation and Plan

4.1 Software Estimation

The cost estimation has been computed using Use Case Point

Unadjusted Use Case Weight (UUCW)

Use Case Number	Use Case Name	Total Transactions/Usecase	Complexity of a Usecase
1	Adding new users (Staffs and managers)	4	10
2	Deleting users (Staffs and managers)	5	10
3	Submit work availability	4	10
4	Assigning staff workload	4	10
5	Staff Rejects workload allocation	2	5
6	Manager processes staff request	5	10
7	Updating training status	5	10
		Total unadjusted use case weight	**65**

Unadjusted Actor Weight (UAW)

Actor	weight	count	product
Simple	1	0	0
Medimum	2	0	0
Complex	3	3	9
		Total	9

Technical Complexity Factors (TCF)

Factor	Description	Weight	Assesment	Product
T1	Distributed system	2	3	6
T2	Response time/performance objectives	1	3	3
T3	End-user efficiency	1	5	5
T4	Internal processing complexity	1	2	2
T5	Code reusability	1	0	0
T6	Easy to install	0.5	0	0
T7	Easy to use	0.5	5	2.5
T8	Portability to other platforms	2	5	10
T9	System maintenance	1	2	2
T10	Concurrent/parallel processing	1	1	1
T11	Security features	1	5	5
T12	Access for third parties	1	3	3
T13	End user training	1	1	1
		Total		40.5
		Technical Complexity Factor		0.6 + 0.01 * 40.5 = 1.005

Environmental Factor (TCF)

Environment	weight	Assessment	Product
Familiar with Development Process	1.5	3	4.5
Part time workers	-1	4	-4
Analyst capability	0.5	4	2
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	2	4
	Total		12
	Environmental Factors		1.4 + (- 0.03 * 12) = 1.04

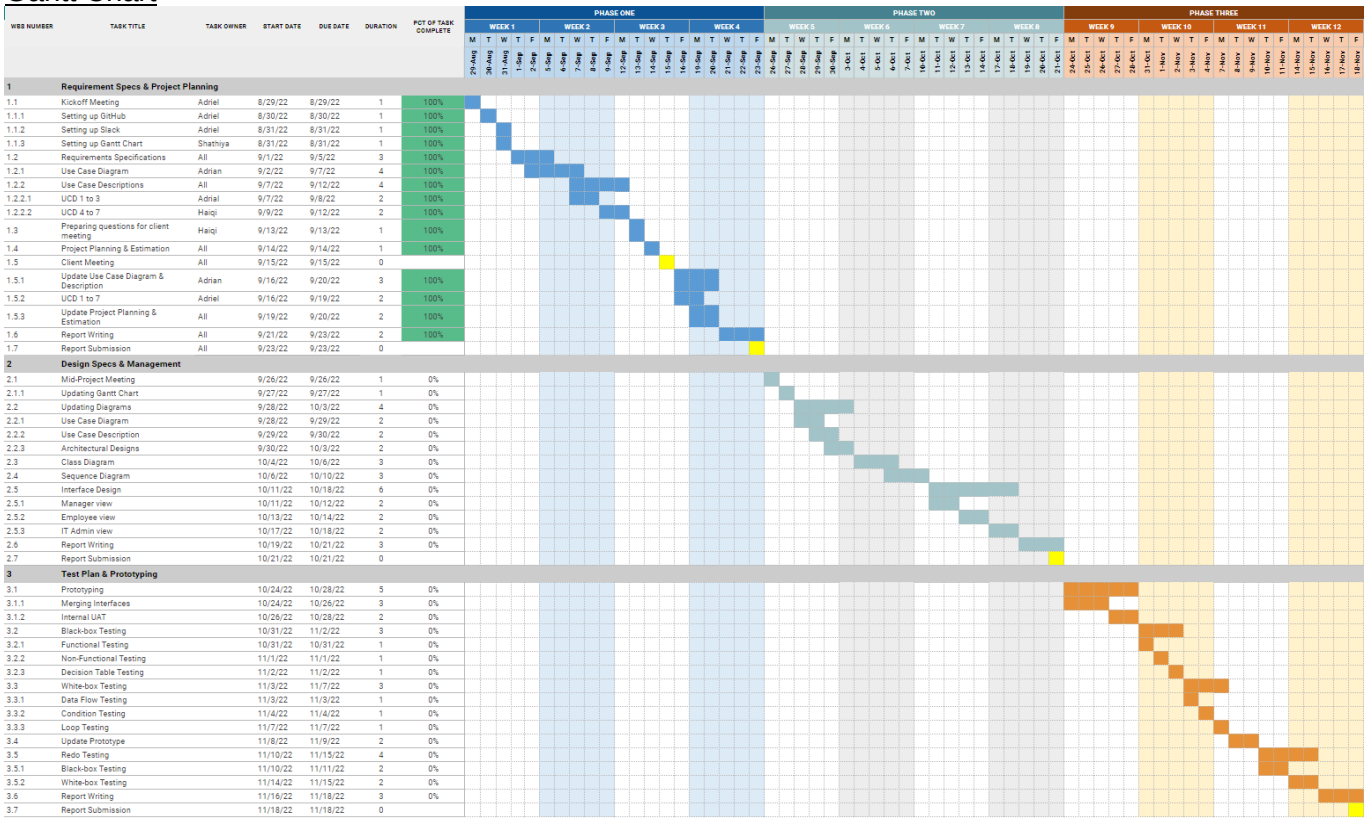
Use Case Point (UCP)

