INF2001: Introduction to Software Engineering

Report Milestone 1

Done by:

Student Name	Student ID
Genisa Gabrielle Lee	2402516
Muhammad Zulfaqar Bin Abdul Hafez	2401578
Teressa Lavanya Aryanto	2401617
Max Tan Chun Yuan	2400562
Ng Aik Seong Sebastian	2400871
Weng Jinrui	2401607

Declaration:

We hereby pledge that this Introduction to Software Engineering task is not plagiarised and
has been written wholly as a result of our own research and compilation of information.

Signed: Genisa, Zulfagar, Lavanya, Max, Jinrui, Sebastian

Dated: 23/09/2025

Team Members and their contributions

Genisa Gabrielle Lee	Gantt chart, Work Breakdown Structure, Gathered the Functional, Non-functional requirements, and Business rules constraints
Muhammad Zulfaqar Bin Abdul Hafez	Requirements elicitation, UML Workload handling, Management segmentation, Sequence diagram adjustment.
Teressa Lavanya Aryanto	Project Planning, Presentation slides creation, UML Diagram adjustments, Use case diagram creation
Max Tan Chun Yuan	Gantt Chart (Support), Report drafting, UML, Presentation slide finessing, Sequence Diagram. Use case elicitation
Weng Jinrui	UML Diagram Drafting, Sequence Diagram drafting, Use Cases drafting, query translation to Functional Requirements
Ng Aik Seong Sebastian	Presentation slide finessing, Project Report finessing, presentation, section of use case and sequence diagram, UML Diagram

1. Introduction

1.1 Purpose

The Music School Workload Management System provides a web-based, interactive platform for **teachers** to view assignments and declare availability, and **managers** to visualise manpower, resolve conflicts, and allocate lessons fairly and efficiently while respecting the school's operating constraints (studios, instruments, opening hours, public holidays, and mandated breaks). The goal is to improve planning transparency, maximize studio utilisation, and support work-life balance through clearer workload visibility and timely scheduling.

1.2 Scope

This report defines the boundaries of a workload management system for a music school, focusing on the capabilities required to plan, coordinate, and oversee instructional work. The scope covers the gathering of planning inputs, the presentation of relevant information to stakeholders, and the governance needed to support routine scheduling decisions within a single institutional context. Activities unrelated to planning and oversight of teaching workload fall outside this scope.

1.3 Product Overview

The product is a centralised platform that unifies the people, resources, and timetable considerations involved in delivering lessons. It provides a coherent, streamlined experience for organising work across planning cycles, making current commitments visible, and supporting fair and consistent decision-making. The design emphasises clarity, reliability, and continuity in day-to-day operations.

2. Requirement Engineering

2.1 Requirement Elicitation Process

2.1.1 Objectives

To discover, clarify, and validate the functional, non-functional, and business-rule requirements for a workload management system tailored to the music school's specific operating context and planning cadence. We began from the Project Description and Initial Requirements as the ground truth, then filled the gaps in with stakeholder engagements.

2.1.2 Stakeholder & Sources

Managers (Administration Teacher)

- Plan allocations
- Monitor capacity
- Ensure fairness

Teachers/Teacher

- Declare availability
- Declare preferences
- Receive assignments

IT Administrators

- Manage Accounts
- Manage Roles
- Operational Security

Primary source materials were derived from the **Music School Brief** and **Initial Requirements**. We will subsequently engage and maintain engaging stakeholders to validate and refine these ideas.

2.1.3 Elicitation Techniques

- Document analysis of the project brief was to surface and extract explicit constraints (studios/instruments, opening hours, public holidays, lesson length, consecutive hours/break rules, manager-on-duty) and the monthly planning cadence (availability by 19th, allocation from 20th). This anchored non-negotiables before we spoke to users.
- Semi-structured Interviews with focused stakeholders of professions, consisting of Managers, Teachers, and IT Admins to probe pain points, decision data, notifications, and security/operational needs. This format balanced consistency with room for discovery.

- Scenario Walkthrough (e.g., double-booking, late availability, public holidays) to elicit edge cases and exception handling concrete terms.

2.1.4 Quality and Governance

- Traceability: Each requirement traces back to a source (brief clause, initial requirement, or interview/scenario), enabling auditability.
- Conflict Resolution: When stakeholder views are diverged (e.g. integration vs standalone), we favoured the brief's constraints and security posture, documenting rationale and marking items as future work where needed.

2.2 Software Requirement Specifications

From the Document analysis, Semi-structured interviews and Scenarios walkthrough, we have compiled and concluded the necessary Requirement specifications feasible for the development of the Software system.

2.2.1 Functional Requirements

ID	Function requirement statement
FR-1	The system shall allow IT Admins to create accounts for new teachers and managers
FR-2	The system shall restrict access by role
FR-3	The system shall display each teachers member's weekly job assignments and monthly workload
FR-4	The system shall allow teachers to manage their availability up to 5 weeks in advance
FR-5	The system shall allow teachers to indicate job preferences for the week
FR-6	The system shall allow teachers to reject assigned jobs, prompting a warning and notifying the manager
FR-7	The system shall display manpower up to one month ahead, including teachers availability, workload, job preference, and studio allocation
FR-8	The system shall allow managers to allocate jobs on week at a time
FR-9	The system shall provide a comparison view of up to three teachers
FR-10	The system shall highlight workload imbalance by: - Showing the top three teachers with the lowest workload - Flagging teachers with over 40 hours in the current week

2.2.2 Business Rules Constraints

ID	Business Rules Constraints
BR-1	Only one teachers may be scheduled per studio at any time
BR-2	The school has 5 studios, only one is equipped for drums
BR-3	Drum lessons shall only be scheduled in the drum studio. Piano, violin, and trumpet may be scheduled in any studio
BR-4	Scheduling must occur only within opening hours: - Mon-Fri: 09:00–21:00 - Sat-Sun: 08:00–21:00 - Exclude public holidays
BR-5	Each class shall have a fixed duration of 30 minutes
BR-6	Teachers shall not be scheduled for more than 4 consecutive hours of teaching
BR-7	After 4 consecutive teaching hours, a teacher shall receive a minimum 1-hour break
BR-8	At least one manager must be scheduled on duty at all times during opening hours

2.2.3 Non-Functional Requirements

ID	Non-Functional Requirements
NFR-1	The system shall be web-based and compatible with modern browsers (Chrome, Edge, Safari, Firefox).
NFR-2	The system shall be responsive for desktop and tablet, with clear visualizations at a glance.
NFR-3	The system shall meet accessibility standards (WCAG 2.1 AA), including sufficient colour contrast and full keyboard navigation
NFR-4	The system shall use secure transport (TLS) and enforce role-based access control to ensure data confidentiality and integrity.
NFR-5	The system shall load dashboard and scheduling pages within 3 seconds under normal network conditions.
NFR-6	The system shall support at least 50 concurrent users (teachers, managers, and administrators) without significant performance degradation.
NFR-7	The system shall achieve 99% uptime during school operating hours (09:00–21:00) to minimise disruption to scheduling activities.
NFR-8	The system shall recover from a restart or outage within 5 minutes without loss of stored data.

NFR-9	The system shall be modular and maintainable, allowing new components (e.g., student tracking or reporting) to be integrated with minimal architectural changes.
NFR-10	The system's source code and documentation shall follow consistent standards, allowing future developers to make minor feature updates within two person-days.
NFR-11	All changes to data (e.g., job allocations, account updates, and schedule edits) shall be recorded in an audit log with timestamp, user ID, and action details.
NFR-12	The system shall maintain transactional integrity; no partial updates shall occur in the event of an interruption or network failure.
NFR-13	The user interface shall follow consistent design patterns (navigation, colour, and iconography) to support intuitive use.
NFR-14	The system shall provide contextual help (tooltips or inline guidance) for key pages such as "Allocate Jobs" and "Manpower Overview."
NFR-15	The system shall send real-time email and in-app notifications for major events (e.g., job rejections, allocations, or late availability submissions).
NFR-16	The system shall perform automatic backups of all operational data at least once daily, with backups retained for a minimum of 30 days.
NFR-17	In the event of database corruption, the system shall be restorable to the most recent backup within one hour.

3. Use Cases

3.1 Actor overview

Teacher:
Responsible for maintaining up-to-date availability and workload preferences, ensuring accurate scheduling and communication with management.

Manager:
Manages operational scheduling by reviewing teacher availability, assigning jobs, and maintaining workload fairness across staff.

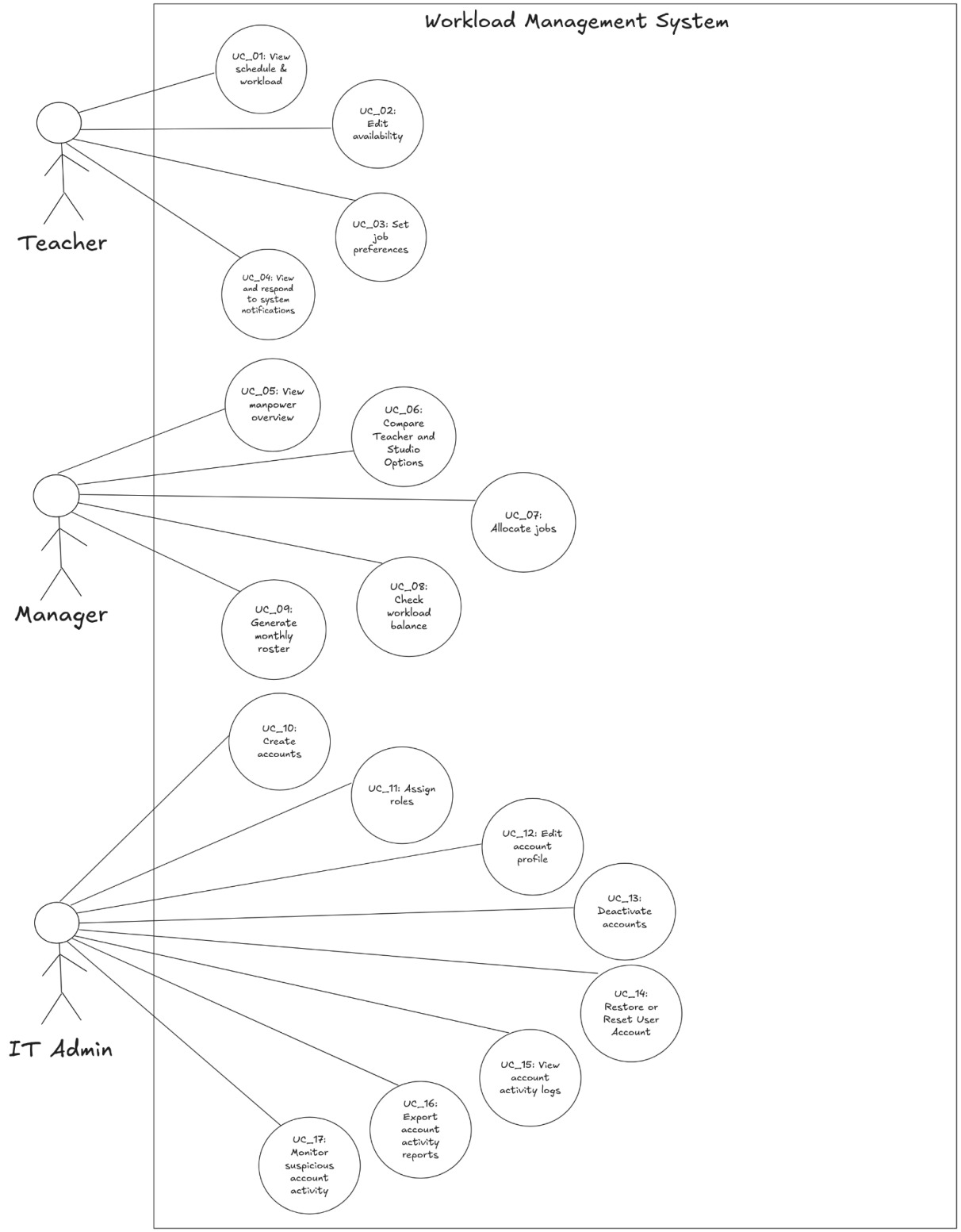
IT Administrator:
Ensures secure and compliant system operation through user account management, role configuration, and activity auditing.

3.2 Formal use cases

ACTORS	USE CASES INITIATED
Teacher	- **UC_01:** View schedule & workload (weekly/monthly) - **UC_02:** Edit availability (up to one month in advance) - **UC_03:** Set job preferences (instrument/level) - **UC_04:** View and respond to system notifications
Manager	- **UC_05:** View manpower overview (availability, workload, preferences, studio usage) - **UC_06:** Compare Teacher and Studio Options - **UC_07:** Allocate jobs (one week at a time) - **UC_08:** Check workload balance (highlight the lowest hours, flag >40 hours) - **UC_09**: Generate monthly roster
IT administrator	- **UC_10:** Create accounts - **UC_11:** Assign roles - **UC_12:** Edit account profile - **UC_13:** Deactivate accounts - **UC_14**: Restore or Reset User Account (locked/forgotten credentials) - **UC_15:** View account activity logs - **UC_16:** Export account activity reports - **UC_17**: Monitor suspicious account activity

For Detailed use cases refer to Section 7: Appendix, Sub-section 1

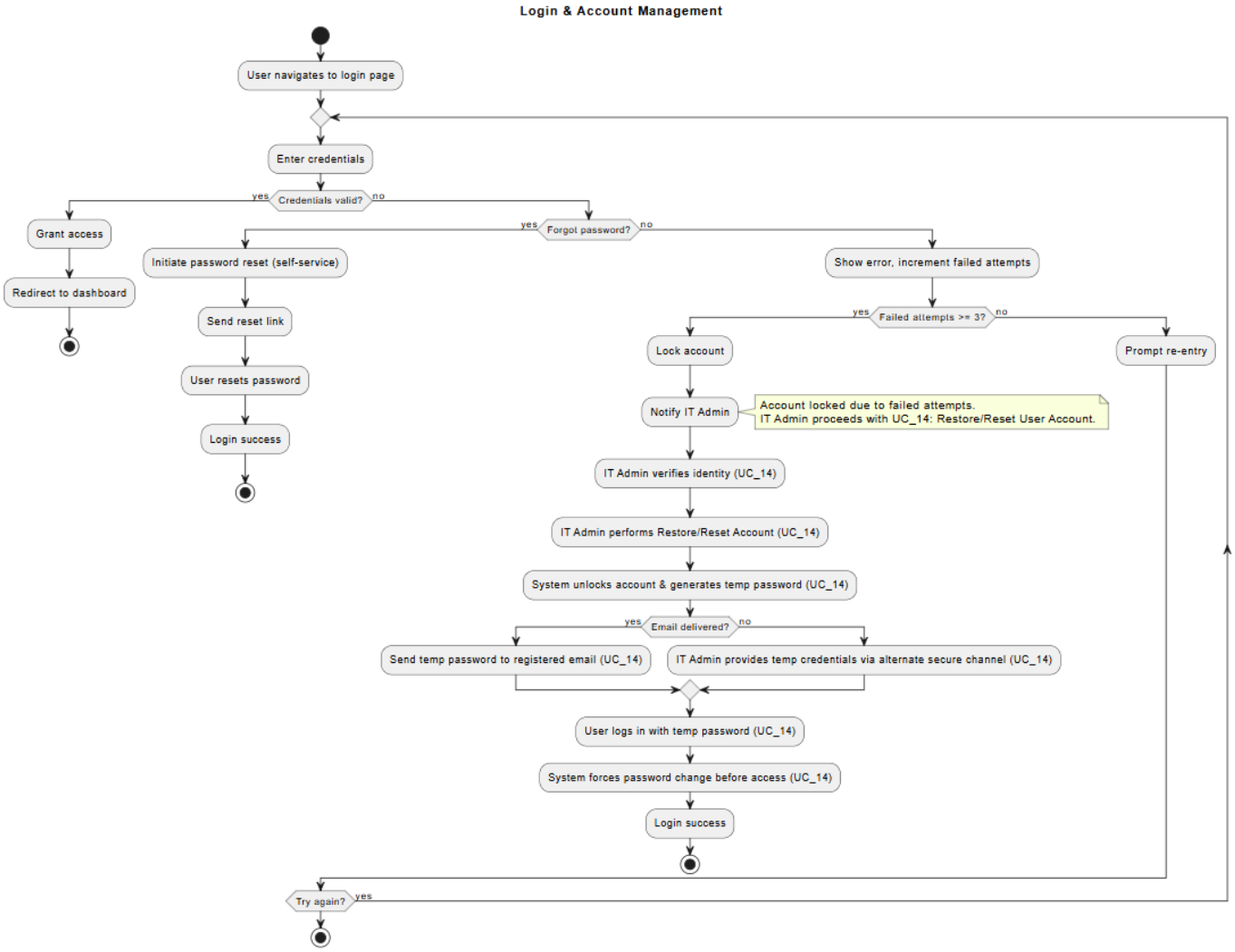
3.2 Use case diagrams



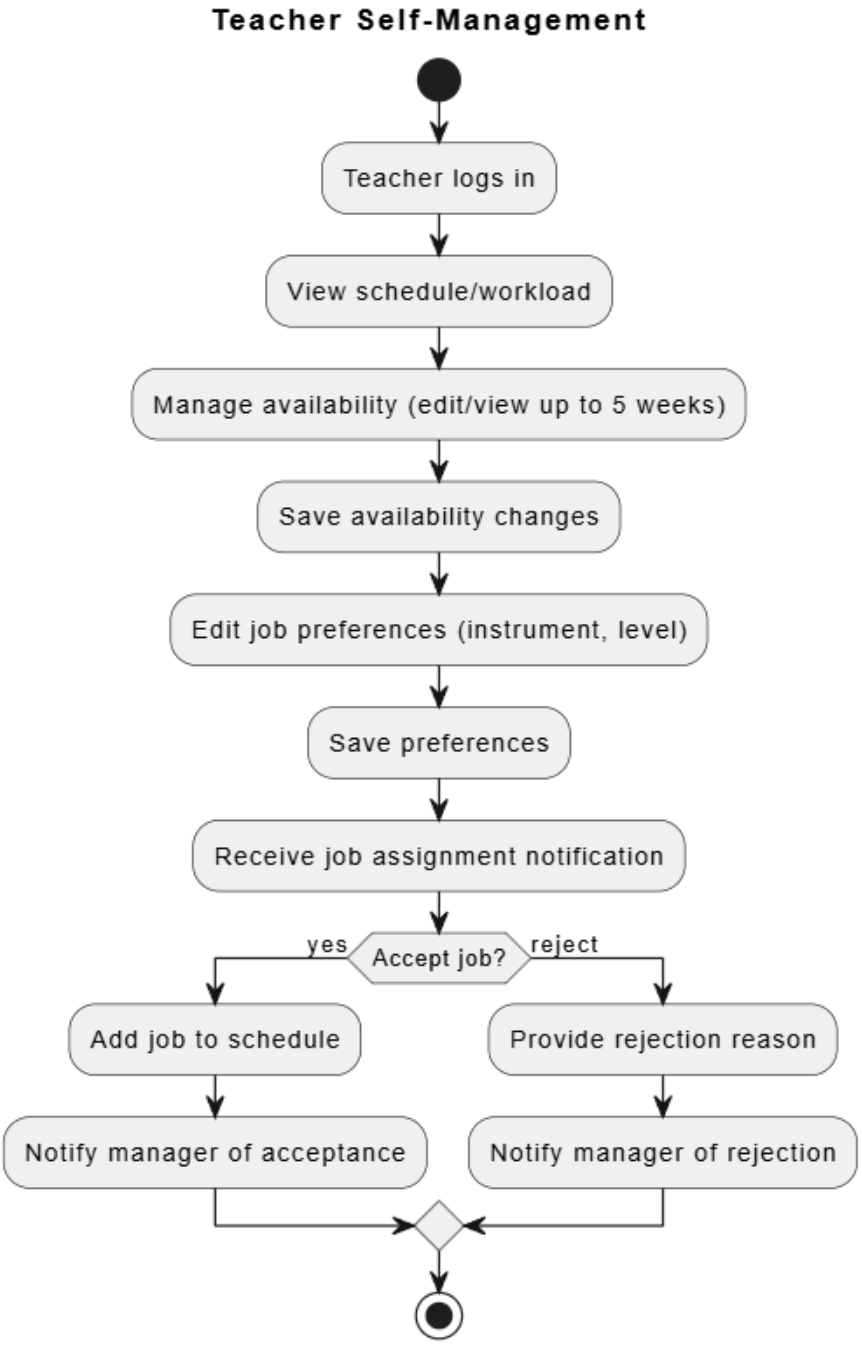
(Fig.1, use case diagram)

