Logo

Description automatically generated

ICT2101/2201

Introduction to Software Engineering

Milestone 1: Software DevelopmentPlan & Specification

for

Airline Cabin Crew (Air Manager)

Prepared by Team 9

Team’s Name:

|  |  |
| --- | --- |
| Gian Meng Rong | 2100657 |
| Jon Tan Jun Ting | 2100787 |
| Wesley Chiau Yan Jun | 2100751 |
| Keefe Christian Lee Kit Onn | 2100829 |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Lab Group: | P*6 - 9* |
| Github handle: | *ICT2201-P6-9* |
| Date: | *<replace with the date of submission here>* |

[1 Introduction [KEEEEFE] 2](#_Toc1797881602)

[1.1 Product Scope (Keefe) 3](#_Toc679061375)

[1.2 Related Background Literature (Wesley) 3](#_Toc1517697117)

[1.2.1 Node.js & MySQL 4](#_Toc2052868261)

[1.3 Intended Audience and Document Overview (Wesley) 5](#_Toc437742157)

[1.4 References and Acknowledgments (Wesley) 5](#_Toc2096863313)

[2 Overall Description 6](#_Toc2129097875)

[2.1 Product Overview (MR) 7](#_Toc914209743)

[2.2 Product Functionality (MR) 7](#_Toc1984704696)

[2.3 Assumptions and Dependencies (MR) 8](#_Toc82456796)

[3 Specific Requirements 9](#_Toc454920409)

[3.1 User Interface Requirements (Wesley) 10](#_Toc1578844389)

[3.1.1 Manager View 10](#_Toc310686534)

[3.1.2 Staff View 11](#_Toc1281104732)

[3.2 Functional Requirements (Wesley + MR) 11](#_Toc1790553823)

[3.3 Use Case Model (Jon) 12](#_Toc1387684130)

[3.4 Non-functional Requirements (Keefe + Jon) 13](#_Toc1848757683)

[3.4.1 Performance Requirements (Keefe + Jon) 13](#_Toc1293087474)

[3.4.2 Safety and Security Requirements (Keefe + Jon) 14](#_Toc598894137)

[3.5 Other Requirements 14](#_Toc585281496)

[4 Project Estimation and Plan 15](#_Toc1800123935)

[4.1 Software Estimation (Keefe + Jon) 16](#_Toc1249010227)

[4.2 Project Management (MR) 17](#_Toc87969121)

[5 Individual Members Task Reflections 17](#_Toc75421418)

[6 Appendix A – Use Case Descriptions 18](#_Toc419500105)

[7 Appendix B – Data Dictionary 28](#_Toc561964515)

# Introduction [KEEFE]

*<Provide a brief introduction to your project, the intended purpose of the project, the impact of the project, and a brief overview of what the reader will find in this section. You can use the concise project description you have created to identify your nouns for your problem statement as a starting point and expand from there.>*

This report details our team’s project, Air Manager. Air Manager is a workload management system delivered through a web application, and it provides a visual and interactive platform for staff to manage their availability and assignments. This application aims to tackle the problem of imbalanced staff workloads and poor work-life balance, ultimately making the work environment more conducive and attractive for staff. This report will cover Air Manager’s overview, specific requirements, and the project estimation and plan.

## Product Scope (Keefe)

<Provide a short description of the software scope, boundaries, and its purpose, including relevant benefits, objectives, and goals.>

Air Manager will provide an interactive and visual way for staff and employees to view staff working hours and job assignments. Staff will be able to indicate their availability, preferences for assignments, assignments which they will be unable to fulfil in advance, and accept or reject assignments given to them. Managers will be able to view employees with the least assigned workload, and give all staff job assignments up to a week in advance. IT administrators can add new staff and managers to the system.

Air Manager's objective is to improve the work-life balance of the staff, which will provide a fairer and more conducive environment for employees to work in. Staff will have more even workloads, more flexibility with their availability, and will receive assignments they prefer more often. Overall, staff will have more welfare and benefits.

## Related Background Literature (Wesley)

Managing work schedules has always been one of the key issues in human resource management. Extra care must be taken to ensure that all employees get enough rest to prevent exhaustion and there is equal distribution of work to each employee to ensure fairness. One of the key issues faced by flight crew was discovered to be the lack of rest for flight crew. As aviation is an industry that is functional 24/7, it is common for flight crew to experience exhaustion. A study conducted in 2010 has shown that over a 3–4-week period, cabin crew personnel typically get an average of 6.3 hours of sleep on off-days and 5.7 hours of sleep on workdays [1].

Thus, it is important to ensure that there is a simple and organized way to allocate staff to their corresponding jobs, to ensure fairness and prevent exhaustion of employees. Cost is also a factor, as poor work scheduling can lead to major staff turnovers or cancellation of flights [2], which can incur significant losses for the company. In the past, before the 2000s, airline crew schedules were usually done on pen and paper. The entire process utilizes a small team of less than 10 individuals, and the planning process takes an excruciating amount of time [3]. It was only until the evolution of the internet that management systems begun to move towards the digital space where there was significant improvement to the overall process.

Diagram

Description automatically generated

One such management system was called the Jeppesen Crew Management system, which was established in 2006. Up to 75% of European aircrews are currently using this system to allocate their work schedules [3]. The entire process consists of four main stages, where managers would consider their staffs’ leaves (availability) and jobs to plan schedules. Crew would also be allowed to bid for certain jobs that they desire, and exchanges can be made, allowing for greater freedom of choice.

In 2016, Singapore Airlines officially started using a system called AirCentre Crew Manager to conduct their work scheduling [4]. The program was created by an American company known as Sabre, and it caters to over 100 airlines internationally [3]. The system is a cutting-edge technology that uses algorithms to determine schedules as well as contingencies when certain crew members become sick, or if an emergency happens. It will also monitor the schedules of crew members to ensure that all crew members are allocated according to laws and regulations set by the company or organizations [5].

With all these factors in mind, the final product that is delivered by the end of this project will assist in solving all the issues faced during the work allocation process. It is crucial for the system to be easy to learn and use as any mistake can lead to significant consequences as mentioned earlier. It is also essential for the system to be fast and stable to accommodate the simultaneous usage of a large number of employees, as well as to consider the possibility of having unforeseen events which requires an immediate response to the situation. Therefore, the team has decided to utilize the following technologies for the different components of the web application, displayed in the table below.

|  |  |
| --- | --- |
| Frontend | React.js |
| Backend | Express.js |
| Database | MySQL |

### Node.js & MySQL

React and Express are libraries derived from the widely known Javascript environment known as Node.js, which was written in JavaScript. Node.js has been known to be a scalable, fast, and lightweight tool for developing web applications [6]. Additionally, due to its popularity, there is excellent documentation available as well as a large community of developers, where developers can ask for help from more experienced developers.

React works by splitting up the code into “components”. Each component will serve a specific purpose on the website, which serve the purpose of being reusable blocks that can act independently of each other [7]. React is also faster when compared to other libraries due to its minimalistic design [8]. Therefore, due to the speed and simplicity of React, it has been chosen for the frontend of the web application.

One of the largest benefits to using Express is speed and stability. Express has been determined to be able to handle 11,202 requests per second, as well as being two times faster than Java in a test conducted by PayPal [9]. It is also beneficial to use similar technologies for the backend and frontend for ease of maintenance and development. There are several instances of web applications having mixed frameworks such as some using Django for the backend and React for the frontend, but additional measures must be taken to ensure compatibility. Thus, the usage of React and Express for the backend and frontend, which are both Node.js libraries, is an excellent choice of technologies used for the website.