Use Cases	Actors	UUCW	TCF	EF	UCP	Effort estimation (in developer hours)
65	9	74	1.005	1.04	77.3448	15 * 77.3448 = 1160.17hours - 30 * 77.3448 = 2320.34hours

Total Time Estimated = 1160 – 2320 hours

4.2 Project Management

Gantt Chart



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Requirement Specs & Project Planning**					
1.1	Kickoff Meeting	Adriel	8/29/22	8/29/22	1	100%
1.1.1	Setting up GitHub	Adriel	8/30/22	8/30/22	1	100%
1.1.2	Setting up Slack	Adriel	8/31/22	8/31/22	1	100%
1.1.3	Setting up Gantt Chart	Shathiya	8/31/22	8/31/22	1	100%
1.2	Requirements Specifications	All	9/1/22	9/5/22	3	100%
1.2.1	Use Case Diagram	Adrian	9/2/22	9/7/22	4	100%
1.2.2	Use Case Descriptions	All	9/7/22	9/12/22	4	100%
1.2.2.1	UCD 1 to 3	Adrial	9/7/22	9/8/22	2	100%
1.2.2.2	UCD 4 to 7	Haiqi	9/9/22	9/12/22	2	100%
1.3	Preparing questions for client meeting	Haiqi	9/13/22	9/13/22	1	100%
1.4	Project Planning & Estimation	All	9/14/22	9/14/22	1	100%
1.5	Client Meeting	All	9/15/22	9/15/22	0	
1.5.1	Update Use Case Diagram & Description	Adrian	9/16/22	9/20/22	3	100%
1.5.2	UCD 1 to 7	Adriel	9/16/22	9/19/22	2	100%
1.5.3	Update Project Planning & Estimation	All	9/19/22	9/20/22	2	100%
1.6	Report Writing	All	9/21/22	9/23/22	2	100%
1.7	Report Submission	All	9/23/22	9/23/22	0	
2	**Design Specs & Management**					
2.1	Mid-Project Meeting		9/26/22	9/26/22	1	0%
2.1.1	Updating Gantt Chart		9/27/22	9/27/22	1	0%
2.2	Updating Diagrams		9/28/22	10/3/22	4	0%
2.2.1	Use Case Diagram		9/28/22	9/29/22	2	0%
2.2.2	Use Case Description		9/29/22	9/30/22	2	0%
2.2.3	Architectural Designs		9/30/22	10/3/22	2	0%
2.3	Class Diagram		10/4/22	10/6/22	3	0%
2.4	Sequence Diagram		10/6/22	10/10/22	3	0%
2.5	Interface Design		10/11/22	10/18/22	6	0%
2.5.1	Manager view		10/11/22	10/12/22	2	0%
2.5.2	Employee view		10/13/22	10/14/22	2	0%
2.5.3	IT Admin view		10/17/22	10/18/22	2	0%
2.6	Report Writing		10/19/22	10/21/22	3	0%
2.7	Report Submission		10/21/22	10/21/22	0	
3	**Test Plan & Prototyping**					
3.1	Prototyping		10/24/22	10/28/22	5	0%
3.1.1	Merging Interfaces		10/24/22	10/26/22	3	0%
3.1.2	Internal UAT		10/26/22	10/28/22	2	0%
3.2	Black-box Testing		10/31/22	11/2/22	3	0%
3.2.1	Functional Testing		10/31/22	10/31/22	1	0%
3.2.2	Non-Functional Testing		11/1/22	11/1/22	1	0%
3.2.3	Decision Table Testing		11/2/22	11/2/22	1	0%
3.3	White-box Testing		11/3/22	11/7/22	3	0%
3.3.1	Data Flow Testing		11/3/22	11/3/22	1	0%
3.3.2	Condition Testing		11/4/22	11/4/22	1	0%
3.3.3	Loop Testing		11/7/22	11/7/22	1	0%
3.4	Update Prototype		11/8/22	11/9/22	2	0%
3.5	Redo Testing		11/10/22	11/15/22	4	0%
3.5.1	Black-box Testing		11/10/22	11/11/22	2	0%
3.5.2	White-box Testing		11/14/22	11/15/22	2	0%
3.6	Report Writing		11/16/22	11/18/22	3	0%
3.7	Report Submission		11/18/22	11/18/22	0	

