

ICT2101/2201

Introduction to Software Engineering



Milestone 2: Software Design Specifications & Management

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: 28/10/2023

Table of Contents

1		
Introduction.....		2
1.1 Product Scope		
Changes.....		3
1.2 References and		
Acknowledgments.....		3
2 Changes in		
Requirements.....		3
3 Software		
Design.....		4
3.1 Component		
Diagram.....		5
3.2 Sequence		
Diagram.....		5
4 Updated Project		
Plan.....		5
5 Appendix A – Updated Use Case		
Descriptions.....		6
6 Appendix C - Overall Class		
Diagram.....		7
7 Appendix C – Data		
Dictionary.....		8

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 pilots' job assignments. 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

In comparison to the project scope in Milestone 1, the purpose of the project remains the same. However, the aim of the project has now slightly shifted to prioritize the work allocation aspect on the manager's end, which is the core requirement of the client. This shift ensures that the project will be more well-focused on the main priority, and it will not over-deliver unnecessary features or requirements.

1.2 References and Acknowledgments

[1] "UML sequence diagram tutorial," Lucidchart,
<https://www.lucidchart.com/pages/uml-sequence-diagram> (accessed Oct. 5, 2023).

2 Specific Requirements

General User Interface Requirements

- The website displays a login page.
- The system confirms credentials.
- The system sends a 2FA code to the user's registered mobile number.
- The system confirms the 2FA code.
- The system links the user to either the pilot or management interface.
- The website displays the respective dashboard.
- The password changing button leads to the change password page.
- The change password page prompts the user to enter the current and new password.
- The change password page contains a confirmation button.
- The edit account page prompts the user to edit their particulars.

Pilots Account

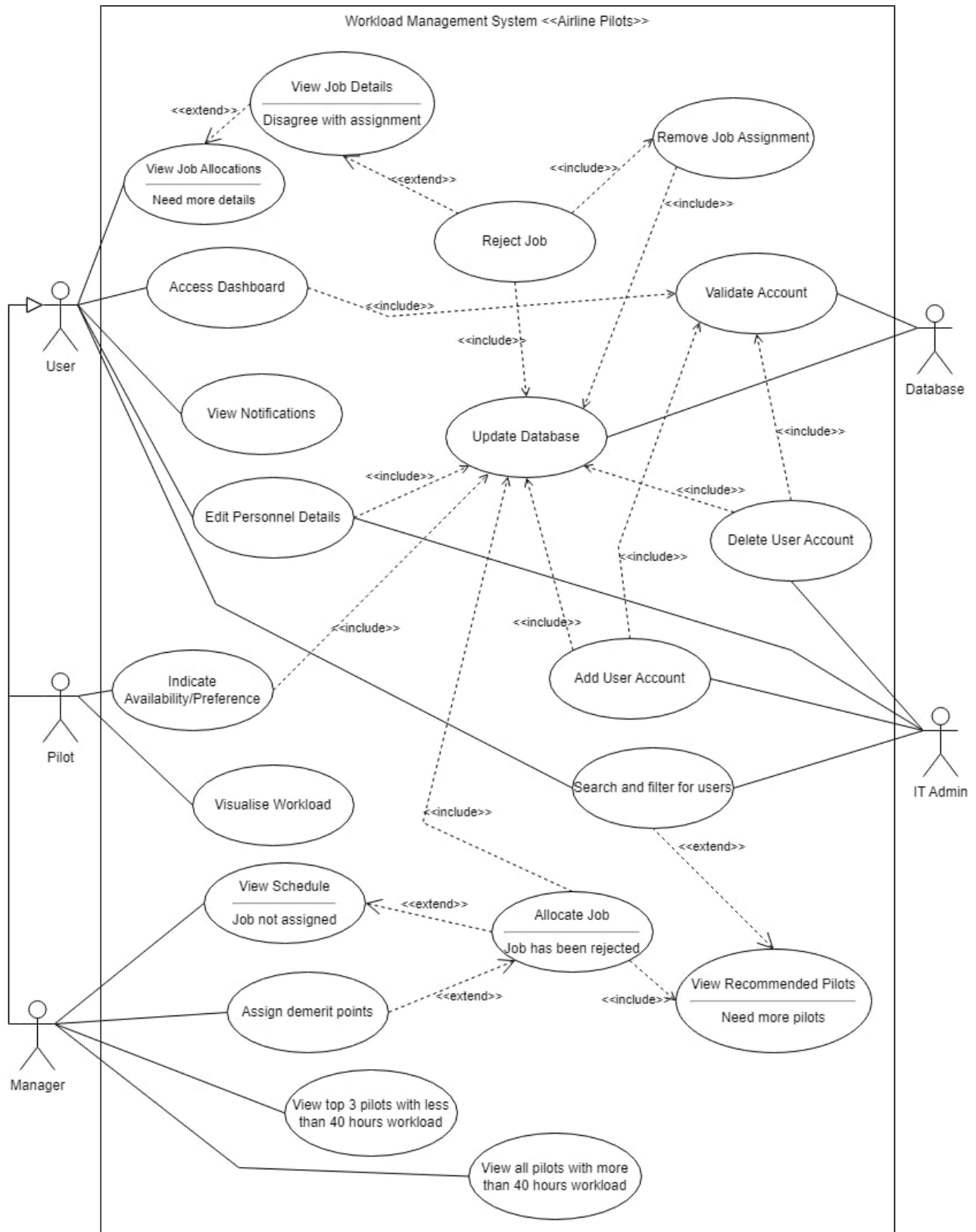
- The website will display the pilot dashboard, which consists of the allocated flight schedule, my training record, pilot performance per week, and schedule update.
- The website will also display the log out, edit account, and password changing buttons.
- The allocated flight schedule includes the pilot's flight schedule with buttons to filter the dates and reject flights.
- The training record table displays the pilot's last training records, which contain a filter year button.
- The schedule update table includes the scheduling update system.
- The scheduling update table shows an unavailable drop-down menu to allow pilots to include their unavailable dates and a "submitted record" button.
- The scheduling update table contains a "submit" button to allow pilots to proceed with their updates.
- The pilot performance per week displays a bar graph that indicates the pilot's performance.

Manager Account

- The website will display the pilot dashboard, which consists of the allocated flight schedule, allocation, pilot training records, assigned demerit points, and pilot performance per week.
- The website will also display the log out, edit account, and password changing buttons.

