
Software Requirements Specification (SRS)

for

COMSYS

University Communication and Services Portal

Section: TT4L

Group: 4

NAME	STUDENT ID
HESHAM NADER DEYAAEEDEN EISA	1221101049
NICKLEIRSCH JAYA RAJ	1231303114
DANESH VERAN A/L BALASUBRAMANIAM	1211109158
LIM XIN YEE	1211109469

One drive link containing proof of execution: [SRE Items](#)

Table of Contents

1	Introduction	3
1.1	Purpose.....	3
1.2	Scope.....	3
1.3	Product Overview.....	4
1.3.1	Product Perspective.....	4
1.3.2	Memory Constraints.....	6
1.3.3	Product Functions	7
1.3.4	User Characteristics.....	8
1.3.5	Limitations.....	9
1.4	Definitions	10
2	References.....	12
3	Requirements.....	13
3.1	Functions	13
3.1.1	F001 Login.....	17
3.1.2	F002 Logout.....	19
3.1.3	F003 Change Language Preference.....	21
3.1.4	F004 Customize Interface.....	23
3.1.5	F005 Customize Session Time-Out	25
3.1.6	F006 View Notification	27
3.1.7	F007 View Tooltip.....	29
3.1.8	F008 Access Help Documentation	31
3.1.9	F009 Send Notification.....	33
3.1.10	F010 Access Calendar.....	36
3.1.11	F011 Use Live Chat	38
3.1.12	F012 View Attendance Record	40
3.1.13	F013 View Academic Record	42
3.1.14	F014 View Class Schedule	44
3.1.15	F015 View Exam Timetable	46
3.1.16	F016 View Billing Information	48
3.1.17	F017 Enrol in Course.....	50
3.1.18	F018 Search Past Announcement	52
3.1.19	F019 Customize Notification Preference.....	54

3.1.20	F020 Set 'Quiet Hours'	56
3.1.21	F021 View Child's Information	58
3.1.22	F022 View University Contact Directory.....	61
3.1.23	F023 Schedule Meeting with University Staff.....	63
3.1.24	F024 Manage Academic Resource	65
3.1.25	F025 View Announcement Read Status	67
3.1.26	F026 Update Student Academic Data.....	69
3.1.27	F027 Manage Communication Template	71
3.1.28	F028 Manage University Contact Directory	73
3.1.29	F029 View System Audit Log	75
3.1.30	F030 Configure Parent Access	77
3.1.31	F031 Authenticate User	79
3.1.32	F032 Send SMS Notification.....	81
3.1.33	F033 Sync with External Calendar.....	83
3.2	Performance Requirements	85
3.3	Usability Requirements	86
3.4	Interface Requirements.....	87
3.4.1	System Interfaces	88
3.4.2	User interfaces.....	89
3.4.3	Software interfaces	91
3.4.4	Communication interfaces	93
3.5	Logical Database Requirements	94
3.6	Design Constraints.....	97
3.7	Software System Attributes	98
3.8	Supporting Information.....	101
4	Verification	104
4.1	Verification Approach	104
4.2	Verification Criteria	106
5	Appendices	107
5.1	Assumptions and Dependencies.....	107
5.2	Acronyms and Abbreviations.....	109
5.3	Glossary.....	110

1 Introduction

1.1 Purpose

The purpose of the University Communication and Services Portal, COMSYS is to provide a centralized, user-friendly platform that facilitates transparent and timely communication between students, lecturers, administrators, and parents. The portal addresses current gaps in academic and administrative information access by integrating with the university's Campus Management System for real-time retrieval of essential student data such as academic performance, attendance records, and billing information. Additionally, the system aims to enhance the effectiveness of critical communications through seamless integration with an SMS Gateway, ensuring urgent updates and important notifications are promptly delivered to students and parents. Ultimately, the portal is designed to improve engagement, streamline access to university services, and foster a more connected campus community.

1.2 Scope

COMSYS is designed to address the fragmentation of current academic and communication platforms used within the university. The system will consolidate grade management, scheduling, billing, announcements, notifications, parental access, and other academic services into a single, secure, and user-friendly portal. The portal will serve as the primary interface for students, parents, lecturers, and administrators to interact with university information and each other, offering customizable user experiences and integrating with existing university systems (e.g., calendars, SMS gateways, Single Sign-On). COMSYS will facilitate timely and relevant communication, automate routine notifications, and provide secure role-based access to information and services for its diverse set of users.

1.3 Product Overview

1.3.1 Product Perspective

COMSYS operates as a core integration point within the university's digital ecosystem. It connects and coordinates the flow of information between students, parents, lecturers, admins, and several external systems. Rather than being a stand-alone product, COMSYS is a crucial element within a larger ecosystem of institutional services.

Related Entities and Their Interactions

1. **Campus Management System (CMS):**

COMSYS exchanges academic information, course updates, grades, user details, and billing data with the CMS. This ensures that students, lecturers, and admins have access to updated and synchronized information.

2. **SMS Gateway:**

COMSYS interfaces with an SMS gateway to send real-time notifications and alerts to users, enabling critical communication outside the portal.

3. **Calendar API:**

The portal pushes calendar information (such as timetables and events) to this API, so users can synchronize with their personal or institutional calendars.

The context diagram (Figure 1.3.1) outlines COMSYS at the centre of all information exchange:

1. Students can access enrolment info, academic records, schedules, billing, notifications, and chat services.
2. Parents receive student attendance, academic, and billing info, along with notifications and chat.
3. Lecturers interact via chat, update grades, access student profiles and timetables, and provide course materials.
4. Admins handle system settings, user management, audit logs, and notifications.
5. External Systems (CMS, SMS Gateway, Calendar API) enable COMSYS to distribute and synchronize institutional data efficiently.

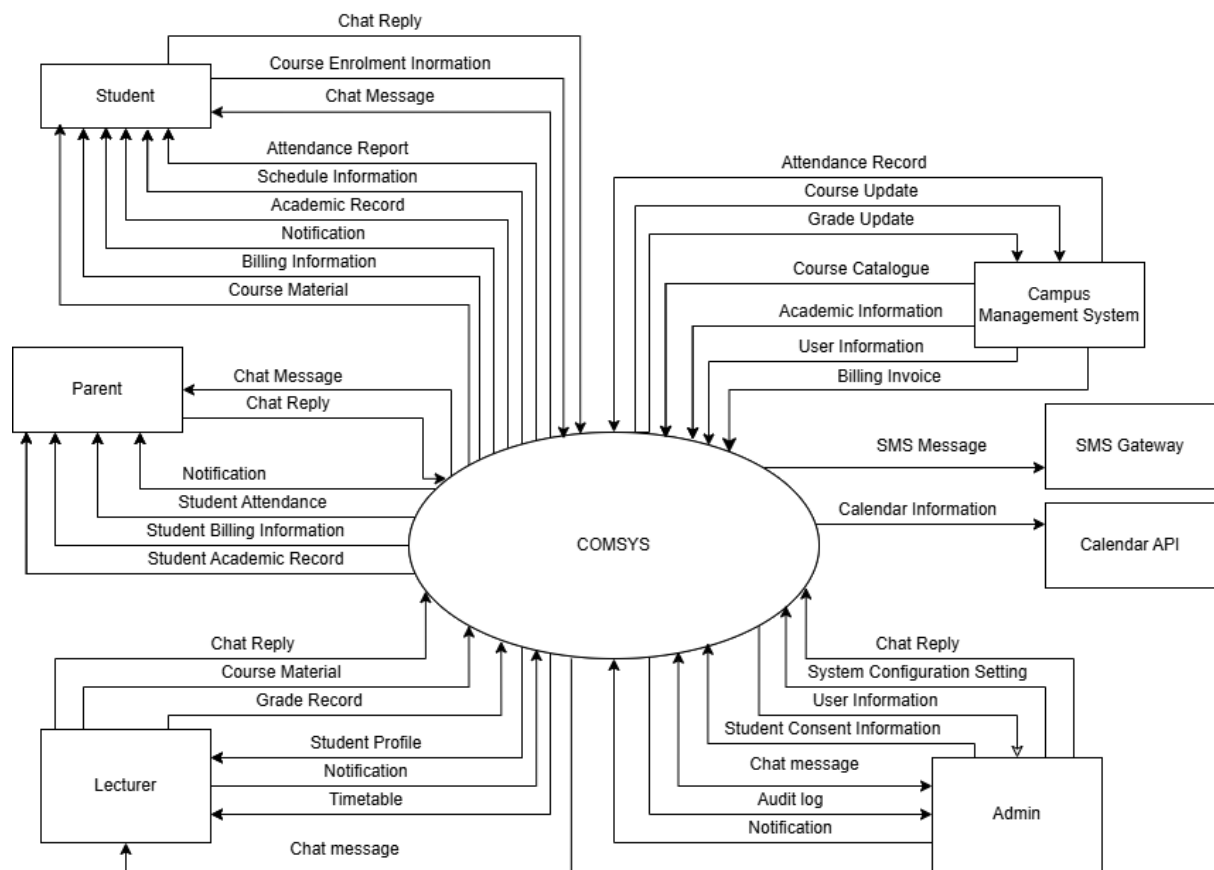


Figure 1.3.1 Context Diagram for COMSYS

1.3.2 Memory Constraints

This section outlines the specific memory requirements and constraints for the university communication and service portal (COMSYS) to ensure stable and reliable operation.

Memory Type	Constraint	Constraint Detail
Primary Memory (RAM)	Minimum Required RAM	8 GB minimum required to support essential services including user authentication, academic data retrieval, and messaging.
	Optimal RAM	16 GB recommended for optimal performance during peak loads, such as semester registration or mass notification events.
	Maximum Memory Usage	System processes should not exceed 70% of available RAM under normal operation to prevent slowdowns and ensure responsiveness.
Secondary Memory (Storage)	Minimum Disk Space	200 GB required to store user profiles, academic performance data, and system configurations.
	Data archiving	Student and staff records older than five years should be archived automatically to maintain at least 40 GB of free disk space at all times.
	Backup and Recovery	At least 20% of total disk space must be reserved for periodic backup snapshots and recovery procedures in case of system failure or data corruption.

By adhering to these memory constraints, COMSYS will remain robust, responsive, and capable of supporting the university's critical communication and service functions.

1.3.3 Product Functions

The University Portal System (COMSYS) provides the following major functions:

Feature Category	Major Functions
Student Information Management	Centralized dashboard for academic records, schedules, and financial information
	Course enrolment
	Grade and attendance tracking
	Financial status monitoring
Multi-Channel Communication System	Integrated notification delivery across email, SMS, and portal
	Customizable notification preferences and quiet hours
	Real-time chat functionality
	Announcement management with read receipt tracking
	Searchable communication history
Parent Engagement Platform	Dedicated parent portal with controlled access to student information
	Compliance-based information sharing
	Communication channels with university administrators
Academic Resource Distribution	Centralized learning material repository
	Unified link generation for academic resources
	Data import/export capabilities
External Calendar System Integration	Synchronization with external calendar services
Multi-language support	Support for multiple languages across the portal interface

Table 1.3.3 Major Functions

1.3.4 User Characteristics

Students are expected to have basic computer literacy and familiarity with common web applications and mobile interfaces. No specialized technical knowledge is required beyond the ability to navigate websites and use basic mobile applications.

Parents may have varying levels of technical proficiency, with only basic computer literacy required. They are expected to access the system less frequently (weekly or monthly) and may prefer simplified interfaces. Some parents may have language preferences other than English.

Lecturers are expected to possess moderate technical proficiency and familiarity with basic educational technology tools. While they should be comfortable with routine computer operations, extensive technical expertise is not required. They will need to manage course materials and student communications regularly, accessing the system daily during academic periods. lecturers are expected to be undergo training to use the new system.

Administrative staff are expected to have moderate to advanced computer skills and will be trained in using administrative functions. They will be regular users requiring proficiency in managing student records, processing requests, and handling bulk operations. They should be comfortable with complex system features and multi-step processes.