Figure 3:Gantt Chart

WBS Diagram

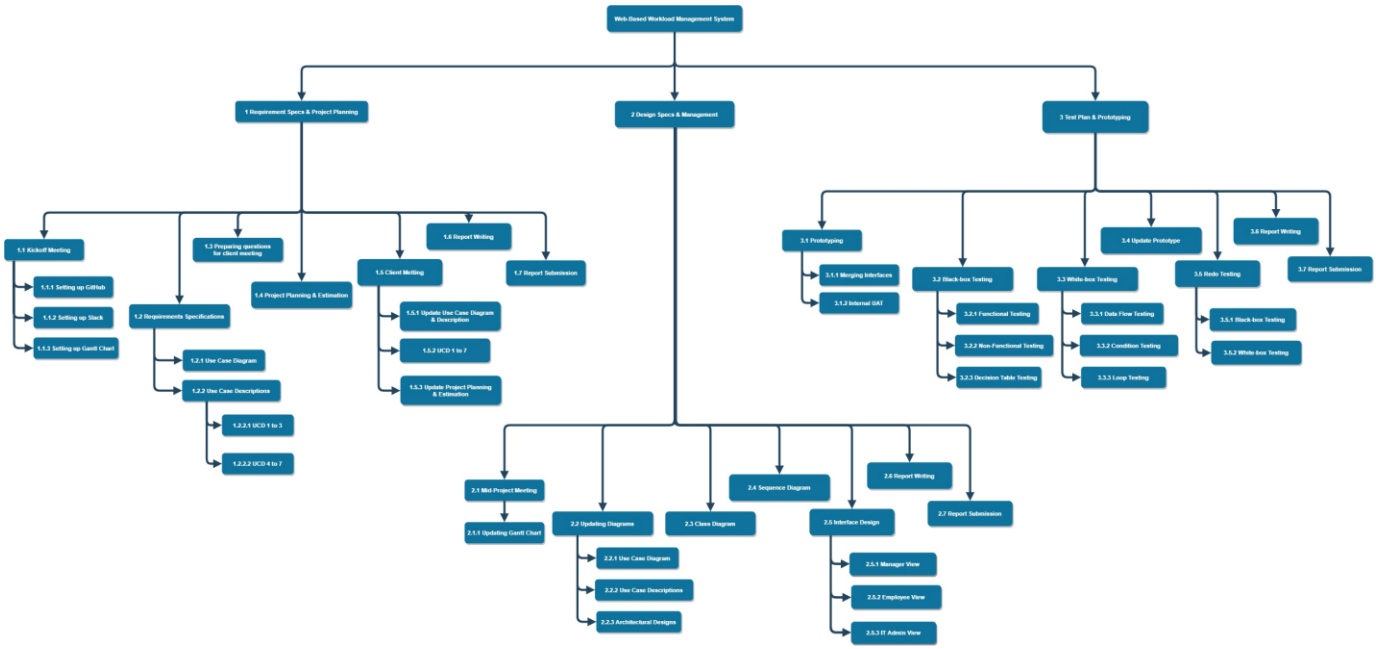


Figure 4: WBS Diagram

Burndown Chart

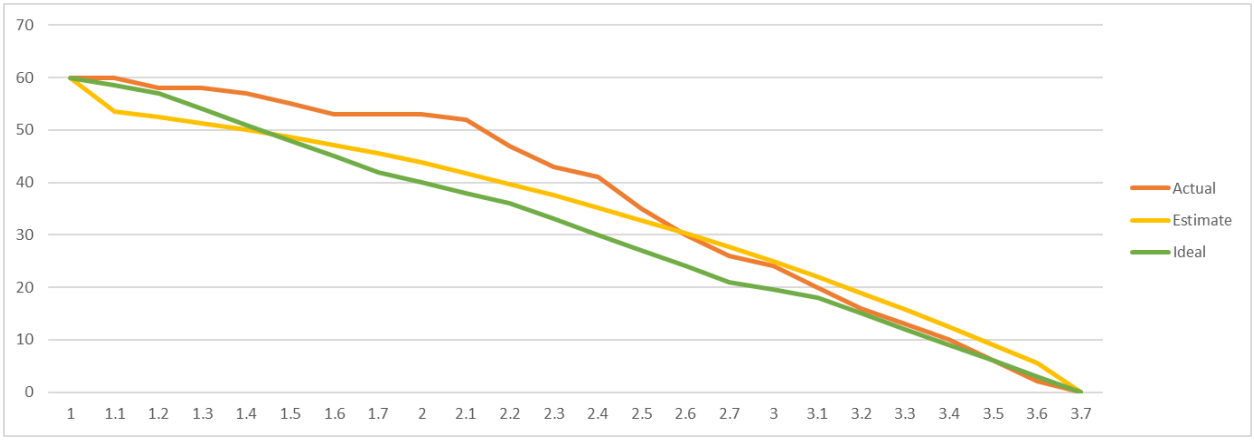


Figure 5: Burndown Chart

5 Individual Members Task Reflections

Adriel Koh Guang Hui: I have learned to appreciate and understand the importance of requirements gathering. If a software is developed without a clear understanding of requirements from the client, the final product will not meet the needs of the client. Hence, the client meeting in week 3 was essential as our team was able to clear our initial doubts and have a clear understanding of the client's requirements.

Given a second chance to do this again, I would have researched deeper into creating a better system architecture, as well as familiarizing better with the technologies that could lead to any dependencies of the product. As a team, we did our best to complete our tasks distributed timely, communicated clearly, and clarified any doubts we had regarding tasks assigned.

He Haigi: I have learnt the importance of planning and gathering requirements before jumping straight into coding. Having a good understanding of the requirements and having a plan would make the project more manageable for everyone involved. Creating the data dictionary helps keep everyone on the same page and avoid confusion between members.

If I were to do this milestone again, I would familiarise myself more with the different diagrams and methodologies before starting work on the report. This would make doing milestone 1 less confusing and more productive. The team communicated well, and tasks were done on time.

Ng Chin Guan Adrian: From this project, I have learnt that the use case diagram, description, and the functional requirements are heavily dependent on one another. For example, if there is a change in the use case diagram, the use case description and the functional requirements must be updated as well. I have also recognised the importance of requirements gathering. When done properly, the use case description will be accurate. It is also important to ensure the group is on the same page as it helps to facilitate the progress of the project

Given a second chance to do this, I would start on planning the GitHub project board first instead of concurrently doing both the report and the project board. I feel that this would allow us to complete the project faster and smoother. Communication within the group was great and everyone completed the task there were assigned.