Finally, MySQL will be used for the database. MySQL is one of the most popular database software available and is compatible with an extensive catalogue of technologies. It can be used for both small and large size projects, and it also has a reputation for being easy to use and scalable.

## Intended Audience and Document Overview (Wesley)

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers (In your case it would probably be the “client” and the professor). Describe what the rest of this document contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

This document is intended to be available to only our clients’ and any party that is granted access to this document by our clients. This document will contain the information necessary for future maintenance of the web application as well as the features that will be included in the final product of the website. The use cases for each of the features will also be available here, which will aid in the testing process as well as future training purposes for users of this website.

Wireframe images of the supposed outcome of the website can also be found in section 3.1. You may refer to the table below for the various sections for the intended audience. The sections have been included with hyperlinks for easy access.

|  |  |
| --- | --- |
| Intended audience | Sections |
| Our clients (Admins) | [**2**](#_Overall_Description) – [2.1](#_Product_Overview_(MR)), [2.2](#_Product_Functionality_(MR)), [2.3](#_Assumptions_and_Dependencies)  [**3**](#_Specific_Requirements) – [3.1](#_User_Interface_Requirements), [3.3](#_Use_Case_Model)  [**4**](#_Project_Estimation_and) – [4.1](#_Software_Estimation), [4.2](#_Project_Management) |
| Users (Managers & Staff) | [**2**](#_Overall_Description) – [2.1](#_Product_Overview_(MR)), [2.2](#_Product_Functionality_(MR))  [**3**](#_Specific_Requirements) – [3.3](#_Use_Case_Model)  [**6**](#_Appendix_A_–) |
| Future developers | [**2**](#_Overall_Description) – [2.1](#_Product_Overview_(MR)), [2.2](#_Product_Functionality_(MR)), [2.3](#_Assumptions_and_Dependencies)  [**3**](#_Specific_Requirements) – [3.3](#_Use_Case_Model)  [**4**](#_Project_Estimation_and) – [4.1](#_Software_Estimation), [4.2](#_Project_Management) |

## References and Acknowledgments (Wesley)

<List any other documents or Web addresses to which this document refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. >

Can find references to microsoft outlook’s calendar

Maybe find some random references to requirement specification

[1] <https://libraryonline.erau.edu/online-full-text/faa-aviation-medicine-reports/AM10-22.pdf>

[2] https://www.kxly.com/sort-of-last-minute-airline-pilot-fatigue-leads-to-flight-delays-cancellations/

[3] https://www.altexsoft.com/blog/crew-management-system-in-airlines/

[4] <https://www.sabre.com/insights/releases/singapore-airlines-to-introduce-real-time-holistic-crew-management-system-from-sabre/>

[5] https://www.sabre.com/page/as-product-dictionary/opr-plt-crew-management/overlay/opr-cw-aircrews/

[6] <https://fireart.studio/blog/why-node-js-is-still-a-good-choice-for-your-startup-in-2020/>

[7] <https://medium.com/@alex_lobera/introduction-to-react-3000e9cbcd26>

[8] https://jelvix.com/blog/is-react-js-fast#:~:text=Google%20is%20an%20example%20of,say%20that%20React%20is%20faster.

[9] https://www.simform.com/blog/node-js-vs-express/#:~:text=Pros%20of%20Express,both%20synchronous%20and%20asynchronous%20codes.

# Overall Description

## Product Overview (MR)

<Describe the context and origin of the product being specified in this document. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the document defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, macke sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.>

TO DO: Provide at least one paragraph describing product perspective. Provide your System’s Architecture diagram to illustrate how your product interacts with the environment and in what context it is being used. This is not a formal diagram, but rather something that is used to illustrate the product at a high level.

Workload management has always been a challenging task for managerial personnel and would be even more so due to increasing desire for work-life balance amongst employees. With that in mind, companies are turning towards technological tools to provide ease in planning for work schedules while providing employees with the ability to achieve their desired work-life balance through choosing their desired working hours.

Our new management product, Air Manager, is a full stack web-based application, which includes a frontend interface for staffs and managers to perform their relevant duties (stated in Section 2.2). It also consists of a backend framework to control the logics of the web application, and to retrieve all relevant data for the dashboard from the database. The interaction between users and product can be modelled as shown in the following diagram.

Diagram

Description automatically generated

Figure 1: System Architecture Diagram

## Product Functionality (MR)

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high-level summary is needed here. These can be at the level given in the project description.>

Provide a bulleted list of all the major functions of the system

**PF1:** The product shall …

**PF2:** The product shall …

The product is expected to allow the different users of the company to perform their relevant duties via the frontend interface of the product. Such duties include but are not limited to,

IT-Admin Functionalities:

1. The product shall allow admins to add new users (staffs and managers) to the system

Manager-Role Functionalities:

1. The product shall allow managers to view all staff workload and other essential details (availability, job preference) on the landing page
2. The product shall allow managers to view all flights on the landing page
3. The product shall allow managers to assign staff to flights for the upcoming week

Staff-Role Functionalities:

1. The product shall allow staff to view their weekly assigned jobs
2. The product shall allow staff to add and edit their upcoming weeks’ availabilities
3. The product shall allow staff to indicate job preference for the upcoming week for managers’ consideration
4. The product shall allow staff to reject jobs assigned to them for the upcoming week for manager to edit their schedules.

## Assumptions and Dependencies (MR)

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.>

TO DO: Provide a short list of some major assumptions that might significantly affect your design.

For the product to work as required, there were a few considerations and factors that might affect the overall requirements and result of the product. In this section we will discuss how these factors may impact the requirements and the assumptions or dependencies we have of these factors to ensure the product meets the expectations of the clients.

#### Fault Tolerance

For the product to perform its functionalities consistently, we need to assume that there will be no server faults, and thus no disruption to the service. However since this realistically unachievable, thus in order to maintain a consistent uptime of service, fault tolerance measures such as having a backup backend server and RAID (redundant array of independent disks) systems for the databases are required to be implemented. The importance and implementation details will be explained further in Section 3.4.1.

#### Server’s Accessibility through the Internet

Depending on client’s decision, the product may only be used while in the company’s own network, which may require staffs to be physically there. Or they might allow having an internet facing interface where their staffs are allowed to connect to the product without being physically restricted by their location. However, under the assumption that there might be events where staffs are on a flight bound for other countries, which is out of their company’s network, they might not be able to attend to urgent requests on the product platform. Thus, we can assume that accessibility to the product’s service through the internet is required.

#### Server Hosting Services

Depending on various situations, clients might be interested in personally taking charge on hosting the product’s web servers on company’s servers instead of a cloud provider such as Microsoft Azure, or Google Cloud. This may lead to a change in requirements from hosting the web service on Microsoft Azure as initially planned and setting it up on the company’s end. In addition, complimentary issues in such decision involves registering a public IP and a domain name for the web service from an ISP, incurring additional cost and requirements.

#### Coding Language and Framework Community Support and Future Maintenance

Lastly, like many other software, our product relies on the consistent update and community support from the coding languages and relevant frameworks that are used in it. Stable releases from the community handling the language and framework lowers the chances of the product malfunctioning which may lead to staff’s inability to perform duties and security concerns as well. Hence, the effectiveness of the product is dependent on the future maintenance of hired IT administrators, which would be required to keep the product up to date and to review the web service’s code to align with changes in business logic or scalability.