3.3 Activity diagrams

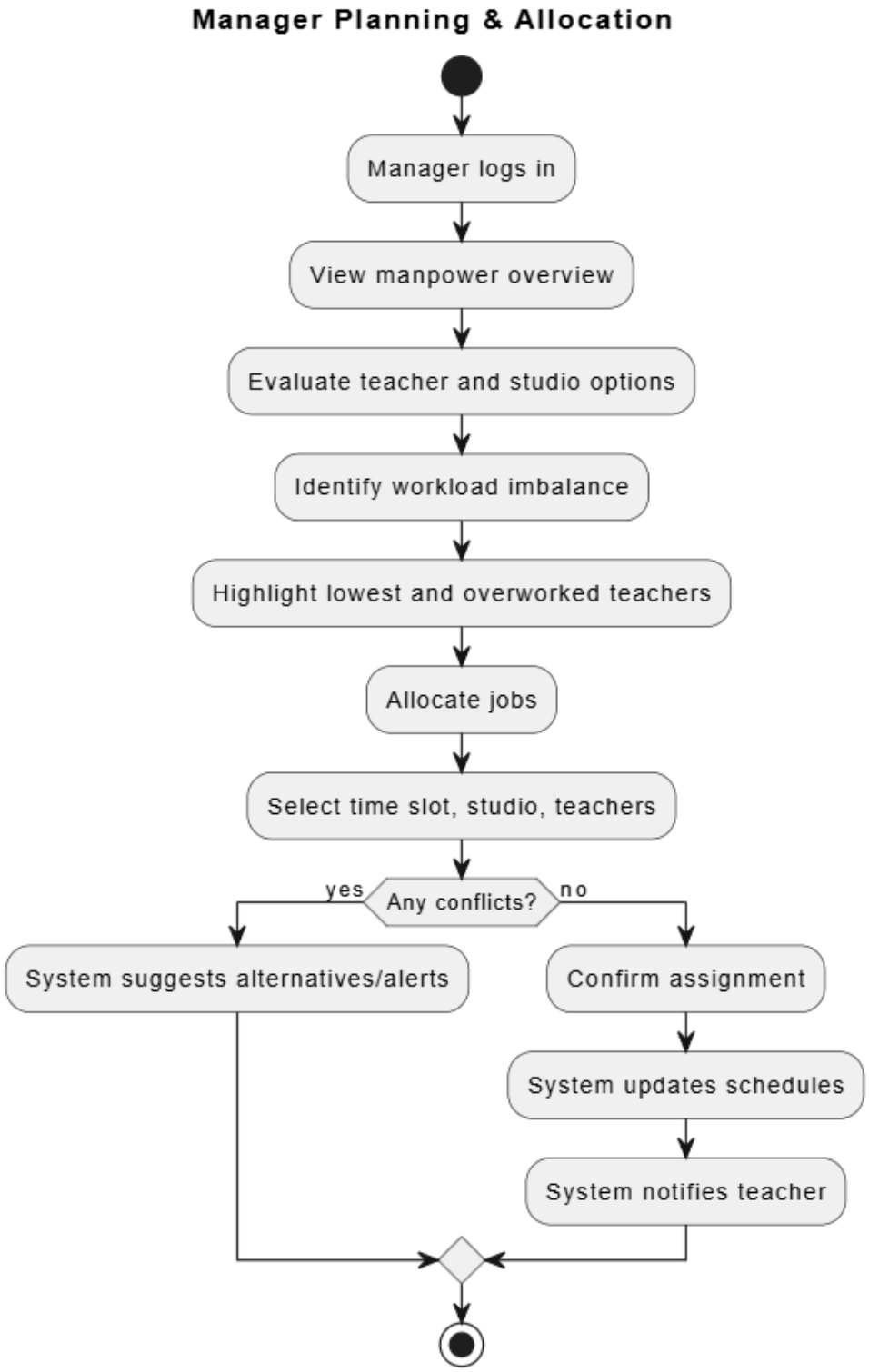
3.3.1 Login & Account Management UC_01, UC_14



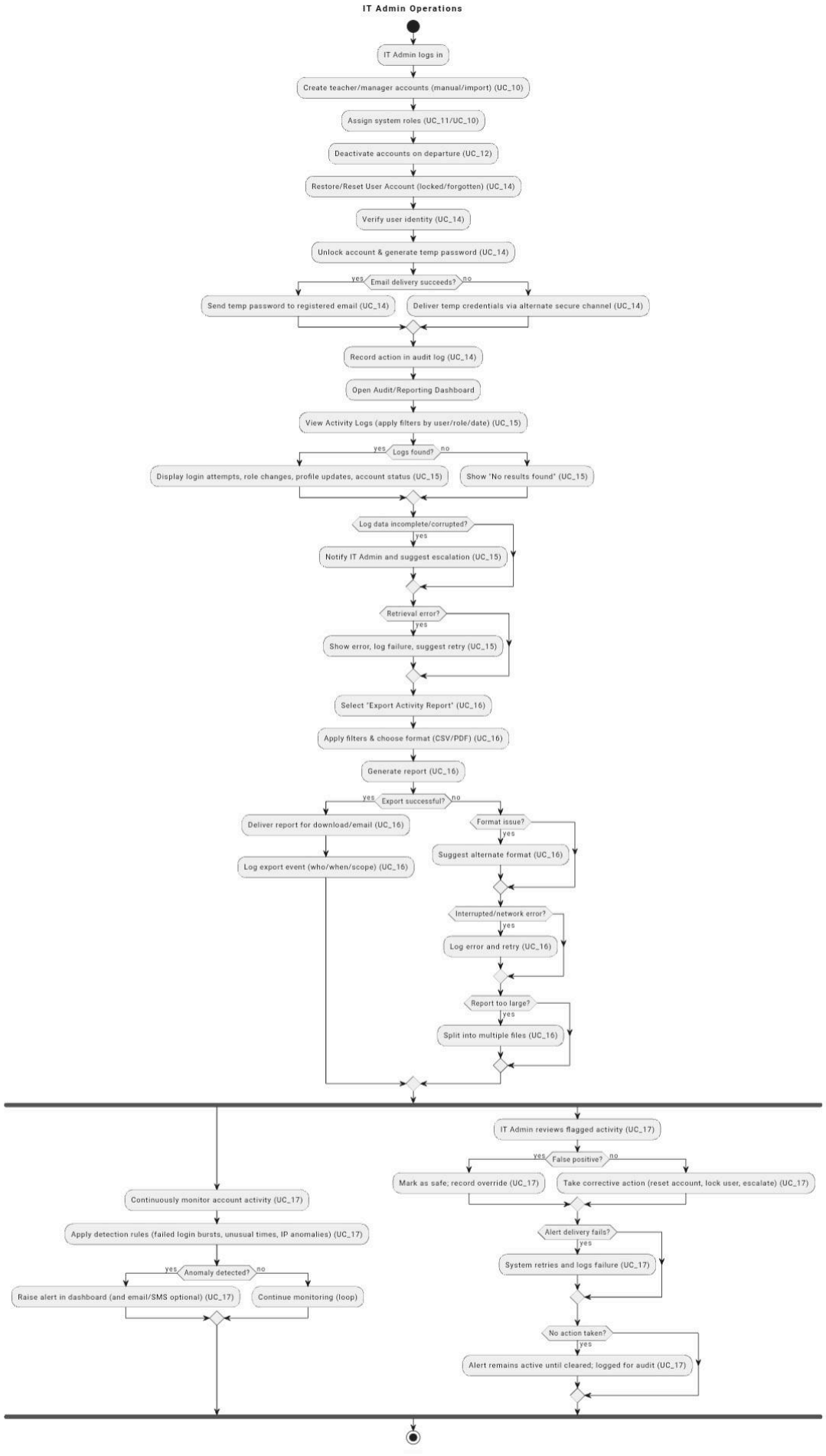
3.3.2 Teacher Self-Management UC_02, UC_03, UC_04, UC_05



3.3.3 Manager Planning & Allocation UC_06, UC_07, UC_08, UC_09



3.3.4 IT Admin Operations UC_10, UC_11, UC_12, UC_13, UC_14, UC_15, UC_16, UC_17



4. Object-oriented Analysis

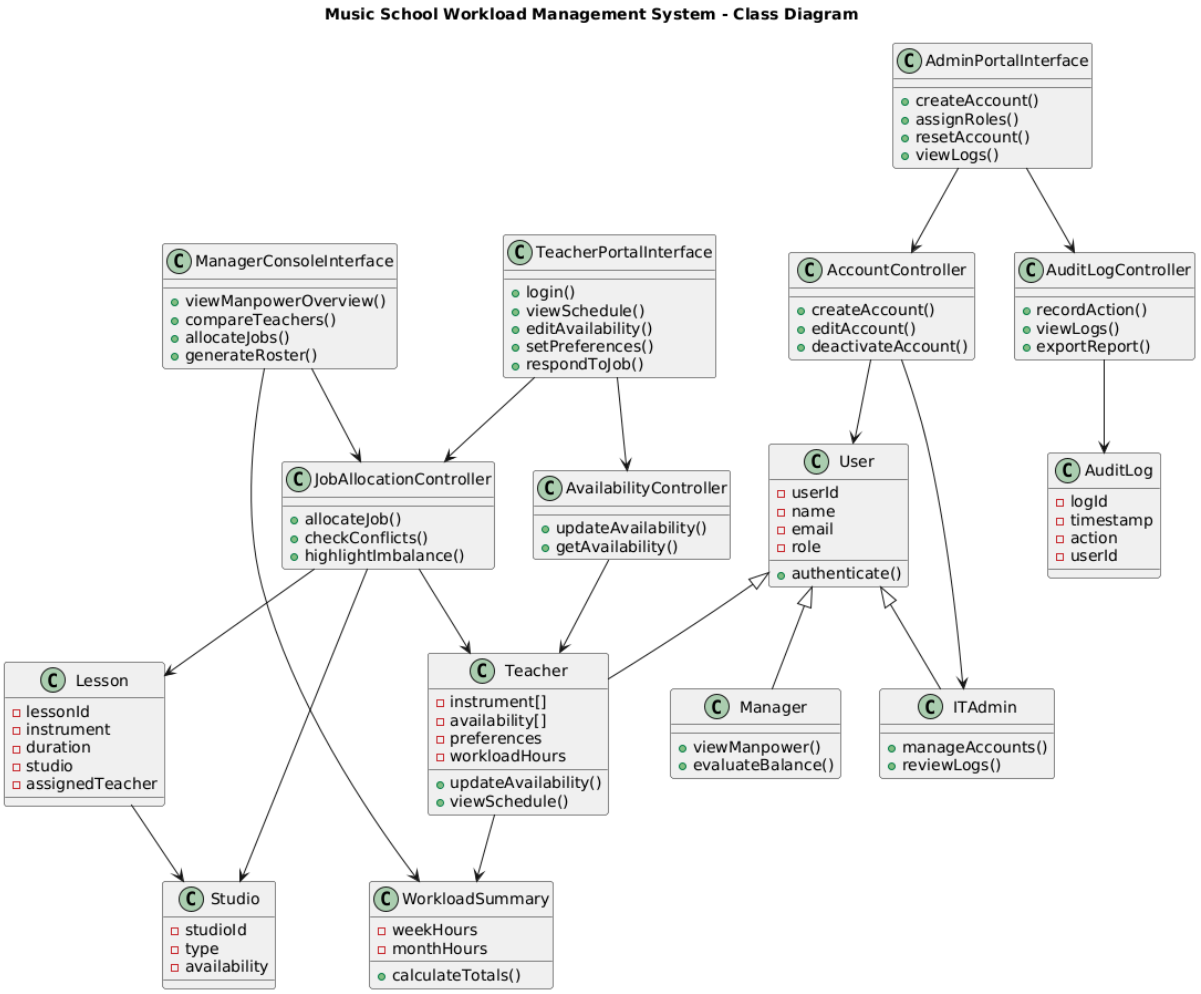
4.1 Identified key classes with diagrams and relationships among the identified classes

Class/ Component	**Responsibility**	**Key Attributes**	**Key Operations**	**Related UCs**
User (abstract)	Common iD & Auth for all roles	userId, name, email, role, status	authenticate(), logout()	UC_01
Teacher (User)	Manage availability & prefs; act on jobs	instruments[], levels[], weekHours, monthHours	viewSchedule(), setAvailability(), setPreferences(), respondToJob()	UC_02 UC_03 UC_04 UC_05
Manager (User)	Plan & allocate lessons; monitor balance	-	viewManpower(), evaluateOptions(), allocateJob(), reviewImbalance()	UC_06 UC_07 UC_08 UC_09
ITAdmin(User)	Account lifecycle & audit reporting	-	createAccount(), modifyAccount(), deactivateAccount(), runAuditReport()	UC_10 UC_11 UC_12 UC_13
Student	Learner for each 30-min lesson	studentId, instrument, level, progressNotes	updateProgress()	UC_07 UC_08 (Context)
Studio	Room scheduling; only one drum kit	studioId, hasDrumKit	isAvailable(timeRange)	UC_06 UC_07 UC_08
Lesson	A scheduled 30-min session	lessonId, startTime, endTime, instrument, status, rejectionReason	assignTo(teacher), updateStatus(), notifyTeacher()	UC_05 UC_07 UC_08
Availability	Future availability slots (≤5 weeks)	date, startTime, endTime, state	upsert(), validateWindow()	UC_03
Preference	Teacher instrument/level preferences	instruments[], levels[], notes	save()	UC_04
Roster	Weekly plan of lessons	weekStart, weekEnd, lessons[]	addLesson(), removeLesson(), getSchedule()	UC_06 UC_07 UC_08
WorkloadSummary	Compute week/month hours, flags	weekHours, monthHours, isOver40h	calcFor(teacher, period)	UC_06 UC_09

Notification	In-app/email notifications	toUserId, type, message, sentAt	notify(user, msg)	UC_05 UC_08
AuditLog	Record who/what/when; export	actorUserId, action, entityType, entityId, timestamp	record(), exportReport()	UC_10 UC_11 UC_12 UC_13
SchedulingPolicy	Enforce rules & constraints	openingHours, classDurationMins (30), maxConsecutiveHours(4), minBreak(60)	isPublicHoliday(date), validate(request)	UC_07 UC_08 (Global)
ManagerShift	Ensure one manager on duty	managerId, startTime, endTime	validateCoverage()	UC_06 (Context)
LessonRequest	Allocation input bundle	studentId, instrument, desiredStart, studioId, teacherId	-	UC_07 UC_08

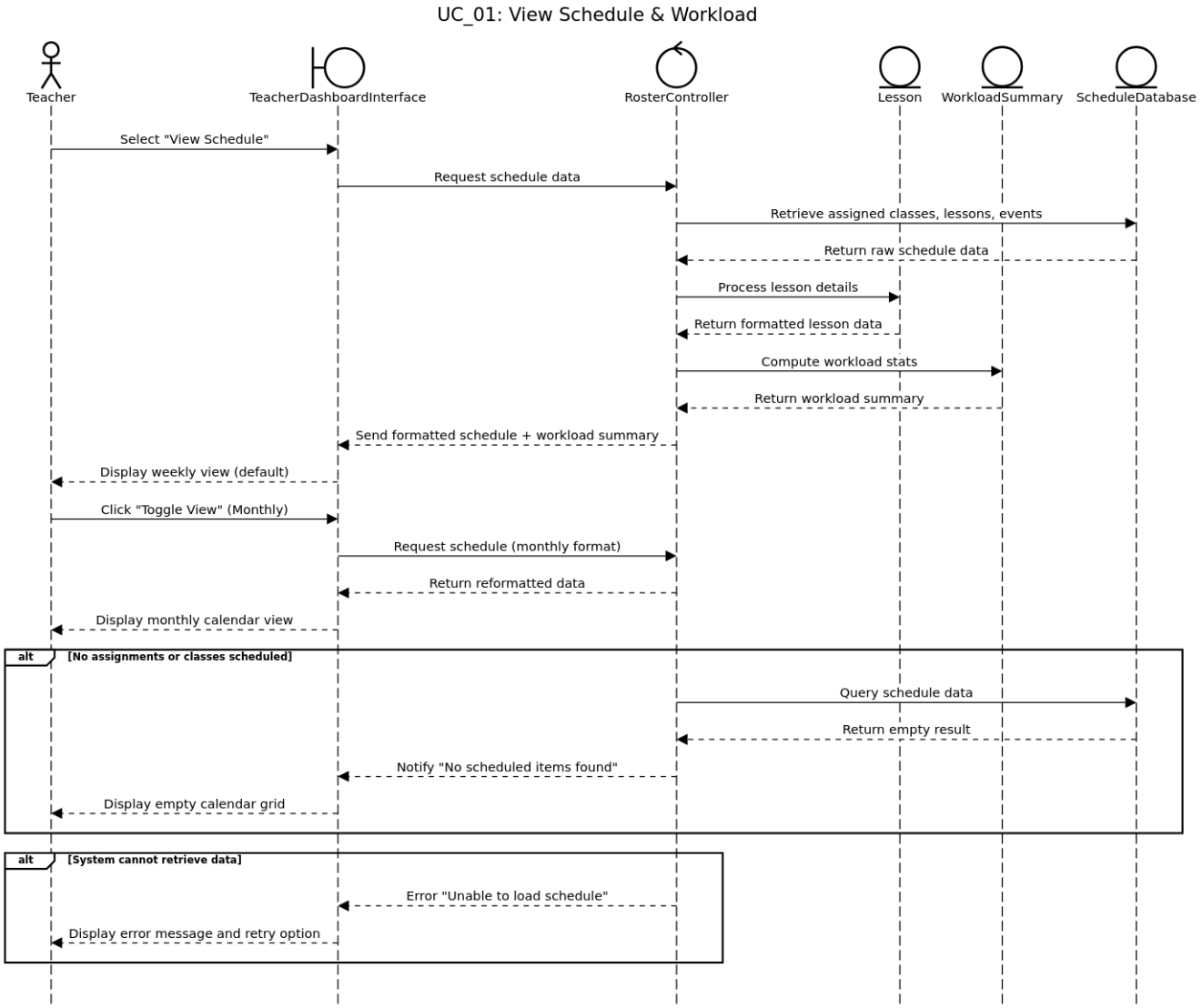
***Context* = supporting domain entity referenced by the UC**
***Global* = cross-cutting policy/service invoked by multiple UCs**