Shathiya Sulthana D/O Shajahan: Up till the first milestone, I have learnt the different methods used in the requirements gathering phase, how we can convert the requirements into functionality and how the different functionality will have an impact in our planning and estimation. I have understood that the different diagram prove varies needs, such for ease of explanation and visualisation.

During this milestone, I would have started by looking through more documents so as to better understand the needs of the company, then moving on GitHub project board then discussing with the group. This would have helped to view the different kinds of perspective available. The communication and understanding between the team were great. Additionally, establishing a common goal at the start to the project as it helped us to think alike in the subsequent meetings.

6 Appendix A – Use Case Descriptions

Use Case ID:	UC-1
Use Case Name:	Adding new users (Staffs and managers)
Description:	IT Administrator adds new staffs and managers
Primary Actor:	IT Administrator
Preconditions:	IT Administrator is logged into the system
Postconditions:	The new user can now log in to the system
Main Success Scenarios:	1. IT Administrator chooses to add a new user 2. The system display lists of credentials for the IT administrator to add 3. The IT administrator submits the information 4. The system shows that the user is successfully added into the system
Alternative Scenarios:	3a. The Company IT Admin did not complete filling up the information 3a1. The system sends an error message which prompts the Company IT Admin to finish the form 3a2. The Company IT Admin can choose to submit the form or exit
Priority	High

Use Case ID:	UC-2
Use Case Name:	Deleting users (Staffs and managers)
Description:	Deleting users that have left the company
Primary Actor:	Company IT Admin
Preconditions:	Company IT Admin is logged in to the system
Postconditions:	User is successfully deleted from the system
Main Success Scenarios:	1. Company IT Admin chooses to delete a user 2. The system display lists of employees for the Company IT Admin to delete 3. Company IT Admin submits the information 4. The system shows a confirmation message before deleting 5. Company IT Admin clicks on “Yes” 6. The system displays a message stating that the user is successfully removed from the system
Alternative Scenarios:	5a. Company IT Admin clicks on “No” 5a1. The system does not delete the user 5a2. The system brings the Company IT Admin back to the list of employees page
Priority	Low