# Specific Requirements

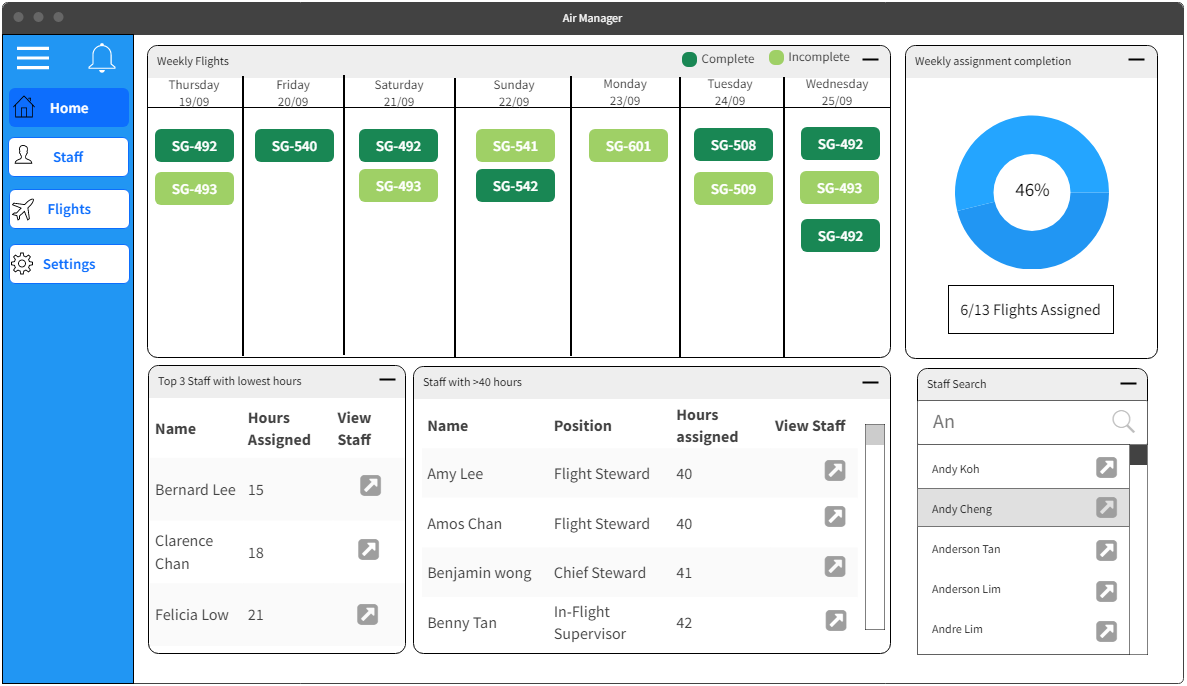
## User Interface Requirements (Wesley)

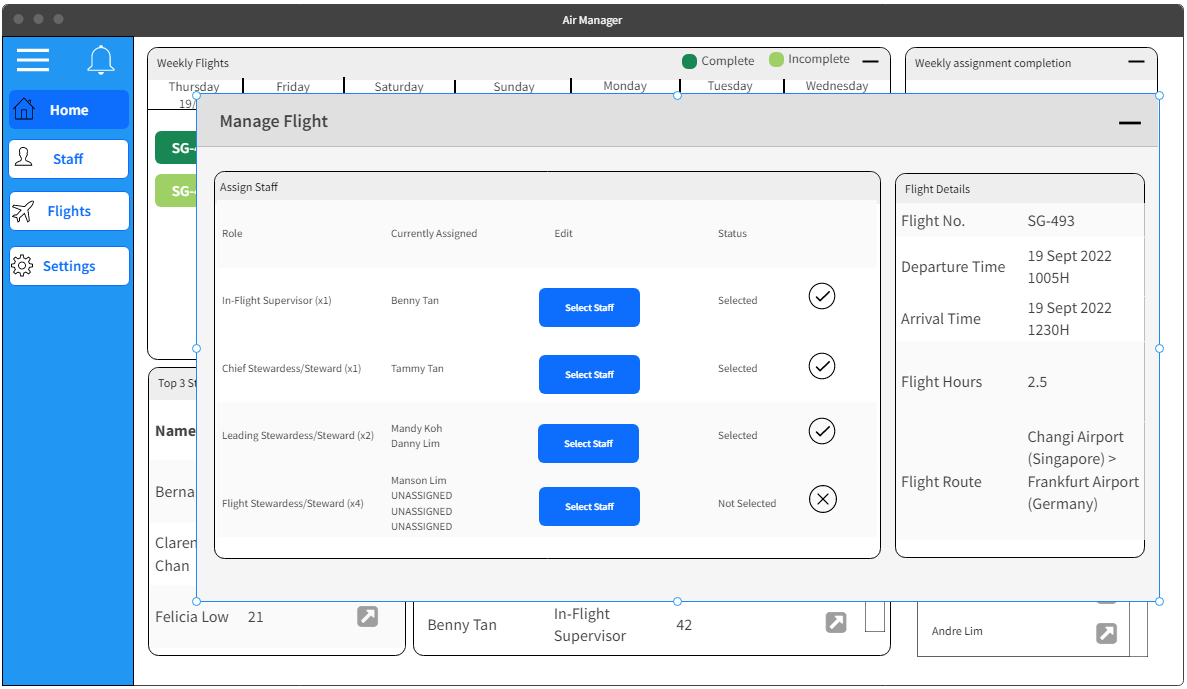
The user interface will be different for each user depending on their role. Administrators, managers, and staff will each have their own individual user interface, with different dashboards and tabs for specific functions in accordance with their roles. Thus, it is essential that the User Interface is easy to understand and navigate. The team has produced website wireframes for each of the various interfaces accessible by Staff and Managers, allowing for better visualization of the outcome of the web application. Staff and managers will have different tabs and dashboards altogether, as they are only permitted to perform specific actions.

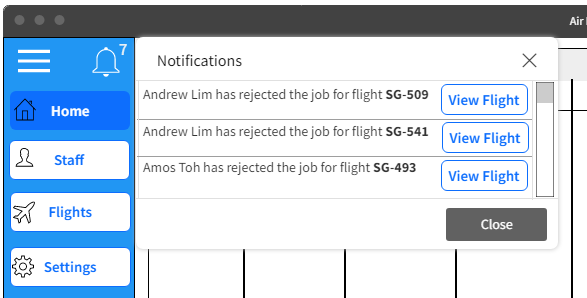
As an additional security measure, the login screen must also contain a 2FA (2-Factor Authentication) feature. When a user accesses the login page and enters in their email and password, an email will be sent to their email account, containing the 6-digit 2FA code. The user will be required to enter in this code to access their account.

### Manager View

Once a manager logs in to the website and the system recognizes that the user is a manager, they will be directed straight to the manager dashboard. The dashboard for the manager consists of 5 main windows, each containing key information/features that will be included on the website.

1. (Top left) **Timeline:** Contains the flights for the week. It is colour-coded based on the status of the job assignment. Once all jobs on a flight are assigned, it will appear as green to indicate completion of the job assignment for the flight.
2. (Top right) **Assignment Status**: Contains a pie chart shows the number of flights that have been assigned for the week. Once all flights have been assigned, the pie chart will appear as 100%, which allows the manager to know that the weekly allocation of jobs is complete.
3. (Bottom left) **Top 3 lowest hours staff**: Contains a table showing the 3 employees with the lowest number of working hours for the week. Buttons have been included to allow the manager to view and manage the specific staff.
4. (Bottom middle) **Staff with more than 40 hours:** Contains a table displaying all staff who have over 40 hours of work assigned to them.
5. (Bottom right) **Staff search:** A search bar with a table to search for certain staff.

To manage a specific flight, managers will be able to click on the specific flights shown in the timeline in the dashboard. This will create a popup panel to view all the currently assigned staff as well as the flight information. Managers will also be able to view and search for flights by navigating to the “Flights” tab which will be included in the menu/navigation bar.