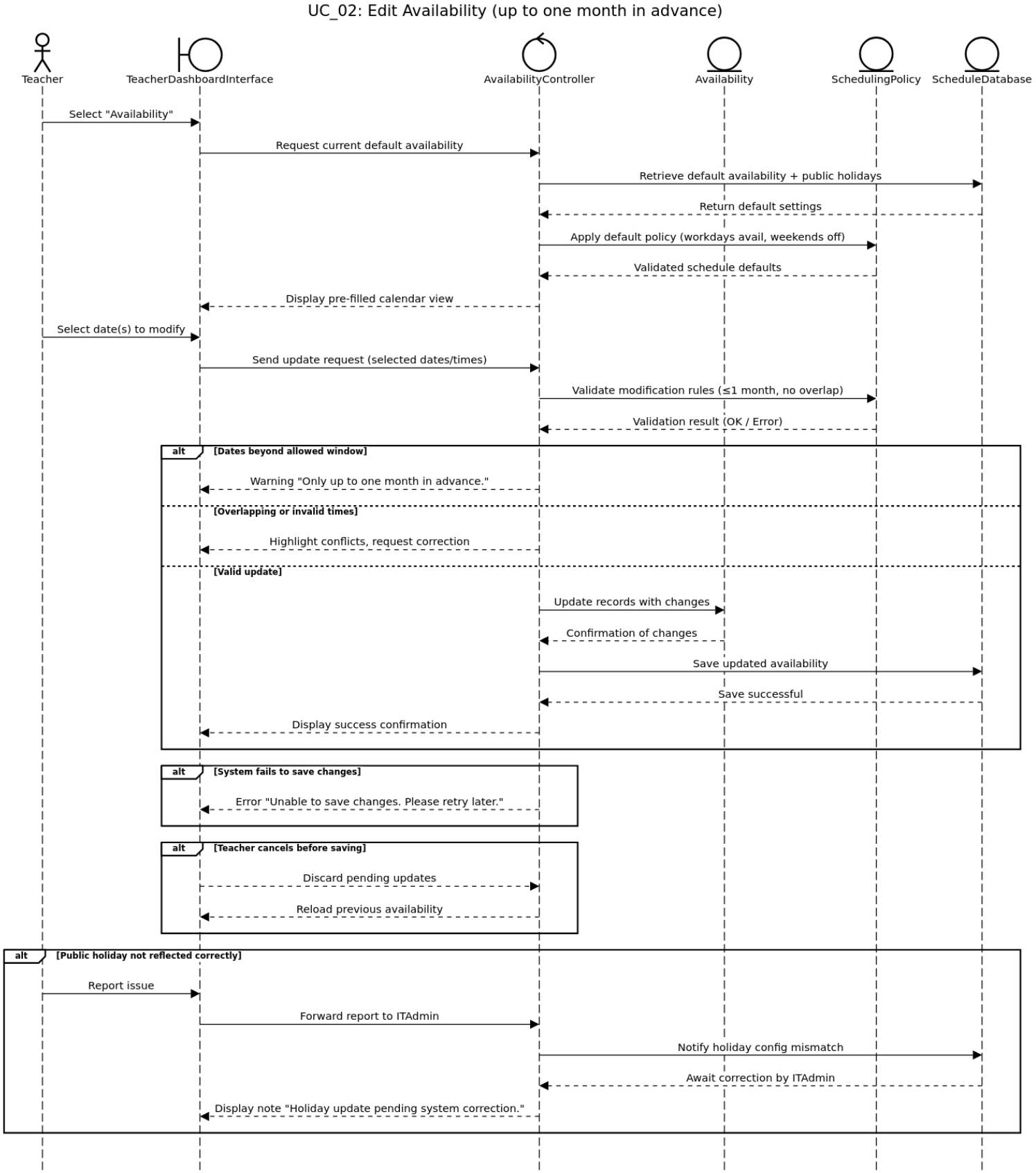
4.2 UML Diagram for Use Cases

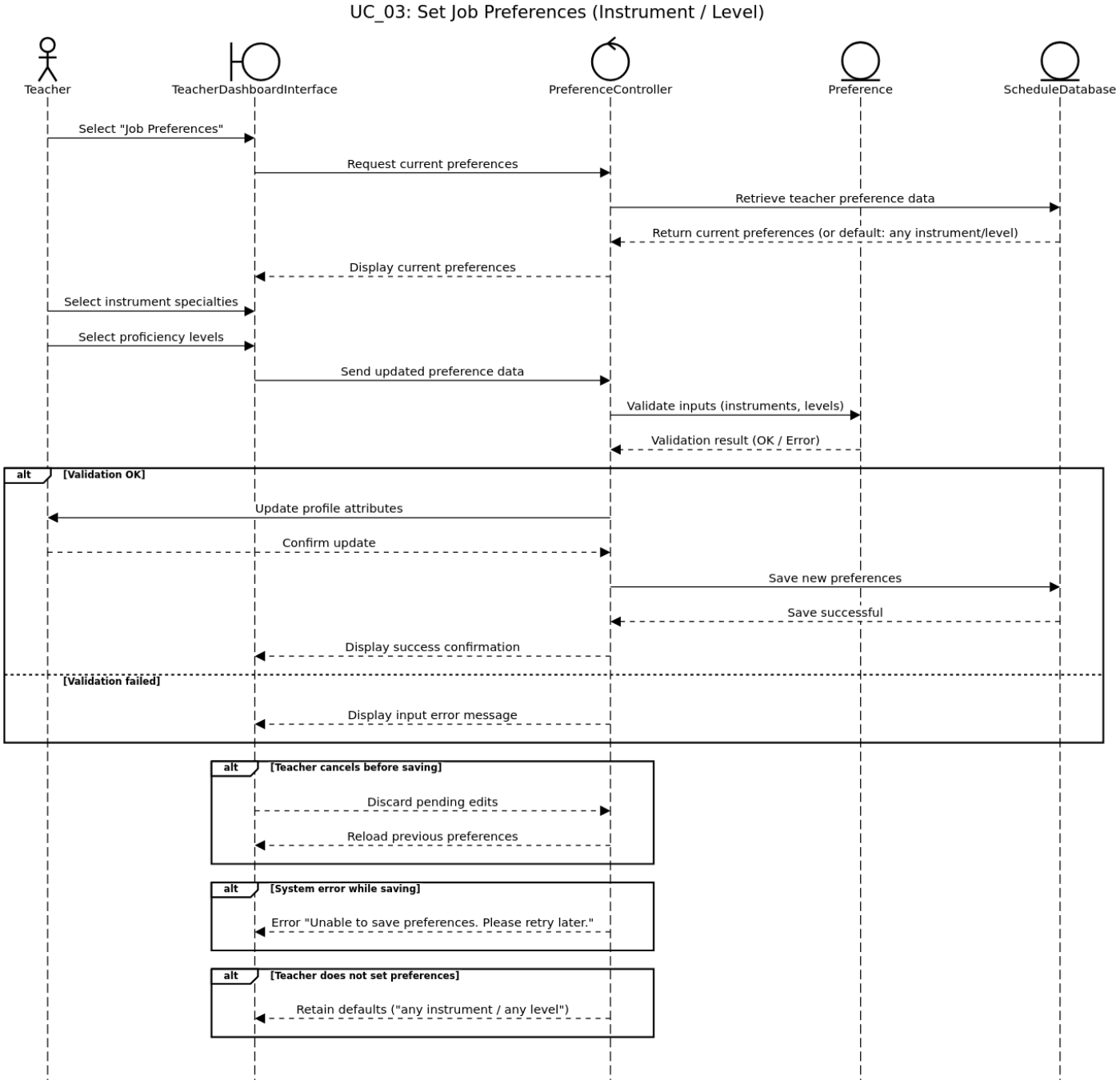


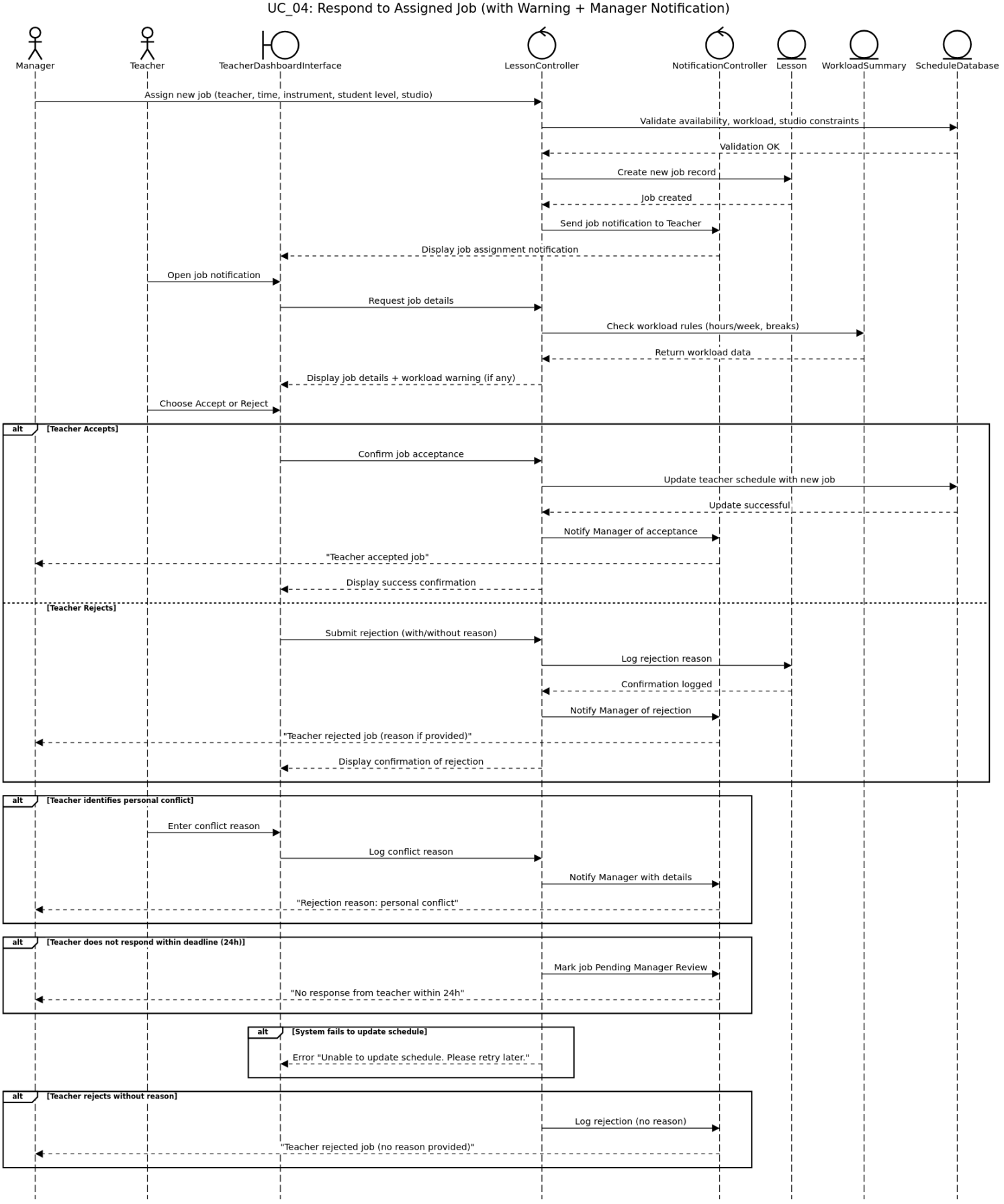
(Fig.2 UML Diagram)

4.2 Sequence diagrams for the use cases with the identified classes

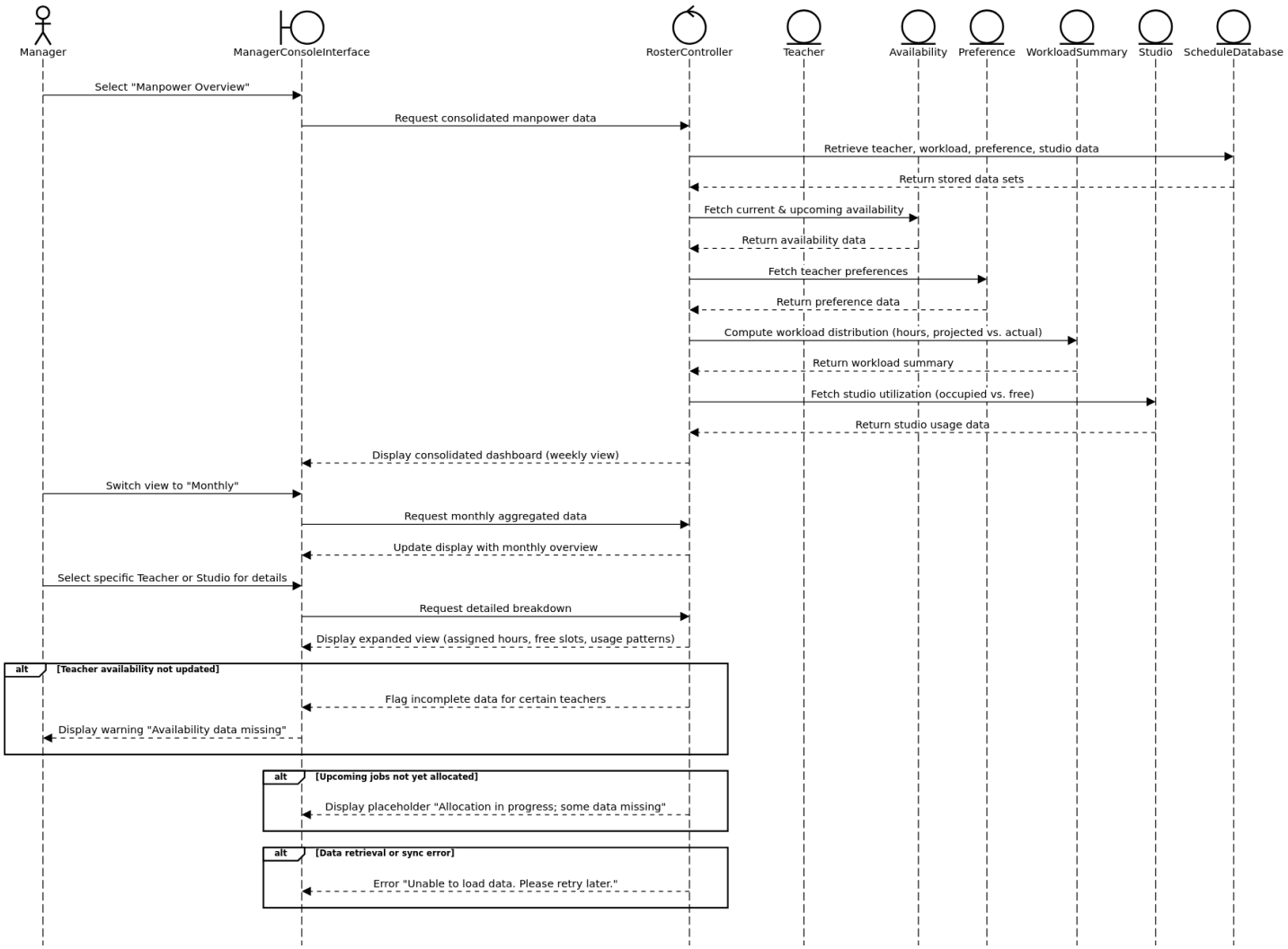


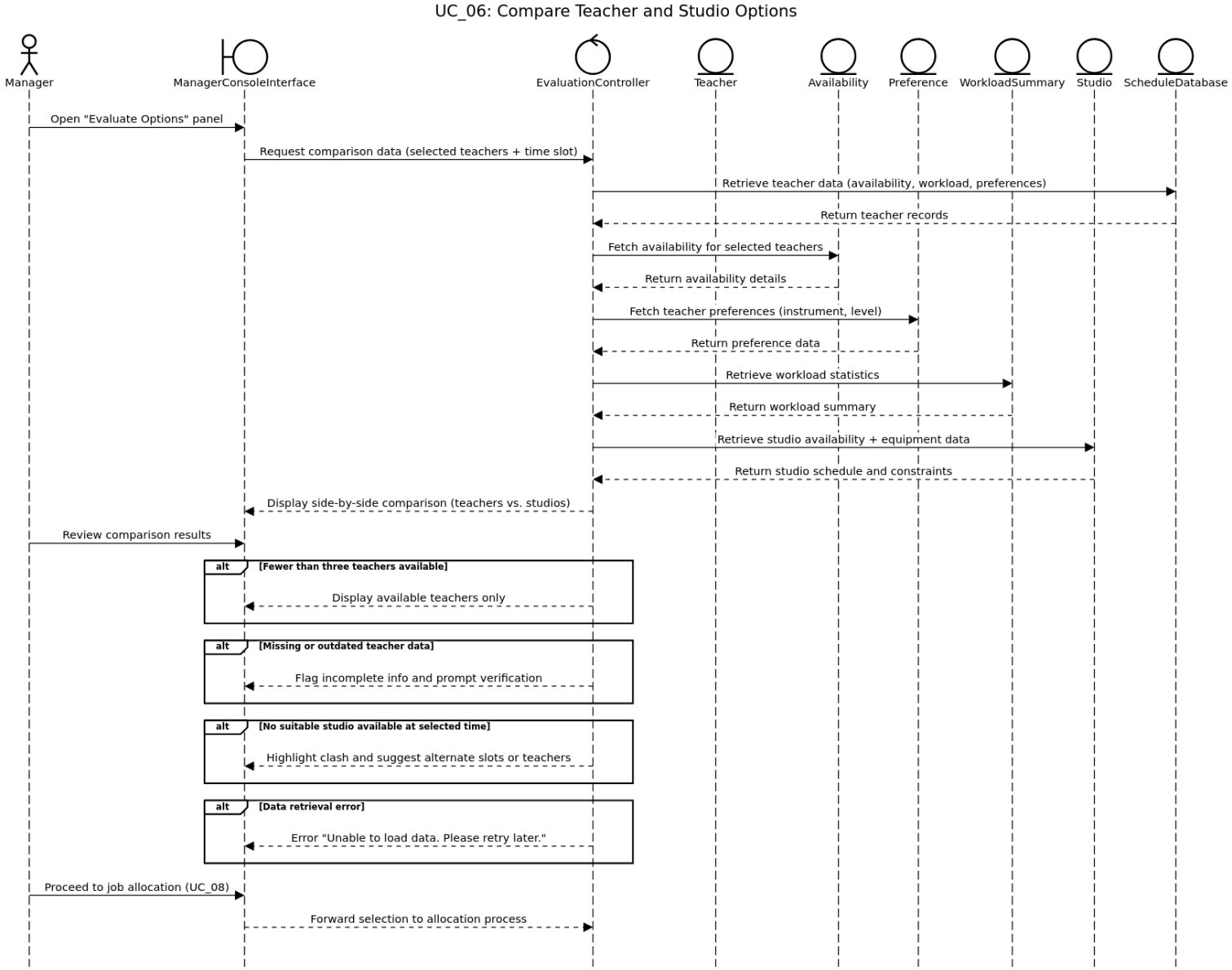

UC_02: Edit Availability (up to one month in advance)
Teacher
TeacherDashboardInterface
AvailabilityController
Availability
SchedulingPolicy
ScheduleDatabase
Select "Availability"
Request current default availability
Retrieve default availability + public holidays
Return default settings
Apply default policy (workdays avail, weekends off)
Validated schedule defaults
Display pre-filled calendar view
Select date(s) to modify
Send update request (selected dates/times)
Validate modification rules (≤1 month, no overlap)
Validation result (OK / Error)
alt
[Dates beyond allowed window]
Warning "Only up to one month in advance."
[Overlapping or invalid times]
Highlight conflicts, request correction
[Valid update]
Update records with changes
Confirmation of changes
Save updated availability
Save successful
Display success confirmation
alt
[System fails to save changes]
Error "Unable to save changes. Please retry later."
alt
[Teacher cancels before saving]
Discard pending updates
Reload previous availability
alt
[Public holiday not reflected correctly]
Report issue
Forward report to ITAdmin
Notify holiday config mismatch
Await correction by ITAdmin
Display note "Holiday update pending system correction."