1.3.5 Limitations

The University Portal System (COMSYS) is subject to the following limitations and constraints:

Regulatory Requirements and Policies

1. Must comply with GDPR and FERPA data protection regulations, which may restrict access, storage, and sharing of personal and academic data.
2. Legal and privacy regulations may limit the extent of data visibility and communication features, particularly for parental access to student information.

Technical Limitations

1. Real-time data synchronization depends on network reliability and the availability of integrated external services (e.g., SMS gateways, calendar APIs, campus management systems).
2. System performance and user experience may be degraded during peak periods or if hardware resource constraints (RAM, storage, etc.) are not met.
3. The platform will require regular routine maintenance and timely updates to address security vulnerabilities, technological advancements, and feature enhancements.
4. Feature rollout—including new modules and multilingual support—may be phased and prioritized based on stakeholder feedback, funding, and resource availability.
5. Integration with legacy or third-party institutional systems may be limited by incompatible data formats, outdated APIs, or insufficient documentation.

Operational Limitations

1. User support and system administration resources may be limited, potentially leading to delays in resolving technical issues or implementing requested enhancements.
2. System scalability and performance may be constrained by the underlying infrastructure, especially if user growth exceeds projected estimates.
3. User training and onboarding resources may be limited, affecting the adoption rate and effective use of the portal.

These limitations should be considered during the planning, implementation, and operation of COMSYS to ensure realistic expectations and ongoing compliance with institutional and regulatory requirements.

1.4 Definitions

1. **Academic Record:** A collection of data representing a student's academic performance, including grades, attendance, enrolment status, and completed courses.
2. **Administrator (Admin):** A university staff member with privileges to manage users, system settings, data, and oversee operations within COMSYS.
3. **Campus Management System (CMS):** The institution's core administrative information system responsible for managing student, course, and billing data.
4. **Consent Management:** A process or feature ensuring that access to sensitive data (e.g., parent access to student records) is granted only with explicit user authorization, in compliance with privacy regulations. This consent can only be changed by having the student physically mail in a letter.
5. **Critical Notification:** A message flagged as essential or urgent (e.g., exam changes, fee deadlines) requiring prompt delivery and user attention.
6. **Dashboard:** The main user interface screen that aggregates and presents key information and actions relevant to the user's role.
7. **Data Synchronization:** The process of ensuring that information is current and consistent across all integrated systems and interfaces.
8. **End User:** Any individual who interacts with COMSYS, including students, parents, lecturers, and administrators.
9. **Notification:** Any automated or manual message sent to users through email, SMS, or portal channels to convey updates, alerts, or reminders.
10. **Parental Access:** Controlled access granted to parents or guardians for viewing their child's academic and billing information, subject to consent and privacy policies.
11. **Portal:** The web-based entry point to COMSYS, providing access to academic, administrative, and communication services.
12. **Role-Based Access Control (RBAC):** A security mechanism restricting system access based on the user's assigned role within the institution.
13. **Single Sign-On (SSO):** An authentication method allowing users to access COMSYS and related university systems with a single set of credentials.
14. **User:** Any individual authorized to interact with COMSYS, including students, parents, lecturers, and administrators.
15. **User Interface (UI):** The set of screens, forms, navigation, and controls through which users interact with COMSYS.
16. **Real Time:** System action occurs and is reflected to the user within 5 seconds of the triggering event, unless otherwise specified for specific features.
17. **Urgent:** Requires user attention or action within 1 hour to avoid negative consequences; see also "Critical."
18. **Consistently:** The required action or state must occur in at least 99% of cases, measured monthly, with no unexplained exceptions.
19. **Proper Termination (Logout):** All session tokens (local and SSO), cookies, and active logins are invalidated, and the user is redirected to the login page.

20. **How is a system determined to be complex:** A system is considered complex if it contains features, workflows, or terminology that are not immediately intuitive to first-time users, require multiple steps to complete, or frequently result in user questions or errors.
21. **Communication channel:** The three communication channels are SMS, email and in-portal channels

2 References

International Organization for Standardization. (2018). ISO/IEC/IEEE 29148:2018: Systems and software engineering—Life cycle processes—Requirements engineering. <https://www.iso.org/standard/72089.html>

Pohl, K. (2010). Requirements engineering: Fundamentals, principles, and techniques (1st ed.). Springer. <http://www.requirements-book.com/>

PCI DSS Guide. (n.d.). *PCI DSS Session Timeout Requirements*. Retrieved May 25, 2025, from <https://pcidssguide.com/pci-dss-session-timeout-requirements/>

International Organization for Standardization. (2015). ISO 9000:2015 Quality management systems—
Fundamentals and vocabulary (4th ed.). <https://www.iso.org/standard/45481.html>

AltexSoft. (n.d.). *Software Requirements Specifications: Best Practices and SRS*. from <https://www.altexsoft.com/blog/software-requirements-specification/>

Internet Engineering Task Force (IETF). (2009). Internet calendaring and scheduling core object specification (iCalendar) (RFC 5545). <https://datatracker.ietf.org/doc/html/rfc5545>

3 Requirements

3.1 Functions

The following table (Table 3.1) contains the list of features to be implemented in COMSYS, separated by its accessible role.

Feature ID	Feature	Actor
F001	Login	Student, Parent, Lecturer, Admin
F002	Logout	Student, Parent, Lecturer, Admin
F003	Change Language Preference	Student, Parent
F004	Customize Interface	Student
F005	Customize Session Time-Out	Student
F006	View Notification	Student, Parent
F007	View Tooltip	Student, Parent
F008	Access Help Documentation	Student, Parent
F009	Send Notification	Lecturer, Admin
F010	Access Calendar	Student, Lecturer
F011	Use Live Chat	Student, Parent, Lecturer, Admin
F012	View Attendance Record	Student
F013	View Academic Record	Student
F014	View Class Schedule	Student
F015	View Exam Timetable	Student
F016	View Billing Information	Student
F017	Enrol in Course	Student
F018	Search Past Announcement	Student
F019	Customize Notification Preference	Student
F020	Set 'Quiet Hours'	Student
F021	View Child's Information	Parent
F022	View University Contact Directory	Parent
F023	Schedule Meeting with University	Parent
F024	Manage Academic Resource	Lecturer
F025	View Announcement Read Status	Lecturer
F026	Update Student Academic Data	Lecturer
F027	Manage Communication Template	Admin
F028	Manage University Contact Directory	Admin
F029	View System Audit Logs	Admin

F030	Configure Parent Access	Admin
F031	Authenticate User	University Portal
F032	Send SMS Notification	SMS Gateway
F033	Sync with External Calendar	Calendar API

Table 3.1 COMSYS Features

Figure 3.1 represents the use case diagram for COMSYS, followed by its overall requirements.

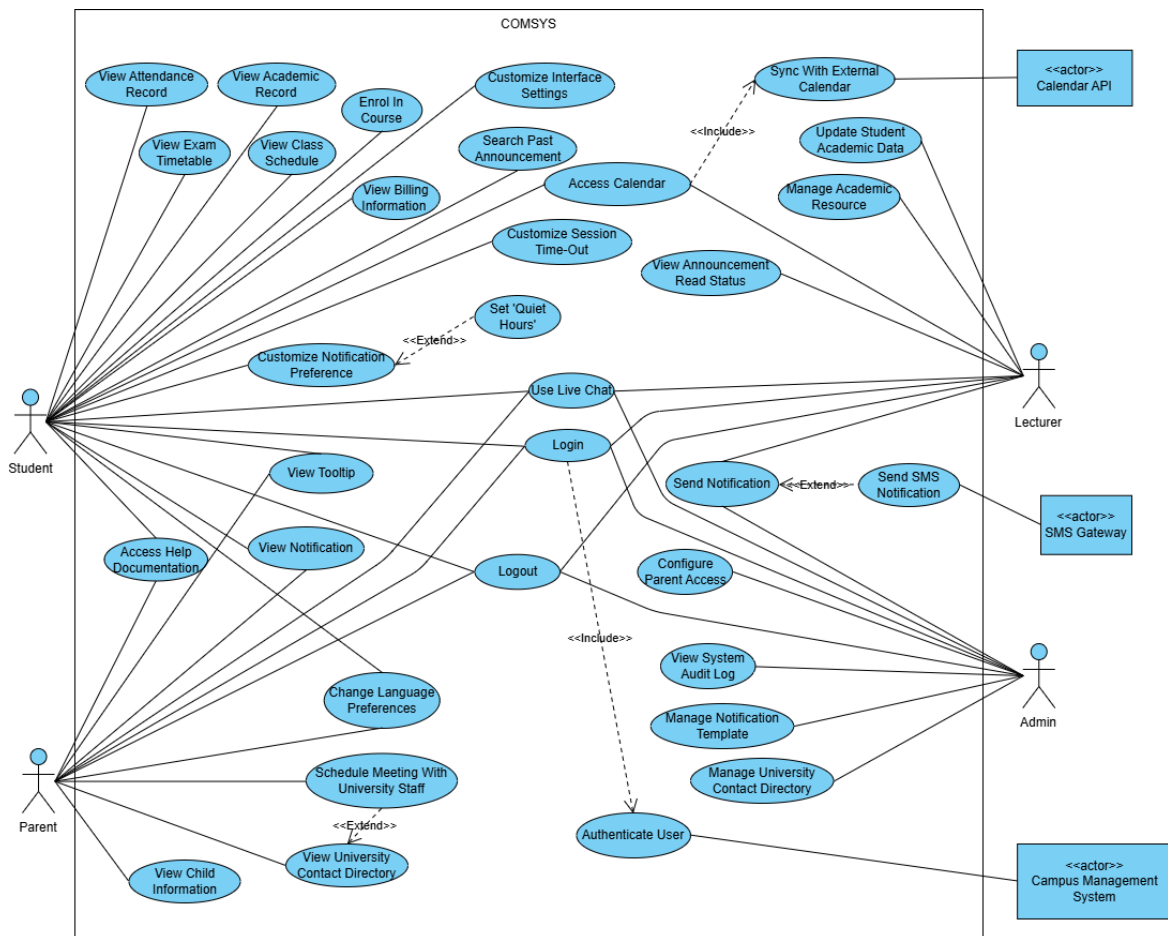


Figure 3.1: Use Case Diagram for COMSYS

Requirement Identifier Format:

Requirement IDs follow the format: **REQ_TXXYY**, where:

- **T**: Type (F = Functional, I = Interface, U = Usability, P = Performance)
- **XX**: Feature number (00 = Overall requirement)
- **YY**: Sequential requirement number within the feature

Example:

REQ_F0602 — Functional requirement, feature 06, second requirement.

The following are the overall requirements for COMSYS:

Requirement ID	REQ_F0001	Version	1.0
Description	The system shall update read status of announcements in real time.		
Author	Nickleirsch		

Requirement ID	REQ_F0002	Version	1.0
Description	The system shall implement features and controls necessary to comply with applicable data privacy and protection regulations, including but not limited to GDPR and FERPA.		
Author	Nickleirsch		

Requirement ID	REQ_F0003	Version	1.0
Description	The system shall implement encryptions for all sensitive data in transit and at rest.		
Author	Nickleirsch		

Requirement ID	REQ_F0004	Version	1.0
Description	The system shall only permit notifications to parents regarding attendance or emergencies in compliance with privacy policies.		
Author	Nickleirsch		

Requirement ID	REQ_F0005	Version	1.0
Description	The system should send automated email digests of attendance status on a bi-weekly basis.		
Author	Nickleirsch		

Requirement ID	REQ_F0006	Version	1.0
Description	The synchronization mechanism shall include academic records, financial transactions, attendance logs, and communication messages.		
Author	Nickleirsch		

Requirement ID	REQ_F0007	Version	1.0
Description	The system shall verify file type and size before accepting uploads and display a clear error if requirements are not met.		
Author	Nickleirsch		

Requirement ID	REQ_F0008	Version	1.0
Description	The system shall ensure that no user receives duplicate notifications for the same event across any communication channel.		
Author	Nickleirsch		

Requirement ID	REQ_F0009	Version	1.0
Description	Access to information and features shall be based on user roles (student, parent, lecturer, admin).		
Author	Nickleirsch		

Requirement ID	REQ_F0010	Version	1.0
Description	The system shall ensure that data changes are consistently reflected across all user-facing interfaces, including the web portal and mobile application.		
Author	Nickleirsch		

3.1.1 F001 Login

The functional requirement(s) for F001 Login:

Requirement ID	REQ_F0101	Version	1.0
Description	The system shall redirect users to role-specific dashboards after successful authentication		
Author	Hesham		

Table 3.1.1 below illustrates the use case for the login functionality (UC001), detailing the process as defined by Requirement REQ_F0101, followed by an activity diagram which represents the process flow.