The flight management page can also be accessed through the notifications menu. The notifications can be accessed by clicking on the notification icon in the manager’s menu, which will create a list of alerts meant for the manager. This will include all alert messages that are meant for the manager, such as when a staff rejects a certain job.

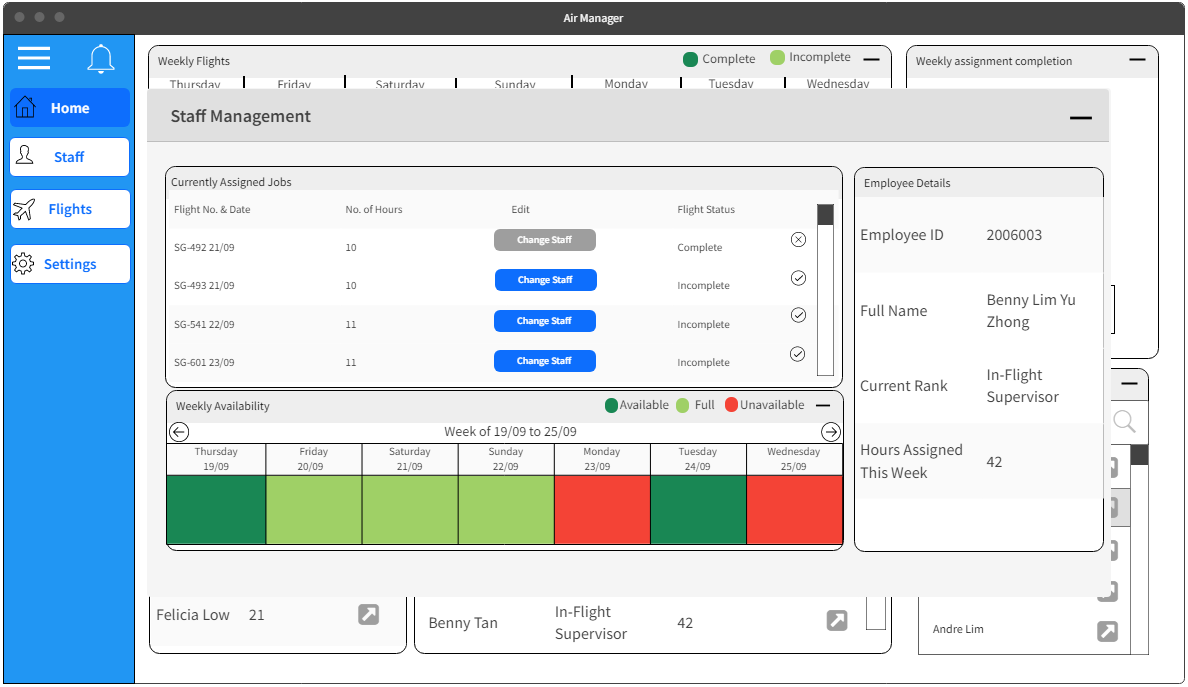
When viewing the flight management window for a specific flight, managers will be able to assign staff to their respective roles by clicking on the “**Select Staff**” button as shown in Figure 3. This will bring up a list of all available staff that can fulfil that role. For roles that require more than 1 personnel, the amount of selected staff will also be shown to the manager.

Graphical user interface, text, application

Description automatically generated

Similarly to flights, managers will also be able to navigate to the “Staff” page to manage and view all staff. The staff overview will contain the following:

1. A list of all currently assigned jobs
2. The staff’s availability for the week
3. The staff’s details

The features will allow managers to easily view all information pertaining to the staff all at once to assist in the job allocation process.

### Staff View

The staff user interface will also contain a dashboard and additional pages to navigate to for other functions. The dashboard for staff will compose of the following:

1. (Top left) **Timeline:** Contains the jobs that were allocated in the current week to the staff that is logged in.
2. (Top right) **Monthly Workload:** Contains a bar chart showing the overall workload of the current staff, in the form of work hours over the past four weeks.
3. (Bottom) **Upcoming jobs:** Contains a list of jobs that are available in the following week. The staff will be able to indicate preference for specific jobs.

Staff will also be able to reject jobs that were assigned to them. By simply clicking on the flight in the dashboard under their weekly schedule, a window will appear displaying the flight information as well as the staff that was assigned to the job. A “reject job” button will be present which will allow staff to reject the job assignment. However, before the request is submitted, a popup will appear, informing the staff user that they must have discussed with their manager first before rejecting the job.

Graphical user interface, application

Description automatically generated

Lastly, one of the most important features that must be present is the declaration of availability for staff. This feature will be accessed by navigating to the “**Availability**” page. The staff user will be able to indicate their availability over the next four weeks by clicking on the selected box and selecting either “**Available**” or “**Unavailable**”. The upcoming week will always be highlighted and placed at the top of the screen, as shown in the figure below.

Calendar

Description automatically generated

## Functional Requirements (Wesley + MR)

**FR1:** The system shall …

**FR2:** The system shall…

…

ADMIN:

FR1: IT administrators shall be able to create new staff and manager accounts (?)

FR2: IT administrators shall be able to edit staff and manager information

FR3: IT administrators shall be able to delete staff and manager accounts

MANAGER:

FR1: Managers shall be able to view all flights.

FR2: Managers shall be able to assign staff to flights.

FR3: Managers shall be able to assign staff to their respective roles for each individual flight.

FR4: Managers shall be able to view any staff’s availability up to one month in advance.

FR5: Managers shall be able to start planning and allocating workload on Thursdays.

FR6: Managers shall be able to view the allocated work schedules of all staff on the dashboard.

FR7: Managers shall be able to view up to three staff’s availability and information at a time

FR8: The staff availability, the workload assigned, staff’s job preference, and

availabilities for the week shall be shown when viewing each individual staff as a manager

FR9: Managers shall be able to view the top three staff with the lowest workload on the dashboard

FR10: Managers shall be able to view the staff with the over 40 hours allocated for the week on the dashboard

STAFF (Steward/Stewardess):

FR1: Staff shall be able to view their weekly assigned jobs (done)

FR2: Staff shall be able to view all available flights (done)

FR3: Staff shall be able to view their overall workload (? – in hours?) for the current month on their landing page

FR4: Staff shall add and edit their availabilities up to 5 weeks ahead of time. (done)

FR5: Stall shall be able to indicate their job preference for the week (? – do we allow to indicate preference for all their 5 weeks advance availabilities stated in FR4?) (done)

FR6: Staff shall be able to reject jobs assigned to them by the managers (done)

FR7: Staff shall be warned to discuss assigned jobs that they do not agree with the manager before proceeding with the rejection (done)

FR8: Staff shall be able to allocate their weekly availability starting from Thursday till Wednesday of the next week (done)

FR9: Staff shall be alerted and warned if they have yet to state availability for the upcoming week by Wednesday

FR10: Staff shall be able to able to still make request but case-by-case basis (?) - ask prof

FR11: Users shall receive a 2FA (2 Factor Authentication) code from their email when attempting to login to the website.