- The allocated flight schedule includes the flight schedules for a particular date with buttons to filter the dates and reject flights.
- The training record table displays the last training records of all the pilots, which contains a filter button for rank.
- The flight allocation table consists of different filters for flight details and rank, allowing managers to allocate schedules more efficiently.
- The pilot performance per week displays a bar graph that indicates the performance of various pilots and includes a ranking filter button.

Updated Use Case Diagram



3 Software Design

3.1 Class Diagram

In this section, the description of the class diagram will be split due to the class diagram being too large. The descriptions will focus on the control and interface classes. The boundary classes only handle the display of information, which is self-explanatory through the function names in the diagram. Please refer to Appendix C for the full class diagram. Classes found in the diagram are shown in Table 1 below.

Please take note of the following:

- The User entity class is inherited by the Pilot, Manager, and IT Admin entity classes.
- The Schedule entity class consists of the WorkingHours entity class.
- The NotificationController control class is a part of the DashboardController control class.

Entity	Control	Boundary	Interface
User	Admin Account Manager	IT Admin UI	IAccountManager
Pilot	User Account Manager	Edit Personnel Details UI	IValidation
Manager	AccountValidation	Login UI	INotification
IT Admin	DashboardController	Notification UI	IDashboard
Notification	NotificationController	Dashboard UI	ISchedule
Dashboard	JobScheduleManager	Schedule UI	IJobSchedule
Schedule	PilotJobController	User Job UI	
WorkingHours	ManagerJobController	Manager Job UI	
Job	ScheduleController		

Table 1: Class Diagram Classes

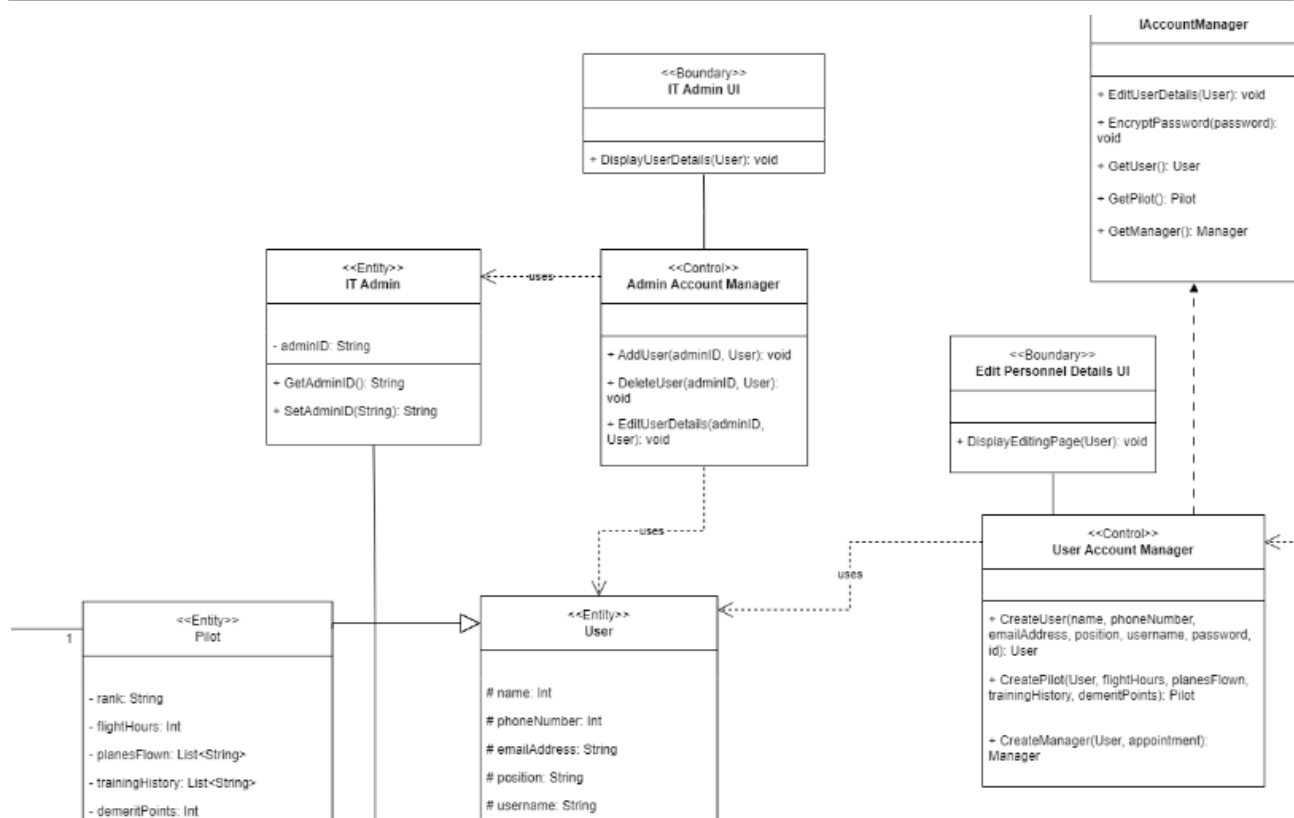


Figure 1: Account Management Classes

The classes shown in Figure 1 are related to account management, where user instances are created and managed.

- **Admin Account Manager control class:** This class is specifically used by IT Admins to add, delete, and edit accounts.
- **User Account Manager control class:** This class primarily creates user instances for the system to use, allowing for better management of user objects.
- **IAccountManager interface:** This interface opens up user object accessibility to other classes.