UC_03: Set Job Preferences (Instrument / Level)
Teacher
TeacherDashboardInterface
PreferenceController
Preference
ScheduleDatabase
Select "Job Preferences"
Request current preferences
Retrieve teacher preference data
Return current preferences (or default: any instrument/level)
Display current preferences
Select instrument specialties
Select proficiency levels
Send updated preference data
Validate inputs (instruments, levels)
Validation result (OK / Error)
alt
[Validation OK]
Update profile attributes
Confirm update
Save new preferences
Save successful
Display success confirmation
[Validation failed]
Display input error message
alt
[Teacher cancels before saving]
Discard pending edits
Reload previous preferences
alt
[System error while saving]
Error "Unable to save preferences. Please retry later."
alt
[Teacher does not set preferences]
Retain defaults ("any instrument / any level")

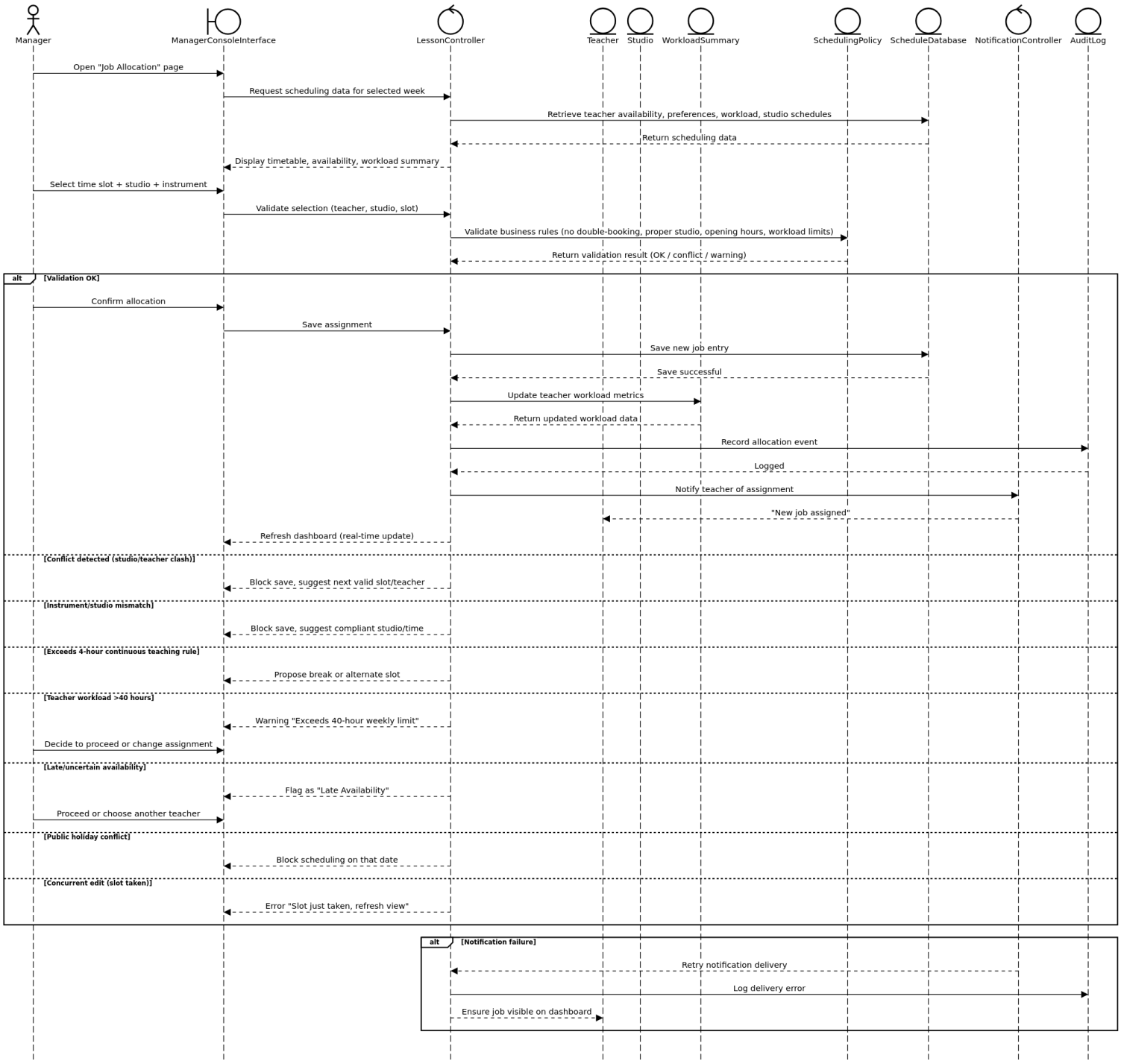

UC_04: Respond to Assigned Job (with Warning + Manager Notification)
Manager
Teacher
TeacherDashboardInterface
LessonController
NotificationController
Lesson
WorkloadSummary
ScheduleDatabase
Assign new job (teacher, time, instrument, student level, studio)
Validate availability, workload, studio constraints
Validation OK
Create new job record
Job created
Send job notification to Teacher
Display job assignment notification
Open job notification
Request job details
Check workload rules (hours/week, breaks)
Return workload data
Display job details + workload warning (if any)
Choose Accept or Reject
alt
[Teacher Accepts]
Confirm job acceptance
Update teacher schedule with new job
Update successful
Notify Manager of acceptance
"Teacher accepted job"
Display success confirmation
[Teacher Rejects]
Submit rejection (with/without reason)
Log rejection reason
Confirmation logged
Notify Manager of rejection
"Teacher rejected job (reason if provided)"
Display confirmation of rejection
alt
[Teacher identifies personal conflict]
Enter conflict reason
Log conflict reason
Notify Manager with details
"Rejection reason: personal conflict"
alt
[Teacher does not respond within deadline (24h)]
Mark job Pending Manager Review
"No response from teacher within 24h"
alt
[System fails to update schedule]
Error "Unable to update schedule. Please retry later."
alt
[Teacher rejects without reason]
Log rejection (no reason)
"Teacher rejected job (no reason provided)"

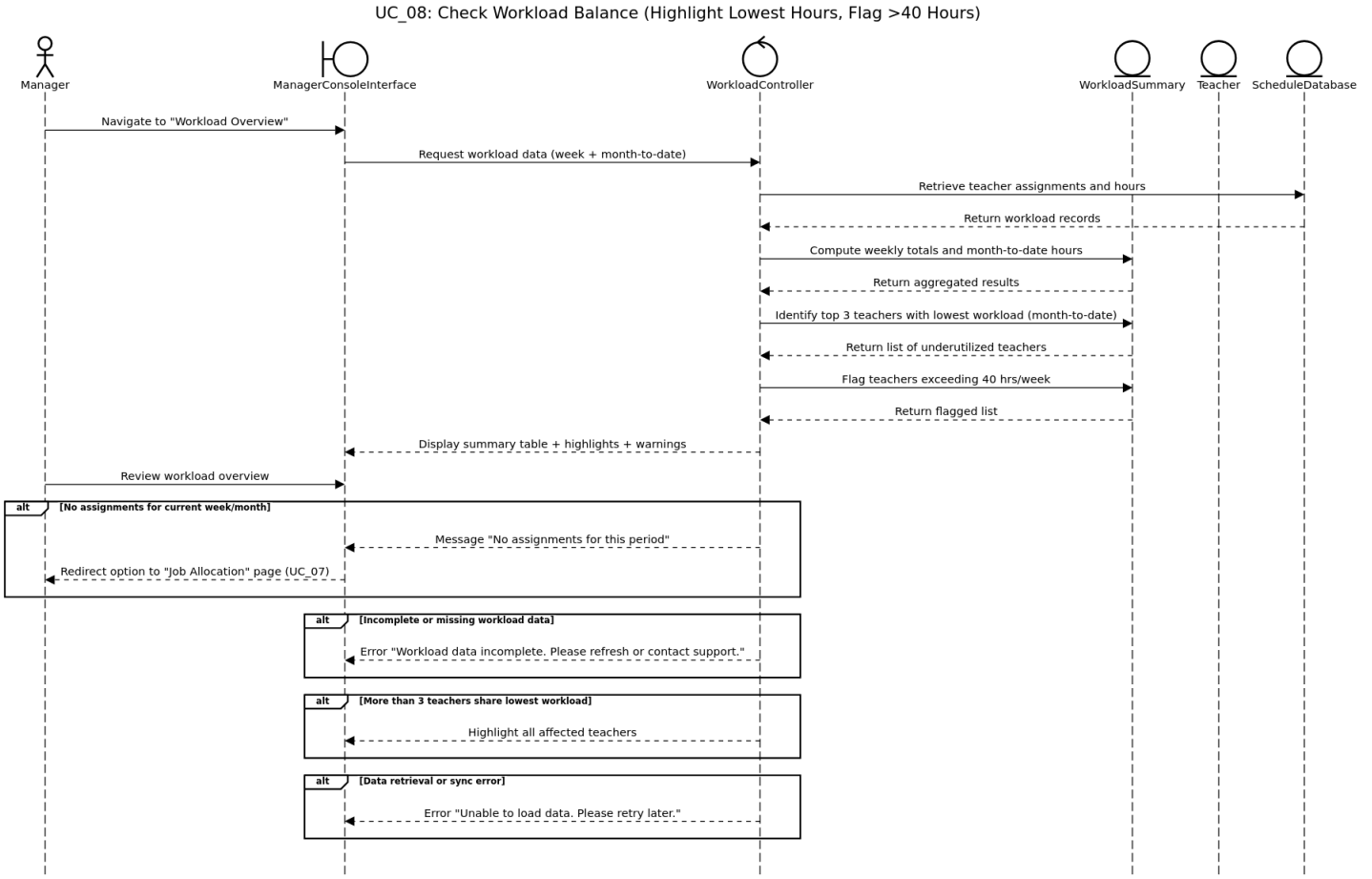
UC_05: View Manpower Overview (Availability, Workload, Preferences, Studio Usage)

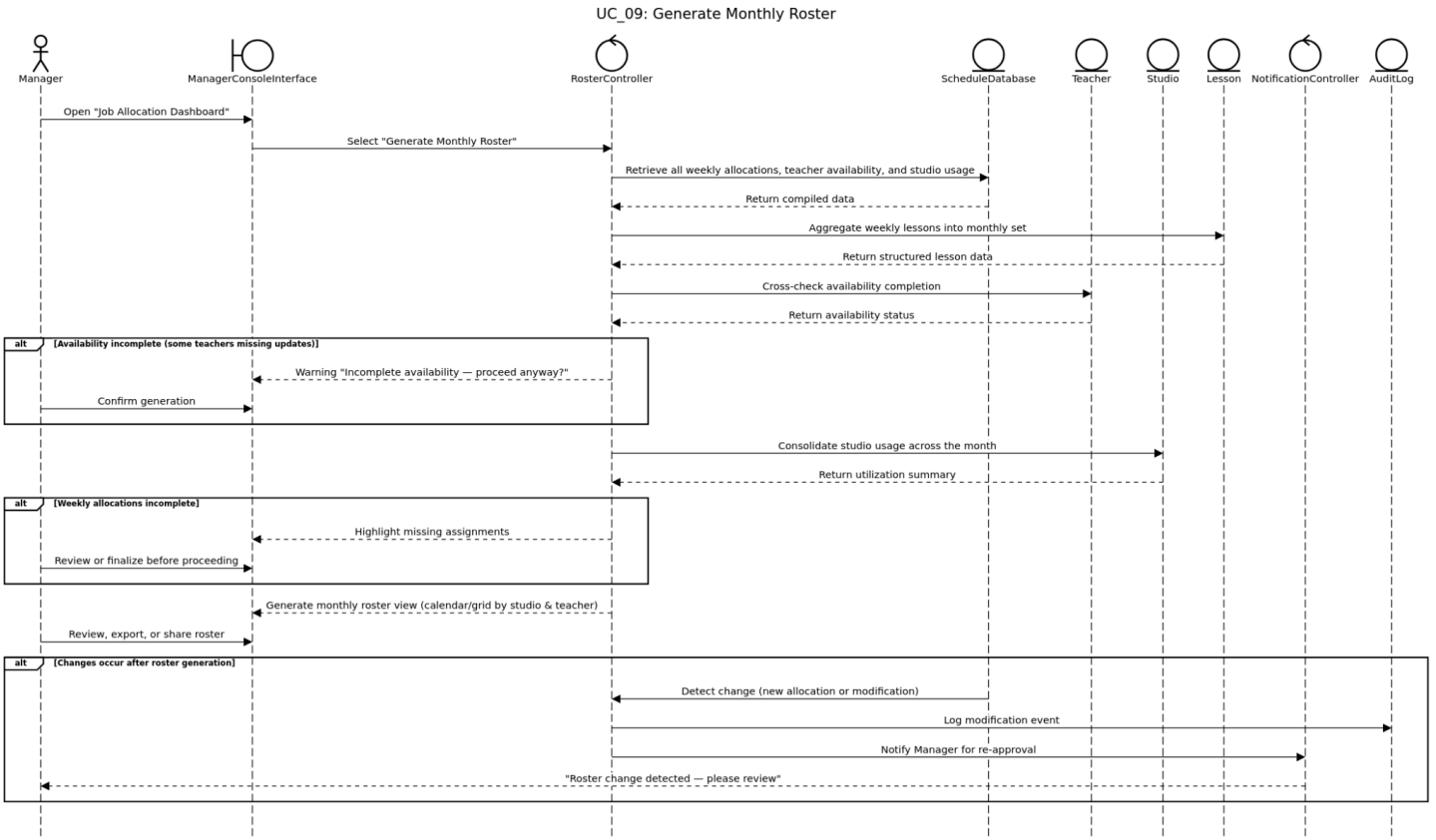



UC_06: Compare Teacher and Studio Options
Manager
ManagerConsoleInterface
EvaluationController
Teacher
Availability
Preference
WorkloadSummary
Studio
ScheduleDatabase
Open "Evaluate Options" panel
Request comparison data (selected teachers + time slot)
Retrieve teacher data (availability, workload, preferences)
Return teacher records
Fetch availability for selected teachers
Return availability details
Fetch teacher preferences (instrument, level)
Return preference data
Retrieve workload statistics
Return workload summary
Retrieve studio availability + equipment data
Return studio schedule and constraints
Display side-by-side comparison (teachers vs. studios)
Review comparison results
alt
[Fewer than three teachers available]
Display available teachers only
alt
[Missing or outdated teacher data]
Flag incomplete info and prompt verification
alt
[No suitable studio available at selected time]
Highlight clash and suggest alternate slots or teachers
alt
[Data retrieval error]
Error "Unable to load data. Please retry later."
Proceed to job allocation (UC_08)
Forward selection to allocation process

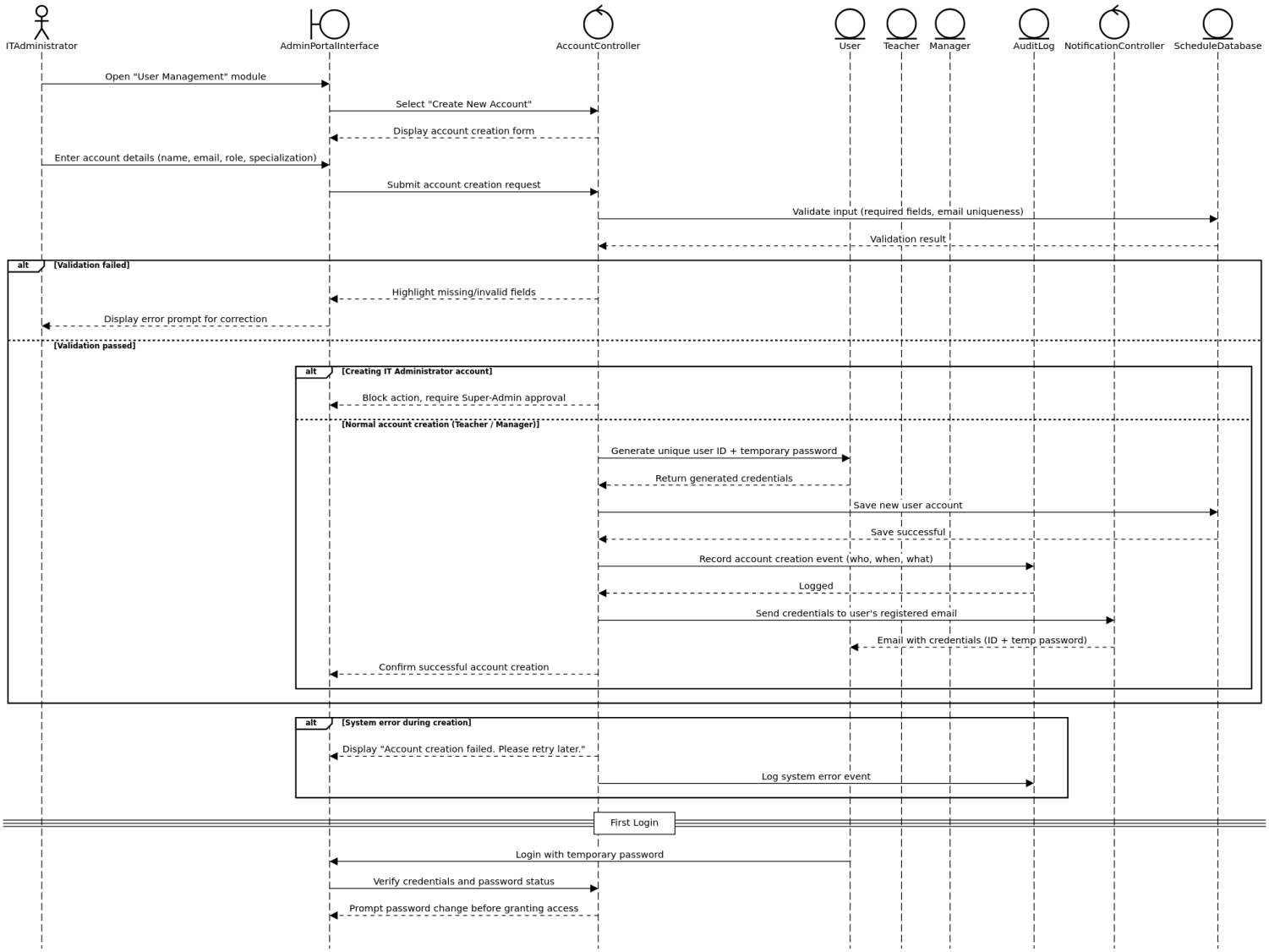
UC_07: Allocate Jobs (One Week at a Time)