FR12: Users shall be required to enter 2FA code when logging in

## Use Case Model (Jon)

<In your Use Case model and explain your use case diagram to aid understanding of your concepts. You can place your use case descriptions in Appendix A. Please inform your readers where to find your use case description. >

## Non-functional Requirements (Keefe + Jon)

*<Include any generic non-functional requirements. Please include a summary of the set of generic non-functional requirements>*

**NFR1**: The system shall provide the feature to present content in multi-lingual.

**NFR2**: The system shall …

NFR1: The system shall support mobile devices -> how to specify this sia

NFR2: The system shall be accessible anytime of the day

NFR3: Staff shall be able to use all the system functions after four hours of training

NFR4: The system database shall meet HIPAA requirements

NFR5: The system shall meet Web Content Accessibility Guidelines 2.1

NFR6: The date format in the system shall be as follows: dd/mm/yyyy

NFR7: The system shall be able to run on Google Chrome ver X, Mozilla Firefox ver X, and Microsoft Edge ver X

NFR8: Managers shall be able to use all the system functions after six hours of training

NFR9: IT administrators shall be able to use all system functions after eight hours of training

To summarise the non-functional requirements, the system shall be accessible at any time of the day, from any of the specified browsers, on the staffs’ mobile devices. The system shall meet the HIPAA and Web Content Accessibility Guidelines 2.1. All users shall be able to use the system after four to eight hours of training depending on their roles.

### Performance Requirements (Keefe + Jon)

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features. >

TODO: Provide performance requirements based on the information you collected from the client/professor. For example, you can say:

**NFR3**: The secondary heater will be engaged if the desired temperature is not reached within 10 seconds

**NFR4**: …

# This is stuff copied from the non-functional requirements

NFR10: The system shall update the total hours assigned to any staff within 5 seconds whenever a change is made

NFR11: The system should allow for at least x number of users concurrently while maintaining optimal performance

NFR12: The system shall be able to store 10,000 tasks at once

NFR13: Webpages in the system shall load within 5 seconds

NFR14: The system shall be able to restart within 10 minutes

### Safety and Security Requirements (Keefe + Jon)

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.>

TODO: Provide safety/security requirements based on your interview with the client - again you may need to be somewhat creative here. At the least, you should have some security for the mobile connection.

**NFR5**: Data transmitted from and received are always encrypted with AES-128.

**NFR6**: All approving roles must use a different factor of authentication from the user authentications (on top of 2FA) for approval processes.

# This is stuff copied from the non-functional requirements

NFR15: Users shall be required to use a complex password, with a minimum length of 8, a number, an uppercase and lowercase letter, and a special character

NFR16: The system shall not be susceptible against the top 10 web application vulnerabilities on OWASP

NFR17: Users should be automatically logged out after 15 minutes of inactivity

NFR18: The system shall only allow a user to be logged in on one device at a time

NFR19: The system database shall be backed up every week on Thursday

NFR20: The system shall prompt new users to change their initially assigned login password immediately after the first login

NFR21: The system shall prompt users to change their passwords every month and prevent users from reusing the previous 3 passwords

NFR22: The system shall use HTTPS to encrypt all traffic

## Other Requirements

<This section is **Optional.** Define any other requirements not covered elsewhere in the document. This might include database requirements, internationalization requirements, legal requirements, HBRA, PDPA, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

In reference to the following requirement:

2. The manager should be able to **visualise the staff workload** immediately on the landing page.

What does staff workload refer to? What should the manager be able to see immediately upon logging in? For example, could there be a scrolling bar to view all staff’s hours for the week? (Wesley)

How many people gonna use the system – for non-functional requirement (JON)

Admin FR:

Ask if Administrator is meant to do every Requirement/Use case the Manager can perform

Manager FR

By when (specific day) does managers need to announce staff’s upcoming assigned flights? Since staff can reject their jobs, need to ensure enough time to accommodate for the change in assigned flights. And maybe need exact cut off dates to state disproval of assigned jobs.

From Requirements

“If employees miss the weekly deadline, requests would be dealt with on a case-by-case basis” What must be done on both the manager and staff side? Does the manager need a system to deal with each case, and does the system need to warn the staff about missing the deadline?

“Hence, all employee’s availabilities must be informed in the system every Wednesday to be considered in the planning”

What

What does this mean “It will help if you consider that the manager will make their assignments such that staff should not need to travel between routes and planes do not stay on the ground for more than three hours.”