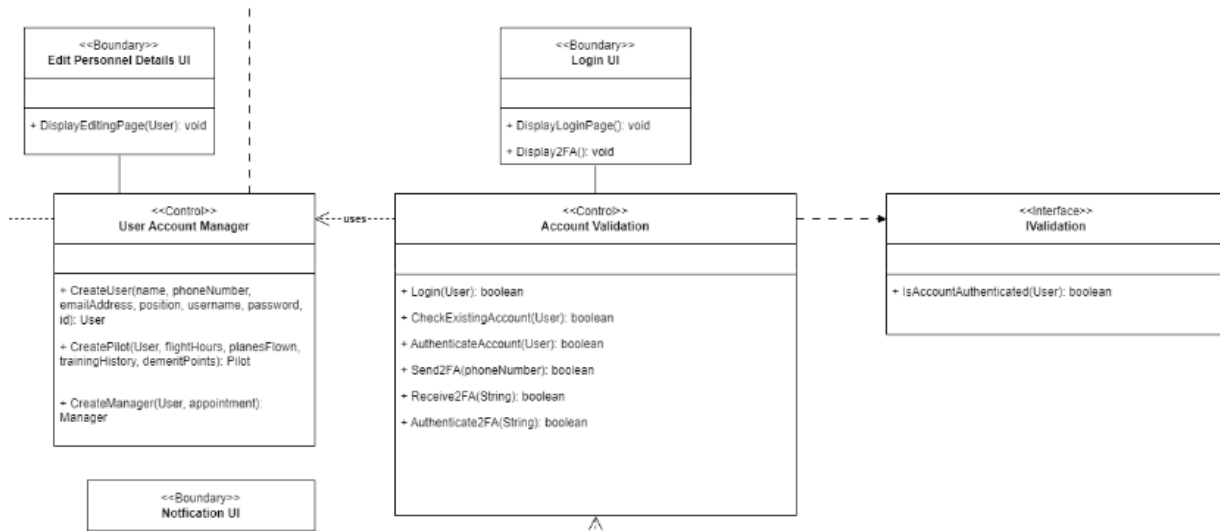
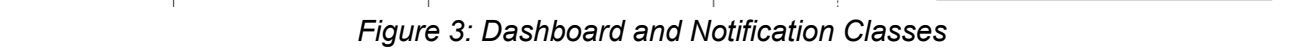


Figure 2: Account Validation Classes

Classes shown in Figure 2 are related to account validation, which authenticates accounts for access to the system.

- **Account Validation control class:** This class handles the validation of user accounts trying to access the system. It utilizes the User Account Manager control class to use user objects for authentication.
- **IValidation interface class:** This interface opens up accessibility to security-sensitive classes that require additional authentication checks before usage.



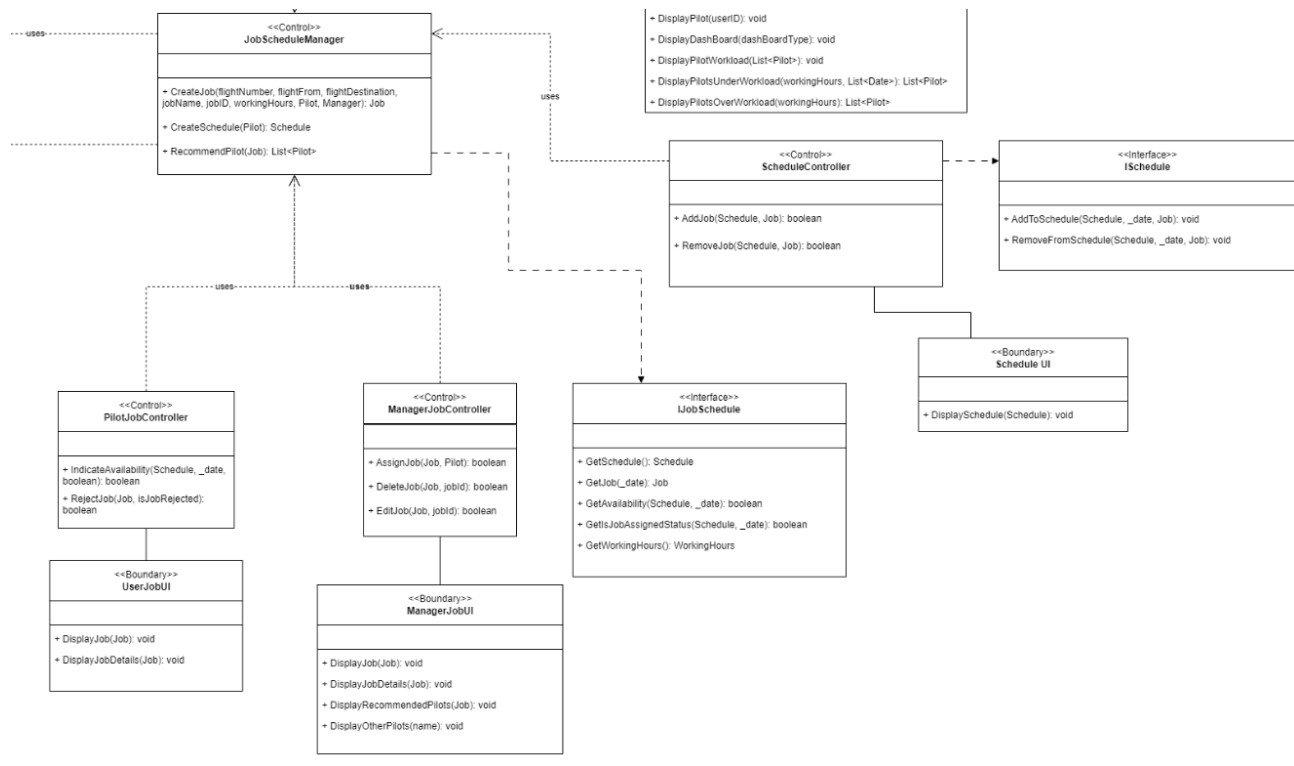


Figure 4: Schedule & Job Classes

The classes shown in Figure 4 are related to how jobs and schedules are handled in the system.

- **JobScheduleManager control class:** This class creates and manages instances of schedules and jobs. It will also handle the functionality of recommendation logic to be used by the Manager entity. This class uses the DashboardController class to aid in the searching of pilot functionality, which will be needed when recommending pilots.
- **ScheduleController control class:** This class handles logic related to adding and removing jobs to the schedule.
- **ISchedule interface class:** This interface opens up accessibility for updates to the schedule to other classes.
- **PilotJobController control class:** This class handles a pilot's ability to indicate availability and rejection of jobs.
- **ManagerJobController control class:** This class handles a manager's job assignment functionality along with removing and editing existing jobs.
- **IJobSchedule interface class:** This interface opens up accessibility to job and schedule information needed by other classes.

3.2 Component Diagram

The component diagram illustrates the different categories of the class diagram as shown in Figure 5. It consists mainly of the Validation, Account, Dashboard, which consists of the notification subcomponent, Job Schedule Manager with job and schedule as its subcomponents, and the database as the external component.

The diagram starts with the Dashboard component, where the notification subcomponent delegates the notification information to the Schedule component through the INotification Interface. On the other hand, the Dashboard component as a whole provides dashboard information to the Job component via the IDashboard Interface. Additionally, it also retrieves access information from the Validation component and receives account information from the Account component through the IAccountManager interface. The dashboard component also receives information from the Job Schedule Manager to create the dashboard.