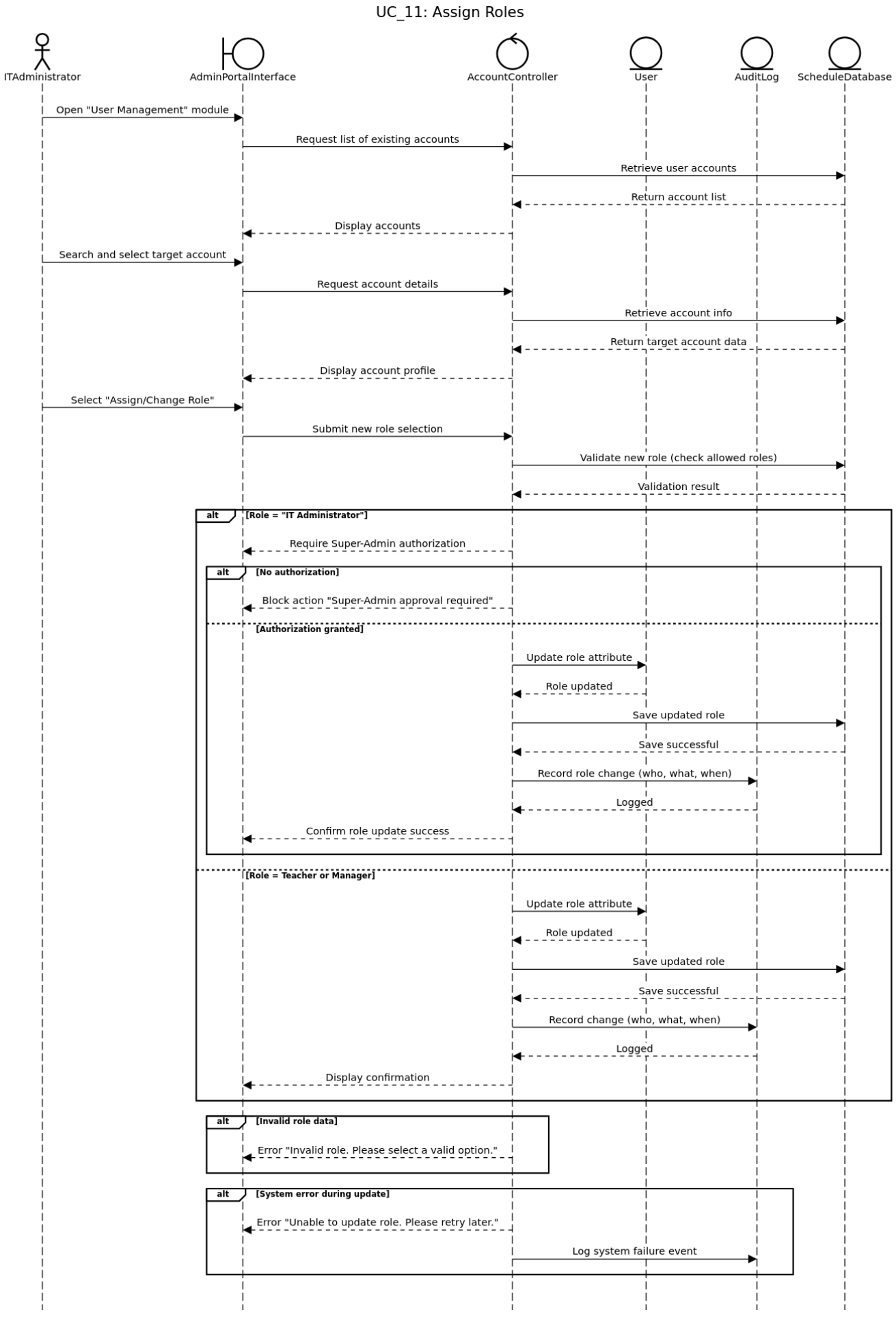



UC_08: Check Workload Balance (Highlight Lowest Hours, Flag >40 Hours)
Manager
ManagerConsoleInterface
WorkloadController
WorkloadSummary
Teacher
ScheduleDatabase
Navigate to "Workload Overview"
Request workload data (week + month-to-date)
Retrieve teacher assignments and hours
Return workload records
Compute weekly totals and month-to-date hours
Return aggregated results
Identify top 3 teachers with lowest workload (month-to-date)
Return list of underutilized teachers
Flag teachers exceeding 40 hrs/week
Return flagged list
Display summary table + highlights + warnings
Review workload overview
alt
[No assignments for current week/month]
Message "No assignments for this period"
Redirect option to "Job Allocation" page (UC_07)
alt
[Incomplete or missing workload data]
Error "Workload data incomplete. Please refresh or contact support."
alt
[More than 3 teachers share lowest workload]
Highlight all affected teachers
alt
[Data retrieval or sync error]
Error "Unable to load data. Please retry later."

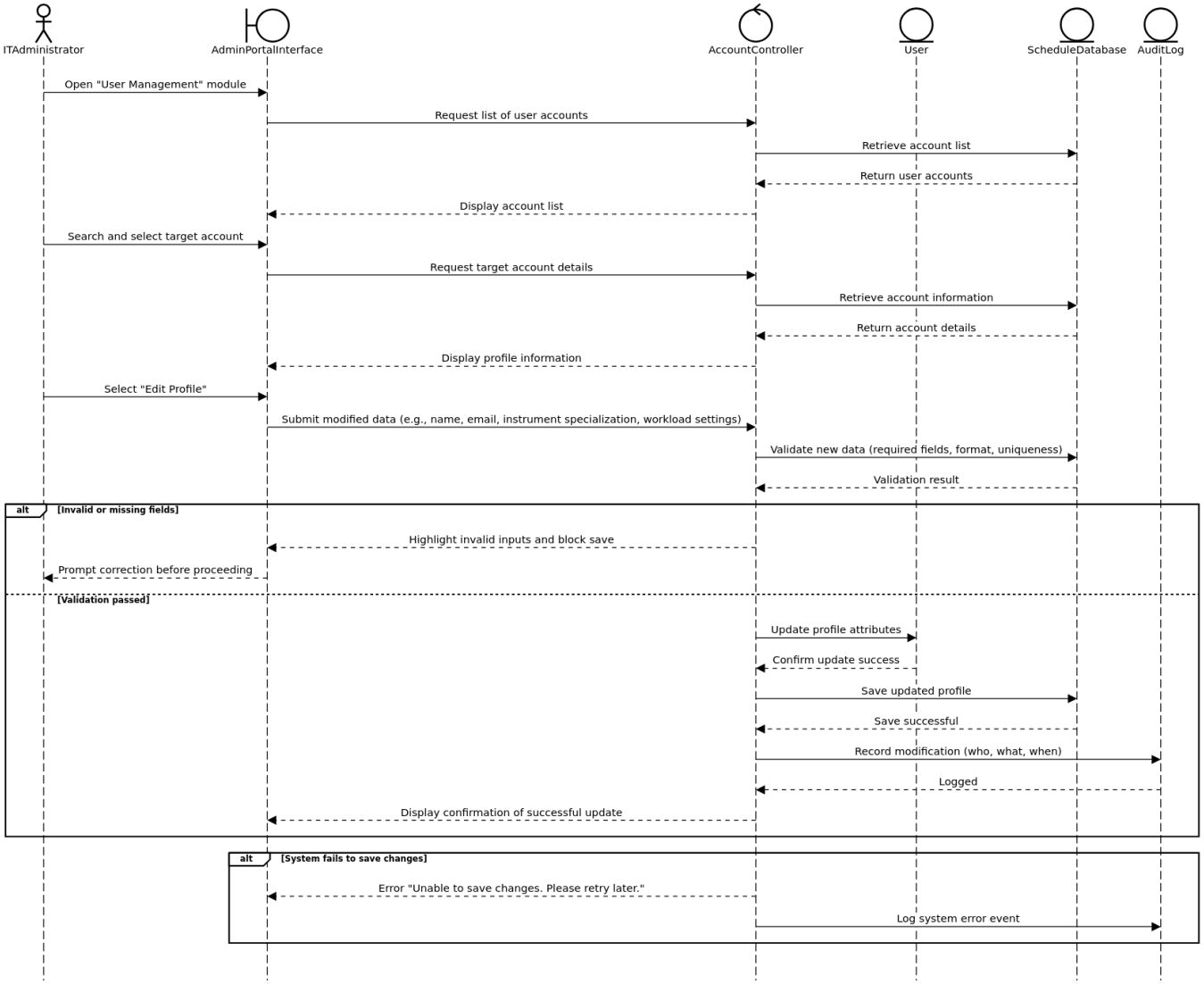

UC_09: Generate Monthly Roster
Manager
ManagerConsoleInterface
RosterController
ScheduleDatabase
Teacher
Studio
Lesson
NotificationController
AuditLog
Open "Job Allocation Dashboard"
Select "Generate Monthly Roster"
Retrieve all weekly allocations, teacher availability, and studio usage
Return compiled data
Aggregate weekly lessons into monthly set
Return structured lesson data
Cross-check availability completion
Return availability status
alt
[Availability incomplete (some teachers missing updates)]
Warning "Incomplete availability — proceed anyway?"
Confirm generation
Consolidate studio usage across the month
Return utilization summary
alt
[Weekly allocations incomplete]
Highlight missing assignments
Review or finalize before proceeding
Generate monthly roster view (calendar/grid by studio & teacher)
Review, export, or share roster
alt
[Changes occur after roster generation]
Detect change (new allocation or modification)
Log modification event
Notify Manager for re-approval
"Roster change detected — please review"

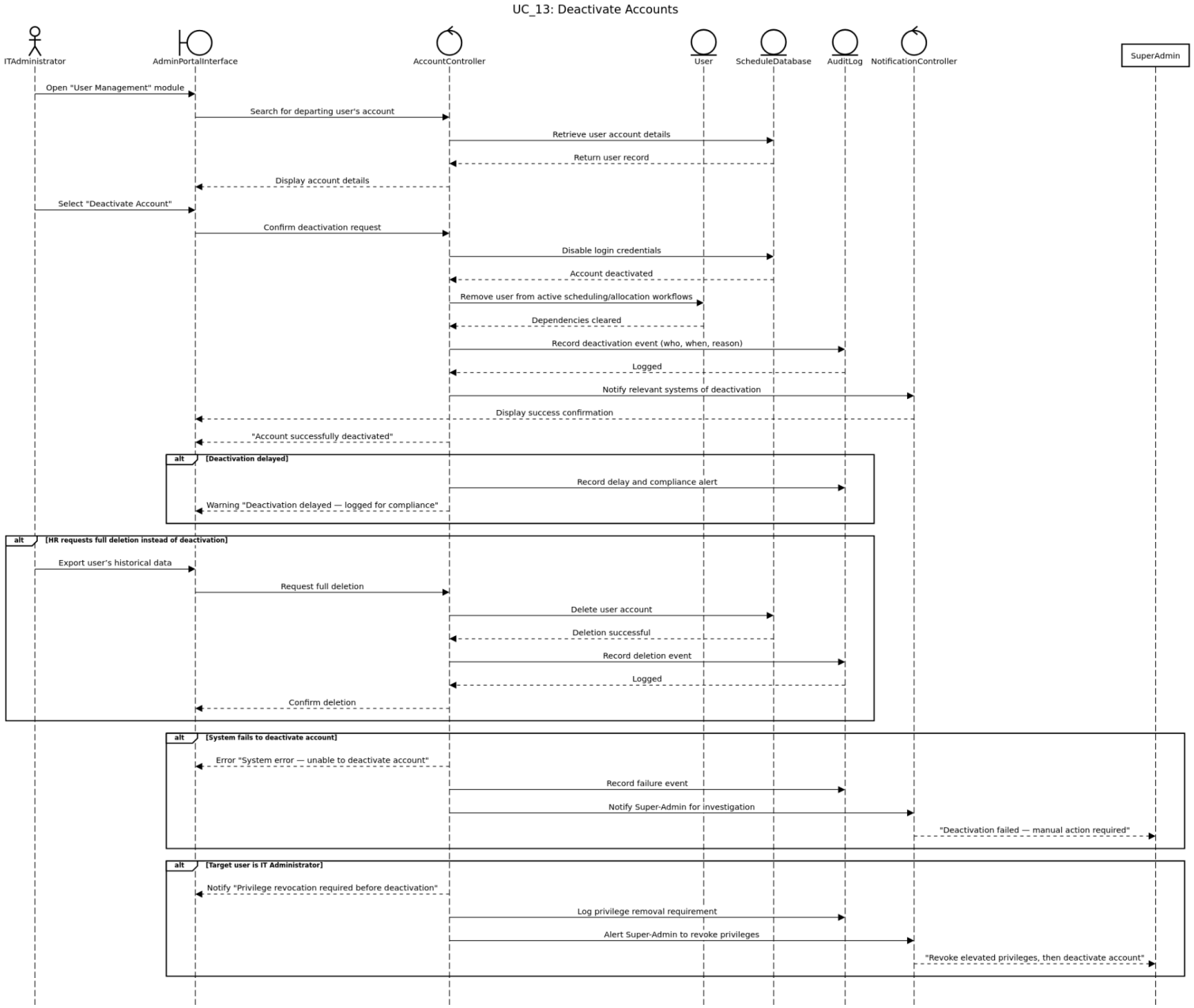
UC_10: Create Accounts

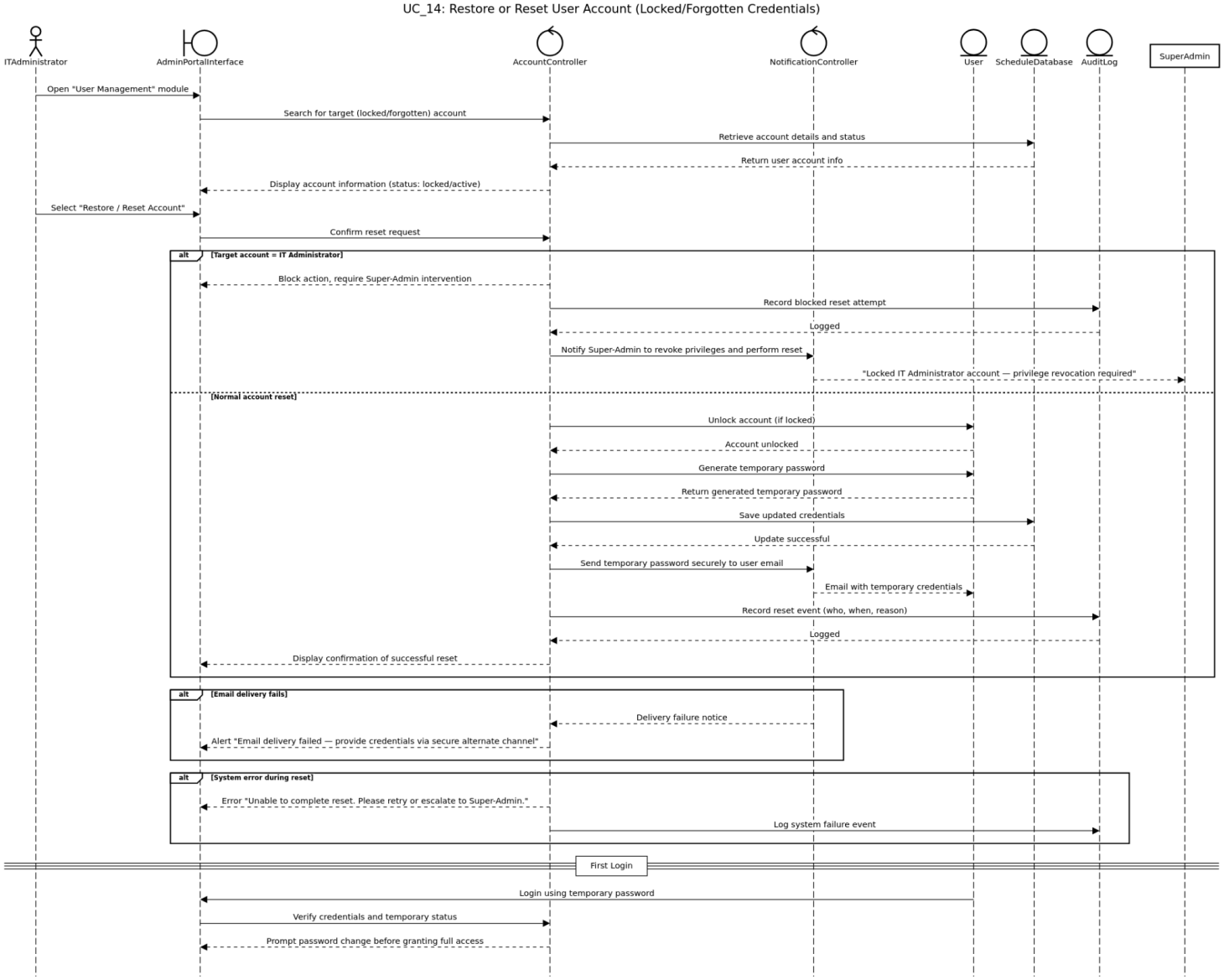



UC_11: Assign Roles
ITAdministrator
AdminPortalInterface
AccountController
User
AuditLog
ScheduleDatabase
Open "User Management" module
Request list of existing accounts
Retrieve user accounts
Return account list
Display accounts
Search and select target account
Request account details
Retrieve account info
Return target account data
Display account profile
Select "Assign/Change Role"
Submit new role selection
Validate new role (check allowed roles)
Validation result
alt
[Role = "IT Administrator"]
Require Super-Admin authorization
alt
[No authorization]
Block action "Super-Admin approval required"
[Authorization granted]
Update role attribute
Role updated
Save updated role
Save successful
Record role change (who, what, when)
Logged
Confirm role update success
[Role = Teacher or Manager]
Update role attribute
Role updated
Save updated role
Save successful
Record change (who, what, when)
Logged
Display confirmation
alt
[Invalid role data]
Error "Invalid role. Please select a valid option."
alt
[System error during update]
Error "Unable to update role. Please retry later."
Log system failure event