Use Case ID	UC001	Version	1.0
Use Case	F001 Login		
Purpose	To authenticate users and provide secure access to the system based on their role		
Actor	Student, Parent, Lecturer, Admin		
Trigger	User attempts to access the system		
Precondition	User has an active account in the system		
Postcondition	1. User is successfully authenticated. 2. User is granted access based.		
Scenario Name	Step	Action	
Main Flow	1	User navigates to login page	
	2	User enters credentials	
	3	System validates credentials and role	
	4	System authenticates the user	
	5	System redirects user to their role-specific dashboard	
Alternate Flow – Wrong credentials are entered	3.1	User enters wrong credentials	
	3.2	System displays error message due to wrong credentials entered	
	3.3	System prompts user for re-entry	
Rules	1. System must implement single sign on [REQ_F3101] 2. System must comply with global privacy/security regulation [REQ_F0002] 3. System must validate student-parent consent to provide consented functionality [REQ_F3001]		
Notes	‘User’ in this use case refers to Student, Parent, Lecturer and Admin		
Author	Hesham		

Table 3.1.1: Use Case UC001 Login

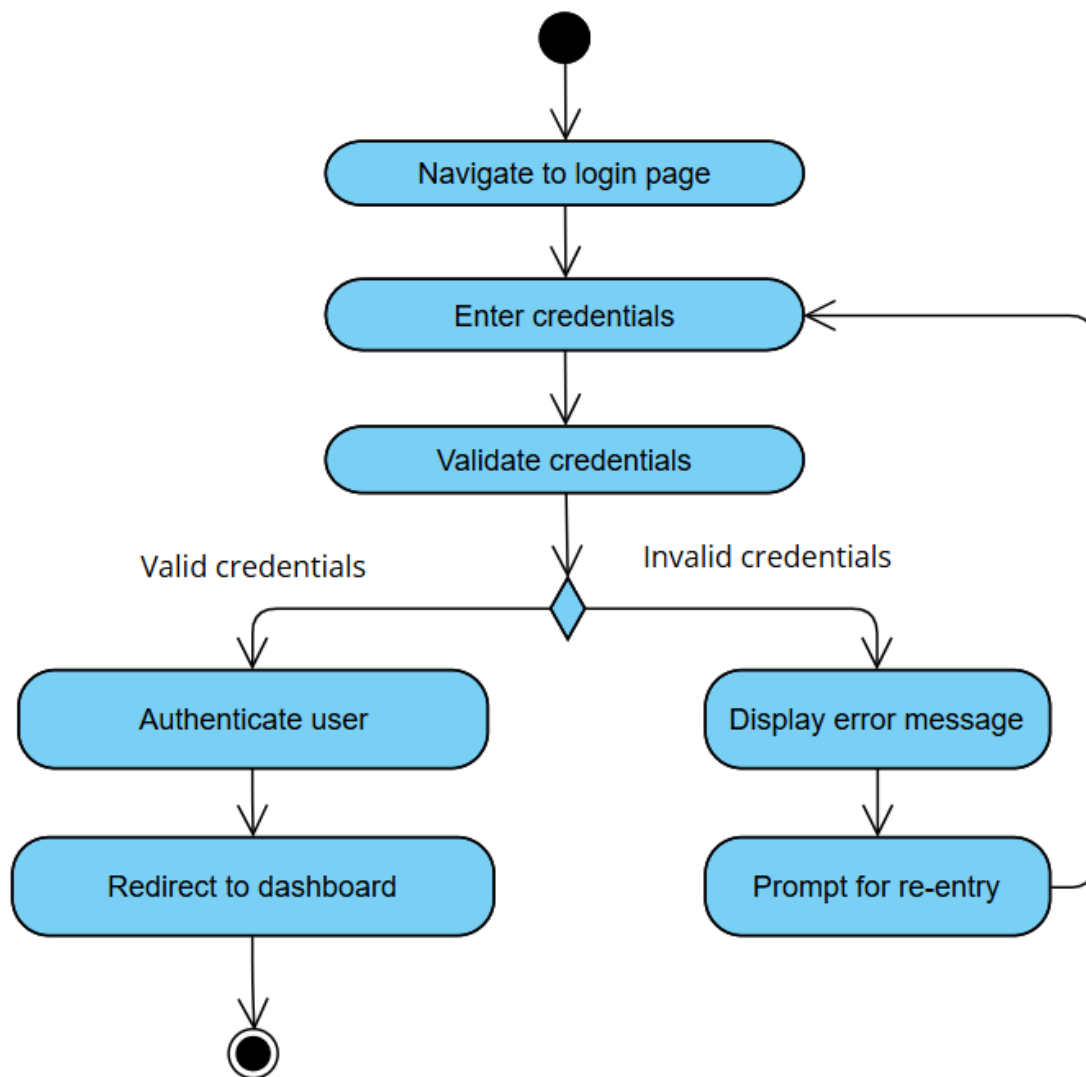


Figure 3.1.1: Activity Diagram for Use Case UC001 Login

3.1.2 F002 Logout

The functional requirement(s) for F002 Logout:

Requirement ID	REQ_F0201	Version	1.0
Description	The system shall provide proper termination of SSO sessions during logout		
Author	Hesham		

Table 3.1.2 illustrates the use case for the logout functionality (UC002), detailing the process as defined by Requirement REQ_F0201, followed by an activity diagram which represents the process flow.

Use Case ID	UC002	Version	1.0
Use Case	F002 Logout		
Purpose	To securely terminate user sessions		
Actor	Student, Parent, Lecturer, Admin		
Trigger	1. User initiates logout action 2. System detects session timeout		
Precondition	User is logged in		
Postcondition	1. User session is terminated 2. User is directed to login page 3. SSO session is terminated		
Scenario Name	Step	Action	
Main Flow	1	User clicks the logout button	
	2	System terminates active session	
	3	User gets redirected to the login page	
Alternate Flow - If Session Timeout Occurs	1.1.1	System shows a timeout warning	
	1.1.2	User can extend session within 30 seconds	
	1.1.3	If no response proceeds with main step 1	
	1.1.4	Else system extends session	
Rules	Session timeout duration must follow user setting (for student only) [REQ_F0501]		
Notes	‘User’ in this use case refers to Student, Parent, Lecturer and Admin		
Author	Hesham		

Table 3.1.2: Use Case UC002 Logout

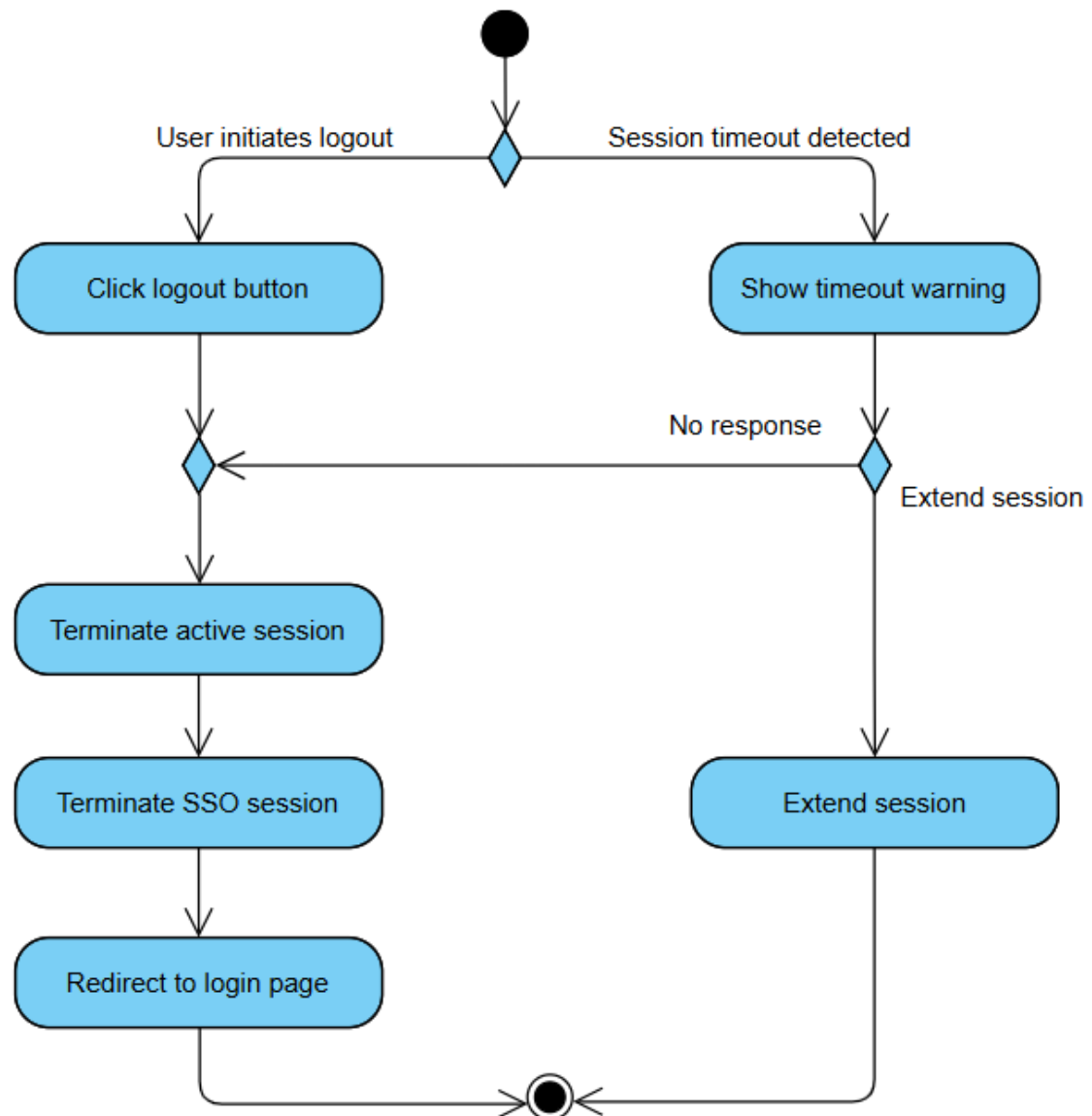


Figure 3.1.2: Activity Diagram for Use Case UC002 Logout

3.1.3 F003 Change Language Preference

The functional requirement(s) for F003 Change Language Preferences:

Requirement ID	REQ_F0301	Version	1.0
Description	The system shall support a multilingual interface for all student and parent facing pages and messages.		
Author	Hesham		

Requirement ID	REQ_F0302	Version	1.0
Description	The system shall allow students and parents to select and save their preferred interface language via their user profile settings.		
Author	Nickleirsch		

Table 3.1.3 illustrates the use case for the changing language preference functionality (UC003), detailing the process as defined by Requirement REQ_F0301 and REQ_F0302, followed by an activity diagram which represents the process flow.

