

Milestone 1: Software Development Plan & Specification

for

Music School

Prepared by

Team's Name: AI

Ang Wee Yi, Alex	2102852
Iphigenia Peh Yureng	2102741
Iphigenia Peh Yuxi	2102744

Lab Group: P2-9

Github handle: ICT2201-P2-9

Date: 24/09/2022

Table of Contents

Introduction	3
Product Scope	3
Related Background Literature	3
Intended Audience and Document Overview	4
References and Acknowledgments	4
Overall Description	5
Product Overview	5
Product Functionality	5
Assumptions and Dependencies	6
Specific Requirements	7
User Interface Requirements	7
Functional Requirements	12
Use Case Model	14
Non-functional Requirements	15
Performance Requirements	15
Safety and Security Requirements	15
Project Estimation and Plan	16
Software Estimation	16
Project Management	19
Individual Members Task and Reflections	20
Appendix A – Use Case Descriptions	22
Appendix B – Data Dictionary	29

1 Introduction

In the music school, work allocations are manually assigned monthly to teachers by the managers. Hence, to facilitate better workload allocation, the team has built a web-based workload management system to show an overview of the company's manpower strength at any time and informative availability of employees and engagement. The system will provide an interactive and informative way for both employees and managers to visualise and allocate their workload optimally which will in turn improve work-life balance and make the workplace environment more conducive and attractive. The following subsections are organised as such: Subsection 1.1 describes the scope of the product. Subsection 1.2 shows the related background literature. Subsection 1.3 discusses the intended audience and the overview of the document. Finally, subsection 1.4 included references and acknowledgements.

1.1 Product Scope

The scope of the product is to create a web-based application that will be utilised by managers, staff and IT administrators for workload allocation. The product provides a framework for managing and organising workload efficiently in such a way that increases work-life balance. The scope is further illustrated in section 2 and section 3 of this document which encapsulates the product functionality as well as the specific features and requirements that characterise this product. The product applies only to the management of workload allocation and is a tool that facilitates decision making, the product does not make decisions for the employees. Section 4 would illustrate the delivery dates of each milestone's tasks as well as the required project estimation to deliver the scope.

The product brings benefits that have the ability to enhance workload management which will in turn improve employee work-life balance. Firstly, managers are able to view timely updates of staff's availability and engagement which allows them to follow up and plan an optimised workload schedule. Also, the product provides automatic validation which would significantly reduce the errors that may occur during work assignment. Furthermore, when compared to paper records, all data stored in the product can be retained and backed up to cloud services in the event of data loss. Section 2 describes further details of the product.

1.2 Related Background Literature

Delegating work to your team may sound like a straightforward task of management, but, in fact, it is complicated [3]. With the ever desire to improve work-life balance and make the workplace environment more conducive and attractive [1], companies are often looking for tools that can ease their workload management as work-life balance is a cycle [2]. However, balancing an employee's workload is always challenging and a sensitive topic in the workplace. It is imperative to consider both the organisation staff strength as well as the staff perception of their workload [4], hence a workload management tool would be beneficial in managing the company's workload, and at the same time providing a complete overview of the staff workload.

1.3 Intended Audience and Document Overview

The intended audience of this document are the client and the professor. In this project, the client refers to an entity that has engaged the team to design and build a custom software according to their requirements and specifications. The professor would provide relevant and essential feedback throughout the project and documentation process for the team and ensure this document meets the stated requirements and specification.

The recommended sequence for reading the document is:

1. Section 1 offers a good overview and understanding of the entire project.
2. Section 6 and 7. These sections provide useful information regarding individual use cases descriptions and a data dictionary that consists of the different terminology used in this document to prevent any misunderstanding.
3. Section 2 illustrates the overview of the product which comprises the product overview, product functionality, assumptions and dependencies.
4. Section 4 would be most pertinent to the client as it encapsulates the project estimation and plan which includes the following subsections: software estimation and project management plan. These sections would give the client a rough estimation of the project cost, resources and time required before the release of the product.
5. Section 3 details the requirements of the product which includes user interface requirements, functional requirements, use case model, non functional requirements, performance requirements, and safety and security requirements.
6. Section 5 consists of individual team reflection regarding this project.

In general, all of the sections are pertinent to the professor for grading and reviewing the team effort and work done.

1.4 References and Acknowledgments

[1] D. E. Guest, "Perspectives on the study of work-life balance - david E. guest, 2002," Sage Journals, Jun-2002. [Online]. Available:
<https://journals.sagepub.com/doi/10.1177/0539018402041002005>.

[2] I. Lupu and M. Ruiz-Castro, "Work-life balance is a cycle, not an achievement," Harvard Business Review, 29-Jan-2021. [Online]. Available:
<https://hbr.org/2021/01/work-life-balance-is-a-cycle-not-an-achievement>.

[3] R. Knight, "Make sure your team's workload is divided fairly," Harvard Business Review, 03-May-2017. [Online]. Available:
<https://hbr.org/2016/11/make-sure-your-teams-workload-is-divided-fairly>.

[4] H. Inegbedion, E. Inegbedion, A. Peter, and L. Harry, "Perception of workload balance and employee job satisfaction in work organisations," Heliyon, 09-Jan-2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405844020300050>.

2 Overall Description

2.1 Product Overview

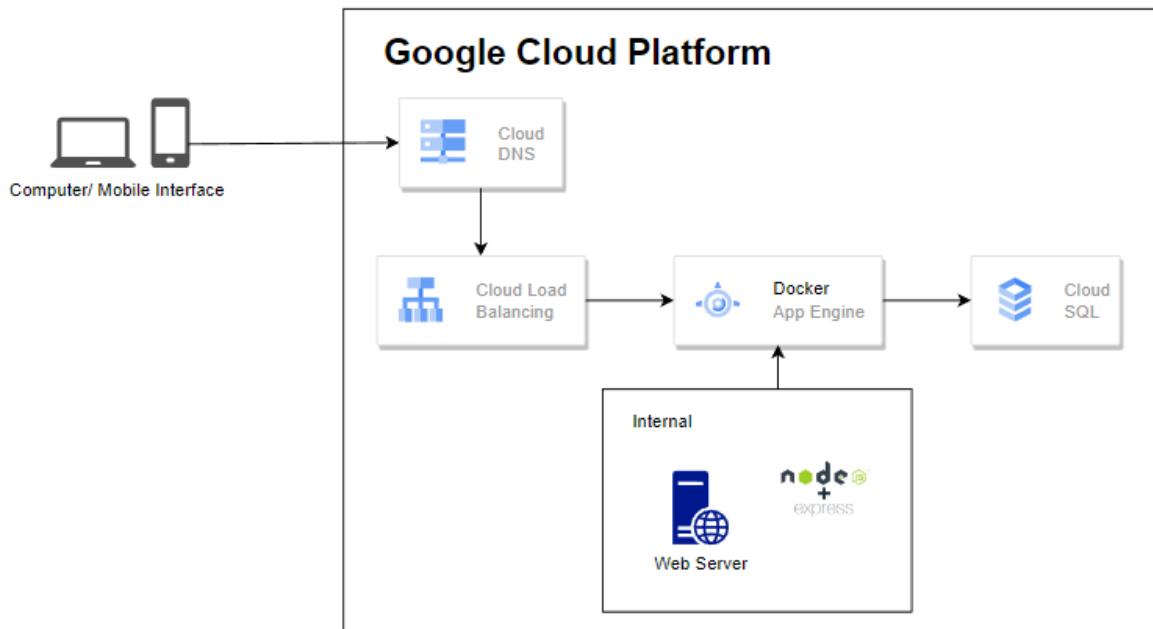


Figure 1: System Architecture Diagram

The product is a web-based application replacing paper records and manual processes. The web-based application will be built with ExpressJS which is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. The web server will be hosted in Application instances run within Docker containers on Compute Engine virtual machines (VM).

Figure 1 presents an overall view of the system architecture. The web-based workload management system provides functionalities that are stated in Section 2.2 Product Functionality. It has interfaces to the external system, such as the Cloud Database System. The web-based workload management system stores all its data in the Cloud Database System and is therefore essential to maintain connection to it. The team would be using Google Cloud as its cloud database as it provides AES256 encryption for data stored in the database. This would ensure that the stored data is more secure and safe for the company.

2.2 Product Functionality

The section consists of the functionality of the product in a bulleted list format.

PF1: The product shall display job assignments to staff

PF2: The product shall allow staff to inform their availability in advance

PF3: The product shall allow staff to indicate any assigned jobs they cannot fulfil

PF4: The product shall allow staff to indicate their job preference

PF5: The product shall allow managers to visualise the manpower availability

PF6: The product shall allow manager to perform and manage job allocations

PF7: The product shall allow managers to view individual staff's workload and availabilities

PF8: The product shall allow IT administrators to manage manager and staff accounts

2.3 Assumptions and Dependencies

The assumed factors include the following:

A1: Management of students is not considered when designing the software.

A2: Allocation of jobs is based on the teacher's qualifications.

A3: Allocation of workload is on a 30-minutes frame basis.

A4: IT administrator role is purely for managing staff and manager accounts.

A5: The studio and lesson are fixed.

A6: Staff can have more than one instrument qualification.

A7: Staff are able to indicate their availability in 30-minutes slots and range.

A8: Staff would be assigned to at least one job a month.

The team will be using Justinmind version 9.9.1 for the creation of a static web page. The web page would consist of the product prototype for the client.

3 Specific Requirements

3.1 User Interface Requirements

The following figures 2-11 show the interface between the users (Staff, Manager and IT Administrator) and the workload management system. The interaction between the users and the system is stated in detail under Section 6 (Appendix A – Use Case Descriptions).

Main User Interface

Figure 2 shows the Login Page which all users will be required to login before entering the workload management system.



Figure 2: Login Page

Manager Interface

Figure 3 shows the Manager Landing Page. For the manager interface, the pages not shown in the figures below include the job allocation page and notification tabs.

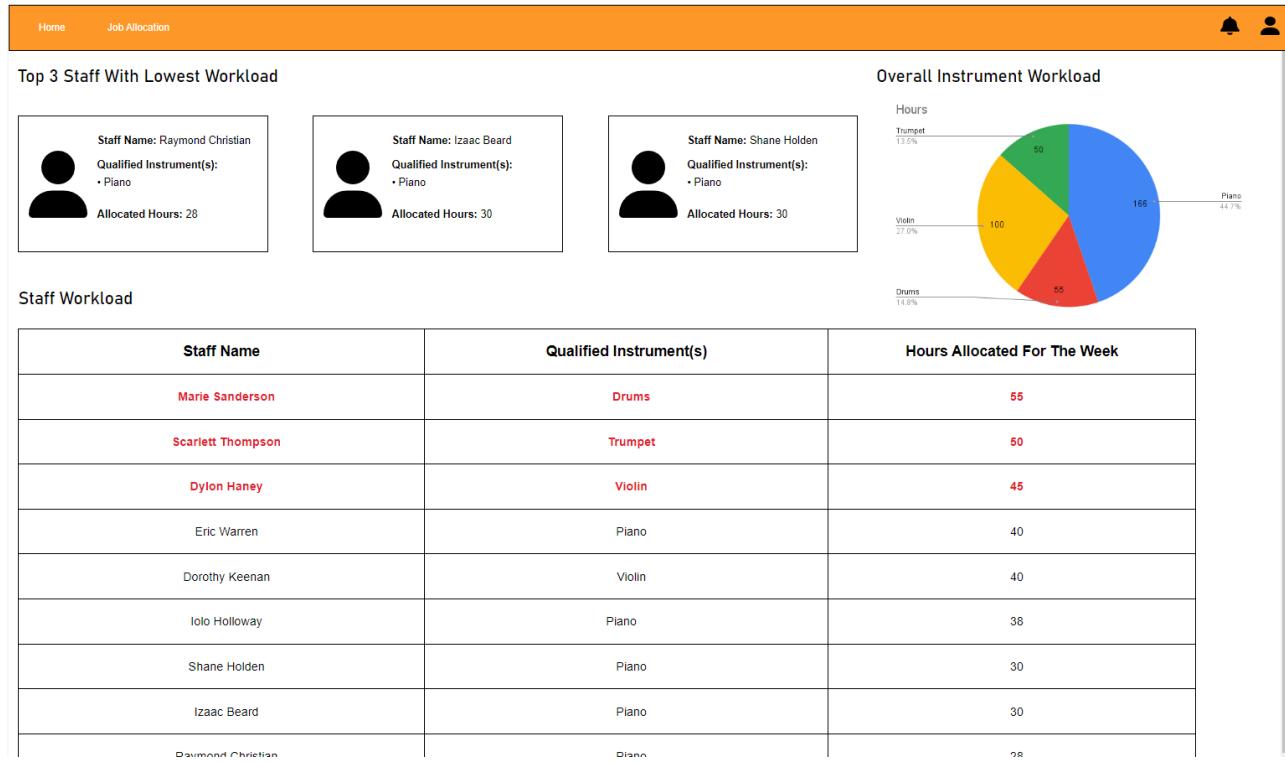


Figure 3: Manager Landing Page

Staff Interface

The following figures 4-7 show the Staff Landing Page, Add Availability, Indicate Preference, View and Edit Availability Page respectively.

Weekly Schedule

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
12/09/22 0900-0930 Piano - Studio 1	13/09/22 Nil	14/09/22 0900-0930 Piano	15/09/22 Nil	16/09/22 Nil	17/09/22 Nil	18/09/22 Nil
0930-1000 Piano - Studio 1		0930-1000 Piano				
1000-1030 Piano - Studio 1		1000-1030 Piano				

Overall Workload for the Month

20 Hours

Assigned Jobs

- Date: 13/9/22
Time: 9.00AM - 9.30AM
Class: Piano
Studio: 1

Reject **Accept**
- Date: 13/9/22
Time: 10.00AM - 10.30AM
Class: Piano
Studio: 1

Reject **Accept**
- Date: 15/9/22
Time: 10.00AM - 10.30AM
Class: Piano
Studio: 1

Reject **Accept**

Figure 4: Staff Landing Page

Timeslots

Date:

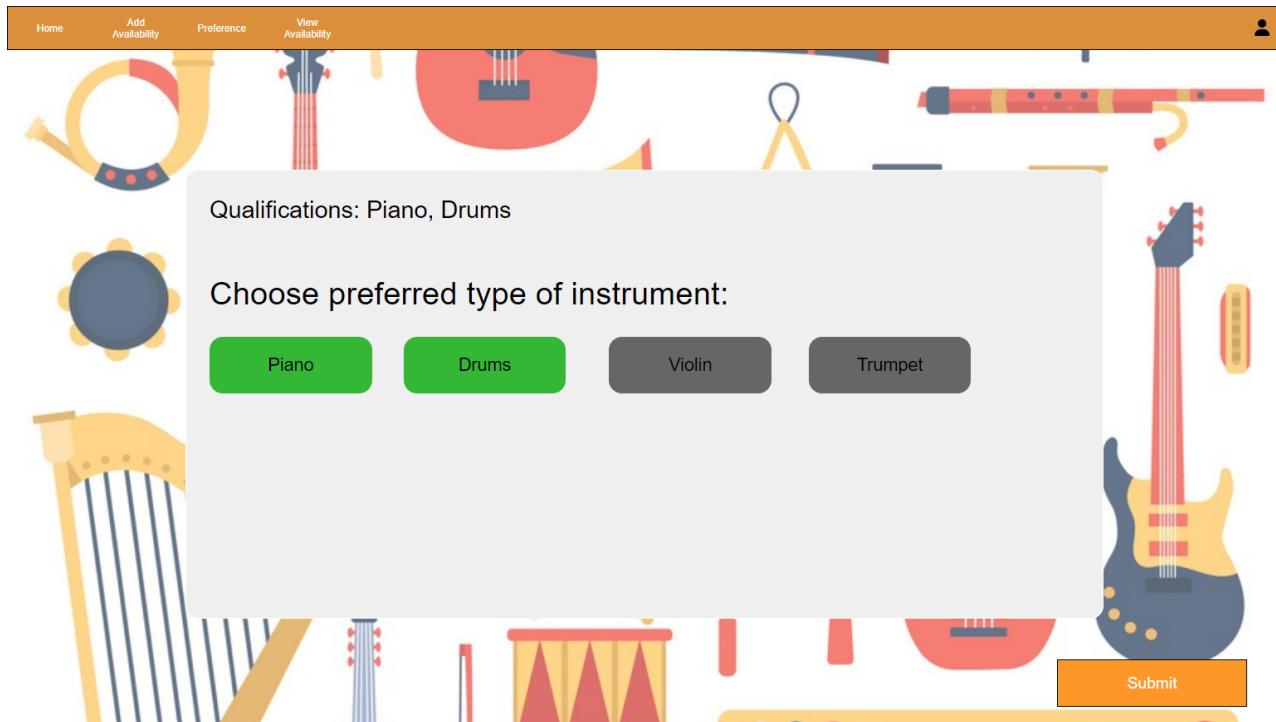
Indicate Preference

9.00AM - 9.30AM	9.30AM - 10.00AM	10.00AM - 10.30AM	10.30AM - 11.00AM	11.00AM - 11.30AM
11.30AM - 12.00PM	12.00PM - 12.30PM	12.30PM - 1.00PM	1.00PM - 1.30PM	1.30PM - 2.00PM
2.00PM - 2.30PM	2.30PM - 3.00PM	3.00PM - 3.30PM	3.30PM - 4.00PM	4.00PM - 4.30PM
4.30PM - 5.00PM	5.00PM - 5.30PM	5.30PM - 6.00PM	6.00PM - 6.30PM	6.30PM - 7.00PM
7.00PM - 7.30PM	7.30PM - 8.00PM	8.00PM - 8.30PM	8.30PM - 9.00PM	

Submit

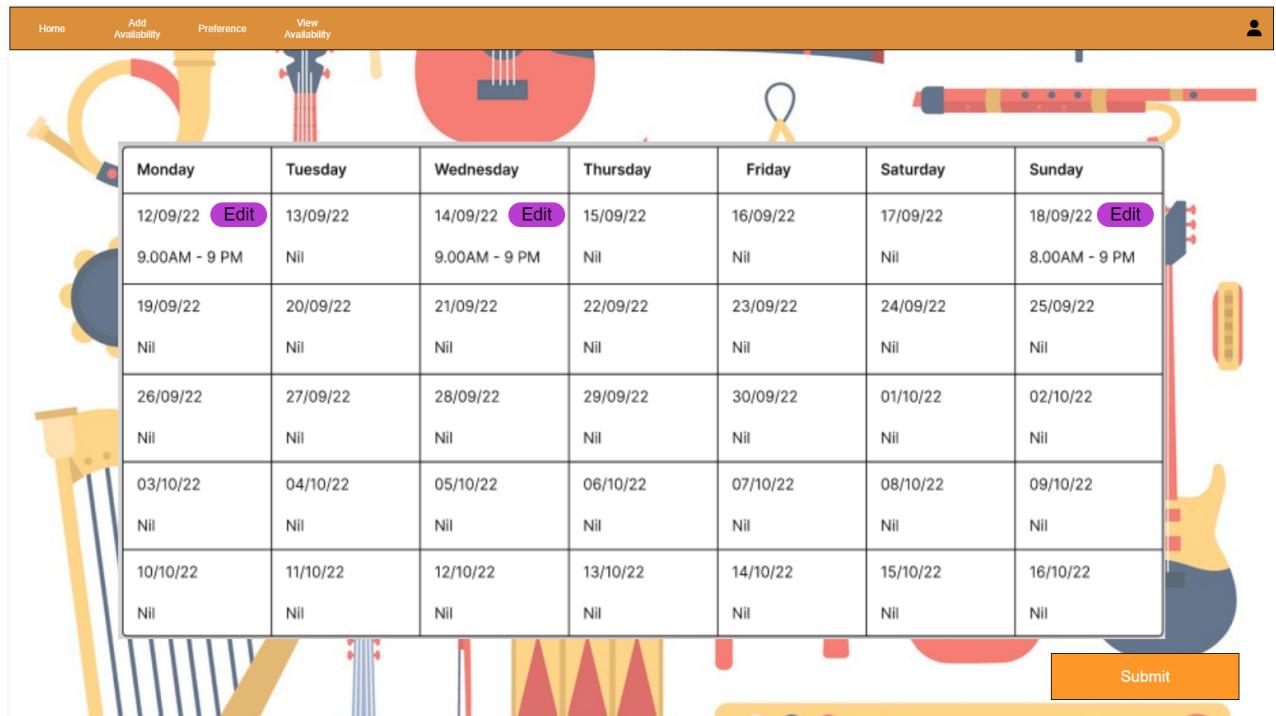
Figure 5: Staff Add Availability Page

Software Development Plan & Specification



This screenshot shows the 'Staff Indicate Preference' page. At the top, there is a navigation bar with links: Home, Add Availability, Preference, View Availability, and a user profile icon. The main content area features a large central box containing the text 'Qualifications: Piano, Drums'. Below this, the text 'Choose preferred type of instrument:' is displayed, followed by four buttons: 'Piano' (green), 'Drums' (green), 'Violin' (grey), and 'Trumpet' (grey). In the bottom right corner of the central box is an orange 'Submit' button. The background of the page is decorated with various colorful illustrations of musical instruments, including a trumpet, a guitar, a drum set, and a piano.

Figure 6: Staff Indicate Preference Page



This screenshot shows the 'Staff View and Edit Availability' page. It has a similar navigation bar at the top. The main feature is a weekly availability grid for the months of September and October. The grid consists of seven columns representing the days of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. Each column contains two rows of data. The first row shows the date and time range, and the second row shows the availability status ('Nil' or 'Available'). An 'Edit' link is located in the top-right corner of each grid cell. In the bottom right corner of the grid is an orange 'Submit' button. The background is decorated with musical instrument illustrations.

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
12/09/22 9.00AM - 9 PM	13/09/22 Nil	14/09/22 9.00AM - 9 PM	15/09/22 Nil	16/09/22 Nil	17/09/22 Nil	18/09/22 8.00AM - 9 PM
19/09/22 Nil	20/09/22 Nil	21/09/22 Nil	22/09/22 Nil	23/09/22 Nil	24/09/22 Nil	25/09/22 Nil
26/09/22 Nil	27/09/22 Nil	28/09/22 Nil	29/09/22 Nil	30/09/22 Nil	01/10/22 Nil	02/10/22 Nil
03/10/22 Nil	04/10/22 Nil	05/10/22 Nil	06/10/22 Nil	07/10/22 Nil	08/10/22 Nil	09/10/22 Nil
10/10/22 Nil	11/10/22 Nil	12/10/22 Nil	13/10/22 Nil	14/10/22 Nil	15/10/22 Nil	16/10/22 Nil

Figure 7: Staff View and Edit Availability Page

IT Administrator Interface

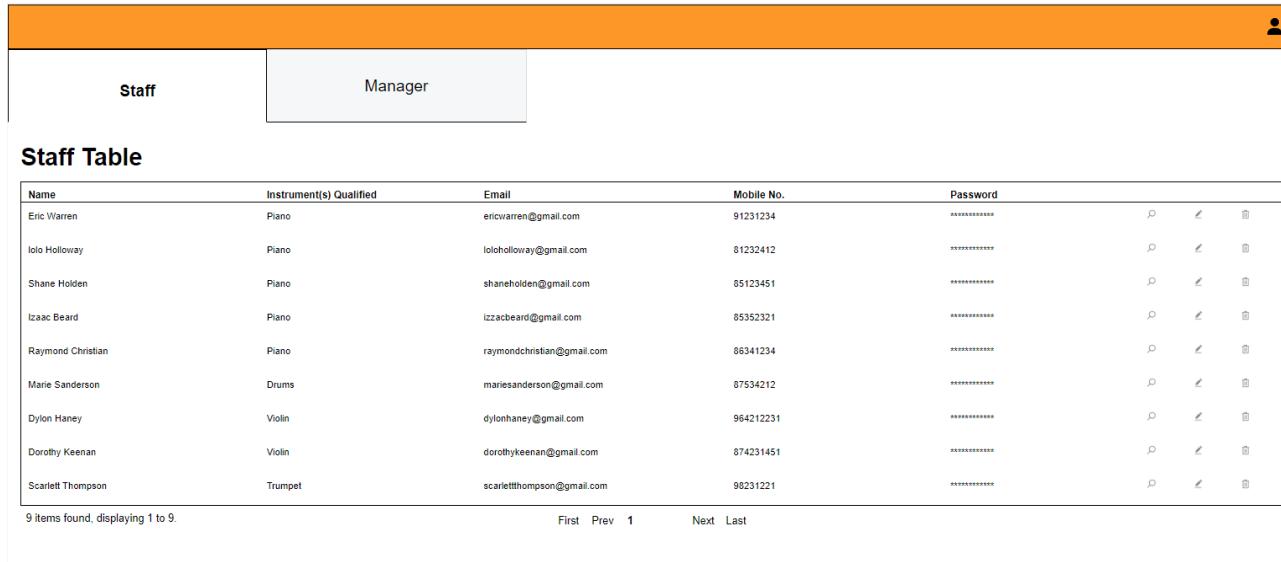
The following figures 8-11 show the IT Administrator Landing Page, Create Accounts, and View, Edit and Delete Account Page respectively.



Figure 8: IT Administrator Landing Page

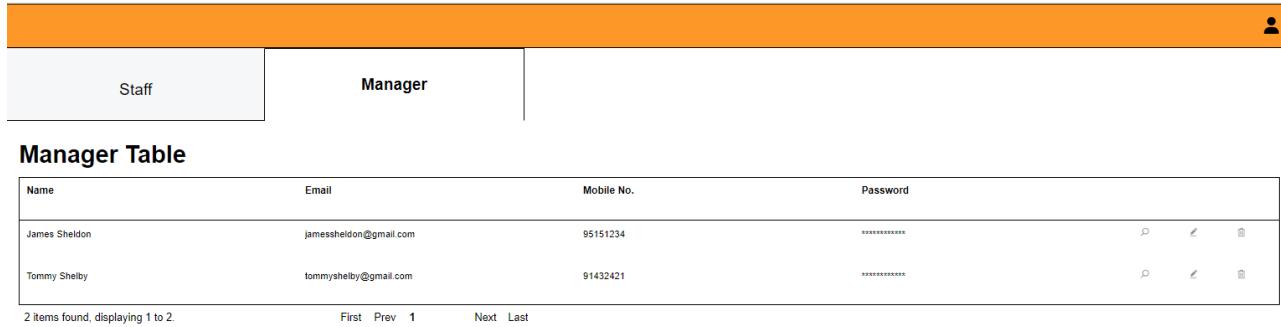
The create account page has a red border and a title "Create Account" at the top. Inside, there are several input fields for staff information: "Staff role" (with "Manager" and "Staff" options), "Staff ID", "Staff qualification", "Staff name", "Staff email", "Staff contact", and "Staff password". Below these is a "Submit" button. At the bottom of the form are three orange buttons: "View Account", "Home", and "Create Account". The background is white with musical instruments.

Figure 9: IT Administrator Create Accounts Page



Staff	Manager
Staff Table	
Name	Instrument(s) Qualified
Eric Warren	Piano
Iolo Holloway	Piano
Shane Holden	Piano
Izaac Beard	Piano
Raymond Christian	Piano
Marie Sanderson	Drums
Dylon Haney	Violin
Dorothy Keenan	Violin
Scarlett Thompson	Trumpet
Mobile No.	Password
91231234	*****
81232412	*****
85123451	*****
85352321	*****
86341234	*****
87534212	*****
964212231	*****
874231451	*****
98231221	*****
	edit
	delete
	copy
9 items found, displaying 1 to 9.	First
	Prev
1	Next
	Last

Figure 10: IT Administrator View, Edit and Delete Staff Account Page



Staff	Manager
Manager Table	
Name	Email
James Sheldon	jamessheldon@gmail.com
Tommy Shelby	tommyshelby@gmail.com
Mobile No.	Password
95151234	*****
91432421	*****
	edit
	delete
	copy
2 items found, displaying 1 to 2.	First
	Prev
1	Next
	Last

Figure 11: IT Administrator View, Edit and Delete Manager Account Page

3.2 Functional Requirements

This section consists of all the functional requirements that capture the intended behaviour of the system.

FR1: The system shall display job assignments in a weekly schedule and overall workload in the landing page to the staff.

FR2: The system shall allow staff to select and submit their available dates with the respective time slots up to 5 weeks in advance.

FR3: The system shall allow staff to reject jobs allocated to them from Thursday to Friday each week to send a prompt to notify the manager.

FR4: The system shall allow staff who are qualified to teach more than one instrument to indicate their preference on which lesson they prefer to teach.

FR5: The system shall display information about the overall workload allocated for different instruments and highlight the top three staff with highest workload in the landing page to the manager.

FR6: The system shall allow managers to allocate jobs to staff in 30 minute sessions while checking if the staff exceeded four consecutive hours of work without rest and an available studio is allocated to the job.

FR7: The system shall allow managers to view staff's workload such as jobs assigned, workload allocated in hours and availability.

FR8: The system shall allow IT administrators to create, read, update and delete (CRUD) manager and staff accounts.

3.3 Use Case Model

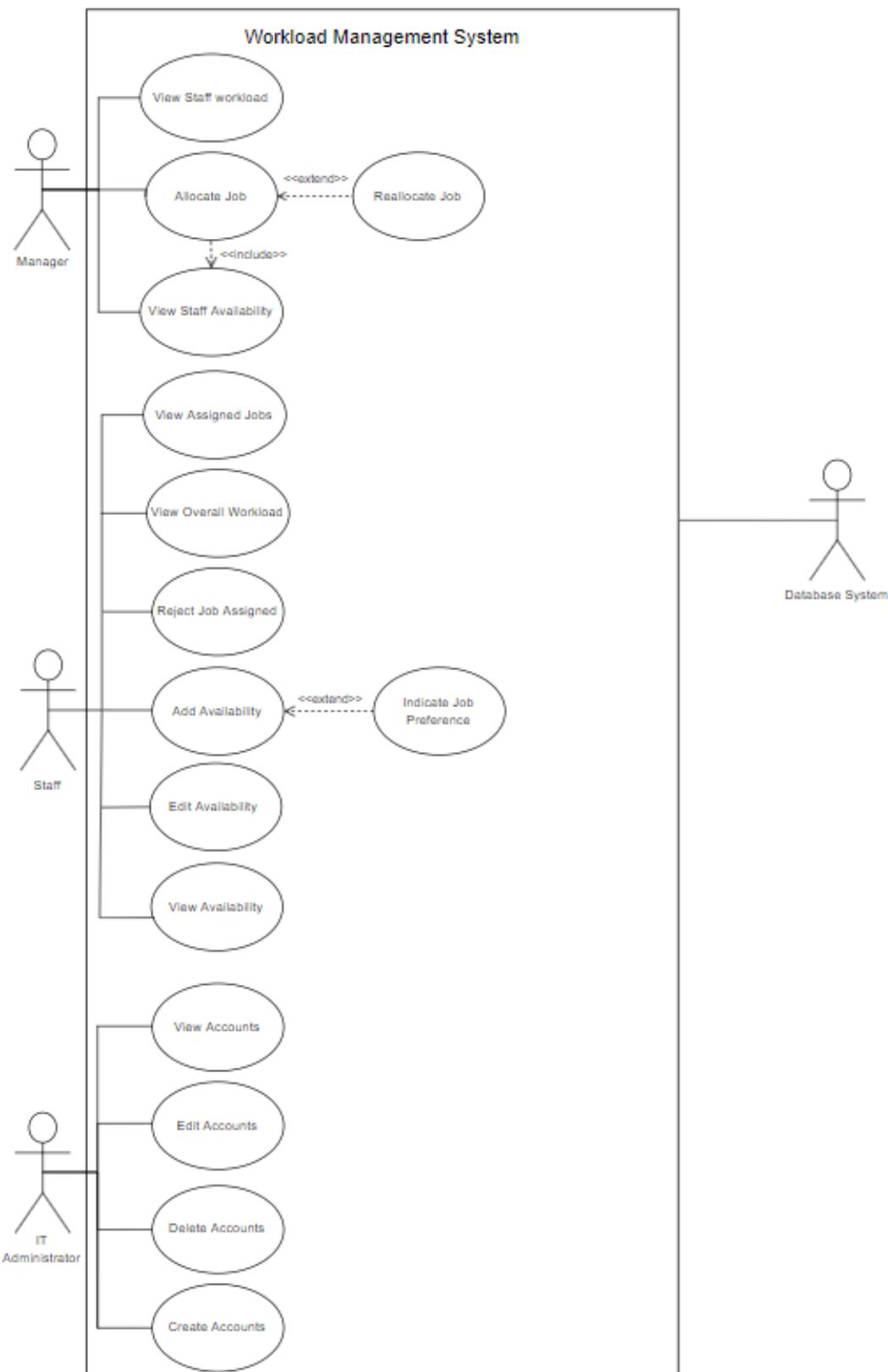


Figure 12: Workload Management System Use Case

3.4 Non-functional Requirements

The non-functional requirements include the following:

NFR1: The system shall present content in English.

NFR2: The system shall be designed with responsive design and mobile-friendly for all types of mobile devices such as mobile phones and tablets.

NFR3: The users shall be able to use all the system functions after one hour of training. After this training, the average number of errors made by experienced users shall not exceed two per hour.

3.4.1 Performance Requirements

The performance requirements include the following:

NFR4: The server will recover after experiencing a crash or denial of service within 24 hours

3.4.2 Safety and Security Requirements

The safety and security requirements include the following:

NFR5: Confidential data stored in the database shall be encrypted with AES-256.

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

NFR7: The system shall have multi-factor authentication (MFA).

NFR8: The system shall enforce strong password policy where passwords have to contain:

- At least 1 special character/ symbol
- At least 8 characters (Mixture of uppercase and lowercase)
- At least 1 number
- Change password every 90 days

NFR9: The data stored in the database shall be backed up to prevent data loss.

NFR9: The website shall have a Secure Sockets Layer (SSL) certificate to secure communications between the web server and client.

4 Project Estimation and Plan

4.1 Software Estimation

The software estimation comprises the estimation of size and estimation of effort. The process of size estimation is shown in the following steps 1-4.

Therefore, the calculation of Use Case Points (UCP) = UUCW * TCF * EF

$$= 101 * 1.0 * 0.77$$

$$= 77.77$$

= 78 (nearest whole number)

The effort was estimated using UCP where an estimation of 20 hours was given to each UCP. The estimated effort in developer-hours of a project of size 78 UCP = 20 * 78

$$= 1560 \text{ hours}$$

Estimation of Size

Step 1: Calculate unadjusted weight of use cases

Table 1 shows the calculation of unadjusted weight of use cases and is summarized as such:

- 12 simple and 3 average use cases
- Unadjusted weight of use cases = 12 (5) + 3 (10) = 90

Use Case Number	Use Case Name	Total Transactions/Use Case	Complexity of a Use Case
1	View Staff Workload	1	5
2	Allocate Job	6	10
3	Reallocate Job	5	10
4	View Staff Availability	3	5
5	View Assigned Jobs	2	5
6	View Overall Workload	2	5
7	Reject Job Assigned	4	10
8	Add Availability	2	5
9	Indicate Job Preference	2	5
10	Edit Availability	2	5
11	View Availability	2	5
12	View Accounts	2	5
13	Edit Accounts	2	5
14	Delete Accounts	2	5
15	Create Accounts	2	5
		Total unadjusted use case weight	90

Table 1: Unadjusted Weight of Use Case

Step 2: Identifying Actors

Table 2 shows the calculation of unadjusted weight of actors and is summarized as such:

- 1 average and 3 complex actors
- Unadjusted weight of actors = $1(2) + 3(3) = 11$

Actor	weight	count	product
Simple	1	0	0
Medium	2	1	2
Complex	3	3	9
Total		11	

Table 2: Unadjusted Weight of Actors

Therefore, the Total Unadjusted Points (UUCW) = $90 + 11 = 101$

Step 3: Determining the Technical Complexity Factors (TCF)

Table 3 shows the calculation of degree of influence for TCF and is summarized as such:

- Degree Of Influence (DI) = 40

Factor	Description	Weight	Assesment	Product
T1	Distributed system	2	0	0
T2	Response time/performance objectives	1	1	1
T3	End-user efficiency	1	2	2
T4	Internal processing complexity	1	3	3
T5	Code reusability	1	5	5
T6	Easy to install	0.5	4	2
T7	Easy to use	0.5	4	2
T8	Portability to other platforms	2	5	10
T9	System maintenance	1	4	4
T10	Concurrent/parallel processing	1	1	1
T11	Security features	1	5	5
T12	Access for third parties	1	0	0
T13	End user training	1	5	5
			Total	40

Table 3: Degree Of Influence (TCF)

Therefore, the TCF = $0.6 + 0.01 \times 40 = 1.0$

Step 4: Environment Factors (EF)

Table 4 shows the calculation of degree of influence for EF and is summarized as such:

- Degree Of Influence (DI) = 21

Environment	weight	Assessment	Product
Familiar with Development Process	1.5	4	6
Part time workers	-1	0	0
Analyst capability	0.5	3	1.5
Application experience	0.5	3	1.5
Object oriented experience	1	4	4
Motivation	1	3	3
Difficult programming language	-1	1	-1
Stable requirements	2	3	6
	Total		21

Table 4: Degree Of Influence (EF)

Therefore, the EF = $1.4 + (-0.03 \times 21) = 0.77$

4.2 Project Management

Figure 13 shows the Gantt chart that illustrates the project schedule.

GANTT CHART

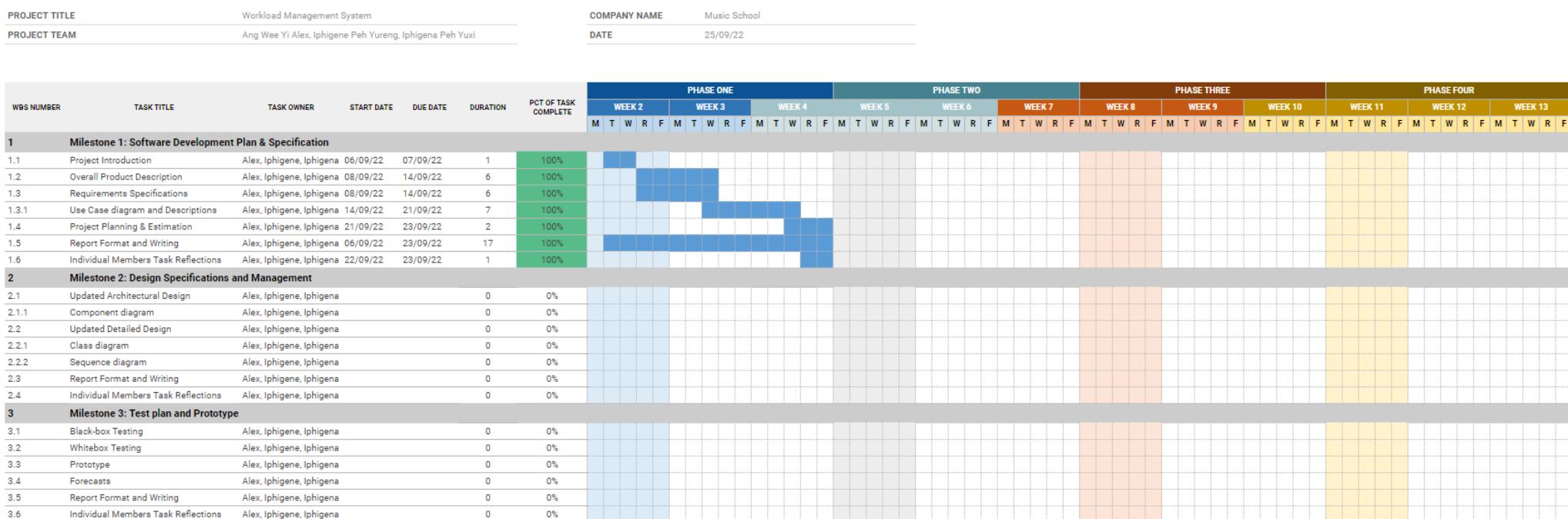


Figure 13: Gantt Chart

5 Individual Members Task and Reflections

Ang Wee Yi, Alex

For milestone one, I wrote down the meeting minutes during the client meeting and summarised the main points we have gathered at the end of the meeting. Using the meeting minutes and the project description, I was able to identify the product functionality and map them to the functional requirements for this project. In addition, during the drawing of the use case model, I have learnt to simplify my diagram to allow higher level users to understand better. Furthermore, I contributed to the use case description, system architecture diagram, calculation for the software estimation and user interface prototype. Throughout this milestone, I realised the importance of requirement specifications as there are many dependencies based on the functional requirements and this would cause a domino effect on the other tasks.

If given another chance, I would clarify my doubts on the use case model during the labatorial to avoid confusion and re-doing the use case model many times. In addition, project management could be done better to identify and allocate the tasks that can be done concurrently so that individually, we can work on different parts of the project more efficiently. As a team, we should have held more group discussions to align our views and perspectives to complete the tasks.

Iphigene Peh Yureng

In milestone one, I gathered the functional and non-functional requirements from the client meeting. In addition, I completed the user interface prototype for the staff, use case model, software estimation and plan, and use case descriptions. I learned the importance of the milestone one report and work done in terms of accountability in the real-world context. This report and work completed are imperative in software engineering and has helped to hone my project management skill.

I believe that we would be able to complete the use case diagram much earlier and accurately at the beginning of the assignment, given a better understanding of the use case model requirements based on our professor's feedback and guidance. As a team, we can work better by having a deeper understanding of the requirements and expectations, and clarifying any questions we have when in doubt.

Iphigena Peh Yuxi

For milestone one, I was in charge of asking questions and recording the client meeting session. Additionally, I have contributed to finishing the report, use case model and description, user interface prototype, and software estimation and plan. I have learned how to apply the knowledge gained from lectures and weekly RAT exercises into practice for this project. This project has corrected my misconception that software engineering is just about coding and development and allowed me to understand the principles of software

engineering, which encompass designing, developing, maintaining, testing, and evaluating software.

If given another chance, I believe that asking better phrased questions during the client meeting session would be more beneficial to the team as we would then receive a more clear and concise answer from the client. Moreover, as a team, I feel that we can work more on our communication because I realized that everyone has a different perspective on the same thing. I suggest that we can organise more group discussions, which could allow us to share our views and suggestions with each other, which could in turn make the team more efficient.

6 Appendix A – Use Case Descriptions

Use Case ID:	UC-1
Use Case Name:	View Staff Workload
Description:	This View Staff Workload use case allows the Manager to view the staff workload on the landing page.
Primary Actor:	Manager
Preconditions:	Manager is logged into the system.
Postconditions:	Staff Workload was displayed successfully.
Main Success Scenarios:	<ul style="list-style-type: none"> 1. System displays the workload in hours allocated to each type of instrument 2. System displays the top three staff with the lowest workload. 3. System displays staff with over 40 hours of jobs allocated.
Alternative Scenarios:	<p>2a. No staff with over 40 hours of jobs allocated</p> <p>2a1. System displays an error message saying no staff with over 40 hours of jobs allocated.</p>

Use Case ID:	UC-2
Use Case Name:	Allocate Job
Description:	The Allocate Job use case allows the Manager to allocate job and view information about the staff availability and other relevant information. Hence, the <<include>> relationship with the View Staff Availability use case.
Primary Actor:	Manager
Preconditions:	Manager is logged into the system.
Postconditions:	System allows managers to view staff assigned jobs after they have allocated the job.
Main Success Scenarios:	<ul style="list-style-type: none"> 1. Manager chooses to allocate a job. 2. System displays the job allocation page. 3. Manager chooses to view staff availability. 4. System displays staff availability. 5. Manager selects the type of lesson. 6. Manager selects the date and time slots. 7. Manager selects the studio. 8. Manager selects the staff. 9. Manager submits the information. 10. System adds the job into the database, and displays a confirmation message.
Alternative Scenarios:	<p>7a. No studios available for allocation.</p> <p>7a1. System displays an error message that no studio is available for the selected time slot</p> <p>8a. No staff available for allocation</p> <p>8a1. System displays an error message that no staff is available for the selected time slot</p>

Software Development Plan & Specification

	10a. Staff exceeds the four hour consecutive work without 1 hour of rest 10a1. System displays an error message that the staff has exceeded the workload limit without rest.
--	---

Use Case ID:	UC-3
Use Case Name:	Reallocate Job
Description:	The Reallocate Job use case allows the Manager to only reallocate the jobs that were rejected by the staff. Hence, the <<extend>> relationship with the Allocate Job use case.
Primary Actor:	Manager
Preconditions:	Manager is logged into the system. Staff has rejected the job that was previously assigned by the manager.
Postconditions:	The rejected job is reallocated successfully.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. System reflects the rejected jobs in the database. 2. Manager chooses to reallocate the rejected job(s). 3. Manager selects the job to allocate. 4. System displays staff availability 5. Manager allocates the job based on staff availability. 6. System updates the job in the database, and displays a confirmation message.
Alternative Scenarios:	4a. No staff is available for the manager to assign the job. 4a1. System displays an error message saying there is no staff available 6a. Staff exceeds the four hour consecutive work without 1 hour of rest 6a1. System displays an error message that the staff has exceeded the workload limit without rest.

Use Case ID:	UC-4
Use Case Name:	View Staff Availability
Description:	The View Staff Availability use case allows the Manager to view the staff availability for job allocation.
Primary Actor:	Manager
Preconditions:	Manager is logged into the system.
Postconditions:	Staff availability was displayed successfully. System allows managers to view staff preference after they have viewed the staff availability.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. Manager chooses to view staff availability. 2. System extracts the list of staff availability, assigned jobs and job preference from the database. 3. System displays the list of staff availability, assigned jobs and job preference.
Alternative Scenarios:	2a. No staff availability is submitted into the database. 2a1. System displays an error message saying no staff available and provides the reason.

Software Development Plan & Specification

Use Case ID:	UC-5
Use Case Name:	View Assigned Jobs
Description:	The View Assigned Jobs use case allows the Staff to view the jobs assigned by the manager.
Primary Actor:	Staff
Preconditions:	Staff is logged into the system. Manager has already assigned the job(s) to the staff.
Postconditions:	Assigned jobs are displayed successfully.
Main Success Scenarios:	<ul style="list-style-type: none"> 1. Staff chooses to view assigned jobs by the manager. 2. System extracts the list of jobs assigned to the staff. 3. System displays the assigned jobs
Alternative Scenarios:	

Use Case ID:	UC-6
Use Case Name:	View Overall Workload
Description:	The View Overall Workload use case allows the staff to view their workload for the current month.
Primary Actor:	Staff
Preconditions:	Staff is logged into the system. System has reflected jobs assigned by managers.
Postconditions:	Overall workload for the staff is displayed successfully.
Main Success Scenarios:	<ul style="list-style-type: none"> 1. Staff chooses to view their overall workload. 2. System extracts the list of jobs assigned to the staff. 3. System displays staff workload for the current month.
Alternative Scenarios:	

Use Case ID:	UC-7
Use Case Name:	Reject Job Assigned
Description:	The Reject Job Assigned use case allows the Staff to reject the job assigned by the manager.
Primary Actor:	Staff
Preconditions:	Staff is logged into the system. Job was previously assigned by managers and recorded into the database.
Postconditions:	Job assigned to staff is rejected successfully.
Main Success Scenarios:	<ul style="list-style-type: none"> 1. Staff chooses to reject jobs assigned. 2. System extracts the jobs assigned to the staff from the database. 3. System displays jobs assigned to the staff. 4. Staff selects the job to reject. 5. System display warning message to staff

Software Development Plan & Specification

	6. System record rejected jobs into the database.
Alternative Scenarios:	4a. Unable to reject job assigned by manager 4a1. System displays an error message saying unable to reject the job and provides the reason.

Use Case ID:	UC-8
Use Case Name:	Add Availability
Description:	The Add Availability use case allows the staff to add their availability up to 5 weeks ahead of time.
Primary Actor:	Staff
Preconditions:	Staff is logged into the system.
Postconditions:	Staff availability was indicated in the system successfully. System allows staff to indicate their job preference.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. Staff chooses to add availability. 2. System displays the available time slots in 30 mins intervals. 3. Staff selects the time slots. 4. Staff submit information. 5. System displays a confirmation message.
Alternative Scenarios:	

Use Case ID:	UC-9
Use Case Name:	Indicate Job Preference
Description:	The Indicate Job Preference use case allows the staff to indicate the instruments they want to teach for the week if they have more than one instrument qualification. If the staff only has one qualification, they will not be allowed to indicate their job preference. Staff indicates their preference while adding their availability. Hence, the <<extend>> relationship with the Add Availability use case.
Primary Actor:	Staff
Preconditions:	Staff is logged into the system. Staff have more than one instrument qualification. Staff is adding their availability.
Postconditions:	Job preference is indicated in the system successfully.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. Staff is adding their availability. 2. System displays the instruments according to the staff qualification. 3. Staff selects the instrument. 4. Staff submits information. 5. System displays a confirmation message.
Alternative Scenarios:	

Software Development Plan & Specification

Use Case ID:	UC-10
Use Case Name:	Edit Availability
Description:	The Edit Availability use case allows the staff to edit their availability up to 5 weeks ahead of time.
Primary Actor:	Staff
Preconditions:	Staff is logged into the system. Staff has viewed their availability.
Postconditions:	Staff availability was updated in the system successfully. System allows staff to indicate their job preference.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. Staff selects the modify option next to the specific date. 2. System displays a popup with time slots selected previously. 3. Staff modifies the information. 4. Staff submit information. 5. System modifies the availability in the database, and displays a confirmation message.
Alternative Scenarios:	

Use Case ID:	UC-11
Use Case Name:	View Availability
Description:	The View Availability use case allows the staff to view their availability up to 5 weeks ahead of time.
Primary Actor:	Staff
Preconditions:	Staff is logged into the system.
Postconditions:	Staff availability was displayed successfully.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. Staff chooses to view availability. 2. System extracts their availability from the database. 3. System displays their availability.
Alternative Scenarios:	3a. Staff did not submit availability. 3a1. System displays a message saying no availability was added.

Use Case ID:	UC-12
Use Case Name:	View Accounts
Description:	The View Accounts use case allows the IT administrators to manage staff and manager accounts by viewing the accounts.
Primary Actor:	IT administrator
Preconditions:	IT administrator is logged into the system.
Postconditions:	Accounts are displayed successfully.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. IT administrator chooses to view accounts. 2. System extracts a list of accounts from the database.

Software Development Plan & Specification

	3. System displays the list of accounts available.
Alternative Scenarios:	2a. No accounts are created for the IT administrator to view. 2a1. System displays an error message saying no accounts created.

Use Case ID:	UC-13
Use Case Name:	Edit Accounts
Description:	The Edit Accounts use case allows the IT administrators to manage staff and manager accounts by editing the account information.
Primary Actor:	IT administrator
Preconditions:	IT administrator is logged into the system. IT administrator has viewed the list of accounts created.
Postconditions:	Account was modified successfully.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. IT administrator chooses to edit an account. 2. IT administrator selects the edit option next to the account. 3. System displays a popup with details about the selected account. 4. IT administrator edits the information. 5. IT administrator submits the information. 6. System modifies the account information in the database, and displays a confirmation message.
Alternative Scenarios:	

Use Case ID:	UC-14
Use Case Name:	Delete Accounts
Description:	The Delete Accounts use case allows the IT administrator to manage the staff and managers account by deleting the accounts.
Primary Actor:	IT administrator
Preconditions:	IT administrator is logged into the system. IT administrator has viewed the list of accounts created.
Postconditions:	Account was deleted successfully.
Main Success Scenarios:	<ol style="list-style-type: none"> 1. IT administrator chooses to delete an account. 2. IT administrator selects the delete option next to the account. 3. System prompts for delete account confirmation. 4. IT administrator confirms by clicking delete. 5. System deletes the account in the database, and displays a confirmation message.
Alternative Scenarios:	

Software Development Plan & Specification

Use Case ID:	UC-15
Use Case Name:	Create Accounts
Description:	The Create Accounts use case allows the IT administrator to manage the staff and managers account by creating the accounts.
Primary Actor:	IT administrator
Preconditions:	IT administrator is logged into the system.
Postconditions:	Account was created successfully.
Main Success Scenarios:	<ol style="list-style-type: none">1. IT administrator chooses to add an account.2. IT administrator selects either “Manager” or “Staff”.3. IT administrator enters the staff/manager ID.4. IT administrator enters the staff/manager name.5. IT administrator enters the staff/manager email.6. IT administrator enters the staff/manager contact information.7. If creating a staff account, IT administrator enters the qualification.8. IT administrator submit information.9. System adds the account into the database, and displays a confirmation message.
Alternative Scenarios:	<p>4a. Email already exists. 4a1. System displays an error message saying the email already exists.</p>

7 Appendix B – Data Dictionary

ExpressJS

Express.js is a back end web application framework, designed for building web applications and RESTful APIs with Node.js. It is released as free and open-source software under the MIT License.

AES256

256-bit AES encryption is a technique that uses a key length of 256 bits for this process. The AES-256 key has the mathematical equivalent of 2²⁵⁶ potential combinations because key combinations increase exponentially with key size. Using 256-bit AES encryption assures the security of your data at rest.

2FA

Two-factor authentication (2FA) is a security mechanism for identity and access management that requires two forms of identification to access services and data.

MFA

Multi-factor authentication is a type of electronic authentication in which a user is only permitted access to a website or application after successfully providing two or more pieces of evidence to an authentication mechanism.

SSL

Secure Sockets Layer (SSL), is an encryption-based Internet security protocol. It was first developed by Netscape in 1995 to ensure privacy, authentication, and data integrity in Internet communications. SSL is the predecessor to the modern TLS encryption used today.

Denial of Service

Denial of service (DoS) is a type of cyber attack that aims to disable, shut down or disrupt a network, website or service.

PF

Product Functionality (PR) is the functions that the product must perform or must let the user perform.

FR

Functional Requirements (FR) captures the intended behaviour of the system. This behavior may be expressed as services, tasks or functions the system is required to perform.

NFR

Non-functional Requirements (FR) includes generic properties of the system that focuses on system usability.