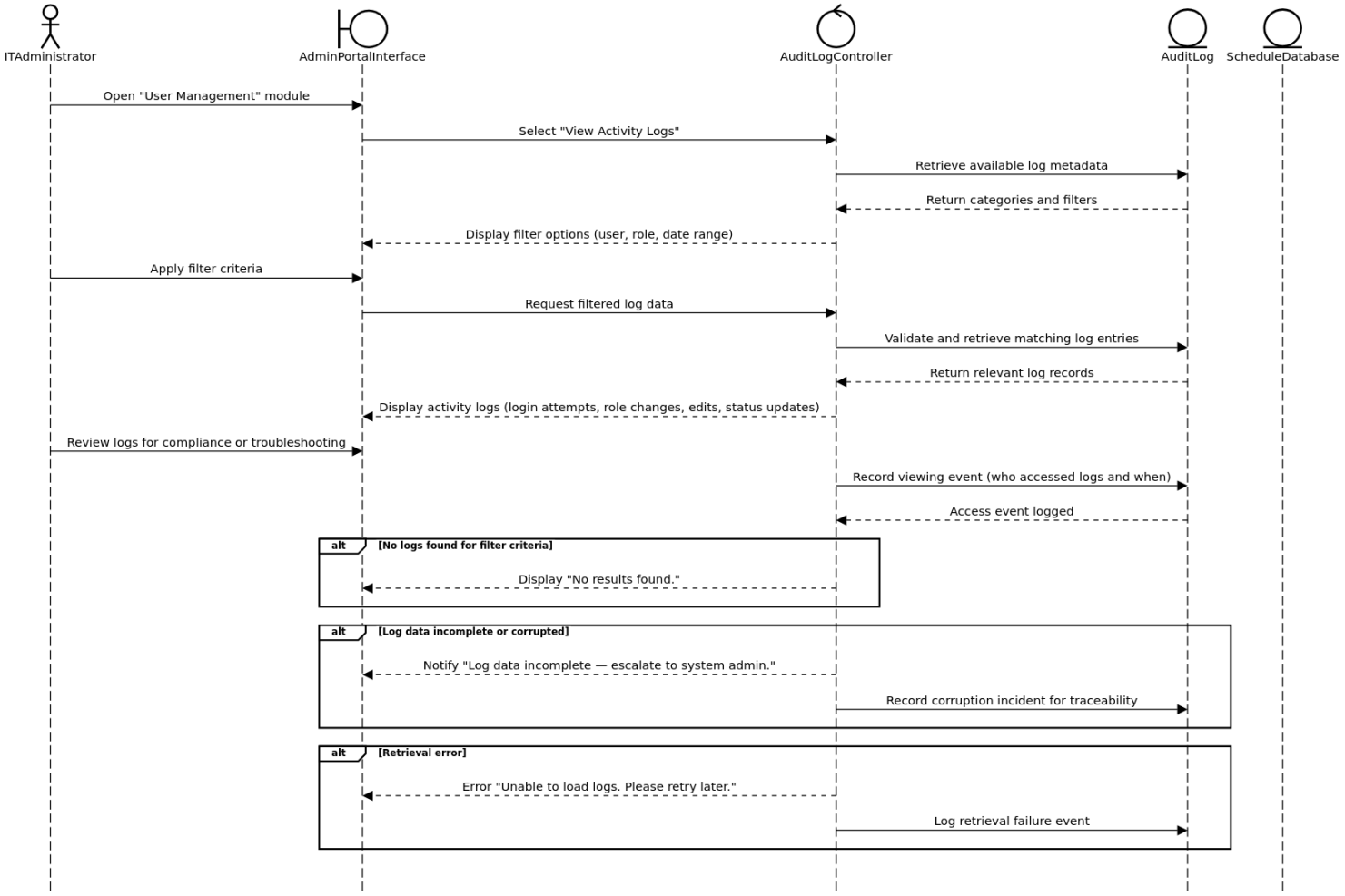
UC_12: Edit Account Profile



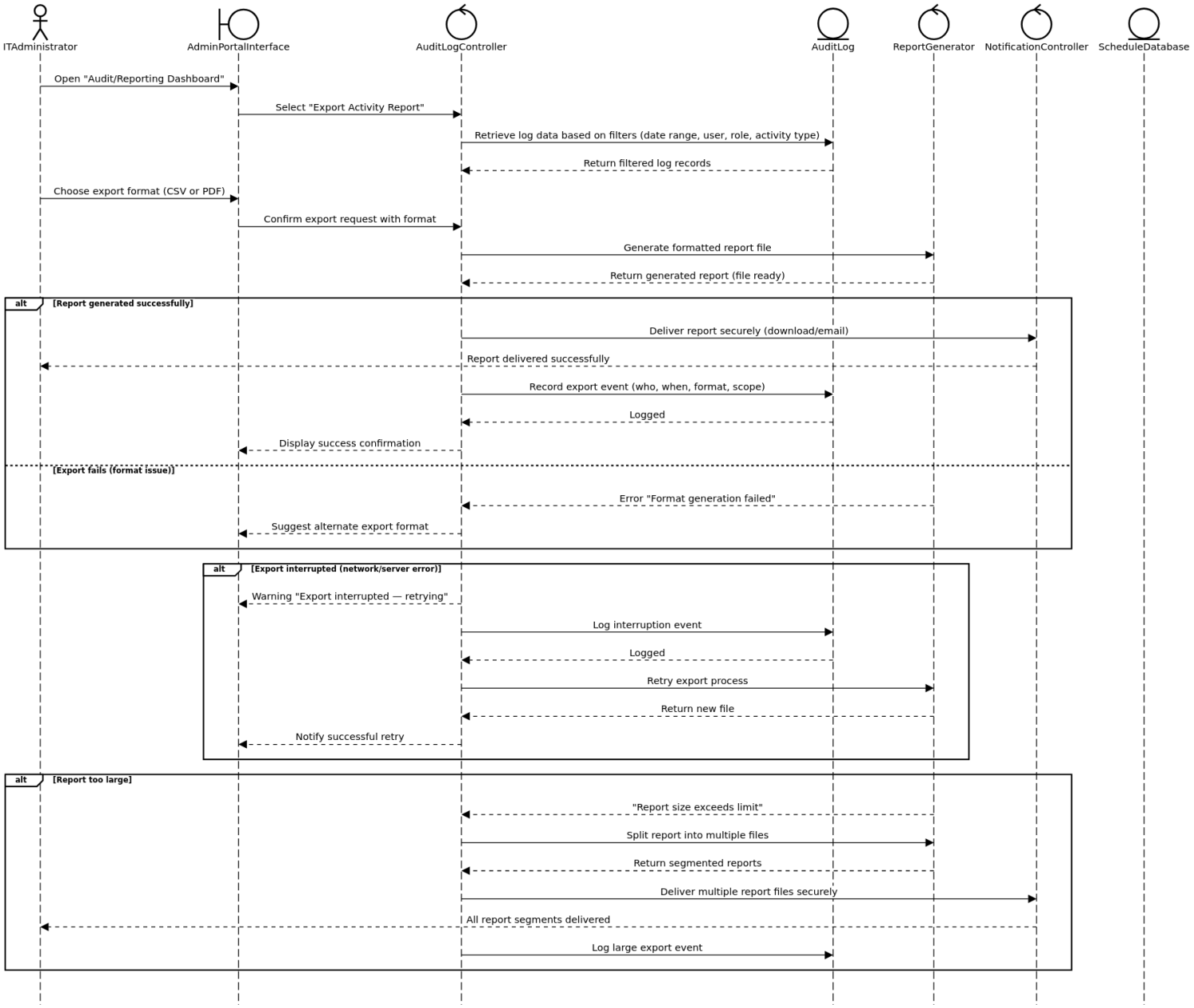

UC_13: Deactivate Accounts
ITAdministrator
AdminPortalInterface
AccountController
User
ScheduleDatabase
AuditLog
NotificationController
SuperAdmin
Open "User Management" module
Search for departing user's account
Retrieve user account details
Return user record
Display account details
Select "Deactivate Account"
Confirm deactivation request
Disable login credentials
Account deactivated
Remove user from active scheduling/allocation workflows
Dependencies cleared
Record deactivation event (who, when, reason)
Logged
Notify relevant systems of deactivation
Display success confirmation
"Account successfully deactivated"
alt
[Deactivation delayed]
Record delay and compliance alert
Warning "Deactivation delayed — logged for compliance"
alt
[HR requests full deletion instead of deactivation]
Export user's historical data
Request full deletion
Delete user account
Deletion successful
Record deletion event
Logged
Confirm deletion
alt
[System fails to deactivate account]
Error "System error — unable to deactivate account"
Record failure event
Notify Super-Admin for investigation
"Deactivation failed — manual action required"
alt
[Target user is IT Administrator]
Notify "Privilege revocation required before deactivation"
Log privilege removal requirement
Alert Super-Admin to revoke privileges
"Revoke elevated privileges, then deactivate account"


UC_14: Restore or Reset User Account (Locked/Forgotten Credentials)
ITAdministrator
AdminPortalInterface
AccountController
NotificationController
User
ScheduleDatabase
AuditLog
SuperAdmin
Open "User Management" module
Search for target (locked/forgotten) account
Retrieve account details and status
Return user account info
Display account information (status: locked/active)
Select "Restore / Reset Account"
Confirm reset request
alt
[Target account = IT Administrator]
Block action, require Super-Admin intervention
Record blocked reset attempt
Logged
Notify Super-Admin to revoke privileges and perform reset
"Locked IT Administrator account — privilege revocation required"
[Normal account reset]
Unlock account (if locked)
Account unlocked
Generate temporary password
Return generated temporary password
Save updated credentials
Update successful
Send temporary password securely to user email
Email with temporary credentials
Record reset event (who, when, reason)
Logged
Display confirmation of successful reset
alt
[Email delivery fails]
Delivery failure notice
Alert "Email delivery failed — provide credentials via secure alternate channel"
alt
[System error during reset]
Error "Unable to complete reset. Please retry or escalate to Super-Admin."
Log system failure event
First Login
Login using temporary password
Verify credentials and temporary status
Prompt password change before granting full access

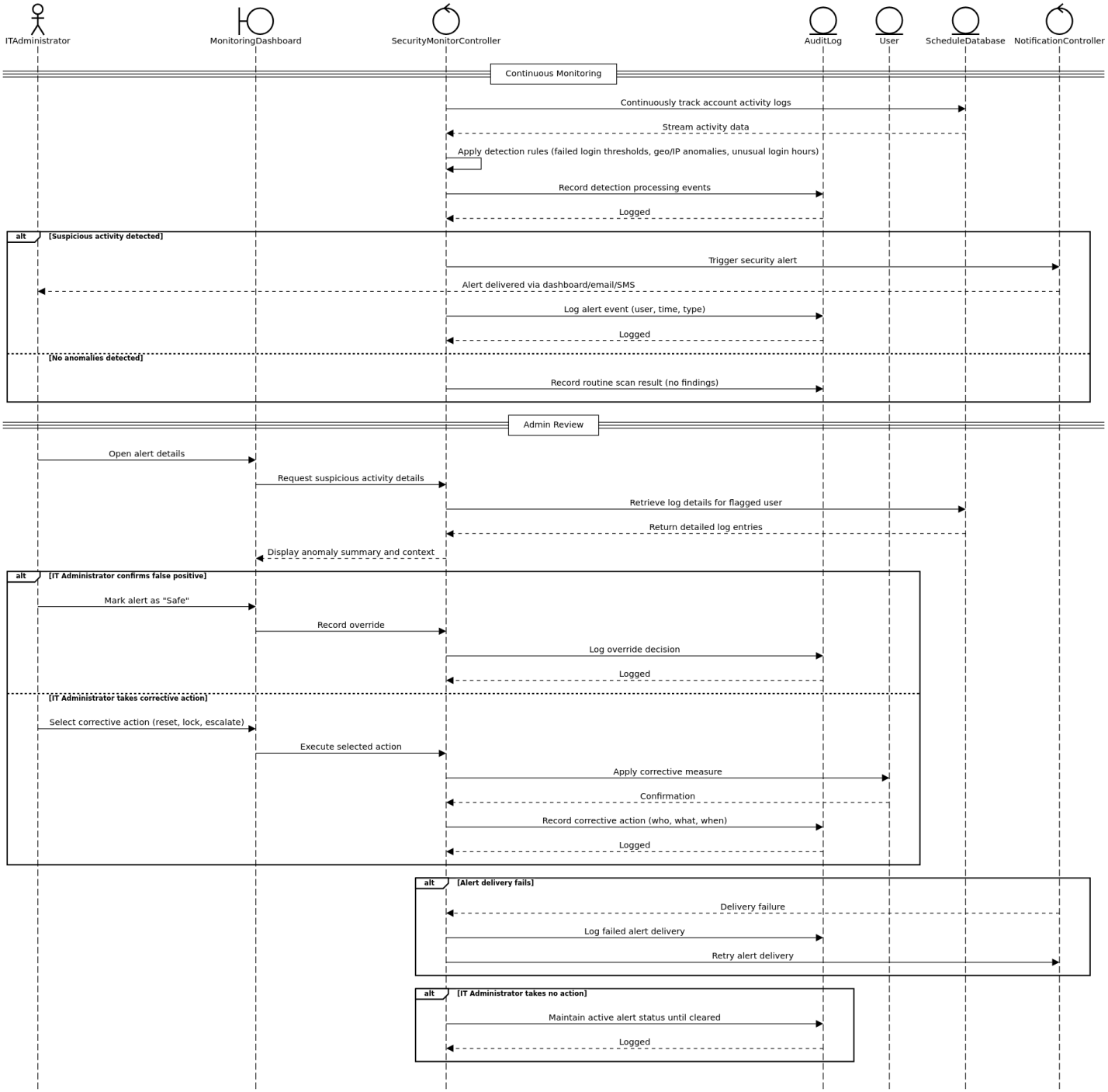
UC_15: View Account Activity Logs



UC_16: Export Account Activity Reports

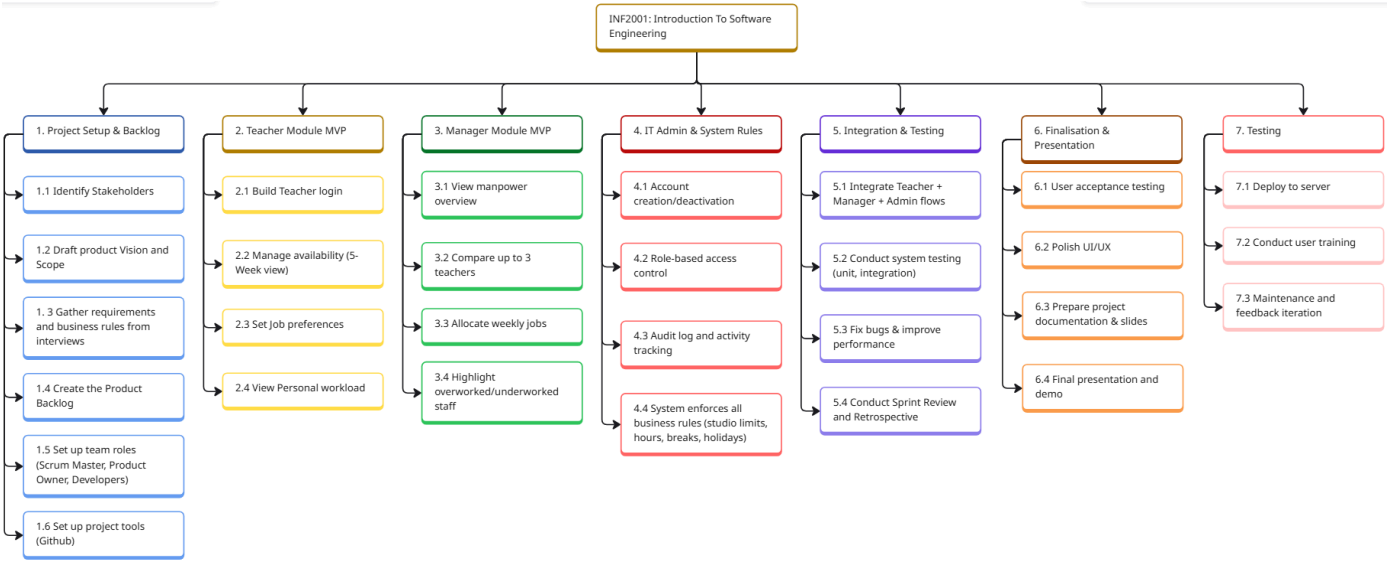


UC_17: Monitor Suspicious Account Activity



5. Project Management

5.1 Work breakdown structure

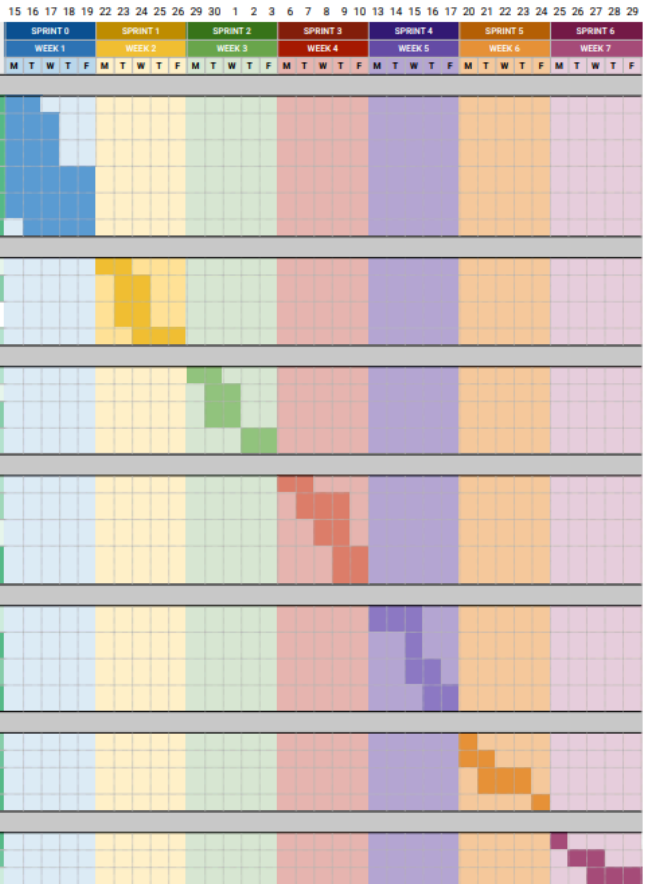


5.2 Project Timeline

GANTT CHART

PROJECT TITLE	INF2001: Introduction to Software Engineering	COMPANY NAME	Singapre Institute of Technology
PROJECT MANAGER	Genisa Lee	DATE	9/15/25

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Setup & Backlog**					
1.1	Identify Stakeholders	Genisa	9/15/25	9/16/25	1	100%
1.2	Draft product Vision and Scope	Lavanya Sebastian	9/15/25	9/17/25	2	100%
1.3	Gather requirements and business rules from interviews	Genisa	9/15/25	9/17/25	2	100%
1.4	Create the **Product Backlog**	Genisa (Lead), Team	9/15/25	9/19/25	4	100%
1.5	Set up team roles (Scrum Master, Product Owner, Developers)	Jin Rui	9/15/25	9/19/25	4	100%
1.6	Set up project tools (GitHub)	Lavanya	9/16/25	9/19/25	3	100%
2	**Teacher Module MVP**					
2.1	Build Teacher login	Genisa	9/22/25	9/23/25	1	70%
2.2	Manage availability (5-Week view)	Genisa (Lead), Jin Rui	9/23/25	9/24/25	1	90%
2.3	Set Job preferences	Jin Rui (Lead), Zulfaqar	9/23/25	9/24/25	1	65%
2.4	View Personal workload	Jin Rui	9/24/25	9/26/25	2	80%
3	**Manager Module MVP**					
3.1	View manpower overview	Max	9/29/25	9/30/25	1	80%
3.2	Compare up to 3 teachers	Jin Rui	9/30/25	10/1/25	1	70%
3.3	Allocate weekly jobs	Zulfaqar Sebestian	9/30/25	10/1/25	1	90%
3.4	Highlight overworked/underworked staff	Zulfaqar	10/2/25	10/3/25	1	80%
4	**IT Admin & System Rules**					
4.1	Account creation/deactivation	Genisa	10/6/25	10/7/25	1	80%
4.2	Role-based access control	Genisa Max	10/7/25	10/9/25	2	85%
4.3	Audit log and activity tracking	Max (Lead), Team	10/8/25	10/9/25	1	70%
4.4	System enforces all business rules (studio limits, hours, breaks, holidays)	Max	10/9/25	10/10/25	1	100%
5	**Integration & Testing**					
5.1	Integrate Teacher + Manager + Admin flows	Jin Rui	10/13/25	10/15/25	2	75%
5.2	Conduct system testing (unit, integration)	Zulfaqar	10/15/25	10/15/25	0	100%
5.3	Fix bugs & improve performance	Max (Lead), Team	10/15/25	10/16/25	1	90%
5.4	Conduct Sprint Review and Retrospective	Max (Lead), Team	10/15/25	10/17/25	2	100%
6	**Finalisation & Presentation**					
6.1	User acceptance testing	Jin Rui	10/20/25	10/20/25	0	90%
6.2	Polish UI/UX	Zulfaqar	10/20/25	10/21/25	1	95%
6.3	Prepare project documentation & slides	Max (Lead), Team	10/21/25	10/23/25	2	100%
6.4	Final presentation and demo	Lavanya	10/24/25	10/24/25	0	100%
7	**Testing**					
7.1	Deploy to server	Lavanya	10/25/25	10/25/25	0	100%
7.2	Conduct user training	Sebastian	10/26/25	10/27/25	1	95%
7.3	Maintenance and feedback iteration	Genisa	10/27/25	10/29/25	2	75%



6. Plagiarism and GenAI usage declarat

6.1 Plagiarism declarations

We hereby declare that this report is the result of our own original work.
We confirm that:

1. No part of this report has been copied from any other student's work, published material, or online source without proper acknowledgement.

2. Any external references, ideas, or supporting material used in this report have been properly cited and acknowledged in accordance with academic standards.

3. We understand that plagiarism is a serious academic offence and accept full responsibility for the content of this submission.