Use Case ID	UC003	Version	1.0
Use Case	F003 Change Language Preference		
Purpose	To allow users to customize their interface language preference		
Actor	Student and Parent		
Trigger	User initiates language change from dashboard settings		
Precondition	User is logged in		
Postcondition	1. User’s language preference is updated 2. Interface language is changed		
Scenario Name	Step	Action	
Main Flow	1	User navigates to language section in settings	
	2	System displays available language options	
	3	User selects desired language	
	4	System prompts for confirmation	
	5	System saves user’s language preferences and updates UI	
Alternate Flow – Cancel Language Selection	4.1	User cancels when prompted for confirmation	
	4.2	Return to Main Flow step 2	
Rules	Language preference must persist across sessions [REQ_F0302]		
Notes	‘User’ in this use case refers to Student and Parent		
Author	Hesham		

Table 3.1.3: Use Case UC003 Change Language Preferences

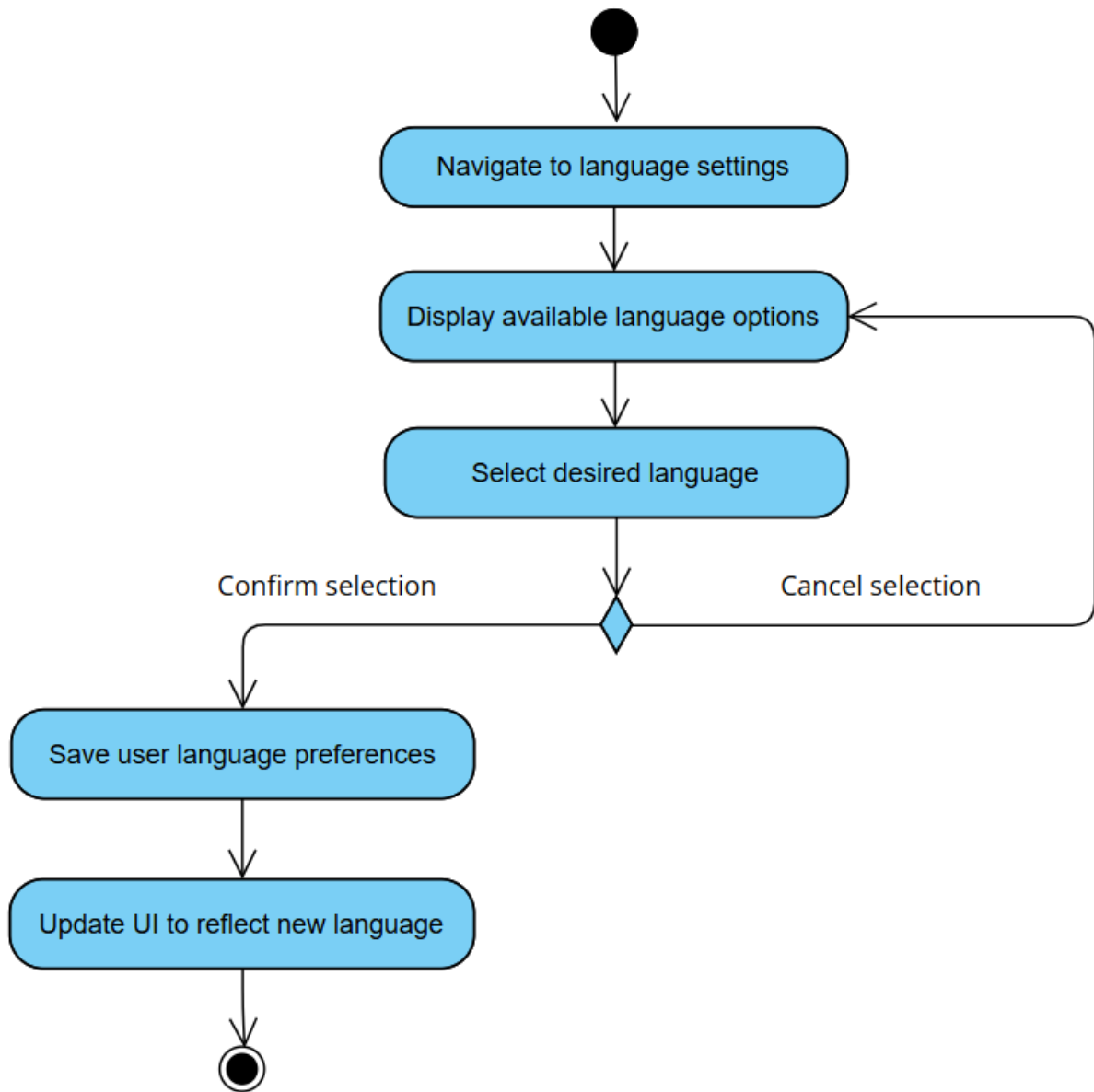


Figure 3.1.3: Activity Diagram for Use Case UC003 Change Language Preferences

3.1.4 F004 Customize Interface

The functional requirement(s) for F004 Customize Interface:

Requirement ID	REQ_F0401	Version	1.0
Description	The system shall provide students with the ability to switch between light and dark display modes		
Author	Hesham		

Requirement ID	REQ_F0402	Version	1.0
Description	The system shall allow students to customize their dashboard layout by moving around widgets.		
Author	Nickleirsch		

Table 3.1.4 illustrates the use case for the customizing interface functionality (UC004), detailing the process as defined by Requirement REQ_F0401 and REQ_F0402, followed by an activity diagram which represents the process flow.

Use Case ID	UC004	Version	1.0
Use Case	F004 Customize Interface		
Purpose	To allow students and parents to personalize their portal interface through theme preferences and widget arrangement		
Actor	Student		
Trigger	Student accesses interface customization settings		
Precondition	Student is logged in		
Postcondition	<ol style="list-style-type: none"> 1. Student interface preferences are changed 2. Dashboard displays customized layout and theme 3. Settings persist across sessions 4. Widgets maintain functionality in new positions 		
Scenario Name	Step	Action	
Main Flow	1	Student access interface settings	
	2	System displays customization options: a) Theme selection (light/dark) b) Widget arrangement interface	
	3	Student selects desired theme	
	4	System previews changes in real-time	
	5	Student saves customization preferences	
	6	System applies and persists changes	
Alternative Flow – Widget Customization	2.1	Student selects widget arrangement	
	2.2	Student rearranges widgets to preferred position	
	2.3	Return to main step 4	

Rules	<ol style="list-style-type: none"> 1. Dashboard layout must maintain responsive design [REQ_I0002] 2. Widget positions must respect screen size constraints [REQ_I0002] 3. Help documentation and tooltips must be available for customization options [REQ_F0701, REQ_F0801]
Author	Hesham

Table 3.1.4: Use Case UC004 Customize Interface

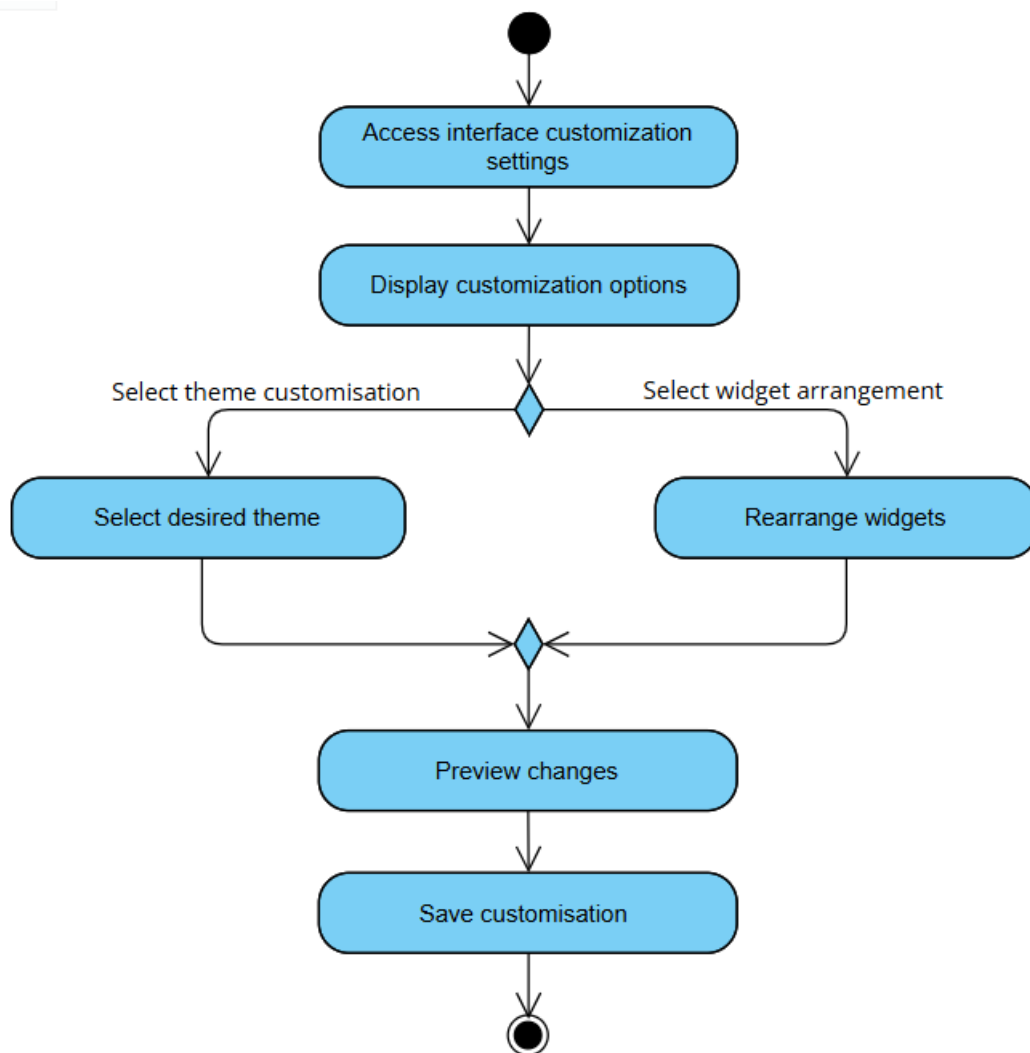


Figure 3.1.4: Activity Diagram for Use Case UC004 Customize Interface

3.1.5 F005 Customize Session Time-Out

The functional requirement(s) for F005 Customize Session Time-Out:

Requirement ID	REQ_F0501	Version	1.0
Description	The system should provide customizable session timeout settings for students		
Author	Hesham		

Requirement ID	REQ_F0502	Version	1.0
Description	The session timeout duration shall be configurable within a secure range, with a minimum of 5 minutes and a maximum of 30 minutes		
Author	Nickleirsch		

Table 3.1.5 illustrates the use case for the customizing session time-out functionality (UC005), detailing the process as defined by Requirement REQ_F0501 and REQ_F0502, followed by an activity diagram which represents the process flow.