The Job Schedule Manager component consists of the Job and Schedule subcomponents. The Job component receives the schedule information from the Schedule subcomponent through the ISchedule interface. As a whole, the Work Management System component receives data from the external database component.

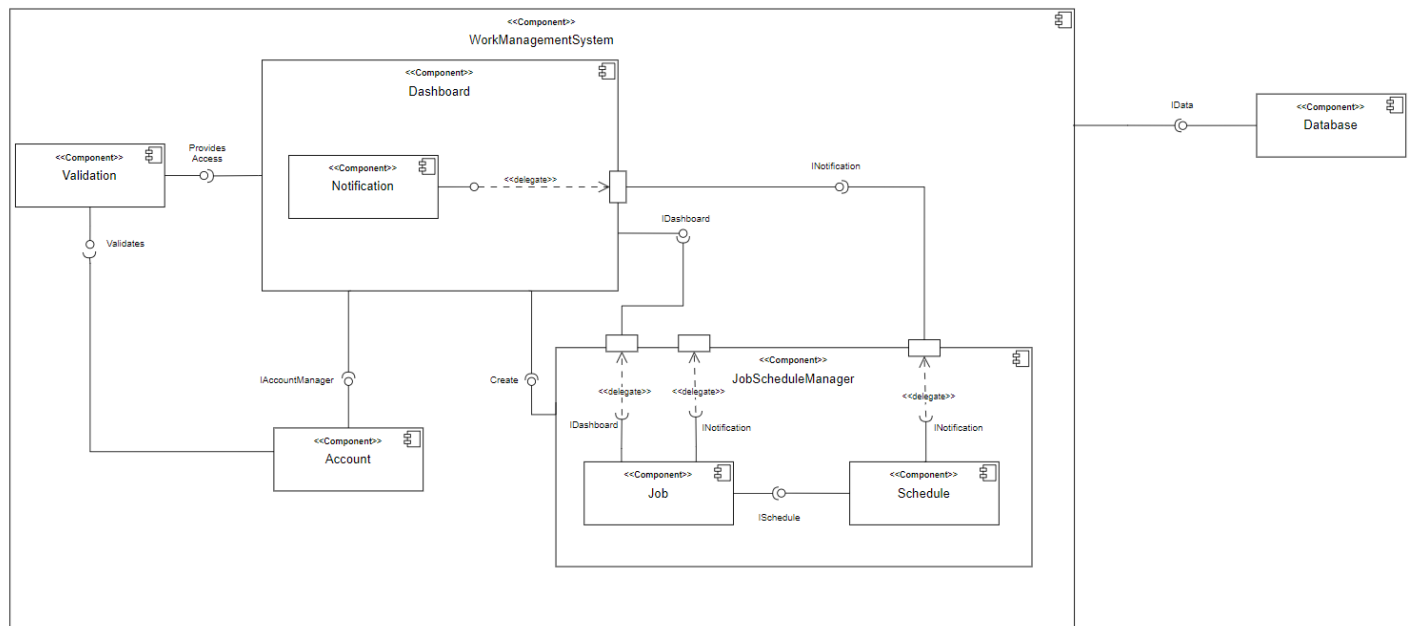


Figure 5: Component Diagram

3.3 Sequence Diagram

Based on the class diagram and use case diagram, specifically use cases 4 and 6, as shown in [Appendix A – Updated Use Case Descriptions](#), were selected as the two prominent uses in this report. These two use cases were chosen from the list of core requirements of this project, which mainly involve the viewing of job allocation and the rejection of job allocation.

The first sequence diagram represents the viewing of job allocation, as shown in Figure 6. It involves the users, representing both managers and pilots, the Dashboard User Interface as the boundary lifeline, the Job Schedule Manager as the control lifeline, the scheduler as the entity lifeline, and the database.

The user initiates the process by creating a job. The Dashboard UI is responsible for displaying all the details on the dashboard, providing information for pilots and managers. The UI displays the dashboard, showing the job allocation schedule by obtaining the schedule from the Job Schedule Manager controller. The controller retrieves the assigned jobs from the Schedule entity. Subsequently, the Database returns the list of jobs to the Dashboard UI, which displays the job details, allowing users to view their respective job allocations.

In an alternative scenario, if the list of jobs is successfully obtained, the Schedule entity returns the list of jobs to the users. However, if there is an error within the system, an error message indicating that no jobs have been found will be displayed.