4. We further confirm that this work has not been previously submitted for academic credit in this or any other course.

6.2 AI usage and declarations

GenAI Declaration

We acknowledge that we used Generative AI (ChatGPT, OpenAI, 2025) to assist in drafting and refining the interview questions for managers, teachers , and IT administrators in **Section 1 – Requirement Engineering** of this report.

- The AI was used to **generate question phrasing and expand ideas** based on the project description and initial requirements provided.

- The AI did **not generate any final answers**; all stakeholder responses were obtained through role-play and group interviews.

- The AI did **not produce any diagrams, functional requirements, or non-functional requirements** directly. These were created and refined by the project team with suggestive assistance and systemic review from AI.

- The project team has reviewed, edited, and validated all AI-generated content to ensure accuracy, appropriateness, and alignment with the course requirements.

- We used a Generative AI assistant (ChatGPT) to draft (i) interview questions for Managers, Teachers, and IT Administrators, and (ii) an initial list of functional requirements for the "Music School" workload management system. All AI-generated content was subsequently reviewed, edited, and validated by our team against the project brief and initial requirements, and with stakeholder feedback captured in our

interviews. No private or personal data was provided to the AI. We accept full responsibility for the final content and any errors.

This declaration ensures transparency in the use of AI tools in our coursework

7. Appendix

1. Detailed use cases

ACTORS	USE CASES INITIATED
Teacher, Manager, IT administrator	- **UC_01:** Login to system
Teacher	- **UC_02:** View schedule & workload (weekly/monthly) - **UC_03:** Manage availability (up to 5 weeks) - **UC_04:** Set job preferences (instrument/level) - **UC_05:** Respond to assigned job (with warning + manager notification)
Manager	- **UC_06:** View manpower overview (availability, workload, preferences, studio usage) - **UC_07:** Evaluate Teacher and Studio Options - **UC_08:** Allocate jobs (one week at a time) - **UC_09:** Identify workload imbalance (highlight teacher with lowest hours; flag teacher >40 hours)
IT administrator	- **UC_10:** Create accounts - **UC_11:** Modify account data (Assign role, edit profile etc) - **UC_12:** Deactivate accounts when teacher leave - **UC_13:** Manage account activity for audit & compliance

(Fig.1a Detailed Use cases)

Use Case Descriptions

ID:	UC_01
Name:	View schedule & workload (weekly/monthly)
Description:	Teachers are able to requests to view their schedule and workload after logging into the system. The system interface displays a calendar view. The system controller retrieves assigned classes, lessons, and events from the schedule database, processes them, and sends the formatted data back to the interface. The interface then presents the schedule in a weekly or monthly format, with the teacher able to switch between views by clicking a button.
Primary Actor:	Teacher
Preconditions:	1. Teacher is successfully logged into the system 2. Teacher has existing work, classes, or events already scheduled in the system 3. System has schedule data stored and accessible
Main Success Scenario:	1. Teacher selects “View Schedule” from the main dashboard 2. System requests schedule data from internal storage 3. System compiles assigned classes, lessons, and events into structured format 4. System displays schedule in a default weekly calendar view with workload summary 5. Teacher clicks toggle to switch between weekly and monthly formats 6. System updates the calendar display accordingly
Alternative Scenarios:	**2a. Teacher does not select a view option** 1. Teacher does nothing after opening the schedule page 2. System defaults to displaying the weekly schedule view **3a. No assignments or classes scheduled** 1. System finds no assigned work, classes, or events in storage 2. System displays a message: “No scheduled items found” along with an empty schedule grid **3b. System cannot retrieve data (server error)** 1. System fails to fetch schedule data from internal storage 2. System displays an error message: “Unable to load schedule. Please retry later or contact IT Administrator.” 3. Teacher may retry or seek support
Postconditions:	1. Teacher can see their current and upcoming schedule in weekly or monthly format 2. Workload information is visible, supporting planning and

	time management
Priority:	High - schedule visibility is essential for teacher to plan lessons and manage workload effectively; without it, teaching operations may be disrupted.

(Fig.1b UC1)

ID:	UC_02
Name:	Edit availability (up to one month in advance)
Description:	This use case allows a teacher to adjust their default availability for the upcoming month. By default, teachers are marked as available on standard workdays and unavailable on weekends and public holidays. The teacher only edits when exceptions occur (e.g., leave, off-days, or volunteering for extra hours). The system records these adjustments and ensures that managers see the updated availability during job allocation.
Primary Actor:	Teacher
Preconditions:	1. Teacher successfully logged into the system 2. System has default availability settings (workdays = available, weekends/public holidays = unavailable)
Main Success Scenario:	1. Teacher selects “Availability” from the main dashboard. 2. System displays the pre-filled calendar (default availability: weekdays marked available; weekends/public holidays unavailable). 3. Teacher selects a date or range within the one-month window. 4. Teacher adjusts availability by either: a. Marking specific days/times as unavailable, or b. Adding extra available slots outside normal hours. 5. System highlights any changes made. 6. Teacher reviews and confirms the updates. 7. System validates entries (check for conflicts, overlaps, or out-of-range dates). 8. System saves the availability changes, updates the teacher’s record, and displays a success confirmation.
Alternative Scenarios:	**2a. Public holiday not reflected correctly (system error)** 1. Teacher notices that a known public holiday is not blocked in the system 2. Teacher contacts IT Administrator to report the issue 3. IT Administrator verifies the holiday error and updates the system holiday/calendar configuration 4. System automatically corrects availability for that holiday across all teachers. **3a. Teacher attempts to select dates beyond one month in advance:** 1. System displays warning: “Availability can only be set up to one month in advance.” **4a. Teacher enters overlapping or invalid timeslots** 1. System highlights the conflict and requests correction before proceeding. **6a. System fails to save changes (e.g., network/server error)** 1. System shows error message and advises retry later.

	6b. Teacher cancels changes before saving 1. System discards unsaved changes and reloads the previous availability view.
Postconditions:	1. Teacher's customized availability overrides the default schedule for the upcoming month. 2. Managers see the updated availability when planning and allocating jobs.
Priority:	High - essential for ensuring accurate workload planning and avoiding scheduling conflicts.

(Fig.1c UC2)

ID:	UC_03
Name:	Set job preferences (instrument/level)
Description:	This use case allows a teacher to update their job preferences through the “Job Preferences” section of their profile. Here, teachers can define their instrument specialties and preferred student proficiency levels. These preferences help managers allocate jobs more effectively. Preferences are not strict restrictions, managers may override them if required to meet scheduling or demand needs.
Primary Actor:	Teacher
Preconditions:	1. Teacher successfully logged into the system
Main Success Scenario:	1. Teacher selects “Job Preferences” from the main dashboard. 2. System displays current instrument specialties and proficiency levels (default = “any instrument / any level” if none set). 3. Teacher selects one or more instrument specialties they are qualified to teach. 4. Teacher selects preferred student proficiency levels (e.g., Beginner, Intermediate, Advanced). 5. Teacher confirms and saves preferences. 6. System validates input, updates the teacher profile, and shows a confirmation message. 7. System makes updated preferences visible to managers during job allocation.
Alternative Scenarios:	**2a. Teacher does not set preferences** 1. System retains default setting (“any instrument / any level”). **4a. Teacher cancels changes before saving** 1. System discards edits and reloads previous preferences or default values. **4a. System error while saving** 1. System displays error message and prompts teacher to retry later.
Postconditions:	1. Teacher’s instrument and level preferences are stored in their profile. 2. Managers can view preferences when assigning jobs.
Priority:	Medium - supports teacher satisfaction and better workload matching, but managers retain override authority to meet operational needs.

(Fig.1d UC3)

ID:	UC_04
Name:	Respond to assigned job (with warning + manager notification)
Description:	This use case allows a teacher to respond to a direct job assignment sent by a manager. The teacher receives a notification through the system, reviews the job details, and either accepts or rejects the assignment. If accepted, the job is added to their schedule, if rejected, the system logs the rejection and notifies the assigning manager. Managers may override or follow up as needed.
Primary Actor:	Teacher
Secondary Actor:	Manager
Preconditions:	1. Teacher successfully logged into the system 2. A manager has assigned a specific job directly to the teacher.
Main Success Scenario:	1. Manager selects a teacher and assigns a new job (time, instrument, student level, studio). 2. System validates the assignment (availability, workload rules, studio constraints). 3. System creates a job notification and delivers it to the teacher's portal. 4. Teacher opens the notification and reviews job details. 5. System checks workload rules again and shows a warning if guidelines are exceeded (e.g., >40 hrs/week, >4 hours without break). 6. Teacher chooses to Accept or Reject. 7. If Accepted: a. System updates teacher's schedule. b. System notifies the assigning manager of acceptance. 8. If Rejected: a. System prompts teacher to enter a reason (optional if policy allows). b. Teacher submits reason or skips. 9. System logs the rejection and notifies the assigning manager.
Alternative Scenarios:	**4a. Teacher identifies personal conflict (e.g., medical appointment, family event) not known to system** 1. Teacher selects Reject and enters the conflict as the reason. 2. System logs the response and notifies the manager with the rejection reason. **6a. Teacher does not respond within deadline (e.g., 24 hours)** 1. System marks the job as Pending Manager Review. 2. System notifies the manager of the non-response.

	7a. System fails to update schedule due to technical error 1. System displays error message to the teacher. 2. Job remains unconfirmed. 3. Teacher retries after issue is resolved. **8a. Teacher attempts to reject without entering reason** 1. System accepts the rejection without requiring a reason. 2. System logs the rejection event and notifies the manager. 3. Manager may follow up directly with the teacher for clarification if needed.
Postconditions:	1. If accepted: Job is added to teacher's schedule, manager notified. 2. If rejected: Response (with/without reason) logged, manager notified, further action determined by manager.
Priority:	Medium - important for flexibility in workload management, but managers remain the final decision-makers.

(Fig.1e UC4)

ID:	UC_05
Name:	View manpower overview (availability, workload, preferences, studio usage)
Description:	This use case allows a manager to access the “Manpower Overview” dashboard to review teacher availability, workload distribution, job preferences, and studio usage. The system consolidates this information into weekly and monthly views, which the manager can switch between. Managers can expand on individual teachers or studios for detailed information. This forward-looking view helps anticipate demand, identify gaps, and balance workloads effectively.
Primary Actor:	Manager
Preconditions:	1. Manager is successfully logged into the system. 2. Teachers have updated their availability and preferences (up to one month in advance). 3. Current or upcoming jobs have been allocated to teachers.
Main Success Scenario:	1. Manager selects “Manpower Overview” from the dashboard. 2. System retrieves and displays a consolidated view of: a. Teacher availability (weekly and monthly outlook) b. Workload distribution (hours per teacher, projected vs. actual), c. Job preferences (instruments, levels), Studio usage (occupied vs. free slots across time horizon). 3. Manager switches between weekly and monthly views. 4. System updates the display according to the selected horizon (e.g., next week vs. full month). 5. Manager selects a specific teacher or studio for more detail. 6. System expands the selection and shows a detailed breakdown (e.g., assigned hours, free slots, usage patterns).
Alternative Scenarios:	**2a. Teacher availability not updated** 1. System highlights missing/expired availability data and flags projections as incomplete. **2b. Upcoming jobs not yet allocated** 1. System displays placeholder: *“Allocation in progress; some data may be missing.”* **2c. Data retrieval or sync error**

	1. System shows error message and suggests retrying later.
Postconditions:	1. Manager can view manpower, workload, preferences, and studio usage for current and upcoming periods. 2. Manager gains foresight to adjust planning, balance workload, and optimize studio utilization.
Priority:	High - forward visibility of manpower and studio usage is essential for smooth scheduling and profitability

(Fig.1f UC5)

ID:	UC_06
Name:	Compare Teacher and Studio Options
Description:	This use case allows a manager to open the “Evaluate Options” panel from the job allocation page to compare teachers and studios side-by-side. The system consolidates teacher availability, workload, and preferences with studio schedules and equipment constraints into a structured comparison view. This enables the manager to identify the most suitable teacher–studio pairing before confirming allocation, supporting fair workload distribution and efficient use of studio resources.
Primary Actor:	Manager
Preconditions:	1. Manager successfully logged into the system 2. Teacher have updated availability and preferences 3. Studios have defined schedules and equipment details
Main Success Scenario:	1. Manager navigates to “Evaluate Options” during job allocation. 2. Manager selects up to three teachers for comparison. 3. System displays, for each teacher: a. Availability (weekly/monthly outlook), b. Current workload (hours assigned; flag if >40 hrs), c. Preferences (instrument, level). 4. System shows studio availability alongside teacher data, including equipment requirements (e.g., drum studio constraint). 5. Manager reviews side-by-side comparison to identify the most suitable teacher–studio pairing. Manager proceeds to job allocation via UC_08: Allocate Jobs.
Alternative Scenarios:	**2a. Fewer than three teachers available:** 1. System displays comparison with only available teachers. **3a. Missing or outdated teacher data:** 1. System flags incomplete information and prompts manager to verify before proceeding. **4a. No suitable studio available at selected time**: 1. System highlights the clash and suggests alternate slots or teachers. **5a. Data retrieval error:** 1. System shows error message and advises retry later.
Postconditions:	1. Manager obtains a clear comparison of teacher and studio options before confirming allocation.
Priority:	Medium - information is already available in the overview, but this

	use case improves decision-making efficiency by presenting teacher and studio details side-by-side..

(Fig.1g UC6)

ID:	UC_07
Name:	Allocate jobs (one week at a time)
Description:	This use case allows a manager to allocate 30-minute lesson jobs to teachers one week at a time through the job allocation page. The manager selects a planning week, chooses a studio and time slot, specifies the instrument, and then assigns a teacher. The system validates the assignment against business rules, ensuring no studio double-booking, correct instrument–studio pairing, compliance with opening hours and public holidays, and teacher workload limits (no more than four consecutive teaching hours without a one-hour break, with warnings for weekly workloads exceeding 40 hours). Once confirmed, the system saves the allocation, updates the teacher's schedule, marks the studio slot as occupied, and refreshes workload indicators. Both managers' dashboards are updated in real time, and the assigned teacher receives a system notification. All changes are logged in the audit trail for accountability.
Primary Actor:	Manager
Preconditions:	1. Manager is authenticated and on the Job Allocation page. 2. The planning week is selected. 3. Teacher availability, preferences, and workload are recorded in the system. 4. System scheduling rules are configured (opening hours, public holidays, workload constraints).
Main Success Scenario:	1. Manager navigates to the job allocation page. 2. System displays: a. Studio timetable (availability and equipment). b. Teacher availability and preferences (instrument/level). c. Weekly workload distribution (highlighting >40 hrs and showing top 3 lowest-workload teachers). 3. Manager selects a time slot and studio (with instrument specified). 4. Manager selects a teacher. 5. System validates the assignment in real time: a. Studio not double-booked. b. Correct studio chosen (e.g., drum lessons only in drum studio). c. Within opening hours and not on a public holiday. d. Teacher available, not exceeding 4 consecutive hours without a 1-hour break. e. Warning if workload exceeds 40 hrs/week. 6. Manager confirms the assignment. 7. System saves the assignment, updates teacher and

	studio schedules, updates workload indicators, and creates an audit log entry. 8. System notifies the teacher of the new job. 9. System updates both managers' dashboards in real time.
Alternative Scenarios:	**5a. Conflict (Studio/Teacher clash):** 1. System blocks save and suggests the next valid slot/teacher. **5b. Instrument/studio mismatch (e.g. drum lesson in non-drum studio):** 1. System blocks save and suggests compliant studio/time options. **5c. Exceeds 4-hour block/breaks rule:** 1. System proposes a break or alternative slot. 2. Manager may select another teacher or slot. **5d. Teacher workload exceeding 40 hours:** 1. System shows warning. 2. Manager may still proceed. **5e. Late/uncertain availability:** 1. System flags as "late availability." 2. Manager may still assign or choose another teacher. **5f. Public holiday:** 1. System blocks scheduling. **5g. Concurrent edit(two managers assign same slot):** 1. System applies concurrency control. 2. System rejects second save with "Slot just taken" message and refreshes view **8a. Notification failure:** 1. System retries delivery and logs error. 2. System ensures teacher still sees job on dashboard.
Postconditions:	1. Job is added to teacher and studio schedules. 2. Workload metrics updated. 3. Teacher notified of new assignment. 4. Audit logs maintained for accountability.
Priority:	High - allocation is central to operations, directly impacting scheduling fairness, teacher satisfaction, and studio profitability.

(Fig.1h UC7)

ID:	UC_08
Name:	Check workload balance (highlight the lowest hours, flag >40 hours)
Description:	This use case allows a manager to review workload balance directly from the workload overview dashboard within the system's allocation workspace. The manager accesses this dashboard to view teacher assignments for the current week and month-to-date. The system automatically highlights the three teachers with the lowest cumulative hours (month-to-date) and flags any teachers scheduled for more than 40 hours in the current week. The information is presented visually through workload charts, indicators, or tables. By consolidating this data in one place, the system helps managers quickly identify underutilized teachers, prevent over-allocation, and make fairer job assignments.
Primary Actor:	Manager
Preconditions:	1. Manager is logged into the system. 2. Teacher job assignment and workload data for both the current week and month-to-date is stored in the system and up to date.
Main Success Scenario:	1. Manager navigates to the Workload Overview section on the dashboard. 2. System displays a summary view of teachers with their assigned hours for the current week and month-to-date. 3. System automatically highlights the top three teachers with the lowest workload for the month-to-date. 4. System flags any teachers whose assigned hours exceed 40 for the current week, using visual indicators (e.g., highlight color, warning icon). 5. Manager reviews the summary and uses the information to guide scheduling decisions or follow-up actions.
Alternative Scenarios:	**2a. No teacher have assigned jobs for the current week or month:** 1. System displays message: "No assignments for this period. Please allocate jobs." 2. System redirects manager to the Job Allocation page (UC_08). **2b. Teacher workload data is incomplete or missing:** 1. System displays an error message and prompts the manager to refresh or contact support. **3a. More than three teacher share the same lowest workload (month-to-date):** 1. System highlights all affected teachers.

	5a. Data retrieval/sync error: 1. System shows an error and offers retry or contact support.
Postconditions:	1. Manager has actionable information to support fair and efficient job allocation.
Priority:	High - ensures fair workload distribution and prevents teacher overwork.

(Fig.1i UC8)

ID:	UC_09
Name:	Generate monthly roster
Description:	This use case allows a manager to generate a consolidated monthly roster from the Job Allocation Dashboard with a button press. The system compiles weekly job allocations, teacher availability, and studio usage into a single monthly view. The roster is typically generated on the 20th of the month, once all teachers have updated their availability. After generation, any changes made to allocations must trigger a notification to the manager to ensure accuracy and control.
Primary Actor:	Manager
Preconditions:	1. Manager is logged into the system. 2. Teachers have updated availability by the 19th. 3. Weekly allocations exist in the system for the planning period.
Main Success Scenario:	1. Manager navigates to the Job Allocation Dashboard. 2. Manager selects “Generate Monthly Roster.” 3. System compiles all weekly allocations, teacher availability, and studio usage for the month. 4. System generates a consolidated monthly roster. 5. System displays the roster in a calendar/grid format, grouped by studio and teacher. 6. Manager can review, export, or share the roster.
Alternative Scenarios:	**3a. Availability not updated by all teachers:** 1. System warns manager that availability is incomplete but allows generation. **3b. Weekly allocations incomplete:** 1. System highlights missing assignments and prompts manager to finalize them first. **5a. Changes made after roster generation:** 1. System logs the change and notifies the manager to review and re-approve.
Postconditions:	1. A monthly roster is available for review, export, or sharing. 2. Any modifications after generation are logged and reported to the manager.
Priority:	High - the monthly roster ensures teachers, managers, and studios are coordinated for the full planning cycle

(Fig.1j UC9)

ID:	UC_10
Name:	Create accounts
Description:	This use case allows an IT Administrator to create new user accounts through the User Management module of the system. The administrator can create Teacher and Manager accounts by entering required details (e.g., name, email, role, specialization). The system validates the input, generates a unique ID and temporary password, stores the account, and delivers the credentials securely to the user's registered email. To maintain security, creation of new IT Administrator accounts requires Super-Admin approval. On first login, new users must change the temporary password to a secure password of their choice. An audit log is maintained to ensure accountability.
Primary Actor:	IT Administrator
Preconditions:	1. IT Administrator is logged into the system. 2. IT Administrator has account management permissions. 3. A valid account creation request has been approved (e.g., new hire, internal role change).
Main Success Scenario:	1. IT Administrator navigates to the User Management page. 2. IT Administrator selects "Create New Account". 3. System displays the account creation form. 4. IT Administrator chooses account type (Teacher / Manager). 5. IT Administrator enters required profile details (e.g., name, email, role, instrument specialization if Teacher). 6. System validates entered details (format, required fields, uniqueness of email/ID). 7. System generates a unique user ID and a temporary password. 8. System saves the new account in the system. 9. System records the account creation event in the audit log (who, when, what). 10. System sends login credentials (user ID and temporary password) securely to the new user's registered email. 11. On first login, the system forces the user to change the temporary password to a secure password before granting full access. 12. System confirms to the IT Administrator that the account was successfully created.
Alternative Scenarios:	**3a. IT Administrator attempts to create a new IT Administrator account:** 1. System blocks the action and requires Super-Admin approval.

	4a. Required fields are missing or invalid: 1. System highlights the errors and prevents account creation until corrected. **5a. Account creation fails due to technical/system error:** 1. System displays an error message, logs the issue, and prompts retry later.
Postconditions:	1. A new user account is stored in the system. 2. The new user receives login credentials by email. 3. On first login, the user must set a secure password before gaining full access. 4. Audit logs record account creation and credential issuance.
Priority:	High - fundamental administrative function, essential for onboarding new Teachers/Managers while maintaining security.

(Fig.1k UC10)

ID:	UC_11
Name:	Assign roles
Description:	This use case allows an IT Administrator to assign or change a user's role within the system (Teacher, Manager, or IT Administrator). The administrator performs this through the User Management module, by opening the target account and selecting "Edit Roles." Role changes determine the user's access rights and available functions. For security, only a higher-level Administrator (Super-Admin) can assign or modify IT Administrator roles.
Primary Actor:	IT Administrator
Preconditions:	1. IT Administrator is logged into the system. 2. IT Administrator has account management permissions. 3. Target account already exists in the system. 4. If assigning an IT Administrator role, Super-Admin approval is required.
Main Success Scenario:	1. IT Administrator selects "Manage Accounts." 2. System displays a list of user accounts. 3. IT Administrator searches for and selects the target account. 4. IT Administrator selects "Assign/Change Role." 5. IT Administrator chooses the new role (Teacher, Manager, or IT Administrator). 6. System validates the role change: a. If role is IT Administrator, requires Super-Admin confirmation. 7. System updates the account with the new role. 8. System logs the modification (who changed what, and when).
Alternative Scenarios:	**5a. IT Administrator attempts to assign/change IT Administrator role without Super-Admin approval:** 1. System blocks the action and requests Super-Admin authorization. **6a. Role assignment request contains invalid data (e.g., role not recognized):** 1. System rejects the change and prompts correction. **7a. System fails to update due to technical error:** 1. System displays error, logs the failure, and prompts retry
Postconditions:	1. User account role is updated and reflected across the system. 2. Audit log records the change for accountability.

Priority:	High - critical for enforcing security and role-based access control.

(Fig.1l UC11)

ID:	UC_12
Name:	Edit account profile
Description:	This use case allows an IT Administrator to update a user's account profile details through the User Management module. The administrator searches for the target account, opens the profile editor, and modifies fields such as name, email, instrument specialization (for teachers), or workload settings. The system validates and saves the changes, ensuring records remain accurate and up to date for scheduling and communication.
Primary Actor:	IT Administrator
Preconditions:	1. IT Administrator is logged into the system. 2. IT Administrator has account management permissions. 3. Target account already exists in the system.
Main Success Scenario:	1. IT Administrator selects "Manage Accounts." 2. System displays a list of user accounts. 3. IT Administrator searches for and selects the target account. 4. IT Administrator chooses "Edit Profile." 5. IT Administrator updates relevant fields (e.g., name, email, instrument specialization, workload settings). 6. System validates the new data. 7. System saves the updates and refreshes the account record. 8. System logs the modification (who changed what, and when).
Alternative Scenarios:	**5a. Required fields are left blank or contain invalid data:** 1. System highlights the errors and prevents saving until corrected. **7a. System fails to save changes due to technical error:** 1. System displays error, logs the failure, and prompts retry.
Postconditions:	1. Account profile data is updated and available across the system. 2. Audit log records the modification for traceability.
Priority:	High - ensures accuracy of user data and reliable scheduling.

(Fig.1m UC12)

ID:	UC_13
Name:	Deactivate accounts
Description:	This use case allows an IT Administrator to deactivate a user account immediately when an employee (Teacher, Manager, or Administrator) leaves the organization. The IT Administrator performs this action through the User Management module by searching for the departing user's account and selecting Deactivate. The system then disables login, removes the user from scheduling and allocation workflows (if applicable), and records the action in the audit log. This ensures unauthorized access is prevented and organizational security and compliance policies are upheld.
Primary Actor:	IT Administrator
Preconditions:	1. IT Administrator is logged into the system. 2. IT Administrator has account management permissions. 3. The resignation or termination of the user has been confirmed by HR. 4. The user's account exists in the system. 5. A higher-level Super-Admin account exists
Main Success Scenario:	1. IT Administrator receives HR confirmation that a user (Teacher, Manager, or IT Admin) has left the organization. 2. IT Administrator logs into the system and navigates to the User Management module. 3. IT Administrator searches for the departing user's account. 4. IT Administrator selects the account and chooses Deactivate. 5. System immediately disables login for the account. 6. System updates scheduling/permissions so the user no longer appears in active workflows (e.g., teaching schedule, job allocation, or administrative actions). 7. System records the deactivation event in the audit log (who, when, reason). 8. System displays confirmation of successful deactivation to the IT Administrator.
Alternative Scenarios:	**5a. Deactivation is delayed:** 1. System raises an alert. System logs the delay for audit/compliance reporting. **5b. HR requests full deletion instead of deactivation:** 1. IT Administrator exports relevant historical data. 2. IT Administrator deletes the account. 3. System records the deletion event in the audit log. **6a. System fails to deactivate account:** 1. System alerts both the initiating IT Administrator and a

	higher-level Administrator (Super-Admin). 2. System records the failure in the audit log. 3. Super-Admin investigates and forcibly deactivates the account. **6b. User to be terminated is an IT Administrator:** 1. Higher-level Administrator (Super-Admin) first revokes the IT Admin's elevated privileges. 2. Once privileges are revoked, the Super-Admin deactivates or deletes the account. 3. System records the privilege removal and account termination in the audit log for compliance.
Postconditions:	1. The user's account is disabled (or deleted if requested by HR) and cannot be used to access the system. 2. For IT Administrators: elevated privileges are revoked before deactivation. 3. Scheduling, allocation, or administrative permissions tied to the account are fully removed. 4. All actions (privilege removal, deactivation/deletion, reason, and who performed them) are permanently recorded in the audit log for compliance and traceability.
Priority:	High - critical for system security, data integrity, and compliance with organizational policies

(Fig.1n UC13)

ID:	UC_14
Name:	Restore or Reset User Account (locked/forgotten credentials)
Description:	This use case allows an IT Administrator to restore or reset a user account that has been locked (e.g., after failed login attempts) or when the user has forgotten their credentials. The IT Administrator performs this action in the User Management module of the system. After verifying the user's identity, the IT Administrator resets the account and issues a temporary password. When the user next logs in, the system forces them to set a new secure password, ensuring continued account security.
Primary Actor:	IT Administrator
Preconditions:	1. IT Administrator is logged into the system. 2. Target account exists in the system. 3. User identity has been verified (e.g., via HR or internal process).
Main Success Scenario:	1. IT Administrator navigates to the *User Management module*. 2. IT Administrator searches for the locked/forgotten account. 3. IT Administrator selects the "Restore/Reset Account" option. 4. System unlocks the account (if locked). 5. System generates a temporary password. 6. System sends the temporary password securely to the user's registered email. 7. User logs in with the temporary password. 8. System forces the user to create a new secure password before accessing the dashboard. 9. System updates the account and records the reset in the audit log.
Alternative Scenarios:	**3a. Resetting an IT Administrator account:** 1. System blocks the action for security. 2. Super-Admin must: a. Revoke the locked IT Administrator's elevated privileges. b. Perform the reset or deletion. c. System records both privilege removal and account reset in the audit log. **4a. Email delivery fails:** 1. System alerts IT Administrator. 2. IT Administrator provides temporary credentials through an alternate secure channel (e.g., in-person or phone verification). **5a. System error during reset:**

	1. System shows error and logs the failure. 2. IT Administrator retries or escalates to Super-Admin if issue persists.
Postconditions:	1. User account is unlocked/reset. 2. User can log in only after setting a new secure password. 3. Audit log contains who performed the reset, when, and why.
Priority:	High - ensures continuity of user access while enforcing strict security controls, especially for sensitive roles like IT Administrators.

(Fig.1o UC14)

ID:	UC_15
Name:	View account activity logs
Description:	This use case allows an IT Administrator to view logs of user account activity through the Audit/Reporting Dashboard in the User Management module. The IT Administrator applies filters (by user, role, or time period) and the system retrieves and displays events such as login attempts, profile edits, role changes, and account status updates. This is done whenever administrators need to check compliance, investigate incidents, or troubleshoot user issues.
Primary Actor:	IT Administrator
Preconditions:	1. IT Administrator is logged into the system. 2. Audit logging is enabled and data exists in the system.
Main Success Scenario:	1. IT Administrator navigates to the *Audit/Reporting Dashboard*. 2. IT Administrator selects “View Activity Logs.” 3. IT Administrator applies filters (e.g., by user, role, date range). 4. System retrieves and validates relevant log data. System displays log entries, including: a. Login attempts (successful/failed) b. Role changes c. Profile updates Account status changes (activation/deactivation) 5. IT Administrator reviews logs for compliance or troubleshooting purposes.
Alternative Scenarios:	**3a. No logs found for filter criteria:** 1. System shows “No results found.” **4a. Log data incomplete or corrupted:** 1. System notifies IT Administrator and suggests escalation. **5a. Retrieval error:** 1. System displays error message, logs failure, and suggests retry.
Postconditions:	1. Logs are successfully displayed for the IT Administrator. 2. All access to logs is itself recorded in the audit trail.
Priority:	High - viewing logs is foundational for compliance, investigations, and troubleshooting.

(Fig.1p UC15)

ID:	UC_16
Name:	Export account activity reports
Description:	This use case allows an IT Administrator to export account activity logs from the Audit/Reporting Dashboard into external formats such as CSV or PDF. The export process is triggered when reports are required for audits, regulatory submissions, or archival purposes. The system generates the report using the selected filters and logs the export event for accountability.
Primary Actor:	IT Administrator
Preconditions:	1. IT Administrator is logged into the system. 2. Audit log data exists and is available for export.
Main Success Scenario:	3. IT Administrator is logged into the system. 4. Audit logging is enabled and data exists in the system.
Alternative Scenarios:	1. IT Administrator navigates to the *Audit/Reporting Dashboard*. 2. IT Administrator selects “Export Activity Report.” 3. IT Administrator applies filters (date range, user, role, activity type). 4. IT Administrator selects export format (CSV, PDF). 5. System generates the report and prepares the file. 6. System securely delivers the report for download or email. 7. System logs the export event (who, when, what scope).
Postconditions:	**4a. Export fails due to format issue:** 1. System suggests alternate format. **5a. Export interrupted (server/network error):** 1. System logs error and retries. **5b. Report too large:** 1. System splits the report into multiple files.
Priority:	1. Exported report is successfully received by the IT Administrator. 2. Export event is logged for accountability.
	Medium - essential for audits and compliance, but secondary to viewing and monitoring logs.

(Fig.1q UC16)

(Fig. 1t UC17)

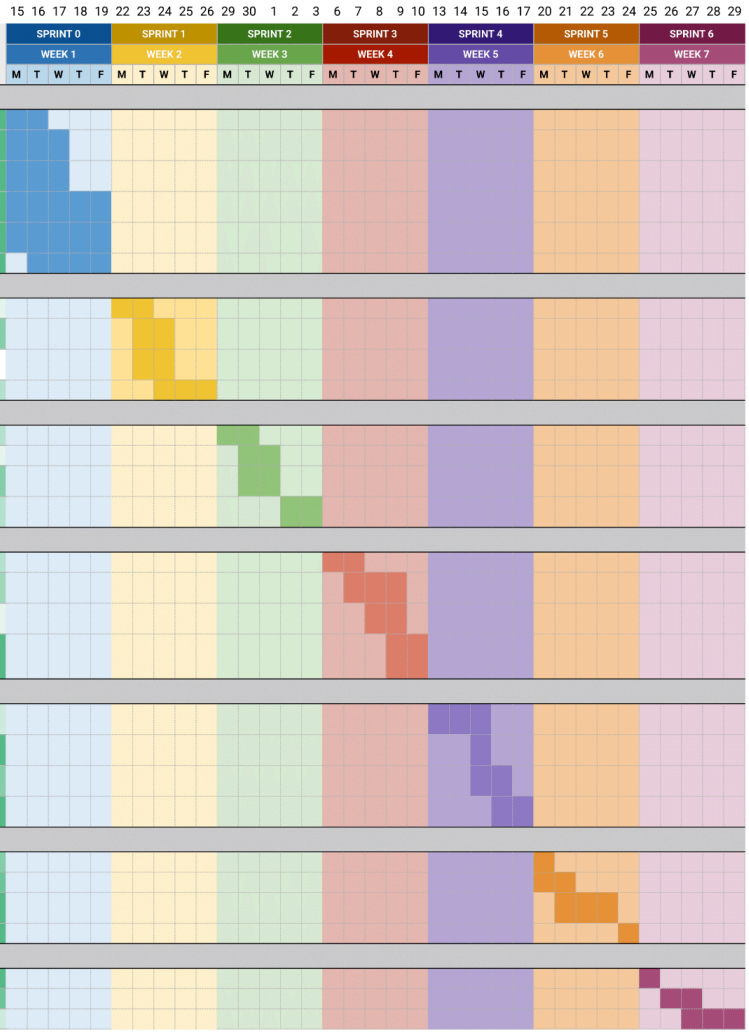
ID:	UC_17
Name:	Monitor suspicious account activity
Description:	This use case allows an IT Administrator to monitor suspicious account activity via the Monitoring Dashboard in the User Management module. The system continuously tracks logs in the background, applies security rules (e.g., multiple failed login attempts, unusual login times, or access from unrecognized IP addresses), and automatically raises alerts in real time. The IT Administrator reviews flagged events and takes corrective action when necessary. This use case is performed continuously, with alerts sent as soon as anomalies are detected, ensuring proactive system security.
Primary Actor:	IT Administrator
Preconditions:	1. IT Administrator is logged into the system. 2. Monitoring rules are configured (failed login thresholds, geo/IP restrictions, etc.).
Main Success Scenario:	1. System continuously monitors account activity. 2. System applies detection rules to identify anomalies. System flags suspicious activity and raises an alert. IT Administrator receives alert via dashboard (and optionally email/SMS). 3. IT Administrator reviews flagged activity in detail. 4. IT Administrator takes corrective action (e.g., reset account, lock user, escalate to Super-Admin).
Alternative Scenarios:	**3a. False positive detected:** 1. IT Administrator marks as safe 2. System records override. **4a. Alert delivery fails:** 1. System retries and logs failure. **5a. IT Administrator takes no action:** 1. Alert remains active until cleared, logged for audit.
Postconditions:	1. Suspicious activities are flagged and visible to IT Administrators. 2. Alerts and admin responses are recorded in the audit log.
Priority:	High - critical for proactive security monitoring and preventing unauthorized access.

(Fig.2 Gantt Chart High Res)

GANTT CHART

PROJECT TITLE	INF2001: Introduction to Software Engineering	**COMPANY NAME**	Singapre Institute of Technology
PROJECT MANAGER	Genisa Lee	**DATE**	9/15/25

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Setup & Backlog**					
1.1	Identify Stakeholders	Genisa	9/15/25	9/16/25	1	100%
1.2	Draft product Vision and Scope	Lavanya Sebastian	9/15/25	9/17/25	2	100%
1.3	Gather requirements and business rules from interviews	Genisa	9/15/25	9/17/25	2	100%
1.4	Create the **Product Backlog**	Genisa (Lead), Team	9/15/25	9/19/25	4	100%
1.5	Set up team roles (Scrum Master, Product Owner, Developers)	Jin Rui	9/15/25	9/19/25	4	100%
1.6	Set up project tools (GitHub)	Lavanya	9/16/25	9/19/25	3	100%
2	**Teacher Module MVP**					
2.1	Build Teacher login	Genisa	9/22/25	9/23/25	1	70%
2.2	Manage availability (5-Week view)	Genisa (Lead), Jin Rui	9/23/25	9/24/25	1	90%
2.3	Set Job preferences	Jin Rui (Lead), Zulfaqar	9/23/25	9/24/25	1	65%
2.4	View Personal workload	Jin Rui	9/24/25	9/26/25	2	80%
3	**Manager Module MVP**					
3.1	View manpower overview	Max	9/29/25	9/30/25	1	80%
3.2	Compare up to 3 teachers	Jin Rui	9/30/25	10/1/25	1	70%
3.3	Allocate weekly jobs	Zulfaqar Sebestian	9/30/25	10/1/25	1	90%
3.4	Highlight overworked/underworked staff	Zulfaqar	10/2/25	10/3/25	1	80%
4	**IT Admin & System Rules**					
4.1	Account creation/deactivation	Genisa	10/6/25	10/7/25	1	80%
4.2	Role-based access control	Genisa Max	10/7/25	10/9/25	2	85%
4.3	Audit log and activity tracking	Max (Lead), Team	10/8/25	10/9/25	1	70%
4.4	System enforces all business rules (studio limits, hours, breaks, holidays)	Max	10/9/25	10/10/25	1	100%
5	**Integration & Testing**					
5.1	Integrate Teacher + Manager + Admin flows	Jin Rui	10/13/25	10/15/25	2	75%
5.2	Conduct system testing (unit, integration)	Zulfaqar	10/15/25	10/15/25	0	100%
5.3	Fix bugs & improve performance	Max (Lead), Team	10/15/25	10/16/25	1	90%
5.4	Conduct Sprint Review and Retrospective	Max (Lead), Team	10/15/25	10/17/25	2	100%
6	**Finalisation & Presentation**					
6.1	User acceptance testing	Jin Rui	10/20/25	10/20/25	0	90%
6.2	Polish UI/UX	Zulfaqar	10/20/25	10/21/25	1	95%
6.3	Prepare project documentation & slides	Max (Lead), Team	10/21/25	10/23/25	2	100%
6.4	Final presentation and demo	Lavanya	10/24/25	10/24/25	0	100%
7	**Testing**					
7.1	Deploy to server	Lavanya	10/25/25	10/25/25	0	100%
7.2	Conduct user training	Sebastian	10/26/25	10/27/25	1	95%
7.3	Maintenance and feedback iteration	Genisa	10/27/25	10/29/25	2	75%



(Fig.3 WBS Chart High Res)