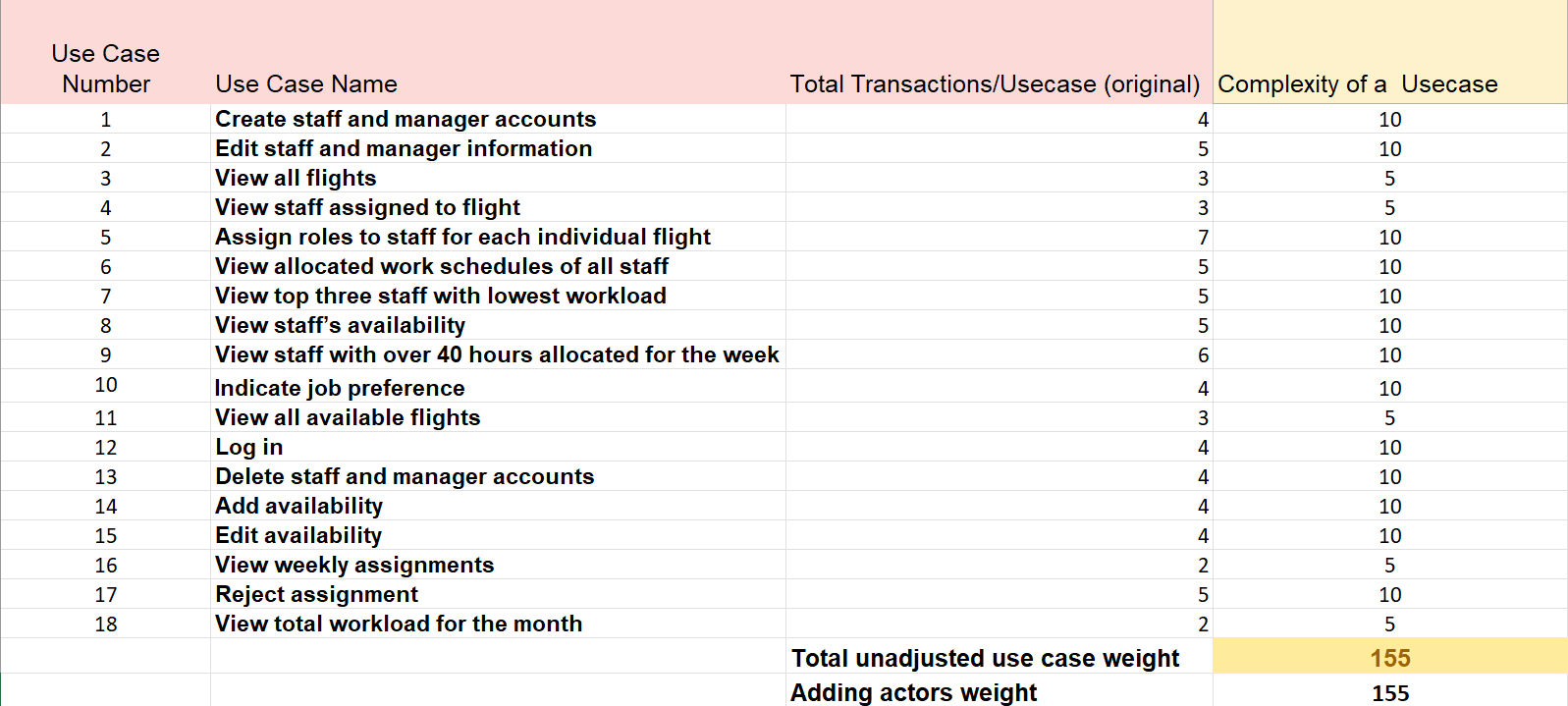
# Project Estimation and Plan

*<Describe* the resources required and overall time required for the project>

## Software Estimation (Keefe + Jon)

Unadjusted use case weight

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Number | Use Case Name | Total Transactions/Usecase (original) | Complexity of a Usecase |
| 1 | **Create staff and manager accounts** | 4 | 10 |
| 2 | **Edit staff and manager information** | 5 | 10 |
| 3 | **View all flights** | 3 | 5 |
| 4 | **View staff assigned to flight** | 3 | 5 |
| 5 | **Assign roles to staff for each individual flight** | 7 | 10 |
| 6 | **View allocated work schedules of all staff** | 5 | 10 |
| 7 | **View top three staff with lowest workload** | 5 | 10 |
| 8 | **View staff’s availability** | 5 | 10 |
| 9 | **View staff with over 40 hours allocated for the week** | 6 | 10 |
| 10 | **Indicate job preference** | 4 | 10 |
| 11 | **View all available flights** | 3 | 5 |
| 12 | **Log in** | 6 | 10 |
| 13 | **Delete staff and manager accounts** | 4 | 10 |
| 14 | **Add availability** | 4 | 10 |
| 15 | **Edit availability** | 4 | 10 |
| 16 | **View weekly assignments** | 2 | 5 |
| 17 | **Reject assignment** | 5 | 10 |
| 18 | **View total workload for the month** | 2 | 5 |
|  |  | **Total unadjusted use case weight** | **155** |

Unadjusted actor weight

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | weight | count | product |
| Simple | 1 |  | 0 |
| Medium | 2 |  | 0 |
| Complex | 3 | 3 | 9 |
|  |  | Total | 9 |

Complexity Factors – WCS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Factor | Description | Weight | Assesment | Product |
| T1 | Distributed system | 2 | 1 | 2 |
| T2 | Response time/performance objectives | 1 | 5 | 5 |
| T3 | End-user efficiency | 1 | 3 | 3 |
| T4 | Internal processing complexity | 1 | 2 | 2 |
| T5 | Code reusability | 1 | 3 | 3 |
| T6 | Easy to install | 0.5 | 1 | 0.5 |
| T7 | Easy to use | 0.5 | 3 | 1.5 |
| T8 | Portability to other platforms | 2 | 1 | 2 |
| T9 | System maintenance | 1 | 4 | 4 |
| T10 | Concurrent/parallel processing | 1 | 3 | 3 |
| T11 | Security features | 1 | 4 | 4 |
| T12 | Access for third parties | 1 | 0 | 0 |
| T13 | End user training | 1 | 3 | 3 |
|  |  |  | Total | 33 |

𝑇𝐶𝐹 = 0.6 + 0.01 × 33 = 0.93

Environmental Factors – WCS

|  |  |  |  |
| --- | --- | --- | --- |
| Environment | weight | Assessment | Product |
| Familiar with Development Process | 1.5 | 4 | 6 |
| Part time workers | -1 | 0 | 0 |
| Analyst capability | 0.5 | 2 | 1 |
| Application experience | 0.5 | 4 | 2 |
| Object oriented experience | 1 | 3 | 3 |
| Motivation | 1 | 5 | 5 |
| Difficult programming language | -1 | 1 | -1 |
| Stable requirements | 2 | 3 | 6 |
|  |  | Total | 22 |

𝐸𝐹 = 1.4 + (−0.03 × 22) = 0.74

UCP = (155 + 9) \* 0.93 \* 0.74 = 113 (round up to integer)

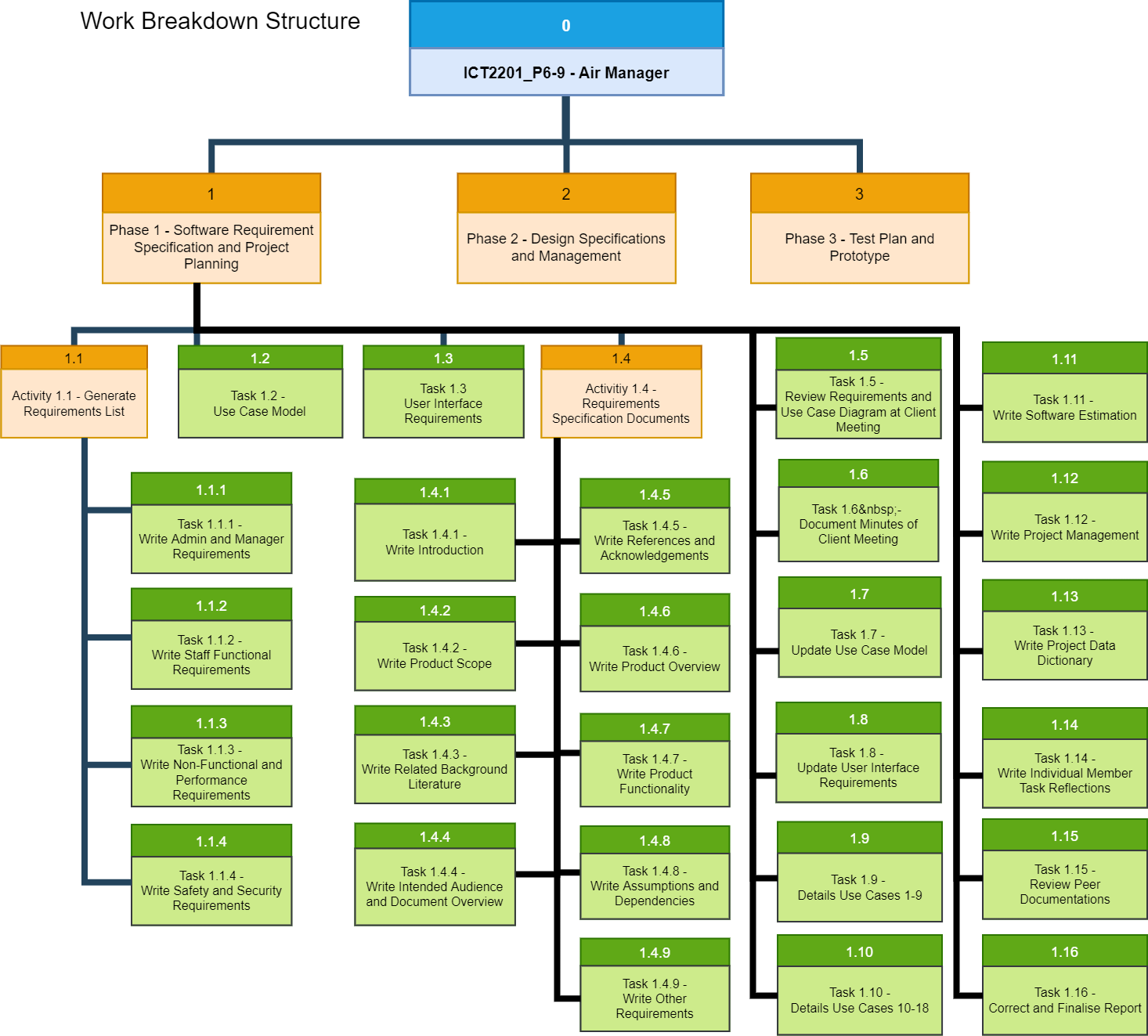
Effort estimation using UCP is 113\*15 = 1695 to 113 \* 40 = 3390 hours.