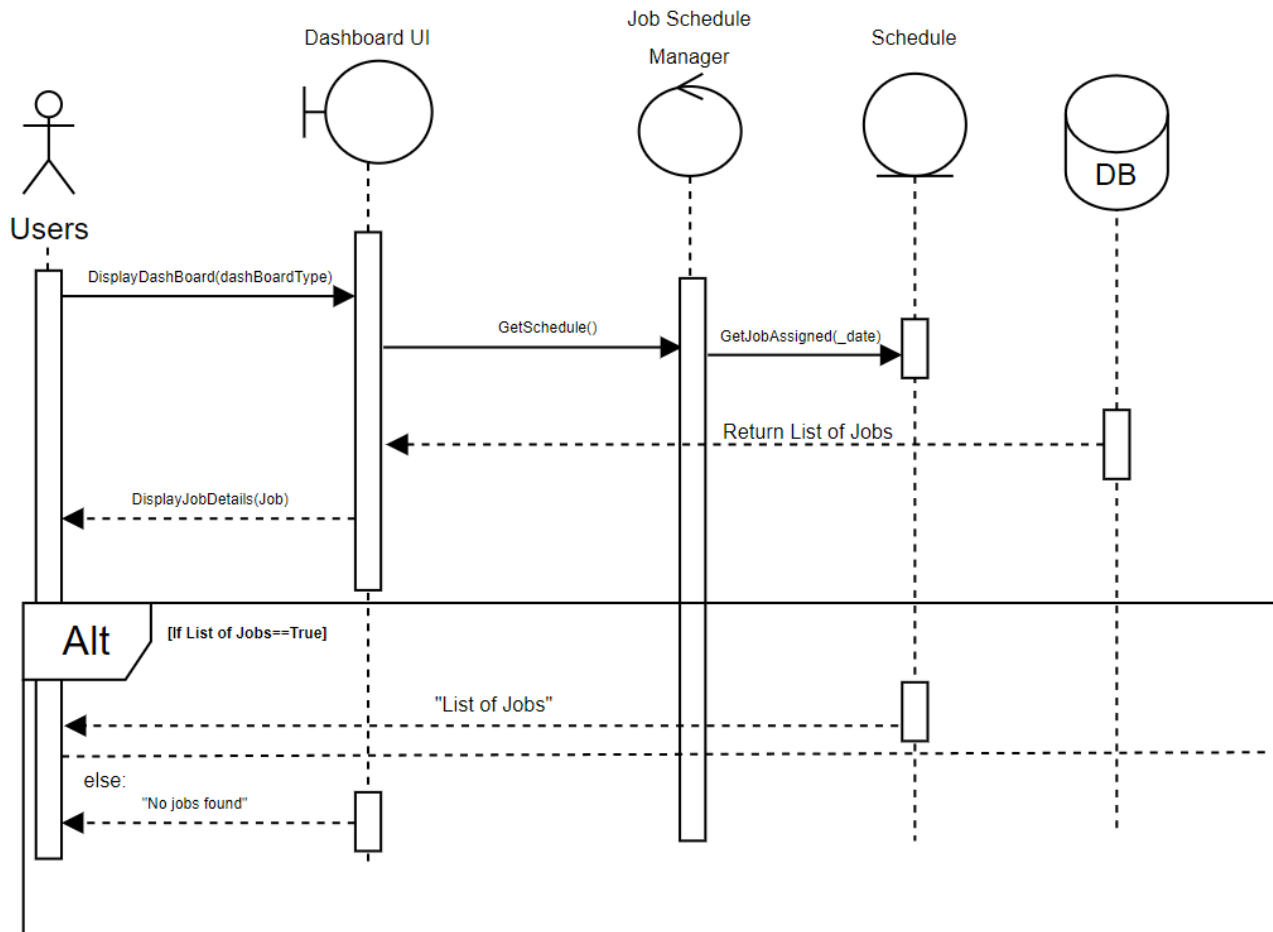


Figure 6: Sequence Diagram 1, View Job Allocation

The second sequence diagram represents the use case for rejecting job allocation, as shown in Figure 7. The diagram includes the pilot as the actor, the User Job User Interface as the boundary lifeline, the Pilot Job controller, the notification controller, the notification entity, and the database.

The diagram begins with the pilot successfully logging into the system. Job details are displayed on the User Job UI. When the pilot initiates the rejection of a job, the reject job function is called from the Pilot Job Controller. Once the system receives the request, the Notification Controller sends a notification using the notify function, and the SetNotification function from the notification entity is used. The set ISRead function then interacts with the database. Subsequently, the notification entity employs the get notification function to send the notification to the User Job UI, which displays the notification when prompted.

In an alternative scenario, if the job rejection is unsuccessful, an error prompt is sent to the User Job UI, which displays a rejection job error message. However, if the job rejection is

successful, a success prompt is sent, and the job rejection message is displayed through the User Job UI.

Additionally, in a nested alternative scenario, if the notification is unable to be delivered, a notification error is sent to notify the pilot. Conversely, if the notification is successfully sent, the pilot will receive the notification message.