Use Case ID	UC005	Version	1.0
Use Case	F005 Customize Session Time-Out		
Purpose	To allow users to personalize their session timeout duration within system-defined security boundaries		
Actor	Student		
Trigger	User accesses session timeout settings in their profile security settings		
Precondition	User is logged in		
Postcondition	1. New session timeout duration is saved 2. Updated timeout is applied to current and future sessions		
Scenario Name	Step	Action	
Main Flow	1	User navigates to security settings	
	2	System displays current timeout settings	
	3	System shows allowed timeout range	
	4	User selects new timeout duration	
	5	System validates selection	
	6	System saves new timeout preference	
	7	System applies for current and future sessions	
Rules	1. Minimum and maximum timeout duration must align with security policy [REQ_F0502] 2. Help documentation and tooltips must explain timeout implications [REQ_F0701, REQ_F0801]		
Author	Hesham		

Table 3.1.5: Use Case UC005 Customize Session Time-Out

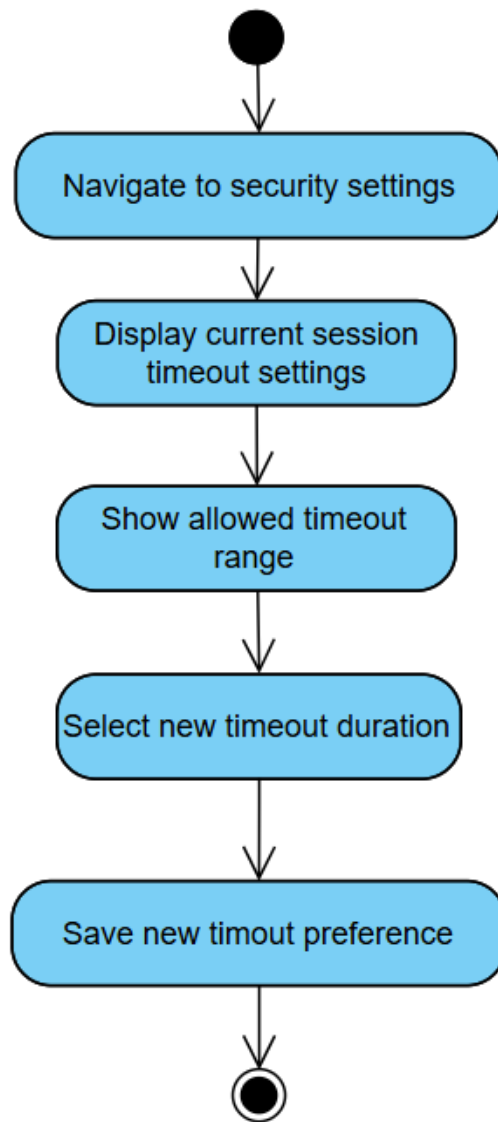


Figure 3.1.5: Activity Diagram for UC005 Customize Session Time-Out

3.1.6 F006 View Notification

The functional requirement(s) for F006 View Notification:

Requirement ID	REQ_F0601	Version	1.0
Description	The system shall support in-portal notifications for students and parents		
Author	Nickleirsch		

Table 3.1.6 illustrates the use case for the view notification functionality (UC006), detailing the process as defined by Requirement REQ_F0601, followed by an activity diagram which represents the process flow.

Use Case ID	UC006	Version	1.0
Use Case	F006 View Notification		
Purpose	To allow students and parents to access their received notifications across different channels		
Actor	Student, Parent		
Trigger	1. New notification is received 2. User clicks on a notification alert		
Precondition	User is logged in		
Postcondition	1. Notification(s) are displayed to user 2. Notification read status is updated 3. Notification(s) are marked as viewed		
Scenario Name	Step	Action	
Main Flow	1	User navigates to notifications section	
	2	System retrieves notifications based on user role and preferences	
	3	System displays notification(s)	
	4	User views notification content	
	5	System updates the notification’s read status	
Alternative Flow – If No Notifications Exist	2.1	System displays “No notifications” message	
	2.2	System redirects to user’s dashboard	
Rules	1. Parents can only view notifications they have consent to access [REQ_F3001] 2. System must respect quiet hours settings if enabled [REQ_F2001] 3. The system must ensure that there are no duplicate notifications sent for the same event [REQ_F0008]		
Notes	‘User’ in this use case refers to Student and Parent.		
	This use case pertains exclusively to the viewing of in-portal notifications within COMSYS. Notifications sent via SMS and email are delivered externally and are not viewable within the COMSYS system.		

Table 3.1.6: Use Case UC006 View Notification

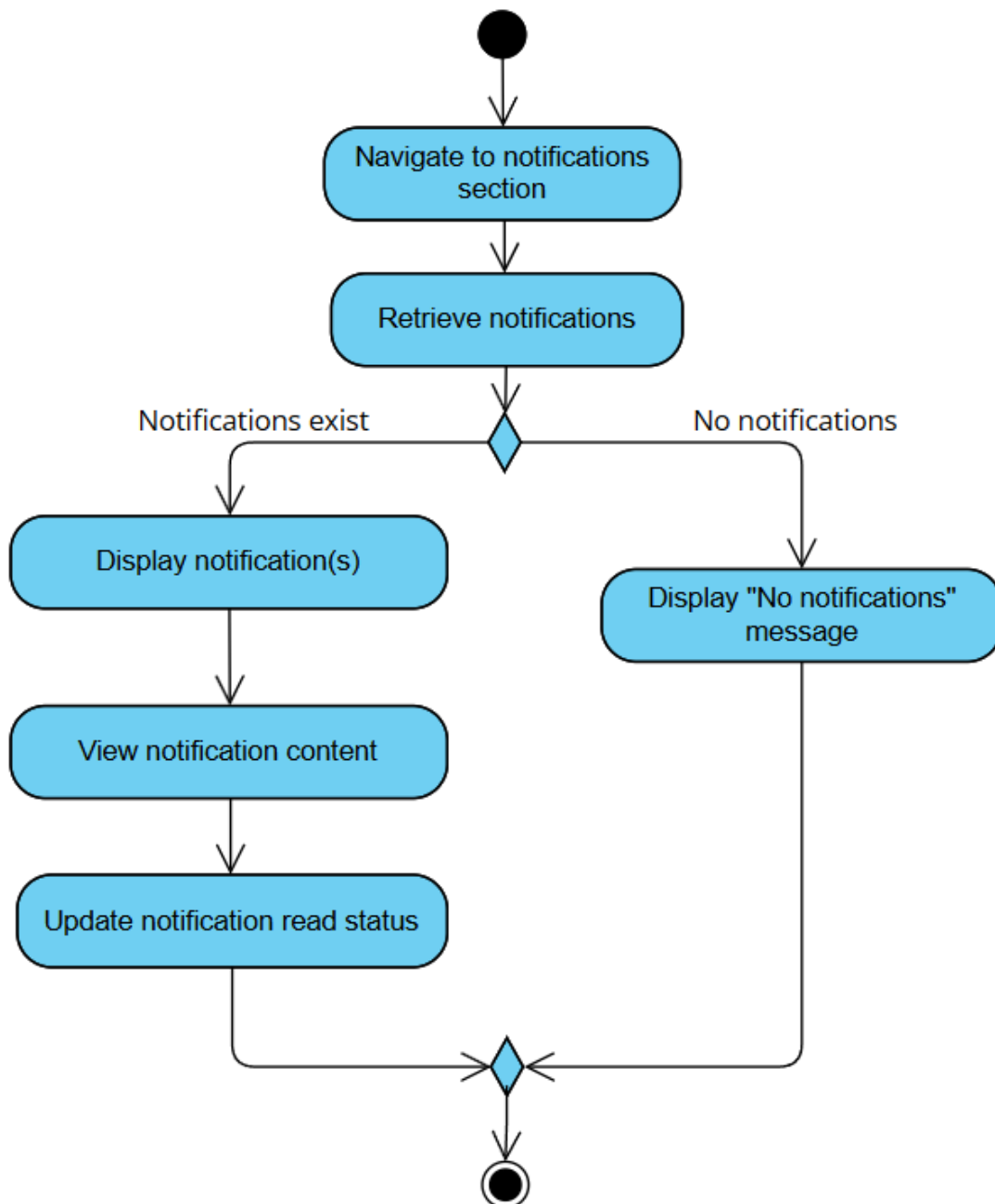


Figure 3.1.6: Activity Diagram for Use Case UC006 View Notification

3.1.7 F007 View Tooltip

The functional requirement(s) for F007 View Tooltip:

Requirement ID	REQ_F0701	Version	1.0
Description	The system shall provide contextual tooltips for students and parents for interface elements on hover		
Author	Hesham		

Table 3.1.7 illustrates the use case for the view tooltip functionality (UC007), detailing the process as defined by Requirement REQ_F0701, followed by an activity diagram which represents the process flow.

Use Case ID	UC007	Version	1.0
Use Case	F007 View Tooltip		
Purpose	To provide students and parents with immediate, contextual help through tooltips for interface elements and features		
Actor	Student, Parent		
Trigger	1. User clicks on tooltip indicator		
Precondition	1. User is logged in 2. User is accessing a feature with tooltip support		
Postcondition	1. Tooltip is displayed 2. User receives immediate contextual help		
Scenario Name	Step	Action	
Main Flow	1	User clicks on an element with tooltip	
	2	System detects tooltip trigger	
	3	System displays relevant contextual help	
Rules	1. Tooltips must be responsive across devices [REQ_I0002] 2. Tooltips must support multiple languages [REQ_F0301, REQ_F0302]		
Notes	'User' in this use case refers to Student and Parent.		
Author	Hesham		

Table 3.1.7: Use Case UC007 View Tooltip

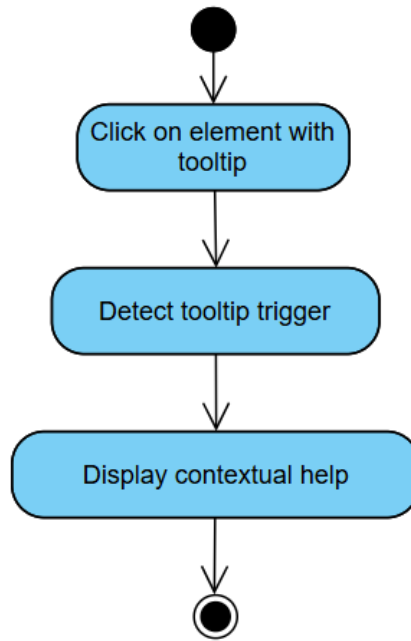


Figure 3.1.7: Activity Diagram for Use Case UC007 View Tooltip

3.1.8 F008 Access Help Documentation

The functional requirement(s) for F008 Access Help Documentation:

Requirement ID	REQ_F0801	Version	1.0
Description	The system shall provide help guides accessible via a help section for students and parents, aimed at explaining system functionalities considered complex.		
Author	Hesham		

Table 3.1.8 illustrates the use case for the access help documentation functionality (UC008), detailing the process as defined by Requirement REQ_F0801, followed by an activity diagram which represents the process flow.

Use Case ID	UC008	Version	1.0
Use Case	F008 Access Help Documentation		
Purpose	To provide students and parents with comprehensive help documentation and user manuals for system features		
Actor	Student, Parent		
Trigger	User navigates to the documentation page		
Precondition	User is logged in		
Postcondition	Help documentation is displayed		
Scenario Name	Step	Action	
Main Flow	1	User navigates to help documentation section	
	2	System displays help categories and search	
	3	User selects topic	
	4	System displays relevant documentation	
Alternate Flow – Search for Help	2.1	User searches for help category	
	2.1	System displays help documentation with filtered category	
Rules	1. Documentation must maintain consistent formatting and be easy to navigate through [REQ_I0001] 2. Documentation must support multiple languages [REQ_F0301, REQ_F0302] 3. System must provide comprehensive guides for complex feature [REQ_F0801]		
Notes	‘User’ in this use case refers to Student and Parent.		
Author	Hesham		