## Project Management (MR)

*<Include your Gant Chart with WBS number and any other required charts like Burndown charts etc. here*>

The project will be handled in three phases:

1. Software Requirements Specification and Project Planning
2. Design Specifications and Management
3. Test Plan and Prototype



Graphical user interface, application, table

Description automatically generatedWhile the individual tasks for each phase are managed and track by a Gantt Chart and the project’s Github Project Board

# Individual Members Task Reflections

*< Each member is required to write no more than 2 paragraphs of what they have done for milestone one, what they have learned, how they were to do it differently if they are given another chance to do it again, and as a team what can be done better. >*

# Appendix A – Use Case Descriptions

|  |  |
| --- | --- |
| Use Case ID: | UC01 |
| Use Case Name: | **Create staff and manager accounts** |
| Description: | Administrator creates either staff or manager accounts |
| Primary Actor: | Administrator |
| Preconditions: | Administrator must be logged in |
| Postconditions: | Main: Staff or manager account is created  2a, 3a, 6a: Staff or manager account is not created |
| Main Success Scenarios: | 1. Administrator chooses to create either staff or manager accounts  2. Administrator enters the account username  3. Administrator enters the account password  4. Administrator enters the employee information  5. Administrator submits the account creation form  6. System displays message stating that the account is created successfully |
| Alternative Scenarios: | 2a. Administrator enters an account username that already exists  2a1. System displays an error message, notifying the administrator that an account with that username already exists  3a. Administrator enters a password that does not meet the security requirements  3a1. System displays an error message, notifying the administrator that the password does not meet the required complexity  6a. System is unable to create account  6a1. System displays an error message containing the details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC02 |
| Use Case Name: | **Edit staff and manager information** |
| Description: | Administrator can edit information of existing staff or managers |
| Primary Actor: | Administrator |
| Preconditions: | Administrator must be logged in and staff or manager must have an account in the system |
| Postconditions: | Main: Staff or manager information is edited  2a, 3a, 5a: Staff or manager information is not edited |
| Main Success Scenarios: | 1. Administrator selects staff or manager from dropdown list  2. System shows staff or manager information  3. Administrator edits staff or manager information  4. Administrator submits the edited account information form  5. System displays a message stating that the account information has been edited successfully |
| Alternative Scenarios: | 2a. System is unable to show staff or manager information  2a1. System displays an error message containing the details of the error  3a. Administrator enters invalid characters or empty account information  3a1. System displays error message, notifying the administrator that the information is invalid  5a. System is unable to edit account information  5a1. System displays an error message containing the details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC03 |
| Use Case Name: | **View all flights** |
| Description: | Manager can view all the flights of the airlines |
| Primary Actor: | Manager |
| Preconditions: | Manager is logged into the system |
| Postconditions: | Main: Manager can view all flights in the system  2b: Manager is unable to view any flights |
| Main Success Scenarios: | 1. Manager chooses to view all flights in the system  2. System displays all flights |
| Alternative Scenarios: | 2a. There are no flights recorded in the system  2a1. System displays an empty table  2b. System is unable to retrieve flights  2b1. System displays error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC04 |
| Use Case Name: | **View staff assigned to flight** |
| Description: | Manager can view all the staff assigned to a particular flight |
| Primary Actor: | Manager |
| Preconditions: | Manager is logged into the system  There are staff assigned to the flight |
| Postconditions: | Main: Manager can view all staff assigned to the flight  2b: Manager is unable to view any staff assigned to the flight |
| Main Success Scenarios: | 1. Manager clicks on a particular flight  2. System displays all staff assigned to the flight |
| Alternative Scenarios: | 2a. There is no staff assigned to the flight  2a1. System displays an empty table  2a2. System displays a list of staff that is available to be assigned to the flight  2b. System is unable to retrieve all staff assigned to the flight  2b1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC05 |
| Use Case Name: | **Assign roles to staff for each individual flight** |
| Description: | Manager can assign roles to staff for each individual flight |
| Primary Actor: | Manager |
| Preconditions: | Manager is logged into the system  Manager has selected a flight to assign staff to  Staff has stated their availability for the day of the flight |
| Postconditions: | Main: Manager assigns staff to a job in the flight  1a, 3a, 3b: Manager is not able to assign any staff to the job  5a: Manager is not able to assign selected staff to the job |
| Main Success Scenarios: | 1. System displays a table of jobs required for the flight, along with the flight information  2. Manager selects a job  3. System displays up to three staff available for the job, with staff preferring the job displayed at the top  4. Manager selects a staff for the job  5. System saves the staff assigned for the job  6. System displays table of jobs required for the flight, with the new staff assigned reflected in the table |
| Alternative Scenarios: | 1a. System is unable to display the jobs required for the flight and the flight information  1a1. System displays an error message containing details of the error  3a. There is no staff available for the job yet  3a1. System displays an empty table to reflect that no staff is available for the job  3b. System is unable to display staff available for the job  3b1. System displays an error message containing details of the error  5a. System is unable to save the staff assigned for the job  5a1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC06 |
| Use Case Name: | **View allocated work schedules of all staff** |
| Description: | Manager can view allocated work schedules of all staff |
| Primary Actor: | Manager |
| Preconditions: | Manager is logged into the system  There are staff with allocated work schedules |
| Postconditions: | Main: Manager can view allocated work schedules of selected staff  2a, 4b: Manager cannot view allocated work schedules of selected staff |
| Main Success Scenarios: | 1. Manager enters the staff page  2. System displays all staff in a table  3. Manager chooses staff to view their work schedule  4. System displays the work schedule of the selected staff |
| Alternative Scenarios: | 2a. System is unable to display all the staff  2a1. System displays an error message containing details of the error  4a. The staff has not been allocated any jobs  4a1. System displays an empty table for the staff’s work schedule  4b. System is unable to display the work schedule of the selected staff  4b1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC07 |
| Use Case Name: | **View top three staff with lowest workload** |
| Description: | Manager can view top three staff with lowest workload in the dashboard |
| Primary Actor: | Manager |
| Preconditions: | Manager must be logged in |
| Postconditions: | Main: Manager can view the top three staff with the lowest workload and what jobs are assigned to them  2a, 4b: Manager cannot view the top three staff with the lowest workload |
| Main Success Scenarios: | 1. Manager enters the dashboard  2. System displays the top three staff with lowest workload  3. Manager clicks on the staff  4. System displays the staff information and the jobs assigned to the staff |
| Alternative Scenarios: | 2a. System is not able to display the top three staff with lowest workload  2a1. System displays an error message containing details of the error  4a. There are no jobs assigned to the staff  4a1. System displays staff information alongside an empty table for the jobs assigned to the staff  4b. System is unable to display the staff information, or the jobs assigned to the staff  4b1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC08 |
| Use Case Name: | **View staff’s availability** |
| Description: | Managers can view staff’s availability on a weekly basis |
| Primary Actor: | Manager |
| Preconditions: | Manager must be logged in  Staff selected should have indicated his or her availability |
| Postconditions: | Main: Manager can view which days the staff is available  2a, 4b. Manager cannot view the staff availability |
| Main Success Scenarios: | 1. Manager enters the staff page  2. System displays all staff in a table  3. Manager selects staff to view their availability  4. System displays selected staff’s availability on a week-by-week basis |
| Alternative Scenarios: | 2a. System is unable to display all the staff  2a1. System displays an error message containing details of the error  4a. Selected staff has no indicated his or her availability  4a1. System displays an empty table for the staff’s availability  4b. System is unable to display the staff’s availability  4b1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC09 |
| Use Case Name: | **View staff with over 40 hours allocated for the week** |
| Description: | Managers can view staff with over 40 hours allocated for the week in the dashboard |
| Primary Actor: | Manager |
| Preconditions: | Manager must be logged in  There must be staff with over 40 hours allocated to them |
| Postconditions: | Main: Manager can view staff with over 40 hours allocated for the week  2b, 4a, 4b: Manager cannot view staff with over 40 hours allocated for the week |
| Main Success Scenarios: | 1. Manager enters the dashboard  2. System displays staff with over 40 hours allocated for the week  3. Manager clicks on the staff  4. System displays the staff information and the jobs assigned to the staff |
| Alternative Scenarios: | 2a. There are no staff with over 40 hours allocated for the week  2a1. System displays an empty table for staff with over 40 hours allocated for the week  2b. System is not able to staff with over 40 hours allocated for the week  2b1. System displays an error message containing details of the error  4a. There are no jobs assigned to the staff  4a1. System displays staff information alongside an empty table for the jobs assigned to the staff  4b. System is unable to display the staff information, or the jobs assigned to the staff  4b1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC10 |
| Use Case Name: | **Indicate job preference** |
| Description: | Staff views the available flights and chooses the flights they would prefer to be assigned. |
| Primary Actor: | Staff |
| Preconditions: | Staff is logged into system |
| Postconditions: | Main: Staff has one or more preferred flights saved in the system.  2a, 5a: Staff does not add any preference. |
| Main Success Scenarios: | 1. Staff chooses to select their preferred flights  2. System displays a list of available flight assignments  3. Staff selects one or more preferred flights  4. Staff submits preferences  5. System saves the staff’s preferences |
| Alternative Scenarios: | 2a. No flights available  2a1. System shows empty list.  5a. System is unable to save the staff’s preferences  5a1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC11 |
| Use Case Name: | **View all available flights** |
| Description: | Staff can view all the available flights that have not been assigned. |
| Primary Actor: | Staff |
| Preconditions: | Staff is logged into system |
| Postconditions: | Main: Staff can view the available flights.  2b: Staff is unable to view the available flights |
| Main Success Scenarios: | 1. Staff chooses to view the available flights that have not been fully assigned  2. System displays all available flights. |
| Alternative Scenarios: | 2a. No flights available  2a1. System displays an empty table  2b. System is unable to display all available flights  2b1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC12 |
| Use Case Name: | **Log in** |
| Description: | Users can log into the system with their own credentials |
| Primary Actor: | All users of the system |
| Preconditions: | User has an account in the system |
| Postconditions: | Main: User accesses their respective account.  4a, 4b, 7a: User is denied access and can try to login again |
| Main Success Scenarios: | 1. User wants to access the system  2. System displays the login page and asks for the user’s credentials  3. User submits their login credentials  4. System verifies the credentials to be correct  5. System prompts user for two-factor authentication code  6. User enters two-factor authentication code  7. System verifies the two-factor authentication code  8. System grants the user access to their landing page |
| Alternative Scenarios: | 4a. System does not recognize user’s credentials  4a1. System displays the user an incorrect login warning  4a2. System remains at login page  4b. System is unable to verify login credentials  4b1. System displays an error message containing details of the errors  7a. System is unable to verify two-factor authentication code  7a1. System displays an error message containing details of the errors |