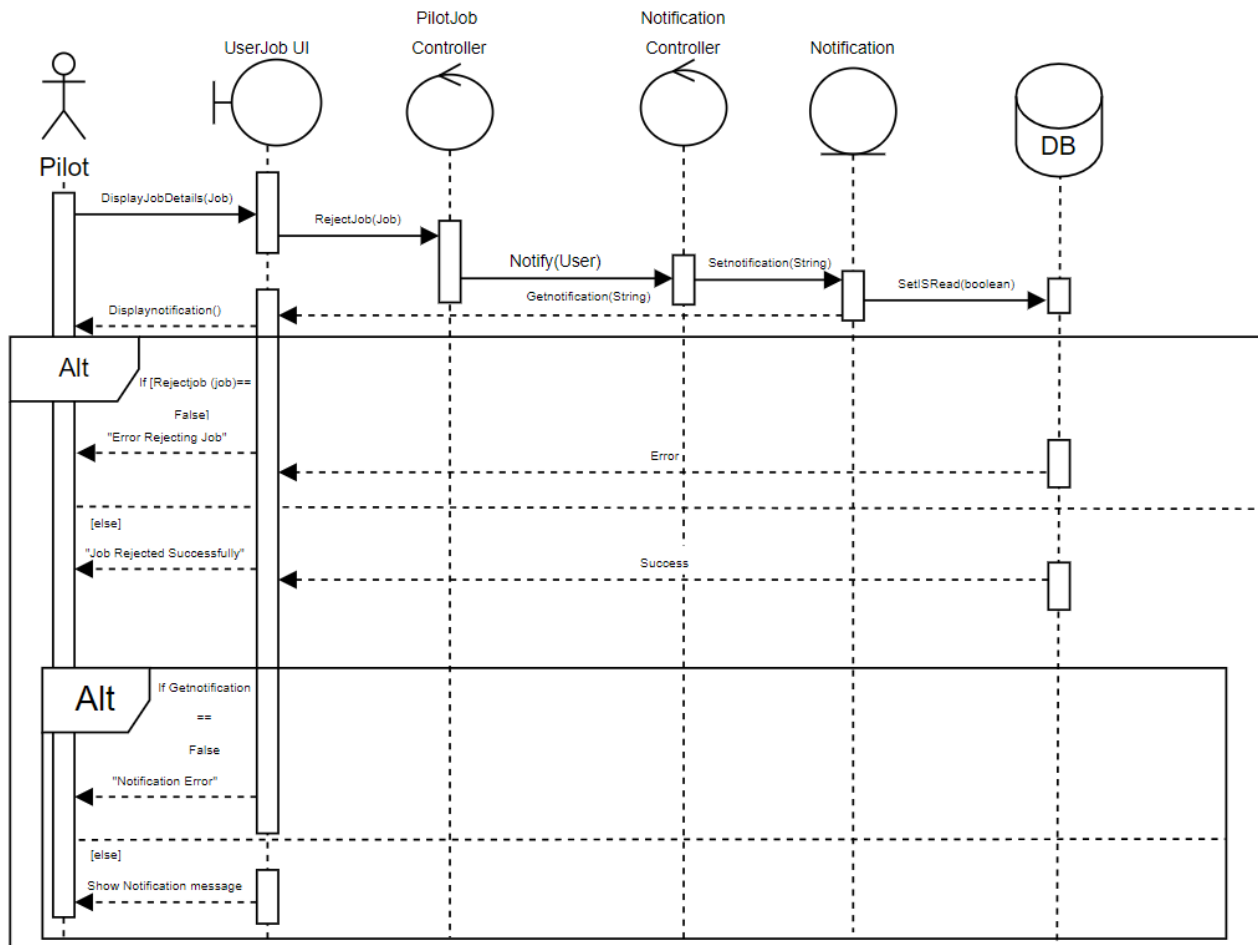


Figure 7: Sequence Diagram 2, Reject Job Allocation

4 Updated Project Plan

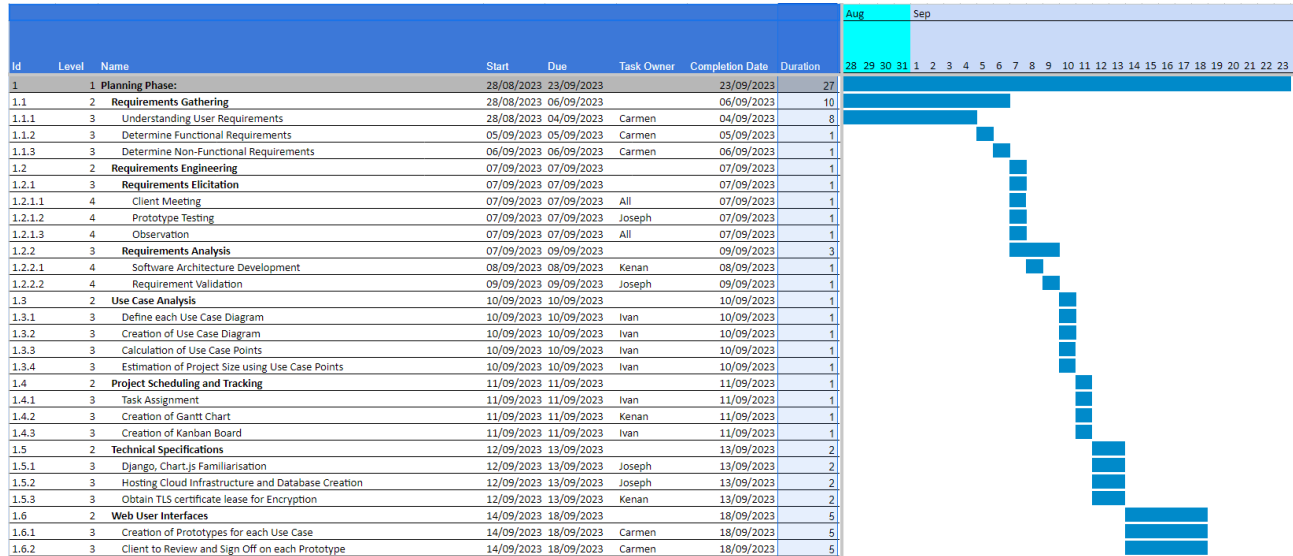


Figure 8: Planning Phase Part 1

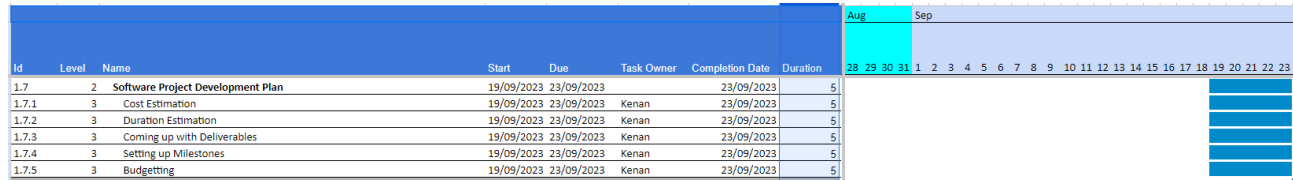


Figure 9: Planning Phase Part 2

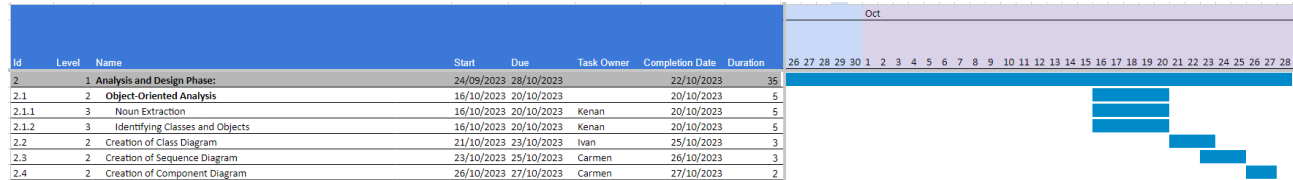


Figure 10: Analysis & Design Phase

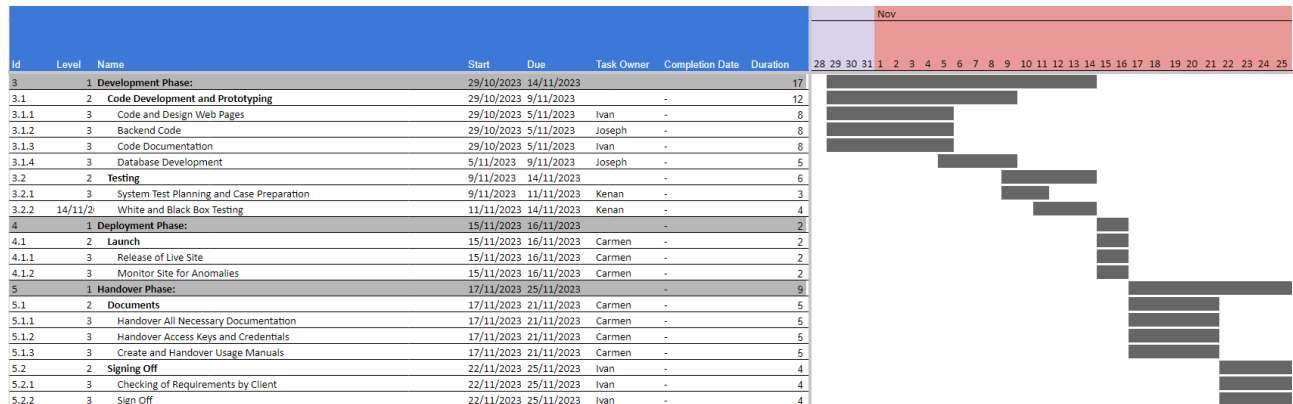


Figure 11: Development, Prototyping & Handover Phase

5 Appendix A – Updated Use Case Descriptions

Only the use case descriptions that have been updated will be shown here.