Table 3.1.8: Use Case UC008 Access Help Documentation

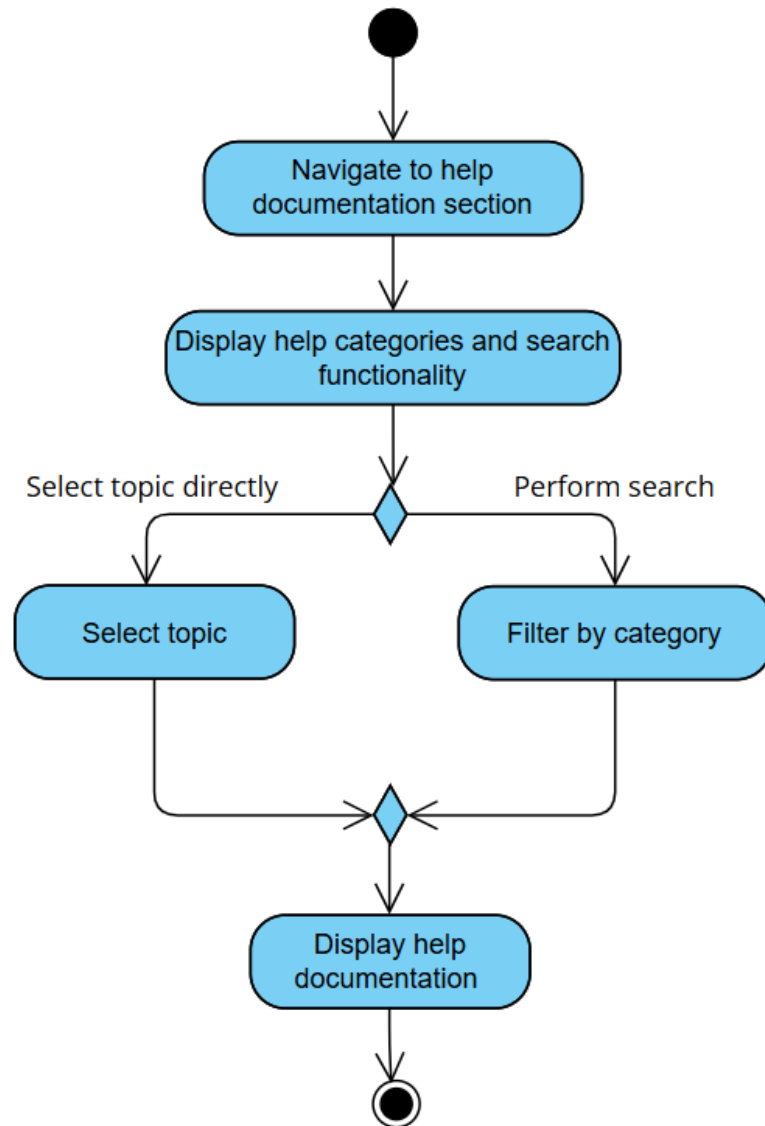


Figure 3.1.8: Activity Diagram for Use Case UC008 Access Help Documentation

3.1.9 F009 Send Notification

The functional requirement(s) for F009 Send Notification:

Requirement ID	REQ_F0901	Version	1.0
Description	The system shall support sending notifications to students and parents via email based on channel preference		
Author	Nickleirsch		

Requirement ID	REQ_F0902	Version	1.0
Description	The system shall allow administrators and lecturers to schedule sending of notifications in advance.		
Author	Nickleirsch		

Requirement ID	REQ_F0903	Version	1.0
Description	The system shall enable sending communications to filtered user groups based on predefined attributes (department, role, program).		
Author	Nickleirsch		

Requirement ID	REQ_F0904	Version	1.0
Description	The system shall allow lecturers and admins to use pre-made templates when creating notifications		
Author	Nickleirsch		

Table 3.1.9 illustrates the use case for the send notification functionality (UC009), detailing the process as defined by Requirement REQ_F0901, REQ_F0902, REQ_F0903 and REQ_F0904, followed by an activity diagram which represents the process flow.

Use Case ID	UC009	Version	1.0
Use Case	F009 Send Notification		
Purpose	Allow lecturers and admins to send notifications to students or groups.		
Actor	Lecturer and Admin		
Trigger	User navigates to send notification page from dashboard		
Precondition	User is logged in		
Postcondition	User has sent or scheduled a notification		
Scenario Name	Step	Action	
Main Flow	1	User navigates to the notification section	
	2	User selects to compose a new notification	
	3	User selects recipients (individuals or groups)	
	4	User chooses notification channel(s) (email, SMS, portal)	
	5	User enters subject and message content	

	6	User confirms and chooses to send the notification immediately
	7	System sends the notification.
	8	System displays a confirmation prompt
Alternate Flow – Schedule Notification	6.1	User chooses to schedule the notification
	6.2	User chooses a future date and time
	6.3	System queues the notification for delivery at the specified time
	6.4	Return to Main Flow step 8
Alternate Flow - Use Notification Template	5.1	User chooses to use a template instead
	5.2	System displays list of available templates
	5.3	System loads the notification template
	5.4	User customizes content as needed.
	5.5	Return to Main Flow step 6
Rules	<ol style="list-style-type: none"> 1. Notification respects user preferences and legal consent [REQ_F1903, REQ_F3001] 2. System prevents duplicate notifications. [REQ_F0008] 3. Critical alerts must be delivered within a minute of their creation [REQ_P0004] 	
Notes	'User' for this use case refers to Lecturer and Admin.	
	<p>Notifications will be delivered through three channels: in-portal, SMS, and email.</p> <ol style="list-style-type: none"> 1. In-portal notifications can be viewed directly within the COMSYS system, as detailed in F006: View Notification. 2. SMS and email notifications are delivered externally to the user's registered phone number and email address. 3. The SMS notification system is implemented via an integrated SMS gateway, as detailed in F032: Send SMS Notification. 	
Author	Nickleirsch	

Table 3.1.9: Use Case UC009 Send Notification

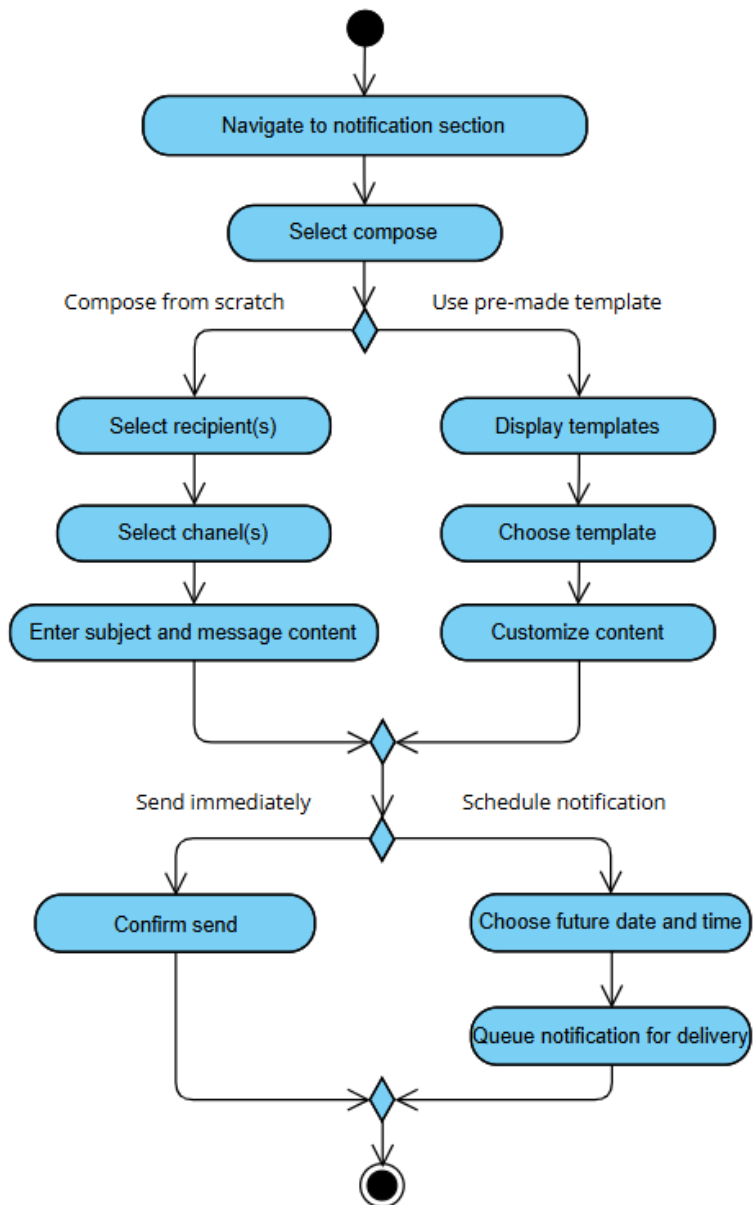


Figure 3.1.9: Activity Diagram for Use Case UC009 Send Notification

3.1.10 F010 Access Calendar

The functional requirement(s) for F010 Access Calendar:

Requirement ID	REQ_F1001	Version	1.0
Description	Students and lecturers shall be able to set personal reminders and manage events or deadlines through the calendar.		
Author	Nickleirsch		

Table 3.1.10 illustrates the use case for the access calendar functionality (UC010), detailing the process as defined by Requirement REQ_F1001, followed by an activity diagram which represents the process flow.

Use Case ID	UC010	Version	1.0
Use Case	F010 Access Calendar		
Purpose	Allow students and lecturers to access their academic calendar, view schedules, upcoming events, deadlines, and other relevant information through the university portal.		
Actor	Student and Lecturer		
Trigger	User navigates to the calendar section.		
Precondition	User is logged in		
Postcondition	<div><div>1. The user sees their personalized academic calendar with relevant events, schedules, and deadlines.</div><div>2. The user can interact with the calendar</div></div>		
Scenario Name	Step	Action	
Main Flow	1	User navigates to the calendar section.	
	2	The system retrieves events, schedules, and deadlines relevant to the user.	
	3	The system displays the calendar	
	4	User views the calendar	
Alternate Flow – Add event	3.1	User chooses to add an event to the calendar	
	3.2	User chooses a date and time	
	3.3	User inputs event type and description	
	3.4	System saves the event	
	3.5	Return to main flow step 4	
Rules	<div><div>1. Calendar must synchronize with external calendars if enabled [REQ_F3301]</div><div>2. Users can set personal reminders and manage events. [REQ_F1002]</div><div>3. Interface must be responsive to screen size. [REQ_I0002]</div><div>4. All calendar data shown must comply with privacy and data security requirements. [REQ_F0002]</div><div>5. Data must be updated within 2 minutes of changes (synchronization). [REQ_P0005]</div></div>		

	6. Event names must be descriptive, not just codes. [REQ_I0003]
Notes	'User' for this use case refers to Student and Lecturer
	The calendar is not the same as the class schedule of a student, for a detailed view of their courses with their instructor, class times and location, students must view their class schedule as detailed in F014 View Class Schedule
Author	Nickleirsch

Table 3.1.10: Use Case UC010 Access Calendar

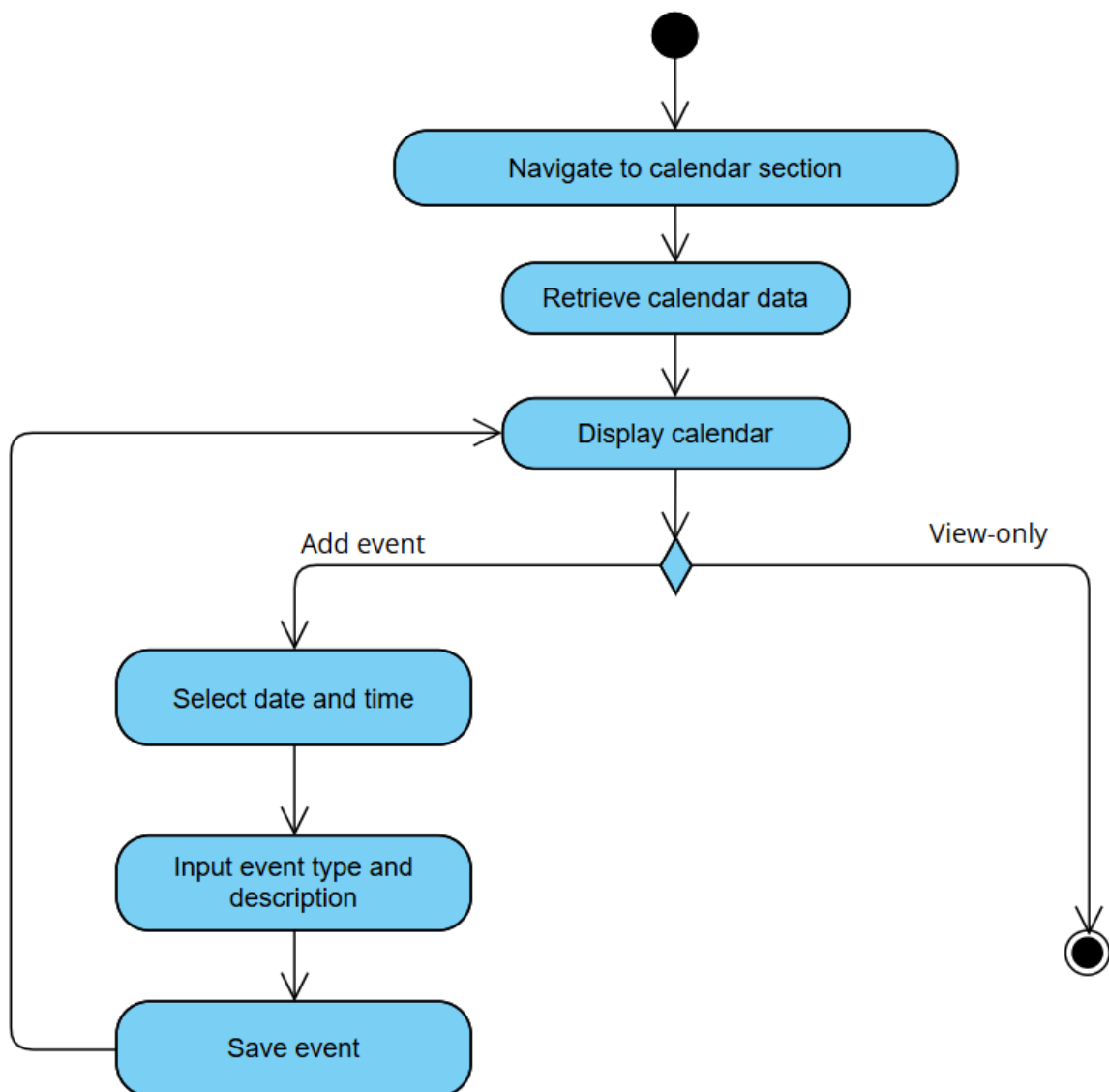


Figure 3.1.10: Activity Diagram for Use Case UC010 Access Calendar

3.1.11 F011 Use Live Chat

The functional requirement(s) for F011 Use Live Chat:

Requirement ID	REQ_F1101	Version	1.0
Description	Students, parents, lecturers and admins shall be able to use the live chat functionality to exchange messages in real time		
Author	Nickleirsch		

Table 3.1.11 illustrates the use case for the use live chat functionality (UC011), detailing the process as defined by Requirement REQ_F1101, followed by an activity diagram which represents the process flow.

Use Case ID	UC011	Version	1.0
Use Case	F011 Use Live Chat		
Purpose	To allow students, parents, lecturers, admins to communicate and exchange messages in real time through a built-in live chat feature.		
Actor	Student, Parent, Lecturer and Admin		
Trigger	1. User accesses the chat section 2. User receives a new message notification and clicks to open chat		
Precondition	User is logged in		
Postcondition	1. Messages are exchanged in real time 2. Chat is updated with the latest conversation		
Scenario Name	Step	Action	
Main Flow	1	User navigates to live chat interface	
	2	System displays conversation history	
	3	User selects a chat box	
	4	System displays past exchanged messages	
	5	User types and sends a message	
	6	System delivers the message to the recipient in real time	
Alternative Flow – Initiate New Chat	3.1	User starts a new chat by searching for a user	
	3.2	System displays search results	
	3.3	Return to Main Flow step 5	
Rules	1. System must load chat boxes within 3 seconds [REQ_P0001] 2. System must ensure chat data is encrypted during transmission [REQ_F0003]		
Author	Nickleirsch		

Table 3.1.11: Use Case UC011 Use Live Chat

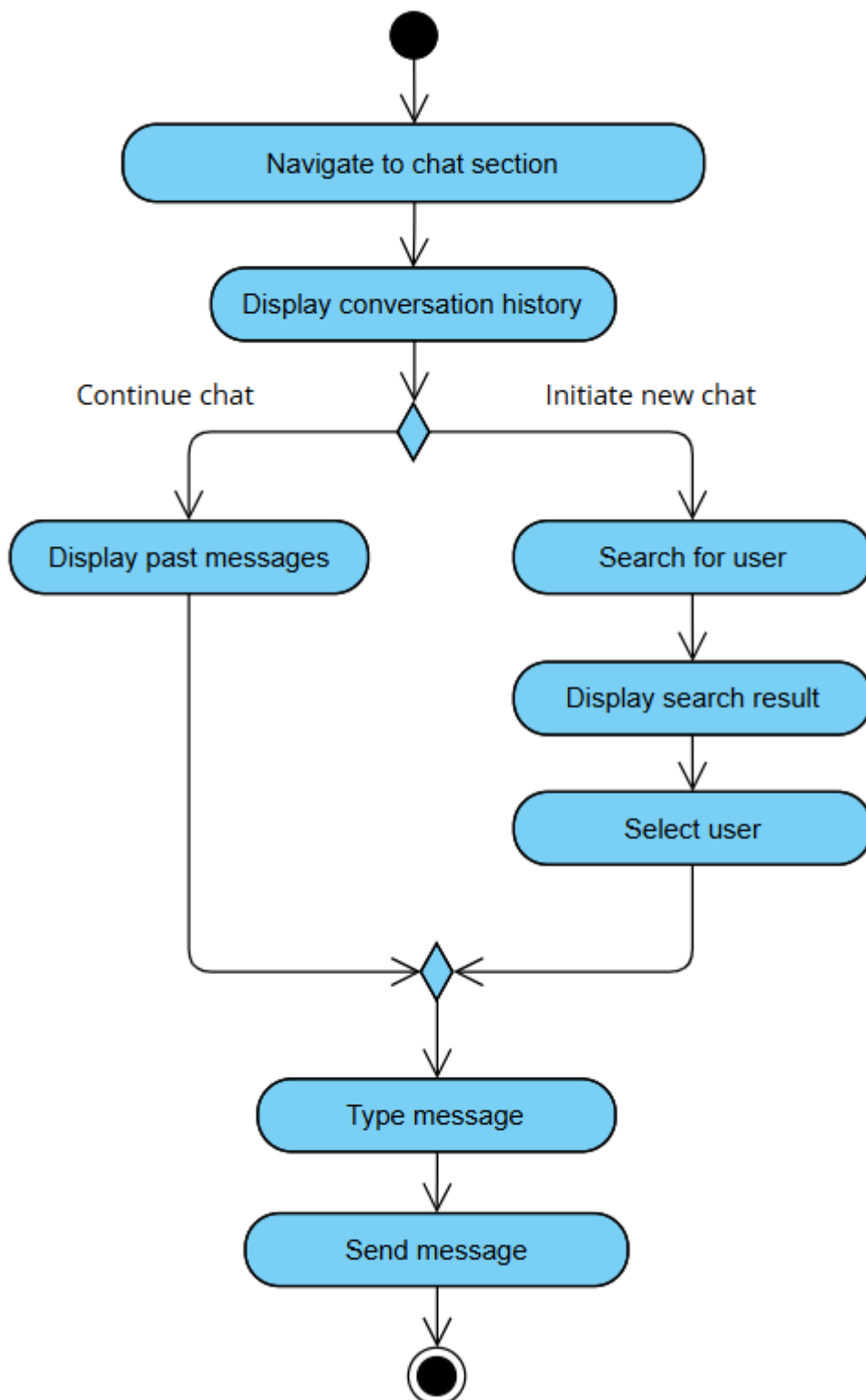


Figure 3.1.11: Activity Diagram for Use Case UC011 Use Live Chat

3.1.12 F012 View Attendance Record

The functional requirement(s) for F012 View Attendance Record:

Requirement ID	REQ_F1201	Version	1.0
Description	The student dashboard shall display an overview of the student's attendance records for each enrolled course.		
Author	Hesham		

Table 3.1.12 illustrates the use case for the view attendance record functionality (UC012), detailing the process as defined by Requirement REQ_F1201, followed by an activity diagram which represents the process flow.

Use Case ID	UC012	Version	1.0
Use Case	F012 View Attendance Record		
Purpose	To allow students to access and monitor their attendance records across all enrolled courses		
Actor	Student		
Trigger	1. Student accesses attendance section 2. Student selects specific course for attendance view 3. Automated attendance digest notification clicked		
Precondition	1. Student is logged in 2. Student has active course enrolments		
Postcondition	1. Attendance records are displayed 2. Any attendance alerts are highlighted as read		
Scenario Name	Step	Action	
Main Flow	1	Student navigates to attendance section	
	2	System retrieves current attendance records	
	3	System displays attendance overview for all courses	
	4	Student views detailed attendance information	
Alternate Flow - Attendance Record Cannot Be Retrieved	2.1	System cannot retrieve attendance record	
	2.2	Display error message	
Rules	1. System must load attendance records within 3 seconds [REQ_P0001] 2. System must provide consistent header formatting [REQ_I0001] 3. System must provide tooltips for attendance calculations [REQ_F0701] 4. Attendance record must synchronize across interfaces within 5 seconds [REQ_P0003, REQ_F0005, REQ_F0010]		
Author	Hesham		

Table 3.1.12: Use Case UC012 View Attendance Record

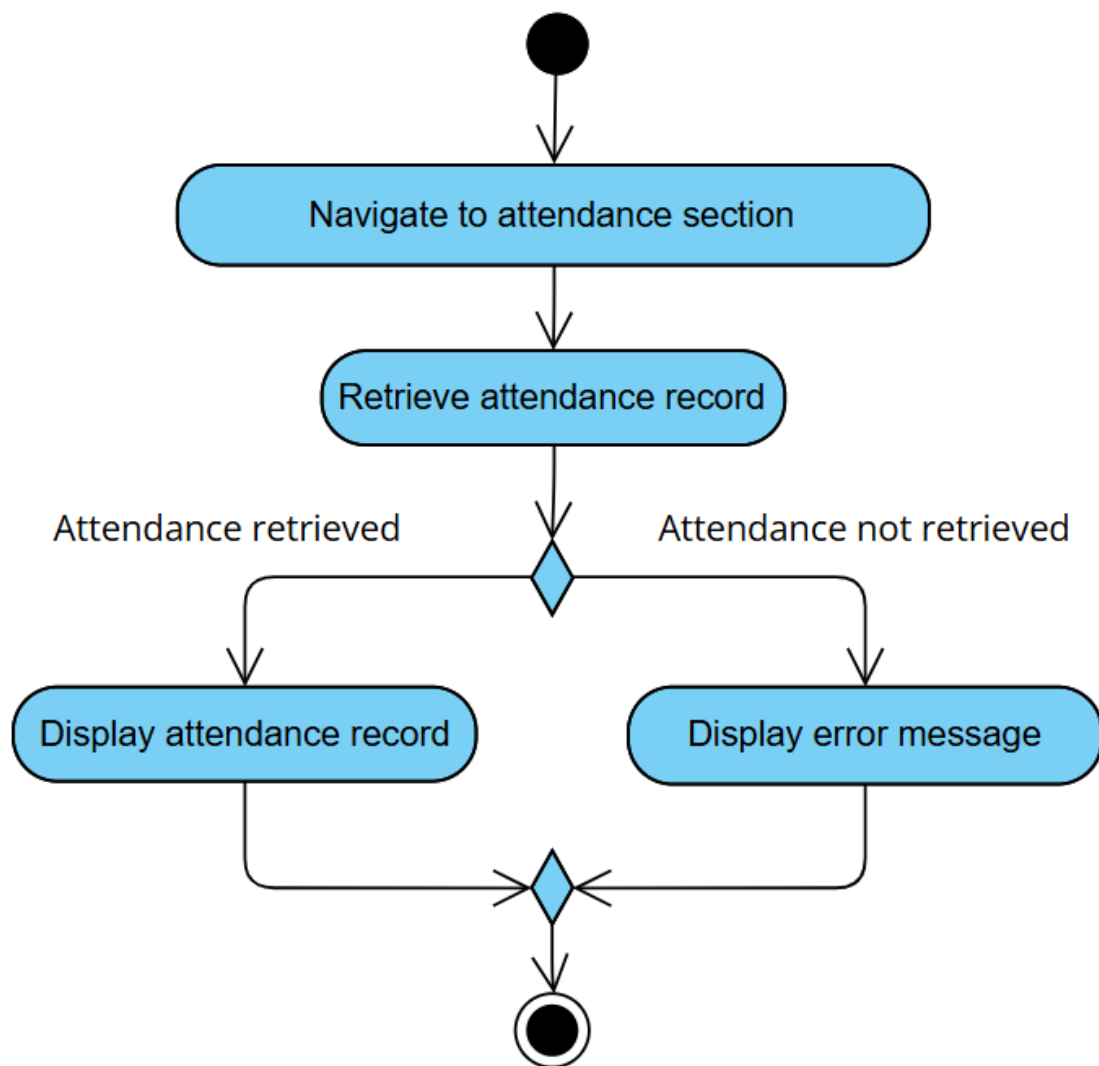


Figure 3.1.12: Activity Diagram for Use Case UC012 View Attendance Record

3.1.13 F013 View Academic Record

The functional requirement(s) for F013 View Academic Record:

Requirement ID	REQ_F1301	Version	1.0
Description	The student dashboard shall display a summary of the student's academic records, including grades and completed courses.		
Author	Nickleirsch		

Table 3.1.13 illustrates the use case for the view academic record functionality (UC013), detailing the process as defined by Requirement REQ_F1301, followed by an activity diagram which represents the process flow.