Use Case ID:	UC-3
Use Case Name:	Submit work availability
Description:	Staff accesses the system and submits the dates that they are not available to work to the manager every Wednesday
Primary Actor:	Staff
Preconditions:	Staff is registered and logged into the system
Postconditions:	The manager will be able to view the updated staff availability
Main Success Scenarios:	1. Staff chooses to allocate the dates available 2. System displays the calendar for the staff to choose 3. Staff selects the dates they are not available and submits the information 4. System registers the staff for the selected dates and sends a success notification
Alternative Scenarios:	3a. Staff chooses a date that is more than 5 weeks from the current date 3a1 System displays error message saying that the selected date is not available (change phrasing later) 4a. System displays an error, provides the reason and offer a retry option 4a1 Staff can retry or quit
Priority	High

Use Case ID:	UC-4
Use Case Name:	Assigning staff workload
Description:	The manager assigns the workload for the staffs every Thursday
Primary Actor:	Manager
Preconditions:	The manager is registered and logged in to the system, the system filters out staff that are ineligible to work (E.g., Staff did not complete training) Staffs have submitted their workload availability
Postconditions:	Staff can view the updated work allocation.
Main Success Scenarios:	1. Manager clicks on the “Assign workload” button 2. The system displays the top 3 staff with the lowest workload and highlight all staffs over 40 hours of job allocated 3. The manager allocates the workload based on the staff’s availability and submits the form 4. The system displays that the workload has been successfully allocated
Alternative Scenarios:	3a. Manager allocates a staff with more than 40 hours of workload in a week 3a1. The system will not allow the workload to be allocated to that staff 3b. There are no staff available 3b1. System prompts a warning message that there is no staff to be assigned
Priority	High

Use Case ID:	UC-5
Use Case Name:	Staff Rejects workload allocation
Description:	Staffs can reject manager's workload allocation
Primary Actor:	Staff
Preconditions:	Staff is logged in to the system. Manager has finished assigning all workload allocation for the week
Postconditions:	The selected date will not be allocated to the staff
Main Success Scenarios:	1. System displays the staff workload allocation for the week 2. Staff selects the day that he wants to reject 3. System notifies the staff to discuss with the manager
Alternative Scenarios:	-
Priority	High

Use Case ID:	UC-6
Use Case Name:	Manager processes staff request
Description:	Manager views pending requests and processes staff's rejection request on Thursday
Primary Actor:	Manager
Preconditions:	Staff has raised request to reject job allocation
Postconditions:	Staff request has been cleared
Main Success Scenarios:	1. Manager opens pending requests tab 2. System displays the pending requests 3. Manager selects staff request and processes it 4. System notifies that request has been completed
Alternative Scenarios:	2a. No pending requests 2a1. System displays message saying there are no requests currently pending 4a. System does not send notification 4a1 System warns that notification was not sent, and offers retry option 4a2 Manager can choose to retry or exit
Priority	Medium

Use Case ID:	UC-7
Use Case Name:	Updating training status
Description:	Manager updates a staff training status when they completed training
Primary Actor:	Manager
Preconditions:	Manager is logged in to the system
Postconditions:	The training status will be updated accordingly
Main Success Scenarios:	1. The manager chooses to get list of staff 2. The system displays a list of staff 3. The manager selects the staff 4. The system displays all the airline fleet 5. The manager selects the airline fleet that the staff is already trained for and submits the form 6. The system returns a success message
Alternative Scenarios:	6a. Success message was not returned 6a1. System warns that an error has occurred, and offers retry option 6a2. Manager can choose to retry or exit
Priority	Medium

7 Appendix B – Data Dictionary

IT Administrator

A person in the client's IT department, responsible for adding new staff members and managers.

Manager

A person in the client's company, responsible for assigning workload to staff. Managers must keep in mind what aircraft a given staff is trained for before assigning workload to the staff.

Staff

A person in the client's company, which can be a aircraft pilot or air steward/ stewardess. Pilots must go through at least two months of training for the make and model before they can be qualified to fly it. Air steward/ stewardess do not have this requirement. A staff should not work more than 40 hours a week.

Job availability

The hours that the staff is able to work in the coming 1 to 5 weeks, not to be confused by job preference. At least 1 week of availability must be indicated by Wednesday, 2359 hours, if not the manager can assume staff is available for all jobs in the following week(s).

Job preference

The aircraft model(s) that the staff wishes to fly for the week, not to be confused by job availability. At least 1 week of availability must be indicated by Wednesday, 2359 hours, if not the manager can assume staff is available for all aircraft suitable in the following week.