Previous Use Case Descriptions

Use Case ID:	UC-2
Use Case Name:	View Dashboard
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:	1. Dashboard displays a summary of workload assigned.
Alternative Scenarios:	

- The name of Use Case 2 has been changed to provide a better understanding of its functionality.

Use Case ID:	UC-8
Use Case Name:	Edit Personal Details
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
Preconditions:	User must have access to his dashboard.
Postconditions:	An email and notification will be sent to the user.
Main Success Scenarios:	<ol style="list-style-type: none">1. User clicks on profile icon.2. Information relating to user will be displayed.3. User clicks on update information.4. User edits information.5. User clicks on submit button to register new information.6. An email is sent to the user.7. A notification is sent to the user.
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</p> <p>6a. Email not sent 6a1. System offers a retry or quit option</p> <p>7a. Notification not sent 7a1. System offers a retry or quit option</p>

- The name of Use Case 8 has been changed to provide a better understanding of its functionality.

Use Case ID:	UC-18
Use Case Name:	Account Management
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:	<ol style="list-style-type: none">1. On the website, the user can search and filter for personnel.2. User clicks on the 'add personnel' button to add a new account.3. User clicks on the 'delete personnel' button to delete the existing account.4. User can click on personnel.5. User edits personnel details.6. User registers information by clicking on the 'submit' button.7. An email is sent to the user involved.8. A notification is sent to the user involved.
Alternative Scenarios:	<p>6a. System could not register information.</p> <p>6a1. System shows error message and provides reason for not being able to register information.</p> <p>6a2. System offers a retry or quit option.</p> <p>7a. Email not sent.</p> <p>7a1. System offers a retry or quit option.</p> <p>8a. Notification not sent.</p> <p>8a1. System offers a retry or quit option.</p>

- Use Case 18 replaced.

Use Case ID:	UC-19
Use Case Name:	Database Management
Description:	IT Admins have access to database for administrative purposes.
Primary Actor:	IT Admin
Preconditions:	User must have administrative access to the database.
Postconditions:	
Main Success Scenarios:	<ol style="list-style-type: none">1. User can search for and edit databases.2. User can search for and edit tables.3. User can search for and edit columns.4. User can search for and edit data.
Alternative Scenarios:	<p>1a Database does not show up. 1a1 Error will be shown to tell the user what is wrong. 1a2 Repeat 1.</p> <p>2a Table does not show up. 2a1 Error will be shown to tell the user what is wrong. 2a2 Repeat 2.</p> <p>3a Column does not show up. 3a1 Error will be shown to tell the user what is wrong. 3a2 Repeat 3.</p> <p>4a Data does not show up. 4a1 Error will be shown to tell the user what is wrong. 4a2 Repeat 4.</p>

- Use Case 19 replaced.

Updated Use Case Description

Use Case ID:	UC-2
Use Case Name:	Access Dashboard
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:	1. Dashboard displays a summary of workload assigned.
Alternative Scenarios:	

- The name has been changed from 'View Dashboard' to 'Access Dashboard'.

Use Case ID:	UC-8
Use Case Name:	Edit Personnel Details
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. Only IT Admins have the permission to edit all personnel details.
Primary Actor:	User, IT Admin
Preconditions:	For users, they must have access to their dashboard. For IT Admins, they must have their account validated.
Postconditions:	An email and notification will be sent to the user. Information will be updated on the database.
Main Success Scenarios:	<ol style="list-style-type: none">1. User clicks on the profile icon.2. Information relating to the user will be displayed.3. User clicks on update information.4. User edits information.5. User clicks on the submit button to register new information.6. An email is sent to the user.7. A notification is sent to the user.
Alternative Scenarios:	<p>5a. System could not register information.</p> <p>5a1. System shows an error message and provides a reason for not being able to register information.</p> <p>5a2. System offers a retry or quit option.</p> <p>6a. Email not sent.</p> <p>6a1. System offers a retry or quit option.</p> <p>7a. Notification not sent.</p> <p>7a1. System offers a retry or quit option.</p>

- The name has been changed from 'Edit Personal Details' to 'Edit Personnel Details.'

Use Case ID:	UC-18
Use Case Name:	View Schedule
Description:	Managers should be able to view the schedule. Upon looking at the schedule, he should be able to determine which days require job allocation.
Primary Actor:	Manager
Preconditions:	Manager must have access to his dashboard.
Postconditions:	
Main Success Scenarios:	<ol style="list-style-type: none">1. Monthly schedules will be displayed.2. Unassigned dates will be highlighted.3. Days requiring allocation will be highlighted.4. The manager will be able to click on a date.5. Clicking on the date brings him to the job allocation page.
Alternative Scenarios:	5a Job allocation page does not load. 5a1 Error will be displayed. 5a2 Error will direct the user to find IT Admin.

- Use Case 18 has been replaced with the functionality of allowing the manager to view the schedule, which will then help him determine which dates require job allocation.

Use Case ID:	UC-19
Use Case Name:	Add User Account
Description:	IT Admin will be able to add more user accounts.
Primary Actor:	IT Admin
Preconditions:	The user must have validated administrative access to the database.
Postconditions:	The database updates its information.
Main Success Scenarios:	<ol style="list-style-type: none">1. User can search for and edit databases.2. User can search for and edit tables.3. User can search for and edit columns.4. User can search for and edit data.5. User can add new data to the above-mentioned 1-4.
Alternative Scenarios:	<p>1a Database does not show up. 1a1 Error will be shown to tell the user what is wrong. 1a2 Repeat 1.</p> <p>2a Table does not show up. 2a1 Error will be shown to tell the user what is wrong. 2a2 Repeat 2.</p> <p>3a Column does not show up. 3a1 Error will be shown to tell the user what is wrong. 3a2 Repeat 3.</p> <p>4a Data does not show up. 4a1 Error will be shown to tell the user what is wrong. 4a2 Repeat 4.</p>

- Use Case 19 has been replaced by the functionality of allowing IT Admins to add new user accounts.