|  |  |
| --- | --- |
| Use Case ID: | UC13 |
| Use Case Name: | **Delete staff and manager accounts** |
| Description: | Administrators can delete staff and manager accounts permanently |
| Primary Actor: | Administrator |
| Preconditions: | Administrator is logged into system |
| Postconditions: | Main: System deletes selected non-administrator account  3a, 4a: Administrator account deletion fails. |
| Main Success Scenarios: | 1. Administrator chooses to remove an account  2. System displays the administrator all account names  3. Administrator selects the non-administrator account to be deleted  4. System deletes selected account |
| Alternative Scenarios: | 3a. Administrator selects administrator account to be deleted  3a1. System prevents deletion of administrator account  3a2. System remains on the same page  3a3. Administrator can choose another account to be deleted  4a. System is unable to delete selected account  4a1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC14 |
| Use Case Name: | **Add availability** |
| Description: | Staff can indicate when they are available for the following week. |
| Primary Actor: | Staff |
| Preconditions: | Staff is logged into system. |
| Postconditions: | Main: Staff has their selected days marked as available in the system.  2a. Staff does not change anything |
| Main Success Scenarios: | 1. Staff wants to indicate when they are available to take assignments  2. System displays a list of dates of the following week  3. Staff selects they days that they are available  4. System saved the selected days for that staff |
| Alternative Scenarios: | 2a. Staff has already selected availability  2a1. System displays their selected availability  2a2. Staff can choose to edit availability instead  4a. System is unable to save the selected days for that staff  4a1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC15 |
| Use Case Name: | **Edit availability** |
| Description: | Staff can edit their availability that was submitted. |
| Primary Actor: | Staff |
| Preconditions: | Staff is logged in and has submitted their availability. |
| Postconditions: | Main: Staff changes their availability schedule.  3a, 4a: Staff does not change their availability schedule. |
| Main Success Scenarios: | 1. Staff chooses to edit the availability that was submitted  2. System displays the availability schedule they had submitted  3. Staff changes the availability for one or more days  4. System saves the new availability to replace the previous one |
| Alternative Scenarios: | 3a. Staff decides not to change anything  3a1. System does not change the availability schedule  4a. System is unable to save the new availability  4a1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC16 |
| Use Case Name: | **View weekly assignments** |
| Description: | Staff can view the assignments that have been given to them for the week. |
| Primary Actor: | Staff |
| Preconditions: | Staff is logged in.  Manager has given assignments to the staff. |
| Postconditions: | Main: Staff views their weekly assignments.  2a: Staff is unable to view their weekly assignments |
| Main Success Scenarios: | 1. Staff chooses to view their weekly assignments  2. System displays their assignments |
| Alternative Scenarios: | 2a. System is unable to display staff’s assignments  2a1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC17 |
| Use Case Name: | **Reject assignment** |
| Description: | Staff can reject assignments given to them. |
| Primary Actor: | Staff |
| Preconditions: | Staff is logged into the system.  Staff has been given assignments. |
| Postconditions: | Main: Staff rejects their assignment and manager is notified.  5a, 6a: Staff does not reject the assignment. |
| Main Success Scenarios: | 1. Staff is viewing their assignments  2. System displays an option beside the assignments to reject the assignment  3. Staff selects the rejection option  4. System provides warning and request for reason  5. Staff provides a reason and submits the rejection  6. System saves the rejection and informs the assigning manager |
| Alternative Scenarios: | 5a. Staff views warning and decides not to proceed with the rejection  5a1. System closes the warning  5a2. System does not save the rejection  6a. System is unable to save the rejection  6a1. System displays an error message containing details of the error |

|  |  |
| --- | --- |
| Use Case ID: | UC18 |
| Use Case Name: | **View total workload for the month** |
| Description: | Staff can view the total hours that have been assigned to them. |
| Primary Actor: | Staff |
| Preconditions: | Staff is logged into the system |
| Postconditions: | Main: Staff views their assigned hours for the week  2a: Staff is unable to view their assigned hours for the week |
| Main Success Scenarios: | 1. Staff chooses to view the total hours that have been assigned to them  2. System shows the calculated number of hours of all the staff’s assignments for the week |
| Alternative Scenarios: | 2a. System is unable to show the staff’s calculated hours  2a1. System displays an error message containing details of the error |

# Appendix B – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. The purpose of the Data dictionary is to layout the definition of terms used in this document and align the readers to the authors perspectives.*

*Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

Term 1: Availability

List the definition of term 1 here. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis purus ex, hendrerit sed mi consequat, dignissim lobortis velit.

Term 2: Assigned

List the definition of term 2 here. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis purus ex, hendrerit sed mi consequat, dignissim lobortis velit.

Term 3: Allocated

List the definition of term 3 here.

Term 4: Preference

Tied to Flight and not Date

Term 3:

Term 3:

Term 3:

Term 3: