INF2001

Introduction to Software Engineering

Milestone 1: Software Specifications & Design

for

Train Services Workload Management System

Prepared by

Team's Name: P13-1

ONG ZHEN YANG	**2303279**
STEPHANIE LING KHAI-MEI	**2302967**
LIU JIAXIN	**2302951**
MUHAMMAD IRFAN NUR ILHAM BIN HILMY IRWAN	**2302890**
MOHAMMAD HAIKAL BIN MOHAMMAD ZAMRI	**2302938**
SUMAIYA SHAH	**2303102**

Lab Group: P13 - 1

Github handle: *https://github.com/GoodbyeKittyy/INF2001-Introduction-to-Software-Engineering-P13-Team-1-*

Date: *<replace with the date of submission here>*

Table of Contents

1 Introduction

Our software development product lifecycle involves developing a web-based train services workload management system accessible via desktop and mobile devices. The system is designed to streamline workload management for a train company operating five Pendolino trains between Singapore and Kuala Lumpur. providing both staff and managers with a user-friendly platform. Staff will be able to view job allocations and report availability, while managers can visualize manpower availability and assign tasks accordingly. The primary goal is to improve efficiency by reducing the difficulties of managing fluctuating staff availability and job assignments, enhancing both staff and management's visibility into work schedules and resources.

1.1 Product Scope

The scope of the train services workload management system encompasses a web-based application designed to streamline task allocation and scheduling for both staff and managers in a company. The client requirements specify that for every train assignment, there must be at least an engine driver at the locomotive, one train conductor and one assistant conductor onboard the train and one cleaner for every two passenger carriages.

The system provides each staff with a unique online dashboard based on their assigned role to view weekly job assignments, manage their availability up to 5 weeks in advance, and indicate preferences for specific tasks. Managers can use the application to monitor staff workloads, assign jobs based on staff availability and preferences, and reallocate tasks when necessary. The product will be accessible across both desktop and mobile devices, ensuring flexibility for all users.

The primary benefits of this system include enhanced efficiency in task management and more equitable workload distribution. Staff will gain greater clarity and control over their schedules, while managers can make informed decisions on task allocation based on real-time data on staff engagement and availability. The benefits associated with efficiency help reduce the risk of staff burnout and enhancing overall productivity.

1.2 Related Background Literature

The literature on workload management highlights the challenges of achieving work-life balance while retaining and attracting staff. A key strategy for this is the use of Workload Management Systems (WMS), also known as batch systems in computing, which have been in use for decades [2]. Studies show that companies are actively seeking tools to ease staff workloads and promote sustainable work-life balance cycles [3].

There are numerous existing workload management tools for individuals and businesses, including Google Calendar, Notion, Asana, and Trello, which help users manage tasks and schedules [4]. However, these platforms can be overwhelming due to their extensive functionalities, which often result in a steep learning curve. This highlights the need for simpler, more intuitive tools [4].

A proposed web-based workload management application aims to improve on existing tools by offering a more user-friendly interface. This product will allow managers to view staff workloads, assign jobs efficiently, and track availability, while staff can manage their assignments and indicate preferences or availability. The project also emphasizes usability by employing heuristic evaluation for a clean user interface [1] and adhering to Web Content Accessibility Guidelines (WCAG) to support accessibility for all users [2].

Our design stage of our software development cycle is based on Singapore's Telecommunication and Security Standards [6]. In addition, we addressed security measures by detailing our report with reference to the Cybersecurity Agency of Singapore's Security-By-Design Framework [7].

1.3 References and Acknowledgments

[1] Hieu, V., & Tai, T. (2020). Strategies and analyzing workload management of staff achievement in an organization. [Volume 4, Pages 1-6]

[2] Sfligoi, I. (2008). An update on the scalability limits of the Condor batch system. Journal of Physics: Conference Series, 119, 062044. [DOI: 10.1088/1742-6596/119/6/062044]

[3] Lupu, I., Ruiz-Castro, M., & Leca, B. (2020). Role distancing and the persistence of long work hours in professional service firms. Organization Studies, advance online publication. [DOI: 10.1177/0170840620934064]

[4] Indeed. (2023, December 2). Managing a team workload effectively (With steps and tips). Retrieved September 23, 2023, from https://www.indeed.com/career-advice/career-development/managing-team-workload.

[5] Mulyani, S., Salameh, A. A., Komariah, A., Timoshin, A., Hashim, N. A. A. N., Fauziah, R. S. P., ... & Ul Din, S. M. (2021). Emotional regulation as a remedy for teacher burnout in special schools: Evaluating school climate, teacher's work-life balance and children behavior. Frontiers in Psychology, 12, 655850. [DOI: 10.3389/fpsyg.2021.655850]

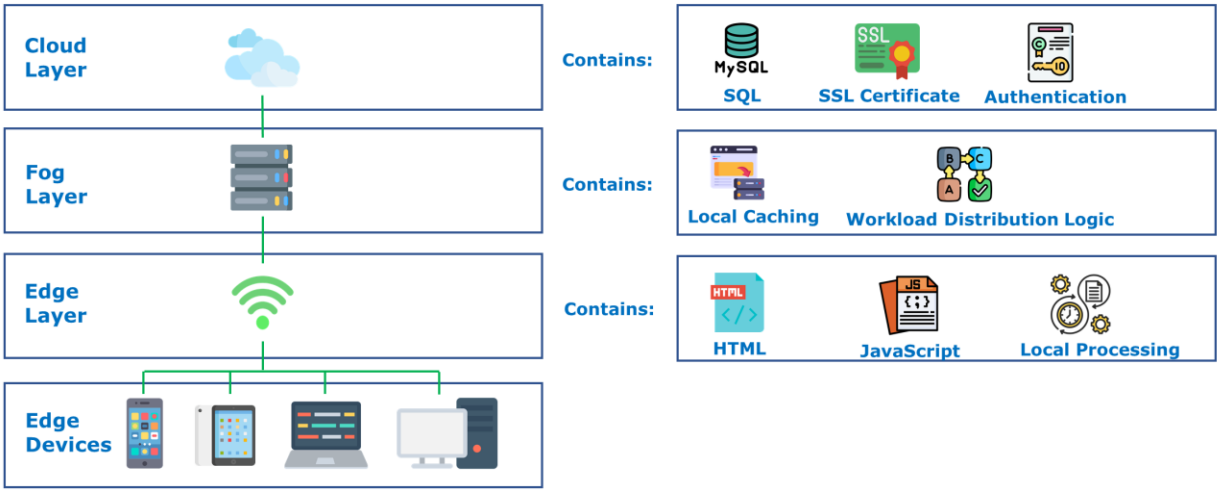
[6] IMDA. (2023, May 3). Telecommunication and security standards. IMDA Telecommunication.

[7] Cyber Security Agency of Singapore. (2017, November 9). Security-by-Design Framework. CSA.

2 Overall Description

The Train Services Workload Management System is a web-based platform that helps staff achieve a better work-life balance. It allows users to input their job availability and preferences, and its architecture is composed of several components that define the sub-system interfaces, connections, and external interfaces. The system will be built using modern web development methodologies inclusive but not limited to HTML and JavaScript for the user interface (boundary classes), and PHP for the server-side related management (MySQL, Cloud Computing e.g.).

Train Services Workload Management System



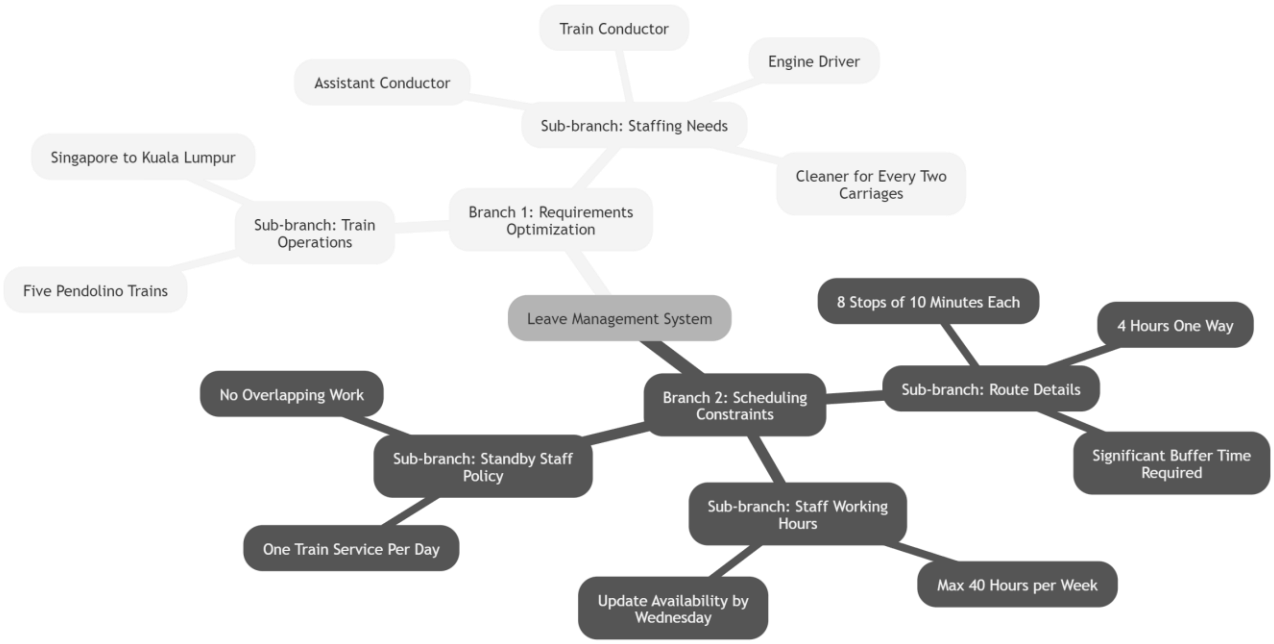
Edge to Fog: When a user inputs availability or preferences via their edge device, the data flows to the fog layer for intermediate processing and caching. For example, regional servers might immediately adjust workloads based on this data.

Fog to Cloud: More complex or large-scale tasks (like recalculating work-life balance metrics for many employees) flow up from the fog to the cloud. The cloud performs deep data processing, stores long-term data, and manages security.

Cloud to Fog: Processed data or updated schedules return to the fog layer for faster distribution to the edge devices.

Fog to Edge: The fog layer then pushes the processed data or decisions back down to the user devices, where users can see their updated schedules or workload adjustments in real-time.

The system also serves as a leave management system, and its implementation is influenced by certain assumptions and dependencies. Speaking of which, we also want to highlight that this system will optimize the following requirements based on active changes. This is based on how we want to optimize the system with reference to what we learned from the client meeting.



1. The client operates five Pendolino trains between Singapore and Kuala Lumpur, all trains must be assigned and fulfilled every month.

2. There must be one engine driver in the locomotive, one train conductor and one assistant conductor onboard the train and one cleaner for every two passenger carriages.

3. Each route is 4 hours one way and will always have 8 stops, each lasting approximately 10 minutes. Each route must be filled with significant buffer time before the same train with the same staff can be scheduled for another train assignment on the same day.

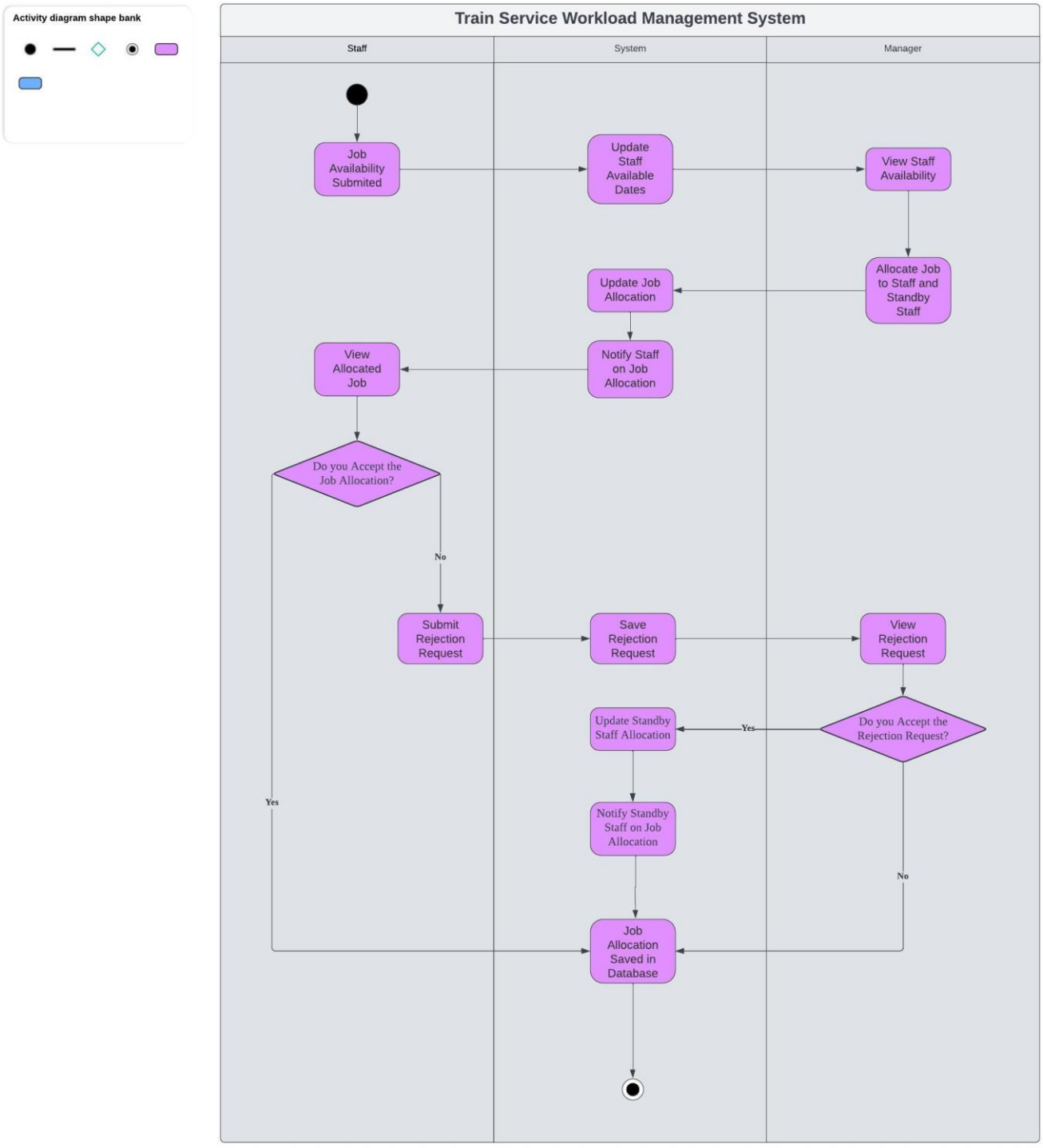
3. Each of the staff has a maximum of 40 working hours per week.

4. Workload allocation planning begins every Thursday. Therefore, employees must update their availability in the system by Wednesday to be included in the planning process.

5. Standby staff cannot be a working staff for another train or the same day AND cannot be standby for more than one train service on the same day.

6. Staff will be able to see their job allocation details based on their role. For example, if they are a train conductor, they can see the direction, departure station etc.

2.1 Product Overall Functionality



System Features Overview

The system is designed to manage staff job assignments, workloads, and availability. It caters to three types of users: staff, managers, and IT administrators. The key features of the system can be summarized as follows:

Staff Features

- **PF1**: The product shall represent a centralized application implementation allowing staff to visualize their job assignments, schedules and workloads on a weekly or monthly basis
- **PF2**: The product shall specify that employee preferences to be managed and will allow specifying up to the next 5 weeks.
- **PF3**: The product shall enable usage to modify any task assignments in the event of any emergency circumstances, pending manager's clearance.
- **PF4**: The product shall use a unique 8-character alphanumeric staff ID to authenticate staff access to the system.

Manager Features

- **PF5**: The product shall provide managers with a comprehensive view of manpower availability up to one month in advance, enabling informed job allocation decisions.
- **PF6**: The product shall facilitate efficient staff search and selection by allowing managers to filter by name, rank, qualifications, location, and flight timings.
- **PF7**: The product shall offer managers a dashboard to monitor staff workloads, highlighting potential issues and opportunities for optimization.
- **PF8**: The product shall enable managers to configure job submission deadlines and manage the job allocation process efficiently.
- **PF9**: The product shall prevent premature job allocation publishing to ensure that all jobs are assigned every Thursday.
- **PF10**: The product shall maintain a standby list of available staff to facilitate rapid allocation in case of job rejections or cancellations.
- **PF11**: The product shall allow managers to create, modify, or cancel jobs and reallocate staff accordingly, ensuring optimal resource utilization.
- **PF12**: The product shall track staff workload patterns, highlighting those with the lowest hours and those exceeding 40 hours.
- **PF13**: The product shall monitor the fulfillment of staff preferences, enabling data-driven decisions to improve job satisfaction.
- **PF14**: The product shall enable managers to export workload data into a CSV file for further analysis and reporting.
- **PF15**: The product shall allow managers to include important details for the staff based on their job allocation. For example, if the staff is a cleaner, what toilets to clean, timings and the order.

IT Admin Features

- **PF15**: The product shall provide IT administrators with a centralized platform to manage user accounts, including creating, deleting, and modifying accounts, as well as generating and assigning staff IDs.

2.2 Assumptions and Dependencies

1. Technical Assumptions

To ensure the system's functionality, the following technical assumptions have been made:

- Robust security features, including firewalls and password encryption, will safeguard the system against potential threats.
- The server will be hosted internally, with data stored outside of cloud-based environments, to maintain control over sensitive information.
- The system's design will prioritize scalability to accommodate future staff growth and expansion.
- Users are expected to have a stable internet connection and basic computer literacy skills.
- In the absence of explicit availability or preference submissions, staff will be assumed to be available for work with no specific preferences.
- Flight schedules will remain consistent on a weekly basis.
- IT administrators will utilize the system to manage user accounts, with no plans for new aircraft acquisitions or staff promotions.
- The necessary infrastructure, including servers and networking components, will be available to support the system's operation.

2. Non-Technical Assumptions

The system's functionality relies on the following non-technical assumptions:

- All users possess a working knowledge of English and will comply with security protocols, including the use of strong passwords.
- The manager has authority over all staff members and access to relevant staff information.
- Each branch will operate independently, with its own workload management system.
- Staff will receive comprehensive training and support to ensure a smooth transition to the new system.

3. Dependencies

The successful implementation and integration of the system are contingent upon:

- The active cooperation and acceptance of all users.
- The existence of a separate system to manage staff promotions and related updates, which will not be handled by this system.

3 Specific Requirements

3.1 User Interface Requirements

Authentication and User Interaction

- **UIR1:** The system shall allow access to the application restricted to logged-in users.
- **UIR2**: The system shall show Incorrect login credentials prompt users to retry.

Staff Features

- **UIR3**: The system shall show a calendar which is the central feature on staff landing pages.
- **UIR4**: The system shall the calendar providing a weekly job schedule and monthly workload overview.
- **UIR5**: The system shall enable staff to notify managers of job availability through the system.
- **UIR6**: The system shall enable staff to track the status of their requests using the system.
- **UIR7**: The system shall enable staff to be receive status updates on the requests.
- **UIR8**: The system shall enable staff to view more details on their job allocations

Manager Features

- **UIR9**: The system shall show managers' landing pages to the three staff members with the lightest workload highlighted.
- **UIR10**: The system shall show managers to identify staff with excessive weekly job assignments (over 40 hours).
- **UIR11**: The system shall show managers to review, approve, or decline staff requests.
- **UIR12:** The system shall show managers to include important details based on the staff's job allocation. For example, if the staff is a cleaner, which cabin are they assigned to.

User Interface Design

- **UIR11**: The system shall show the company logo prominently displayed throughout the application.
- **UIR12**: The system shall display a consistent color scheme applied across all pages.
- **UIR13**: The system shall display font styles and sizes uniform throughout the interface.

3.2 Functional Requirements

Staff Functionalities:

- FR1: The staff shall have access to a personalized dashboard displaying their job assignments and workload, including train, shift and other specific details.
- FR2: The staff shall be able to have the ability to switch between various calendar representations of their assigned jobs and roles.
- FR3: The staff shall propose their current availability to their manager for job assignments up to 5 weeks in advance, with the option to edit or delete their submissions.
- FR4: The staff shall have up to 5 weeks before the job assignment to indicate their preferences for the work, including the day and time.
- FR5: The staff shall be able to request job reassignments, providing a reason, with a minimum of 3 days' notice.
- FR6: The staff shall be able to cancel job assignments by Wednesday each week.
- FR7: The staff shall have a personal calendar view of their workload which they can export into a PNG image.
- FR8: The staff shall be required to acknowledge receipt of job assignments.

Manager Functionalities:

- FR9: The manager shall allocate jobs to staff members every working week.
- FR10: The manager shall have visibility into staff members' workloads and latest details.
- FR11: The manager shall be able to manually push “High Importance” notification to staff members to do their submissions.
- FR12: The manager shall be able to create new job assignments, specifying details such as standby staff.
- FR13: The manager shall be able to edit existing submissions and modify or cancel them, but the system will automatically push the job to the standby staff if this happens less than 24 hours of the working day.
- FR14: The manager shall be able to filter and view staff workloads based on various criteria.
- FR15: The manager shall maintain a standby list of staff members for job allocation, with the ability to approve or deny job reassignments.
- FR16: The manager shall be able to monitor staff members' working hours, including those exceeding 40 hours or with the lowest hours.
- FR17: The manager shall have the option to export staff workloads in PDF, Excel, and PNG formats.
- FR18: The manager shall indicate and modify the training status of all staff.

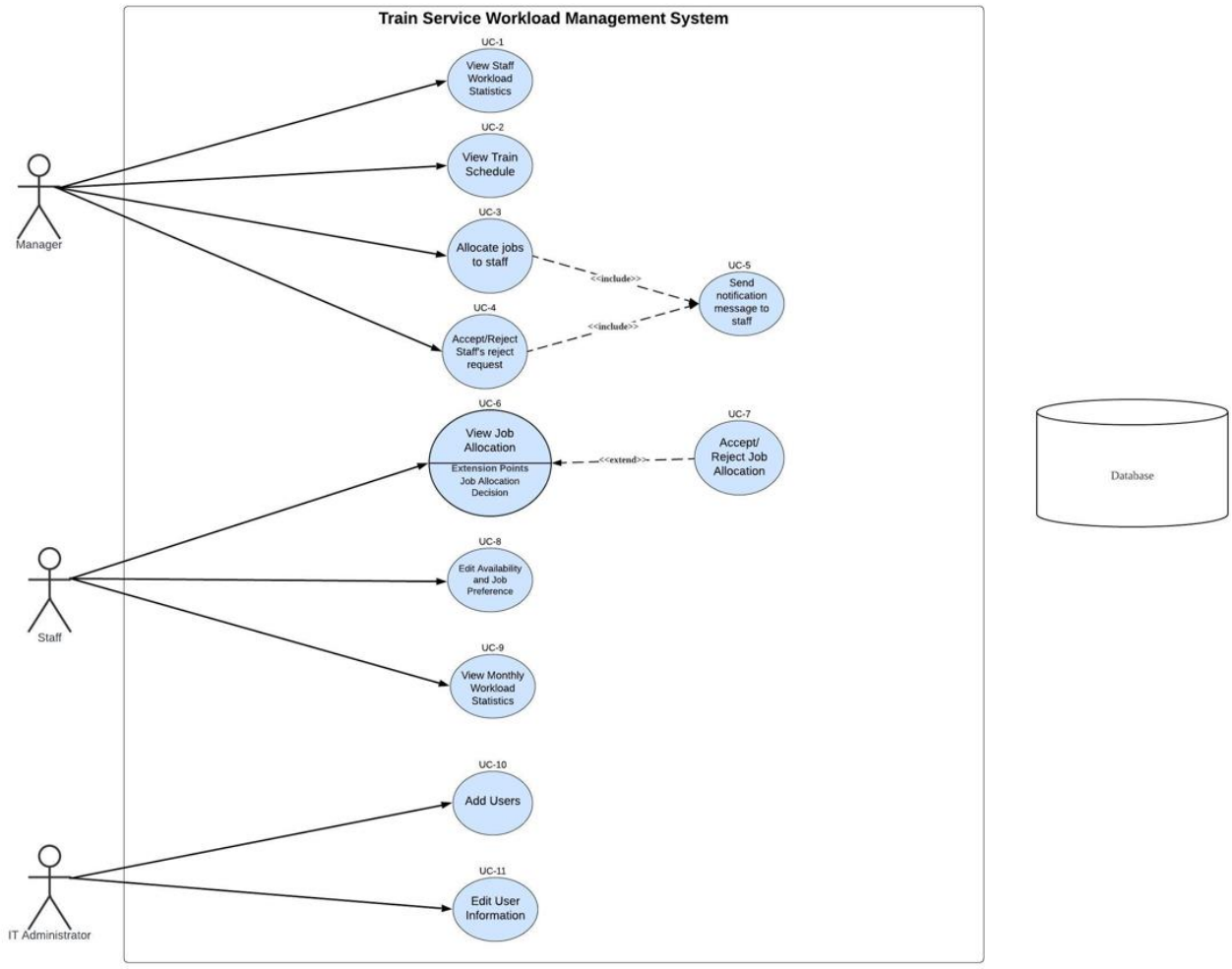
IT Administrator Functionalities:

- FR19: The IT administrator shall be authorized to create staff and manager accounts and assign them
- FR20: The IT administrator shall be authorized to edit and permanently delete any existing staff and manager account or information with no constraints.
- FR21: The IT administrator shall be able to automatically generate and assign unique non-existent staff IDs.

System Constraints:

- FR22: The system shall uniquely identify staff members using a combination of their six-digit staff ID and alphanumeric staff ID.
- FR23: The system shall accept staff availability submissions until every Wednesday for job allocations made every Thursday.

3.3 Use Case Model



3.4 Non-functional Requirements

Accessibility Requirements

- NFR1: The system must function correctly on various web browsers, including Chrome, Firefox, and Safari, to ensure compatibility and usability.
- NFR2: The system should communicate with users in their preferred language (English, 中文, Bahasa Malayu, தமிழ்) to serve a varied user base.

Performance Requirements

- NFR3: Each department (train service) will have its own independent workload management system but with the same interface.
- NFR4: The system should respond quickly to user interactions.
- NFR5: The system should support concurrent users, ranging from 10 to 2500 users.
- NFR6: The system must maintain a high uptime of at least 99.99% over any given month, with scheduled maintenance pre-informed at least 2 weeks in advance.
- NFR7: The job allocation interface must load within 3 seconds to provide a smooth and responsive user experience.
- NFR8: The system shall save availability changes in under 3 seconds and reflect them immediately in the “View job availabilities” in the Manager’s landing page.
- NFR9: The system shall retrieve and display the workload data within 2 seconds.
- NFR10: Adding a new user should take no more than 5 seconds after the submission of the form.
- NFR11: The system shall ensure that changes to user information are reflected in the database within 3 seconds.
- NFR12: The system must prevent job scheduling conflicts and notify managers of any issues in real-time.
- NFR13: The system must handle a growing number of staff and job assignments without a noticeable decline in performance.
- NFR14: The system should handle an increasing number of staff and workload data without degrading performance.

3.4.1 Safety and Security Requirements

Safety and Security Requirements

- NFR15: Every single piece of information must be encrypted to prevent unauthorized access.

- NFR16: Approving roles should use multi-factor authentication, in addition to 2-factor authentication, to ensure that approval processes are secure.

- NFR17: The system should have data recovery capabilities to restore data in case of loss or failure.

- NFR18: The system should include failover and disaster recovery mechanisms to minimize downtime during unexpected events.
- NFR19: Users should create strong passwords with at least 12 characters, including numbers and special characters, and use two-factor authentication for password changes.
- NFR20: All data transmissions should be encrypted and not transmitted in plaintext to avoid man-in-the-middle attacks.
- NFR21: Sensitive information (e.g., train crew details) must be protected from unauthorized access using encryption and role-based access control.
- NFR22: The "Add Users" functionality must be restricted to authorized IT administrators, with role-based access control. Sensitive information such as passwords must be encrypted and stored securely.
- NFR23: All job allocations and changes should be logged for auditing purposes.
- NFR24: The system must log all actions taken by IT administrators, including user creation, to ensure transparency and allow for tracking of administrative actions.

3.5 Other Requirements

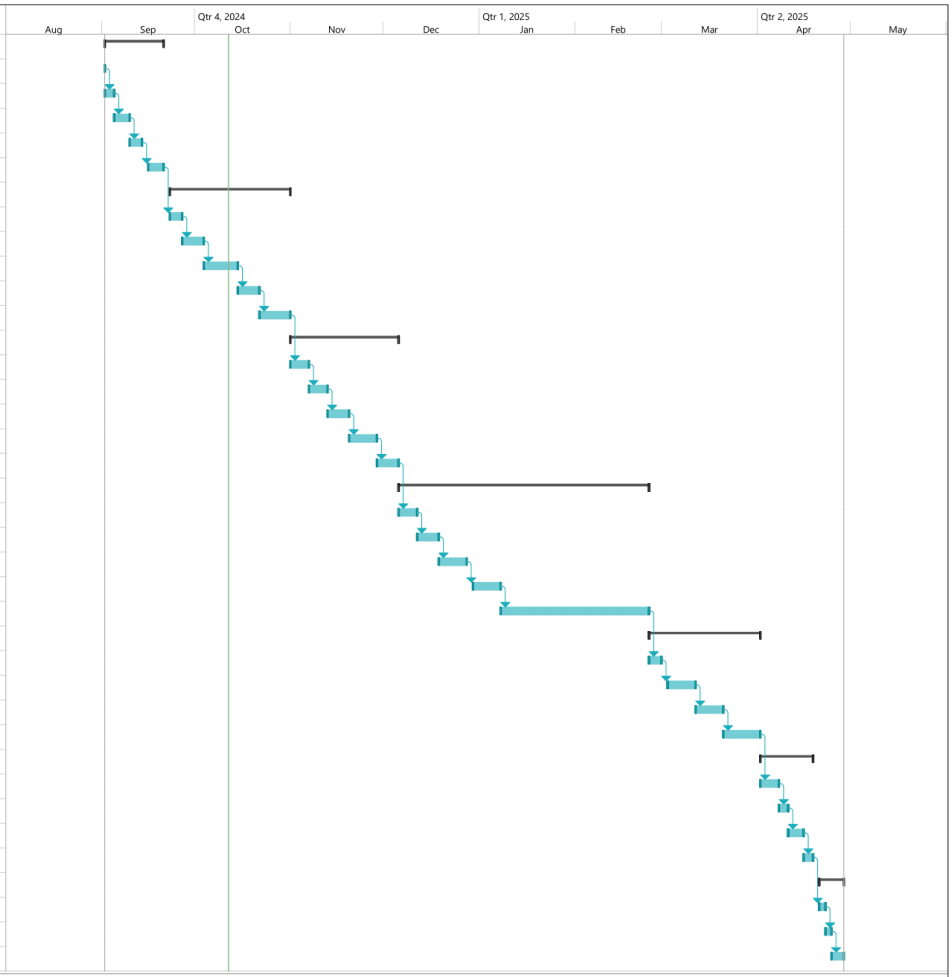
- NFR25: The interface must be intuitive and easy to use, with visual aids (e.g., colour coding or icons) to highlight scheduling conflicts or overworked staff.
- NFR26: The system shall highlight the total working hours in red when the total working hours is more than or equal to 40.
- NFR27: The user management interface should provide clear feedback on missing or incorrect fields. The system should help the administrator quickly correct any issues, such as a duplicate username.
- NFR28: The interface should be user-friendly and intuitive, with visual aids (e.g., colour coding or *icons) to highlight scheduling conflicts or overworked staff, allowing managers to assign jobs efficiently.*

- NFR29: The schedule must be displayed in an easy-to-read format, with clear options to filter by date, route, and time. The interface should be mobile-friendly.
- NFR30: The schedule data must be 100% accurate and updated in real-time to reflect any changes.
- NFR31: The workload data must be accurate, preventing inconsistencies such as missing or incorrect data.
- NFR32: The system must comply with GDPR and PDPA regulations.
- NFR33: The system should be subject to an annual security audit by an external contractor.
- NFR34: The system must adhere to all company policies and regulatory compliance requirements.
- NFR35: The system must be available 24/7 to allow managers to view workload data at any time.
- NFR36: The system's notification functionality must be available 24/7, ensuring that critical job updates or schedule changes can be communicated to staff at any time.
- NFR37: If the notification fails due to external system errors (e.g., email or SMS service provider issues), the system should provide the manager with a clear error message and allow for manual resending after the issue is resolved.
- NFR38: The system shall prevent duplicate user entries (e.g., identical usernames or emails).

4 Software Design

4.1 Gantt Chart

ID	Task Mode	Task Name	Duration	Start	Finish
1		**Project Initiation**	**15 days**	**Mon 9/2/24**	**Fri 9/20/24**
2		Project Kickoff Meeting	1 day	Mon 9/2/24	Mon 9/2/24
3		Define Project Scope and Objectives	3 days	Mon 9/2/24	Wed 9/4/24
4		Identify Stakeholders and Their Roles	3 days	Thu 9/5/24	Mon 9/9/24
5		Develop Project Charter	4 days	Tue 9/10/24	Fri 9/13/24
6		Establish Project Team	5 days	Mon 9/16/24	Fri 9/20/24
7		**Requirements Analysis**	**29 days**	**Mon 9/23/24**	**Thu 10/31/24**
8		Gather Requirements from Stakeholders	4 days	Mon 9/23/24	Thu 9/26/24
9		Conduct User Interviews and Surveys	5 days	Fri 9/27/24	Thu 10/3/24
10		Analyze Requirements and Identify Gaps	7 days	Fri 10/4/24	Mon 10/14/24
11		Develop Use Cases and User Stories	5 days	Tue 10/15/24	Mon 10/21/24
12		Create Requirements Documentation	8 days	Tue 10/22/24	Thu 10/31/24
13		**System Design**	**25 days**	**Fri 11/1/24**	**Thu 12/5/24**
14		Develop System Architecture	4 days	Fri 11/1/24	Wed 11/6/24
15		Design Database Schema	4 days	Thu 11/7/24	Tue 11/12/24
16		Create User Interface (UI) Design	5 days	Wed 11/13/24	Tue 11/19/24
17		Develop System Components and Modules	7 days	Wed 11/20/24	Thu 11/28/24
18		Create System Design Documentation	5 days	Fri 11/29/24	Thu 12/5/24
19		**Implementation**	**57 days**	**Fri 12/6/24**	**Mon 2/24/25**
20		Develop Front-end Components	4 days	Fri 12/6/24	Wed 12/11/24
21		Develop Back-end Components	5 days	Thu 12/12/24	Wed 12/18/24
22		Integrate Front-end and Back-end Components	7 days	Thu 12/19/24	Fri 12/27/24
23		Develop Database and Storage Components	7 days	Mon 12/30/24	Tue 1/7/25
24		Conduct Unit Testing and Integration Testing	34 days	Wed 1/8/25	Mon 2/24/25
25		**Testing**	**26 days**	**Tue 2/25/25**	**Tue 4/1/25**
26		Develop Test Plans and Test Cases	4 days	Tue 2/25/25	Fri 2/28/25
27		Conduct System Testing and Integration Testing	7 days	Mon 3/3/25	Tue 3/11/25
28		Conduct User Acceptance Testing (UAT)	7 days	Wed 3/12/25	Thu 3/20/25
29		Identify and Fix Defects	8 days	Fri 3/21/25	Tue 4/1/25
30		**Deployment**	**13 days**	**Wed 4/2/25**	**Fri 4/18/25**
31		Plan Deployment Strategy	4 days	Wed 4/2/25	Mon 4/7/25
32		Conduct Deployment	3 days	Tue 4/8/25	Thu 4/10/25
33		Configure and Test Deployment Environment	3 days	Fri 4/11/25	Tue 4/15/25
34		Conduct Post-Deployment Review	3 days	Wed 4/16/25	Fri 4/18/25
35		**Project Closure**	**6 days**	**Mon 4/21/25**	**Mon 4/28/25**
36		Document Lessons Learned	2 days	Mon 4/21/25	Tue 4/22/25
37		Evaluate Project Success	2 days	Wed 4/23/25	Thu 4/24/25
38		Obtain Final Acceptance from Stakeholders	2 days	Fri 4/25/25	Mon 4/28/25

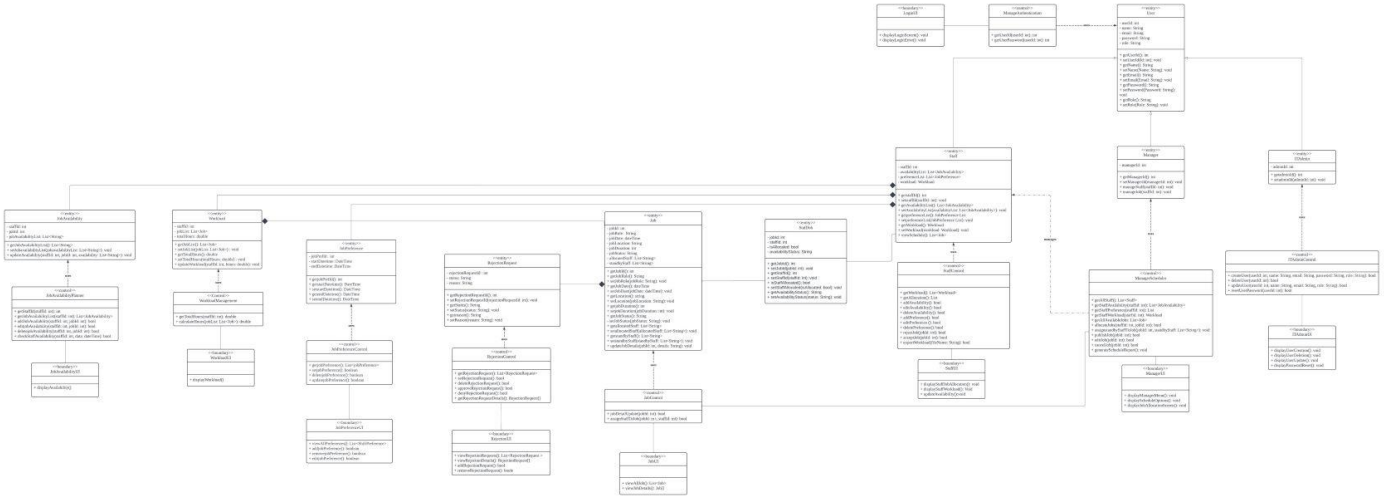


Project: softwareeng
Date: Sat 10/12/24

Task, Split, Milestone, Summary, Project Summary, Inactive Task, Inactive Milestone, Inactive Summary, Manual Task, Duration-only, Manual Summary Rollup, Manual Summary, Start-only, Finish-only, External Tasks, External Milestone, Deadline, Progress, Manual Progress

Page 1

4.2 Class Diagram



Please click here for a clearer view:

Entities: User, Manager, ITAdmin, Staff, Job, StaffJob, Workload, JobAvailability, RejectionRequest, JobPreference

Each entity holds its attributes and methods to manipulate data specific to the object. For example, the Staff entity stores details like staff ID, workload, and job preferences, interacting with other modules for job allocation and workload updates.

Control: ManageAuthentication, Manager Scheduler, ITAdminControl, StaffControl, JobControl, Workload Management, JobAvailabilityPlanner, RejectionControl and JobPreferenceControl.

The control classes handle business logic, facilitating the interaction between entities and the user interface (UI). For example, the WorkloadManagement control class computes and updates staff workload based on job assignments, while ManagerScheduler controls job scheduling, job allocation, and staff management.

Boundary: LoginUI, ManagerUI, ITAdminUI, StaffUI, JobUI, WorkloadUI, JobAvailablityUI, RejectionUI, JobPreferenceUI.

The UIs for Manager, Staff, and ITAdmin display relevant screens for their respective operations, including job allocation, workload viewing, and user management. For example, the ManagerUI displays options for scheduling and job allocation, which interact with the ManagerScheduler control class.

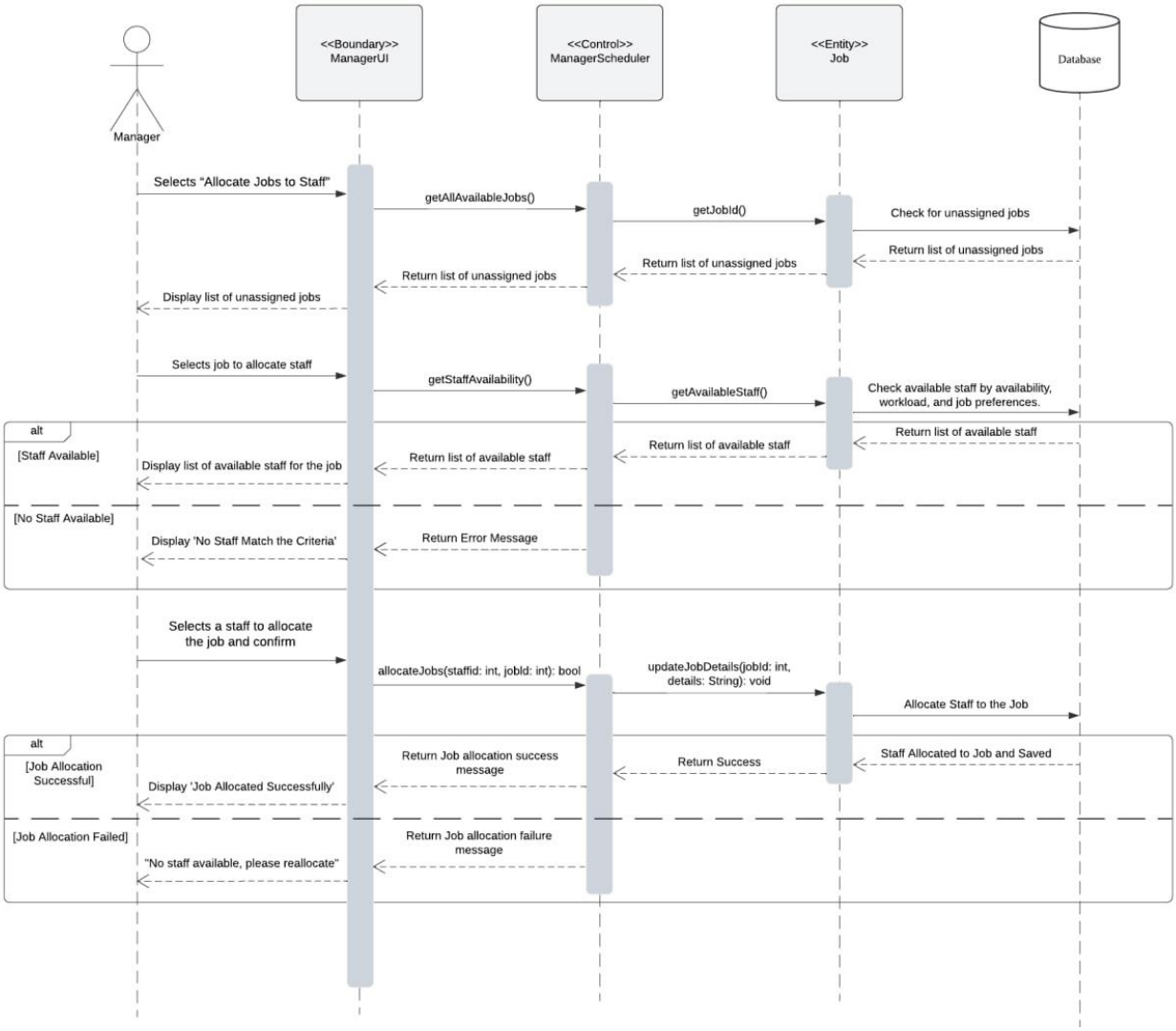
Module Communication:
Entity-Control Interaction where control classes such as JobControl and WorkloadManagement utilize the entities like Job and Workload to update the data or retrieve information. For instance, when assigning a job, the JobControl uses the Staff entity to update the staff member's allocated jobs and workload

Control-Boundary Interaction where boundary classes trigger methods in the control classes to perform actions. For example, ManagerUI will invoke ManagerScheduler methods such as allocateJobs() when the manager makes a scheduling decision.

Control-Control Interaction where control classes in the system interact with each other to perform coordinated tasks, where one control class might depend on another to complete an operation. This interaction ensures that complex business logic can be distributed across multiple controls, promoting modularity and separation of concerns. For example, ManagerScheduler and JobControl whereby ManagerScheduler is responsible for overseeing staff scheduling and job allocation. However, it delegates the specific task of assigning jobs to the JobControl class.

4.3 Sequence Diagram

1) UC-3 – Allocate Jobs to staff (Manager)



Actors:

- Manager: The actor who initiates the job allocation process.

System/Components:

- ManagerUI (Boundary): The user interface where the manager interacts with the system to allocate jobs with standby.
- ManagerScheduler (Control): This controller manages the logic between the UI and the system, handling job requests and staff availability.
- Job (Entity): This entity represents the actual job data, managing its details and allocation.
- Database: Stores all the necessary job, standby, staff, and allocation data.

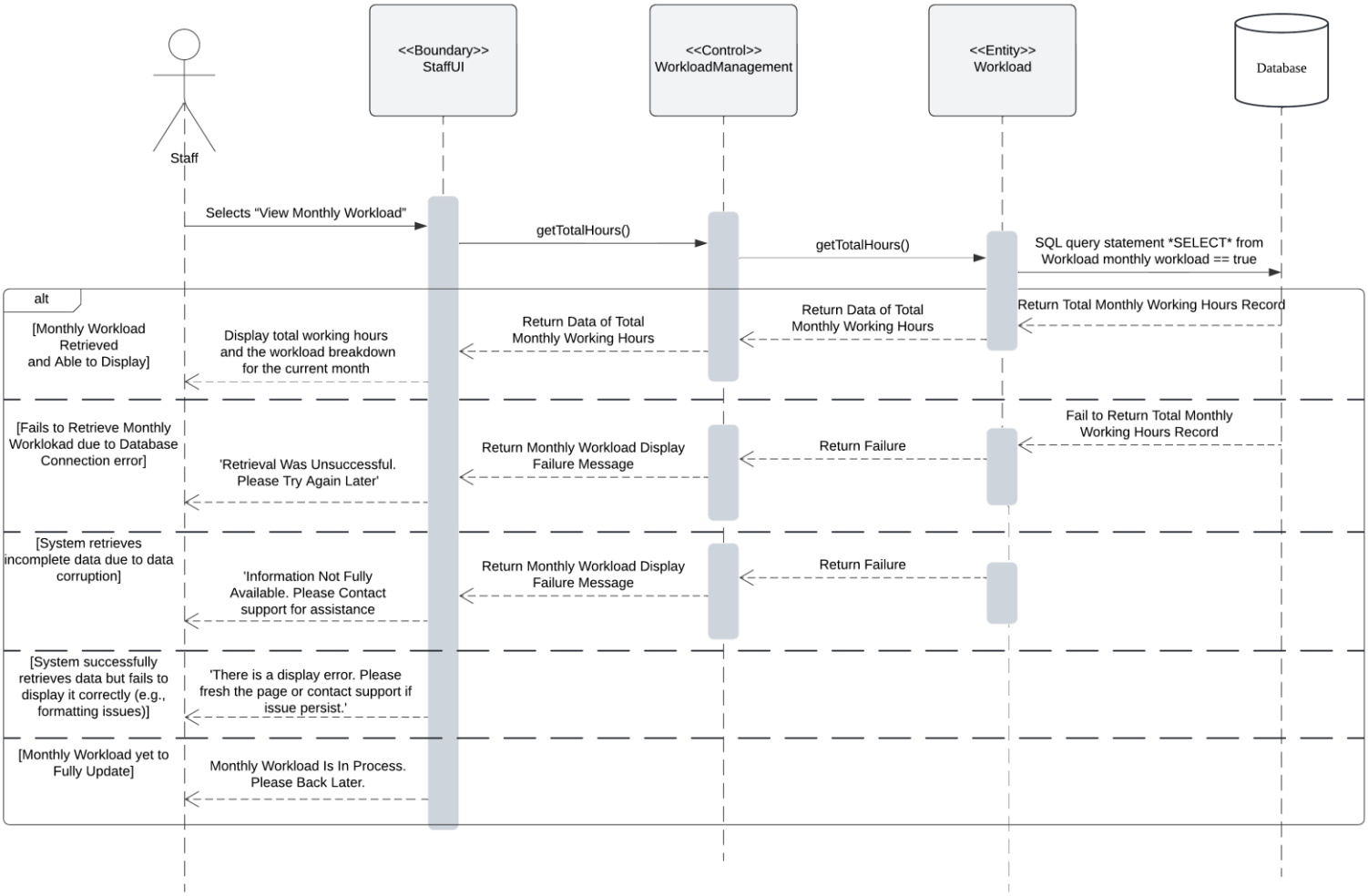
Messages and Flow:

1. Job Selection:
   - The sequence starts with the Manager selecting the "Allocate Jobs to Staff" option in the ManagerUI.
   - The ManagerUI calls the getUnassignedJobs() function in ManagerScheduler to fetch the list of unassigned jobs.
   - The Job entity then interacts with the Database using an SQL query to retrieve unassigned jobs.
   - Once retrieved, the ManagerScheduler sends the job data back to the ManagerUI, which displays the available jobs for selection.

2. Viewing Staff Availability:
   - The Manager selects a job to view the available staff for that job.
   - The ManagerUI sends a request to the ManagerScheduler through functions getStaffAvailability(), getStaffPreference(), and getStaffWorkload() to check staff availability based on several criteria (availability, preferences, workload). This also applies to the standby.
   - The Job entity then queries the Database for staff records who meet the availability criteria using another SQL statement.
   - If staff are available, their data is returned to the ManagerUI and displayed for the Manager to choose from.
   - ALT: If no staff match the availability criteria, an error message is displayed: "No Staff Match the Criteria."

3. Allocating a Job to Staff:
   - Once the Manager selects a staff member, they confirm the allocation of the job.
   - The ManagerUI sends the allocation request allocateJobs(staffId, jobId) to the ManagerScheduler.
   - The ManagerScheduler attempts to update the job details by calling updateJobDetails() to Job entity which then interacts with the Database to record the staff-job assignment. The SQL INSERT INTO statement updates the job allocation in the database.
   - Upon successful allocation, the ManagerScheduler returns a success message to the ManagerUI, which displays "Job Allocated Successfully."
   - ALT: If a staff member is overbooked or unavailable, the system will display "Staff exceeds workload, Please Reallocate"
   - ALT: If the Job Allocation encounters a system error, the system will display "Job Allocation Failed, Please Try Again"
   - These alternative paths handle potential issues during the allocation process and prompt the Manager to try again or reassign the job.

2) UC-9 – View Monthly Workload Statistics



Actors:

- Staff: The actor (user) who initiates the process of viewing their monthly workload.

System/Components:

- StaffUI (Boundary): The user interface where staff interacts with the system to view their workload.
- WorkloadManagement (Control): This controller manages the logic between the UI and the system, handling requests related to workload data.
- Workload (Entity): Represents the workload data, managing the details of the staff's monthly working hours.
- Database: Stores all the necessary workload and staff data, including total monthly working hours.

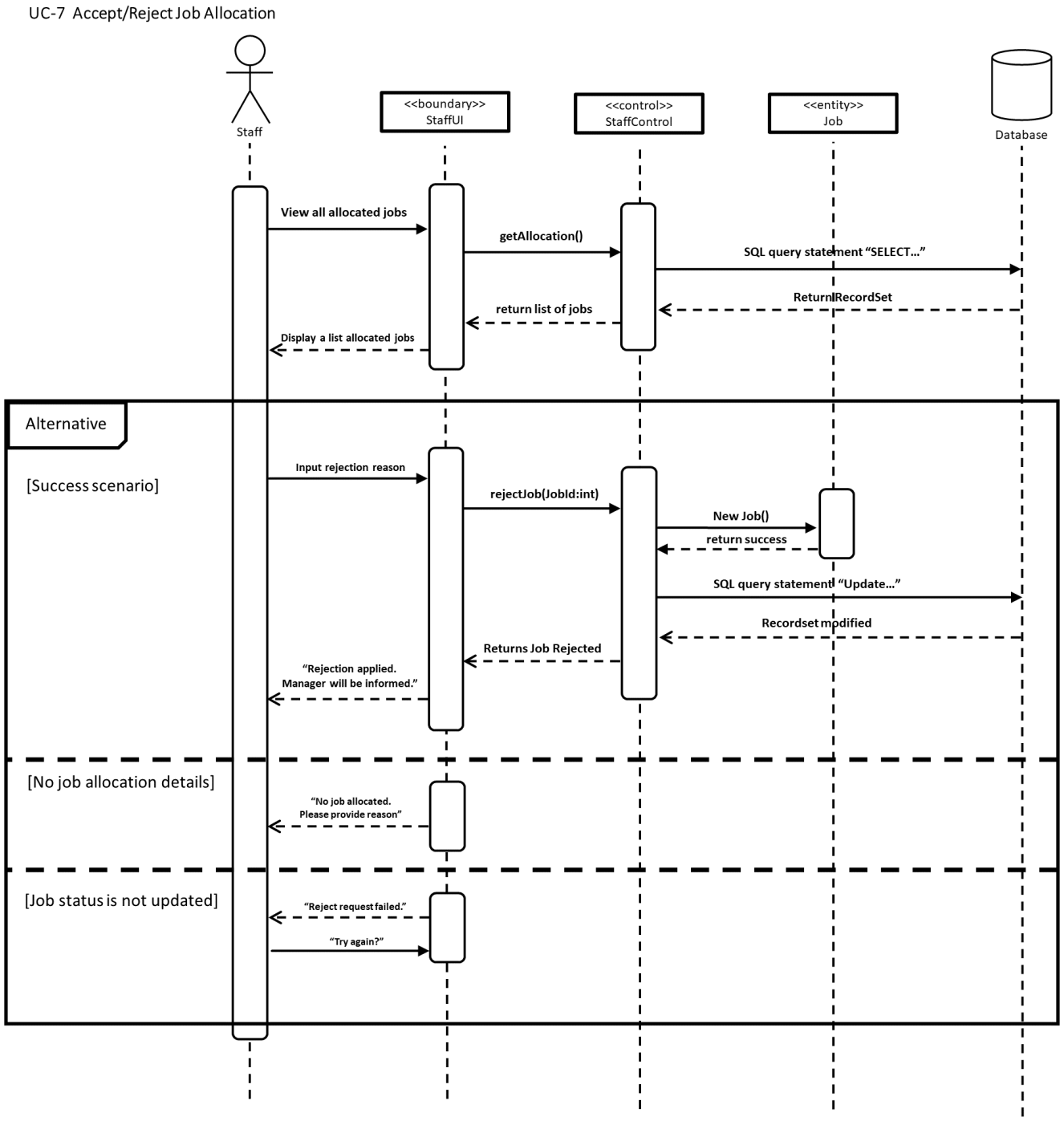
Messages and Flow:

1. View Monthly Workload:
   - The Staff starts by selecting the "View Monthly Workload" option in the StaffUI.
   - The StaffUI sends a request (getTotalHours()) to the WorkloadManagement to retrieve the total monthly working hours for the current month.
   - The WorkloadManagement then calls for getTotalHours to Workload entity.
   - The Workload entity interacts with the Database using an SQL query: "SELECT * FROM Workload WHERE monthly_workload == true".
   - The Database returns the total monthly working hours record to Workload entity, which then passes the data to the StaffUI.

2. Displaying Monthly Workload:

   - If the workload data is successfully retrieved, the StaffUI displays the total working hours and a breakdown of the workload for the current month to the Staff.
   - ALT: If the Database fails to return the total monthly working hours due to a connection error, the system returns a failure message. The StaffUI displays the message: "Retrieval Was Unsuccessful. Please Try Again Later."
   - ALT: If the system retrieves incomplete or corrupted data, it returns a failure message, and the StaffUI displays: "Information Not Fully Available. Please Contact Support for Assistance."
   - ALT: If the data is successfully retrieved but there is a display issue (e.g., formatting problems), the StaffUI displays an error message: "There is a display error. Please refresh the page or contact support if the issue persists."
   - ALT: If the monthly workload data has not been fully updated in the system yet, the StaffUI shows the message: "Monthly Workload Is in Process. Please Check Back Later."

3) UC-7 – Accept/Reject Job Allocation (Staff)



Actors:

- Staff: The user who initiates the process of rejecting a job allocation.
- StaffDashboard (Boundary): The interface that the staff uses to interact with the system.
- StaffControl (Control): The controller that manages the interaction between the boundary and entity objects.
- Job (Entity): Represents the job allocation information in the system.
- Database: Stores the job allocations and other relevant data.

Messages and Flow:

- The sequence starts with the Staff viewing all allocated jobs through the StaffDashboard. The getAllocation() method is called, retrieving a list of jobs from the Job entity using an SQL query. The results are returned to the StaffDashboard, which then displays the allocated jobs to the Staff.

Alternative Paths:

- Success Scenario: The Staff inputs a rejection reason, and StaffControl processes this with rejectJob(), which calls the Job entity to update the database. Upon success, a confirmation message is displayed to the Staff indicating that the rejection was applied, and the manager will be informed.
- No Job Allocation Details: If no job allocation details are present, the StaffDashboard displays an error message asking the Staff to provide a reason.

Job Status Not Updated: If there's an error in updating the job status, the StaffDashboard informs the Staff of the failure and prompts them to try again.

-

5 Appendix A – Use Case Descriptions

Use case ID:	UC-1
Use case Name:	View Staff Workload Statistics
Description:	The manager views a dashboard displaying staff workload statistics, helping to make informed decisions for allocating jobs. The dashboard shows the workload assigned, staff's job preference, staff's location at a particular date, and availabilities for the week.
Primary Actor:	Manager
Preconditions:	• The manager is logged in the system with appropriate rights. • The system has up-to-date workload statistics
Postconditions:	The manager effectively views the staff's workload, enabling better job allocation.

Main Success Scenarios:	1. The manager selects “View Staff Workload Statistics” from the system’s main menu. 2. The system displays the staff workload dashboard, showing completed and assigned jobs, hours worked, and available capacity. 3. The manager filters the workload statistics to focus on staff details (the workload assigned, staff’s job preference, staff’s location at a particular date, and availabilities for the week.) by what the manager chooses to view. 4. The system updates and displays the filtered statistics accordingly. 5. The manager views detailed statistics for specific staff.
Priority:	High
Alternative Scenarios: (only for system not manager)	2a. The system does not display the staff workload dashboard, showing completed and assigned jobs, hours worked, and available capacity: 2a1. If the system fails to retrieve workload data, it displays an error message. 2a2. The manager retries or contacts the support. 4a. The system fails to update and display the filtered statistics: 4a1. If the system fails to process the filtering request due to missing or corrupt data, it displays an error message, explaining that the statistic could not be updated. 4a2. The system suggests reapplying the filter, refreshing the data, or contacting support.

Non-Functional Requirements:	• NFR1: The system must function consistently across all major web browsers to ensure managers can access the workload dashboard from any platform (Chrome, Firefox, Safari). • NFR7: The system must retrieve and display workload statistics within 3 seconds to ensure a responsive and smooth experience for the manager. • NFR18: The dashboard should provide an intuitive interface with clear filtering options, enabling managers to easily navigate, filter, and analyze the workload statistics. • NFR19: The workload data must be accurate and up to date, preventing any inconsistencies such as missing or incorrect data when the manager views the dashboard. • NFR20: The system should scale to handle an increasing number of staff members and their workload data without degrading performance, ensuring it remains efficient even as the team grows. • NFR21: Only authorized managers should have access to the workload dashboard, with proper role-based access control to protect sensitive staff information. • NFR22: The system must be available 24/7, allowing managers to view workload data anytime, ensuring they can make decisions even outside regular working hours.

Use case ID:	UC-2
Use case Name:	View Train Schedule
Description:	The manager can view train schedules, including details such as routes, departure times, and staffing needs, to assist with planning job allocations.
Primary Actor:	Manager
Preconditions:	• The manager must be logged into the system • The system must have the up-to-date train schedule
Postconditions:	• The manager has access to the current train schedule, helping with staff job allocation decisions.
Main Success Scenarios:	1. The manager selects the "View Train Schedule" option from the main menu

	2. The system retrieves and displays the current train schedule 3. The manager filters the schedule by date, route, train service or time-period. 4. The system updated the schedule to reflect the filter criteria. 5. The manager selects a specific train schedule to view more details. 6. The system retrieves and displays detailed information about the selected train service, including staff requirements and route stops.
Alternative Scenarios:	2a. The system does not display the current train schedule: 2a1. If the train schedule is unavailable due to system errors, the system displays a message indicating that the schedule cannot be retrieved. The manager can refresh the schedule or contact support. 4a. The system fails to update the schedule based on the filter criteria: 4a1. If the system cannot process the filter request due to missing or incorrect data, it displays an error message explaining that the filter could not be applied. 4a2. If the refresh fails, the system displays the unfiltered schedule and continues to operate without applying the filter. 6a. The system fails to retrieve the detailed information about the selected train service: 6a1. If the system cannot retrieve the detailed train information due to a data retrieval issue, it logs the error and displays only the basic train schedule without details. 6a2. The system retries the data retrieval in the background without interrupting the current view of the train schedule.
Priority:	High

Non-Functional Requirements	• NFR8: The system should retrieve and display the train schedule within 3 seconds, ensuring fast access to schedule data even with multiple users accessing it concurrently. • NFR15: The schedule data should only be accessible by authorized users, with role-based access control in place to protect sensitive information such as train crew details. • NFR23: The schedule should be displayed in an easy-to-read format, with clear filtering options for date, route, and time. The interface should also be mobile-friendly for managers who are on-the-go and need to access schedules from their devices. • NFR24: The train schedule data must be 100% accurate and updated in real-time, reflecting any changes to routes or timings, to ensure the manager has the latest information for planning job allocations. • NFR25: The system should maintain a 99.9% uptime to ensure that managers can access train schedules at any time of day without disruption.

Use Case ID:	UC-3
Use Case Name:	Allocate Jobs to staff
Description:	The manager assigns jobs to staff based on availability, preferences, and workload. The manager is allowed to assign jobs one week at a time.
Primary Actor:	Manager
Preconditions:	• The manager must be logged into the system. • The system must have up-to-date information on staff availability, preferences, and workload data. • The job allocation is for the upcoming week only. • The manager is viewing the list of unassigned jobs for the upcoming week.
Postconditions:	• The manager successfully allocates jobs to staff for the week, ensuring optimal workload distribution. • The system updates the job allocation records in the database.
Main Success Scenarios:	1. The manager selects “Allocate Jobs to Staff” from the main menu. 2. The system displays a list of available jobs for the upcoming week. 3. The manager selects a job to allocate staff.

	4. The system retrieves from database and displays a list of available staff, considering their availability, workload, and job preferences. 5. The manager selects a staff and allocates the job. 6. The system validates the job allocation to ensure there are no conflicts such as (e.g., schedule overlaps). 7. The manager confirms the job allocation. 8. The system updates the job allocation and notifies the staff of their job allocation. 9. The manager decides to exit the job allocation page or continues allocating more jobs.
Alternative Scenarios:	2a. The system does not display the list of unassigned jobs for the upcoming week. 2a1. If no staff are available for a selected job, the system notifies the manager that no staff match the criteria. 2a2. The manager chooses to adjust the job's time, or searches for alternate staff: 4a. The system retrieves from database and displays a list of available staff, considering their availability, workload, and job preferences: 4a1. If there is a system error while updating the job allocation, the system logs the error and displays a failure message. 4a2. The manager retries the job allocation or contacts' support.
Priority:	High
Non-Functional Requirements	• NFR7: The job allocation interface must load in under 3 seconds to ensure a smooth and responsive user experience for managers. • NFR8: The interface should be user-friendly and intuitive, with visual aids such as color coding or icons to highlight scheduling conflicts, overworked staff, or other important conditions. This ensures the manager can assign jobs efficiently without confusion. • NFR16: Only authorized managers should have access to allocate jobs, with role-based access control. All actions related to job allocations (e.g., assigning jobs and standby, making changes) should be logged for auditing purposes to ensure accountability. • NFR26: The system must accurately prevent job scheduling conflicts, such as overbooking staff or overlapping schedules. Any scheduling errors or conflicts should be clearly communicated to the manager in real time to allow quick resolution.

	• NFR27: The system should be able to handle a growing number of staff and job assignments without degrading performance, supporting hundreds of staff members and job allocations simultaneously. • NFR28: The system must be accessible 24/7, allowing managers to allocate jobs (with standby) at any time and providing real-time job allocation functionality, especially for critical operations like train services. • NFR29: All job allocations and changes made by managers should be logged in detail for future auditing purposes, including information on who assigned the job, what the job was, and how any conflicts were resolved. • NFR30: After a job is allocated, the system must immediately notify the assigned staff with the job details through multiple communication channels (e.g., email, SMS, mobile app notifications) to ensure staff are informed in real time about their assignments.

Use Case ID:	UC-4
Use Case Name:	Accept / Reject Staff's reject request
Description:	The manager can choose to accept or reject a staff member's job rejection request, after the staff member declines a job assignment.
Primary Actor:	Manager
Preconditions:	• The manager must be logged into the system. • The staff must have already rejected a job that was allocated to them. • The manager is viewing the staff that rejected the job allocation in the "View Staff Workload Statistics" dashboard.
Postconditions:	• The manager chooses to accept/ reject the staff's reject request. • The reject request status is updated in the database based on the manager's decision.
Main Success Scenarios:	1. The manager selects the "View Reject Request" button. 2. The system displays the staff's reason of the reject request and provides options to accept or reject. 3. The manager chooses to either accept or reject the staff's reject request. 4. The system saves the decision (accept/reject) in the database. 5. The manager exists the review or moves on to review the next staff rejection request.

Alternative Scenarios:	2a. The system does not display the staff's reason for the rejection request: 2a1. The system displays an error message stating that the reason could not be retrieved and prompts the manager to either retry viewing the reject request or contact support. 4a. The system fails to save the manager's decision: 4a1. The system displays an error message and prompts the manager to try again or contact support.
Priority:	High
Non-Functional Requirements	• NFR7: The system must load the job allocation interface within 3 seconds. • NFR8: The interface must be user-friendly, using intuitive visual aids like color coding to help manage workload distribution and conflicts. • NFR16: Only authorized users should have access to the job allocation system, with role-based access control and logging of all allocation actions for auditing purposes. • NFR26: The system must ensure that job scheduling conflicts (e.g., overbooking staff) are automatically detected and clearly communicated to the manager in real time. • NFR27: The system must scale efficiently to handle large numbers of staff and job assignments without performance degradation. • NFR28: The system must be available 24/7 to allow managers to allocate jobs at any time. • NFR29: The system must log all job allocation actions for auditing, recording details like who assigned the job and what changes were made. • NFR30: After job allocation, the system must immediately notify staff via multiple channels (e.g., email, SMS) with job details to ensure timely communication.

Use Case ID:	UC-5
Use Case Name:	Send notification message to staff
Description:	The manager initiates the use case by performing actions such as job assignments, updates, or other communication, after which the system sends a notification message to the staff, ensuring they are informed of the relevant details.
Primary Actor:	Manager / system (?)
Preconditions:	• The manager must have successfully logged into the system. • The staff must be registered in the system with valid contact details (e.g., email or phone) • A valid job assignment or updates had been made, or decision regarding a reject request is finalized.
Postconditions:	The notification message is sent to the staff's landing page. The system logs the notification action for tracking and auditing purposes.
Main Success Scenarios:	1. The manager performs an action in the system, such as allocating jobs or handling a staff reject request. 2. The system identifies the need to send a notification message based on the manager's action. 3. The system generates the relevant notification message 4. The system sends the message to the appropriate staff member using their preferred communication channel 5. The staff receives the notification 6. The system logs the notification action in the system for audit n tracking
Alternative Scenarios:	4a. Invalid staff contact details: 4a1. This occurs when the system is about to send notification, but the staff's contact details are invalid. Instead of sending, the system alerts the manager to correct the details before proceeding. 5a. Notification failure: 5a1. This occurs when the system has sent the notification, but it fails to be delivered to the staff. The system will retry sending or log the failure.

	2a. No action required: 2a1. If the manager mistakenly performs an action that does not require a notification. The system detects that no notification is necessary and skip step 3-5.
Priority:	high
Non-Functional Requirements:	• NFR9: The system shall retrieve and display the workload data within 2 seconds. • NFR12: The system must prevent job scheduling conflicts and notify managers of any issues in real-time. • NFR13: In case of notification failure, the system should retry sending the message up to 3 times before logging a failure. • NFR15: The system shall ensure that the workload data displayed is always 99.9% accurate and consistent with the database, avoiding outdated or corrupt information. • NFR17: Every single piece of information must be encrypted to prevent unauthorized access. • NFR23: All job allocations and changes should be logged for auditing purposes. • NFR36: The system's notification functionality must be available 24/7, ensuring that critical job updates or schedule changes can be communicated to staff at any time. • NFR37: If the notification fails due to external system errors (e.g., email or SMS service provider issues), the system should provide the manager with a clear error message and allow for manual resending after the issue is resolved. • NFR38: The system shall prevent duplicate user entries (e.g., identical usernames or emails).

Use Case ID:	UC-6
Use Case Name:	View Job Allocation
Description:	Staff can view detailed information about their allocated jobs, including job role, date, time, and location.
Primary Actor:	Staff
Preconditions:	• The staff is logged into the system. • The staff has navigated to the landing page.
Postconditions:	• The notification message is sent to the staff's preferred communication channel (e.g., email, in-app notification, or SMS). • The system logs the notification action in the database for tracking purposes.
Main Success Scenarios:	1. The staff selects the "View Job Allocation" option from the landing page. 2. The system displays the job allocations and the details (date, time, location, job role). 3. Staff views the list of job allocations and the details.
Alternative Scenarios:	2a. No job allocations are available for the staff: 2a1. System displays a message indicating that there are currently no job allocations assigned to the staff member.
Priority:	High
Non-Functional Requirements	• NFR7: The job allocation interface must load within 3 seconds to provide a smooth and responsive user experience. • NFR9: The system shall retrieve and display the workload data within 2 seconds. • NFR15: The system shall ensure that the workload data displayed is always 99.9% accurate and consistent with the database, avoiding outdated or corrupt information. • NFR17: Every single piece of information must be encrypted to prevent unauthorized access. • NFR23: All job allocations and changes should be logged for auditing purposes. • NFR25: The interface must be intuitive and easy to use, with visual aids (e.g., colour coding or icons) to highlight scheduling conflicts or overworked staff.

	• NFR29: The schedule must be displayed in an easy-to-read format, with clear options to filter by date, route, and time. The interface should be mobile-friendly. • NFR36: The system's notification functionality must be available 24/7, ensuring that critical job updates or schedule changes can be communicated to staff at any time.

Use Case ID:	UC-7
Use Case Name:	Accept/Reject Job Allocation
Description:	The staff decides to accept or reject the jobs that has been assigned to them by the manager.
Primary Actor:	Staff
Preconditions:	• Staff must be logged into the system. • The system must have the job allocation available for review. • Staff must be viewing the job allocations.
Postconditions:	• The staff's decision to accept or reject the selected job allocation is saved in the database.
Main Success Scenarios:	1. The staff selects the job allocation that they wish to review. 2. The system displays the details of the selected job, along with options to accept or reject the job allocation. 3. The staff chooses to either accept or reject the selected job allocation. 4. The system saves the staff's decision in the database. 5. The system displays a confirmation message.
Alternative Scenarios:	2a. The system does not display the details of the job and the options to accept or reject the job: 2a1. The system displays an error message and notifies the staff to refresh the page.
Priority:	High
Non-Functional Requirements:	• NFR7: The job allocation interface must load within 3 seconds to provide a smooth and responsive user experience. • NFR9: The system shall retrieve and display the workload data within 2 seconds. • NFR15: The system shall ensure that the workload data displayed is always 99.9% accurate and consistent with the database, avoiding outdated or corrupt information.

	• NFR17: Every single piece of information must be encrypted to prevent unauthorized access. • NFR23: All job allocations and changes (accept/reject decisions) should be logged for auditing purposes. • NFR25: The interface must be intuitive and easy to use, with visual aids (e.g., colour coding or icons) to highlight scheduling conflicts or overworked staff. • NFR36: The system's notification functionality must be available 24/7, ensuring that critical job updates or schedule changes can be communicated to staff at any time. • NFR11: The system shall ensure that changes to user information (such as accepting or rejecting job allocations) are reflected in the database within 3 seconds. • NFR37: If the notification fails due to external system errors (e.g., email or SMS service provider issues), the system should provide the manager with a clear error message and allow for manual resending after the issue is resolved.

Use Case ID:	UC-8
Use Case Name:	Edit Availability and Job Preference
Description:	The staff can edit their availability and job preferences by selecting dates from a calendar view, indicating their available timings, and their job preferences. This can be done up to one month in advance.
Primary Actor:	Staff
Preconditions:	• The staff must have logged into the system. • The staff has navigated to the "Edit Availability and Job Preferences" page. • The staff is viewing the calendar interface displaying the current and upcoming weeks.
Postconditions:	• The staff's availability and job preferences, including dates, times, job roles, and locations, are updated in the database. • The edited availabilities and job preferences are saved in the database.
Main Success Scenarios:	1. The staff selects a specific date from the calendar interface to edit their availability and job preferences. 2. The system displays an option to edit availability and job preferences for that date. 3. The staff chooses to edit availability and job preferences for the selected date. 4. The system displays a list of available timings and job roles with associated locations for that day. 5. The staff selects their available timings and preferred job roles with locations. 6. The system displays a preview of the selected timings and job preferences. 7. The staff reviews, confirms the selections, and submits the changes.

	8. The system saves the changes and updates the staff's availability and job preferences in the database, confirming the update with a message.
Alternative Scenarios:	2a. No available timings for the selected date: 2a1. The system displays a message stating there are no available timings or job roles for the selected date. 2a2. Staff can either choose a different date or exit the page. 4a. Unavailable timing or job role: 4a1. The system displays an error message indicating that the selected timing or job role is no longer available. 4a2. Staff is prompted to select different timings or job roles from the available options. 4a3. Staff confirms and submits the new selections. 5a. Deadline passed for changes: 5a1. System displays an error message indicating that the deadline has passed for making changes to that date. 5a2. Staff can view existing preferences or exit the page. 6a. Invalid input: 6a1. The system displays an error message indicating the issue with the selected timing or preferences (e.g., overlapping or invalid timing). 6a2. Staff is prompted to correct the input and resubmit. 6b. Preview not confirmed by staff: 6b1. Staff exits the preview without submitting, and the system discards any unsaved selections 6b2. System reverts to the previous page, showing the original availability and job preferences.
Priority:	High

Non-functional requirements:	• NFR9: The system shall retrieve and display the workload data (availability and job preferences) within 2 seconds. • NFR11: The system shall ensure that changes to user information (such as availability and job preferences) are reflected in the database within 3 seconds. • NFR15: The system shall ensure that the data displayed is always 99.9% accurate and consistent with the database, avoiding outdated or corrupt information. • NFR17: Every single piece of information must be encrypted to prevent unauthorized access, especially for sensitive data like job preferences. • NFR25: The interface must be intuitive and easy to use, with visual aids such as a calendar view to simplify availability and job preference selection. • NFR30: The workload data (availability and job preferences) must be accurate, preventing inconsistencies such as missing or incorrect data. • NFR29: The calendar interface must be easy to navigate, with clear options to filter and select available timings and job roles. The interface should be mobile-friendly to ensure staff can edit their preferences on the go. • NFR37: If the system fails to save the availability or job preferences due to a database error or system issue, it should display an error message and allow the staff to retry.

Use Case ID:	UC-9
Use Case Name:	View Monthly Workload Statistics
Description:	The staff can use this dashboard to view a summary of their monthly workload, including job allocations and total working hours. This helps the staff to make informed decisions regarding their availability and job preferences for the coming weeks.
Primary Actor:	Staff
Preconditions:	• The staff is logged into the system. • The staff has navigated to the landing page. • The job allocation data must be available for the current month.
Postconditions:	• The staff can view a summary of their job allocations and total working hours for the month. • The total working hours for the month are shown, and any updates in job allocations or availability are reflected in the dashboard.
Main Success Scenarios:	1. The staff selects the “View Monthly Workload Statistics” option. 2. The system displays the job allocations and total working hours for the current month, including job roles, dates, and times, and the remaining hours left before they exceed 40 working hours. 3. The staff reviews the monthly job allocations and total working hours.
Alternative Scenarios:	2a. The system does not display the workload data: 2a1. The system displays an error message suggesting the staff to retry or contact support. 2b. There is no available workload data for that month: 2b1. The system displays a message that no job allocation data is available.
Priority:	Medium
Non-Functional Requirements:	• NFR9: The system shall retrieve and display the workload data within 2 seconds. • NFR15: The system shall ensure that the workload data displayed is always 99.9% accurate and consistent with the database, avoiding outdated or corrupt information. • NFR25: The interface must be intuitive and easy to use, with visual aids (e.g., colour coding or icons) to highlight scheduling conflicts or overworked staff. • NFR29: The schedule (monthly workload statistics) must be displayed in an easy-to-read format, with clear options to filter by date, route, and time. The interface should be mobile-friendly.

	• NFR36: The system's notification functionality must be available 24/7, ensuring that critical updates or changes to the monthly workload can be communicated to staff at any time. • NFR37: If the system fails to retrieve the workload data, it should display an error message and provide options for retrying or contacting support. • NFR30: The workload data must be accurate, preventing inconsistencies such as missing or incorrect data. • NFR26: The system shall highlight the total working hours in red when the total working hours is more than or equal to 40.

Use Case ID:	UC-10
Use Case Name:	Add Users
Description:	The IT administrators use this function to add new users to the system, such as managers or staff members.
Primary Actor:	IT administrator
Preconditions:	• The IT administrator must be logged into the system with administrative privileges. • The system must allow access to the user management module.
Postconditions:	The new user is successfully added to the system and ready for access.
Main Success Scenarios:	1. The IT administrator selects the "Add Users" from the administration menu. 2. The system displays a form for entering user details. 3. The IT administrator enters new user details, including role assignment. 4. The system validates the input and checks for unique usernames/ employee ID. 5. The IT administrator confirms the user details and submits the form. 6. The system creates the new user and assigns the appropriate access rights. 7. The system displays a confirmation message and updates the user list.
Priority:	Medium
Alternative Scenarios:	2a. The system displays a form for entering user details: 2a1. If the system encounters an error while loading the user entry form, it displays a message to the administrator, suggesting a retry. 4a. The system validates the input and checks for unique usernames/ employee ID: 4a1. If the system detects that the username or email is already in use, an error message is displayed. The IT Administrator can input a different username or email. 6a. The system creates the new user and assigns the appropriate access rights: 6a1. If the system fails to create the new user due to a technical issue such as database error, it displays an error message and logs the issue for later review. The administrator is prompted to retry.
Non-Functional Requirements:	• NFR10: Adding a new user should take no more than 5 seconds after the submission of the form.

	• NFR11: The system shall ensure that changes to user information are reflected in the database within 3 seconds. • NFR17: Every single piece of information must be encrypted to prevent unauthorized access. • NFR22: The "Add Users" functionality must be restricted to authorized IT administrators, with role-based access control. Sensitive information such as passwords must be encrypted and stored securely. • NFR23: All job allocations and changes should be logged for auditing purposes. • NFR37: If the system fails to create the new user due to external system issues (e.g., database error), the system should provide an error message and allow the administrator to retry.

Use Case ID:	UC-11
Use Case Name:	Edit User Information
Description:	The IT administrator can update the information of users (both staff and managers) in the system. This includes updating user details, or removing users from the system.
Primary Actor:	IT administrator
Preconditions:	• The IT administrator must be logged into the system with administrative privileges. • The system displays the user management interface.
Postconditions:	The user's information is updated or removed from the system.
Main Success Scenarios:	1. The IT administrator selects the "Edit User Information" option. 2. The system displays a list of current users. 3. The IT administrator selects a user to edit. 4. The system displays the editable details of the selected user. 5. The IT administrator modifies the necessary field (e.g., name, user role (staff or manager role, is it?) or chooses to remove the user. 6. The IT administrator submits the changes or confirms the removal of the user. 7. The system updates the user's information in the database or successfully removes the user from the database. 8. The IT administrator receives a confirmation message that the changes have been saved or the user has been removed.
Alternative Scenarios:	5a. The IT Administrator attempts to edit a user's information, but the system detects invalid input (e.g., incomplete data): 5a1. The system displays an error message, highlighting the invalid fields, and prompts the administrator to correct the input. 6a. The IT Administrator attempts to remove a user who still has assigned tasks: 6a1. The system displays a warning message indicating the user cannot be removed while they have active job assignments and suggests unassigning the tasks first. 7a. The system fails to update the database due to a connection issue: 7a1. The system displays an error message and suggests retrying later.
Priority:	Medium
Non-Functional Requirements:	• NFR11: The system shall ensure that changes to user information are reflected in the database within 3 seconds.

	• NFR17: Every single piece of information must be encrypted to prevent unauthorized access. • NFR22: The "Add Users" and "Edit User Information" functionality must be restricted to authorized IT administrators, with role-based access control. Sensitive information such as passwords must be encrypted and stored securely. • NFR23: All changes to user information (e.g., edits or removals) should be logged for auditing purposes. • NFR37: If the system fails to update the user information due to a connection or database error, the system should provide an error message and allow the administrator to retry. • NFR25: The interface must be intuitive and easy to use, with clear prompts and feedback to guide the IT administrator through the edit or removal process. • NFR38: The system shall prevent duplicate user entries (e.g., identical usernames or emails) during user information edits.

6 Appendix B – Data Dictionary

Job Availability

The specific dates and times when staff are available to work. This information helps managers in assigning tasks based on staff availability.

Staff Job Preference

The preferred location and job role a staff member would like to work in. This includes specific locations and roles that a staff member selects as their preference for job assignments.

Workload

The cumulative schedule of jobs allocated to staff, including total working hours, job assignments, and related job details (e.g., job roles, locations, dates, and times).

Job Allocation

The process whereby the manager assigns the job to staff.

Standby Staff

The staff available for assignment by the manager in case of unforeseen circumstances (e.g. staff cancels the job allocation last minute or medical leave etc.)

Train Schedule

A schedule that displays the planned timetable for a train, including details like train number, departure and arrival times, stations, route.

Reject Request

A formal submission made by a staff member to decline a job assignment, including the reason for rejection, submission date, and the request's status (Pending, Accepted, Rejected)

Job Rejection

The process by which a staff member declines or rejects a job allocation that has been assigned to them. This could occur due to a conflict in availability or other reasons provided by the staff.

Roles

The specific role or position a user holds within the system. This includes roles such as IT Administrator, Staff, or Manager, each with different permissions and access levels within the system.