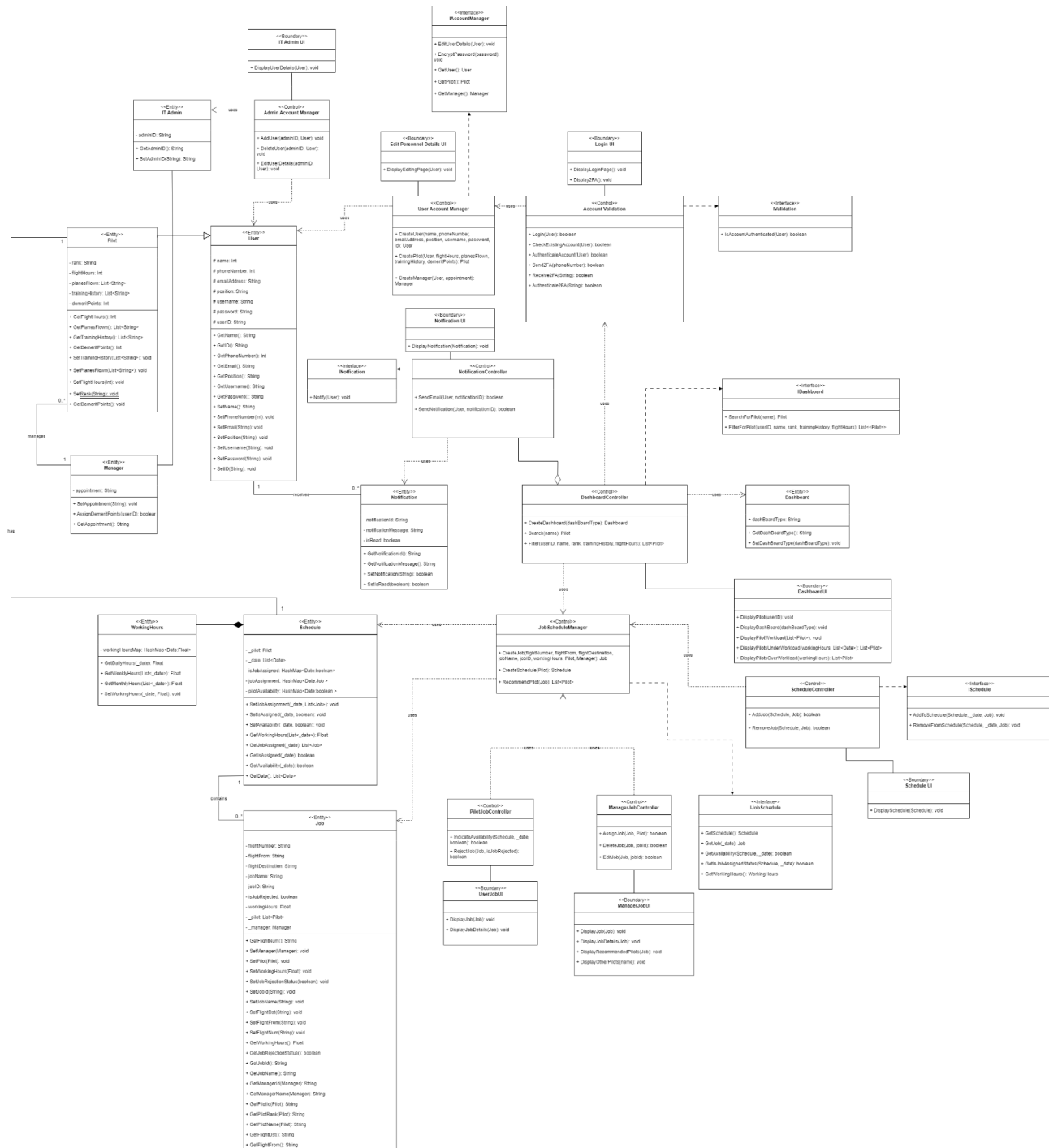
Use Case ID:	UC-20
Use Case Name:	Delete User Account
Description:	IT Admin will be able to add more user accounts.
Primary Actor:	IT Admin
Preconditions:	The user must have validated administrative access to the database.
Postconditions:	The database update its information.
Main Success Scenarios:	<ol style="list-style-type: none">1. User can search for and edit databases.2. User can search for and edit tables.3. User can search for and edit columns.4. User can search for and edit data.5. User can remove data from above mentioned 1.-4.
Alternative Scenarios:	<p>1a Database does not show up. 1a1 Error will be shown to tell the user what is wrong. 1a2 Repeat 1.</p> <p>2a Table does not show up. 2a1 Error will be shown to tell the user what is wrong. 2a2 Repeat 2.</p> <p>3a Column does not show up. 3a1 Error will be shown to tell the user what is wrong. 3a2 Repeat 3.</p> <p>4a Data does not show up. 4a1 Error will be shown to tell the user what is wrong. 4a2 Repeat 4.</p>

- Use Case 20 has been added to demonstrate that an IT Admin has the authority to delete existing accounts.

Use Case ID:	UC-21
Use Case Name:	Update Database
Description:	The database should always contain updated information to reflect accurate data. The database will also be used to validate user accounts whenever they log into the system.
Primary Actor:	Database, IT Admin
Preconditions:	The database is running.
Postconditions:	The information is accurate.
Main Success Scenarios:	<ol style="list-style-type: none">1. IT Admin looks up information on the database.2. The database displays information accurately.3. The information shown on the database matches what is displayed on the dashboard.4. IT Admin is responsible for maintaining the database.
Alternative Scenarios:	<p>1a Information not found. 1a1 Repeat 1.</p> <p>1b Information is corrupted. 1b1 IT Admin replaces corrupted information with a backup.</p> <p>2a Information is not displayed. 2a1 Repeat 1-2.</p> <p>3a Information does not match. 3a1 IT Admin troubleshoots. 3a2 IT Admin updates information manually.</p>

- Use Case 21 has been added to show that a database update will occur when information is changed.

6 Appendix C - Overall Class Diagram



7 Appendix C – Data Dictionary

2FA - *Two-Factor Authentication, a method of identity and access management*

User Interface (UI) - *The design of machine and software user interfaces*

Pilots/Users - *This term refers to staff who are either under training or are trained to operate the various planes of FLYSG Airlines*

Managers/Management/Users - *This term refers to any administrative staff or managers, as well as any staff within the human resource management level at FLYSG Airlines*

IT Administrator - *This term refers to the staff members who belong to the IT department of FLYSG Airlines and are responsible for managing the company's IT systems and matters*

IT - *Information Technology*