Use Case ID	UC013	Version	1.0
Use Case	F013 View Academic Record		
Purpose	To allow students to access and review their comprehensive academic records including grades, transcripts, and academic progress		
Actor	Student		
Trigger	1. Student accesses academic records section 2. New grade notification received and clicked		
Precondition	1. Student is logged in 2. Student has academic records in the system		
Postcondition	1. Academic records are displayed 2. Any academic alerts are highlighted as read		
Scenario Name	Step	Action	
Main Flow	1	Student navigates to academic section	
	2	System retrieves academic records	
	3	System displays comprehensive academic performance	
	4	Student views detailed academic information	
	5	Student can export their academic data if they wish to	
Rules	1. Academic data must sync within 5 seconds of lecturer updates [REQ_P0003, REQ_F0005, REQ_F0010] 2. System must maintain consistent header formatting [REQ_I0001] 3. System must provide tooltips for GPA calculations and academic standings [REQ_F0701] 4. Academic data must be exportable in Excel/CSV format [REQ_F2601]		
Author	Nickleirsch		

Table 3.1.13: Use Case UC013 View Academic Record

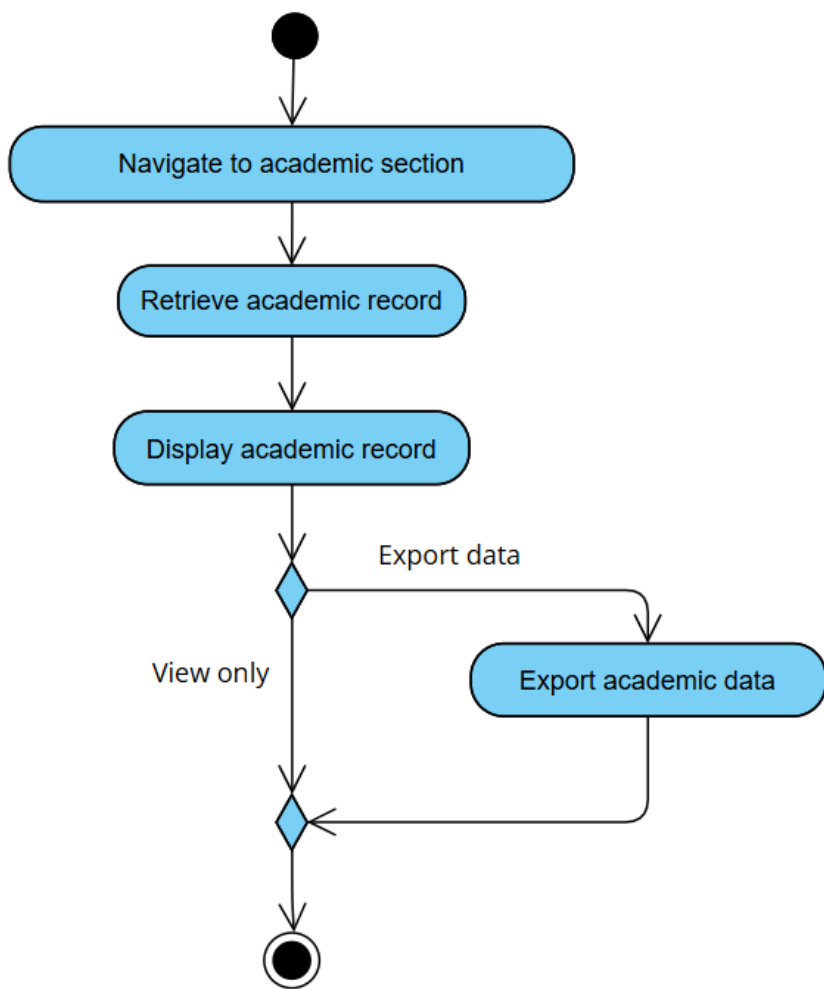


Figure 3.1.13: Activity Diagram for Use Case UC013 View Academic Record

3.1.14 F014 View Class Schedule

The functional requirement(s) for F014 View Class Schedule:

Requirement ID	REQ_F1401	Version	1.0
Description	The system shall present the student's current class schedule with course names, times, and locations.		
Author	Hesham		

Table 3.1.14 illustrates the use case for the view class schedule functionality (UC014), detailing the process as defined by Requirement REQ_F1401, followed by an activity diagram which represents the process flow.

Use Case ID	UC014	Version	1.0
Use Case	F014 View Class Schedule		
Purpose	To allow students to view their current class schedule		
Actor	Student		
Trigger	1. Student navigates to class schedule section 2. Schedule change notification received and clicked		
Precondition	1. Student is logged in 2. Student has active course enrolments		
Postcondition	Current class schedule is displayed		
Scenario Name	Step	Action	
Main Flow	1	Student navigates to class schedule section	
	2	System retrieves current class schedule	
	3	System displays comprehensive schedule view	
	4	Student views detailed schedule information	
Alternate Flow - Schedule Cannot Be Retrieved	2.1	System cannot retrieve schedule data	
	2.2	Display error message	
Rules	1. System must maintain consistent header formatting [REQ_I0001] 2. System must provide tooltips for schedule features [REQ_F0701] 3. Schedule changes must sync across interfaces within 5 seconds [REQ_P0003, REQ_F0005, REQ_F0010] 4. Descriptive course names [REQ_I0003]		
Notes	The class schedule is different from the calendar; the class schedule is a list of every course the student is enrolled in, detailed each class, the instructor, time and location. The calendar on the other hand is an integrated external calendar where events and reminders can be saved too, detailed in F010 Access Calendar		
Author	Hesham		

Table 3.1.14: Use Case UC014 View Class Schedule

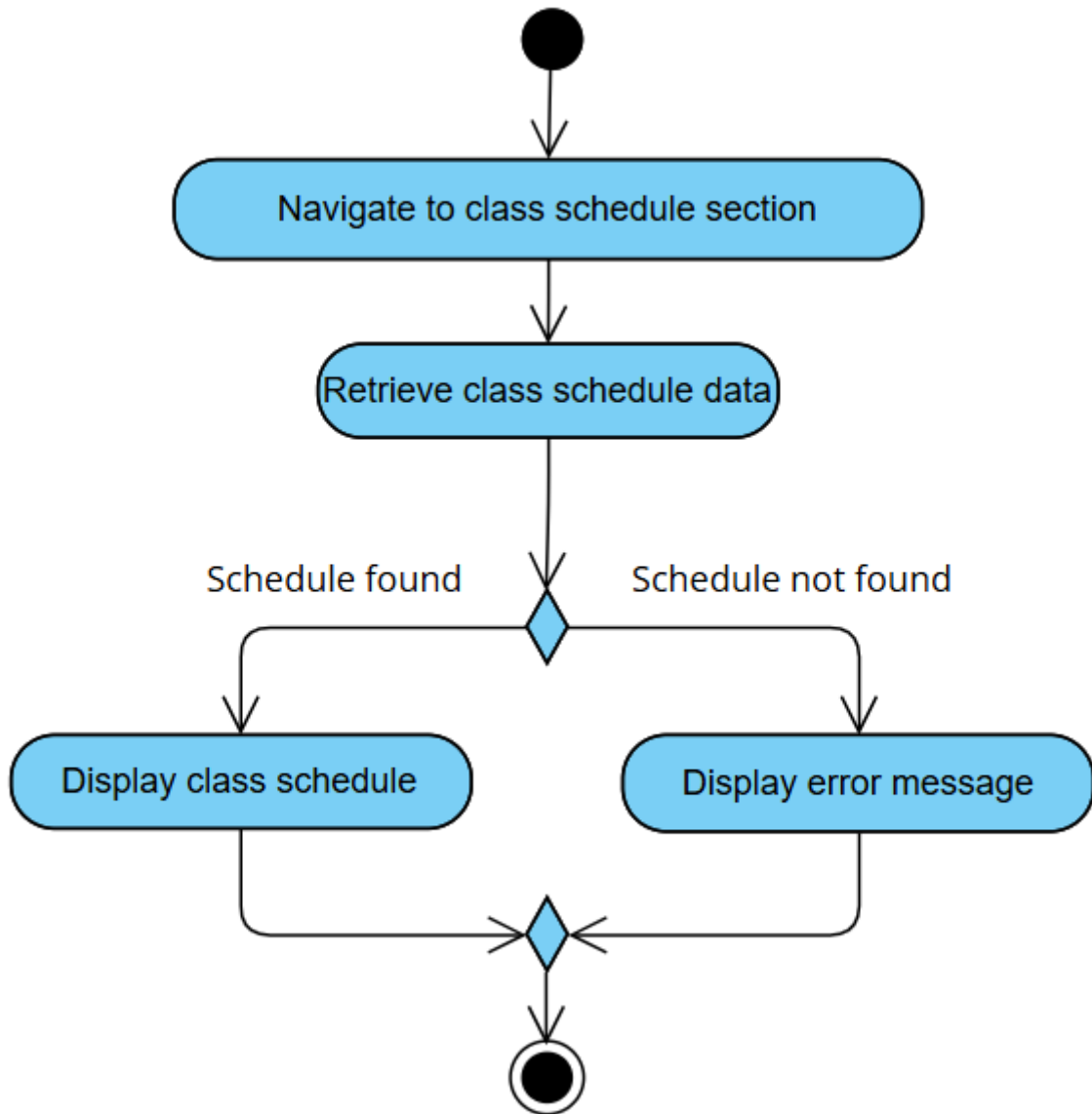


Figure 3.1.14: Activity Diagram for Use Case UC014 View Class Schedule

3.1.15 F015 View Exam Timetable

The functional requirement(s) for F015 View Exam Timetable:

Requirement ID	REQ_F1501	Version	1.0
Description	The student dashboard shall display upcoming exam dates and times relevant to the student's courses.		
Author	Hesham		

Table 3.1.15 illustrates the use case for the view class schedule functionality (UC015), detailing the process as defined by Requirement REQ_F1501, followed by an activity diagram which represents the process flow.

Use Case ID	UC015	Version	1.0
Use Case	View Exam Timetable		
Purpose	To allow students to access and view their examination schedule, including dates, times, locations, and exam requirements		
Actor	Student		
Trigger	1. Student accesses exam timetable section 2. Exam schedule notification received and clicked		
Precondition	1. Student is logged in 2. Student has active enrolments with exams 3. Exam schedule is published		
Postcondition	Exam timetable is displayed		
Scenario Name	Step	Action	
Main Flow	1	Student navigates to exam timetable section	
	2	System retrieves current exam schedule	
	3	System displays comprehensive exam timetable	
	4	Student views detailed exam information	
Alternative Flow – If Exam Timetable Data Cannot Be Retrieved	2.1	System cannot retrieve exam data	
	2.2	System displays error message	
Rules	1. System must maintain consistent header formatting [REQ_I0001] 2. Exam dates must sync across interfaces within 5 seconds [REQ_P0003, REQ_F0005, REQ_F0010]		
Author	Hesham		

Table 3.1.15: Use Case UC015 View Exam Timetable

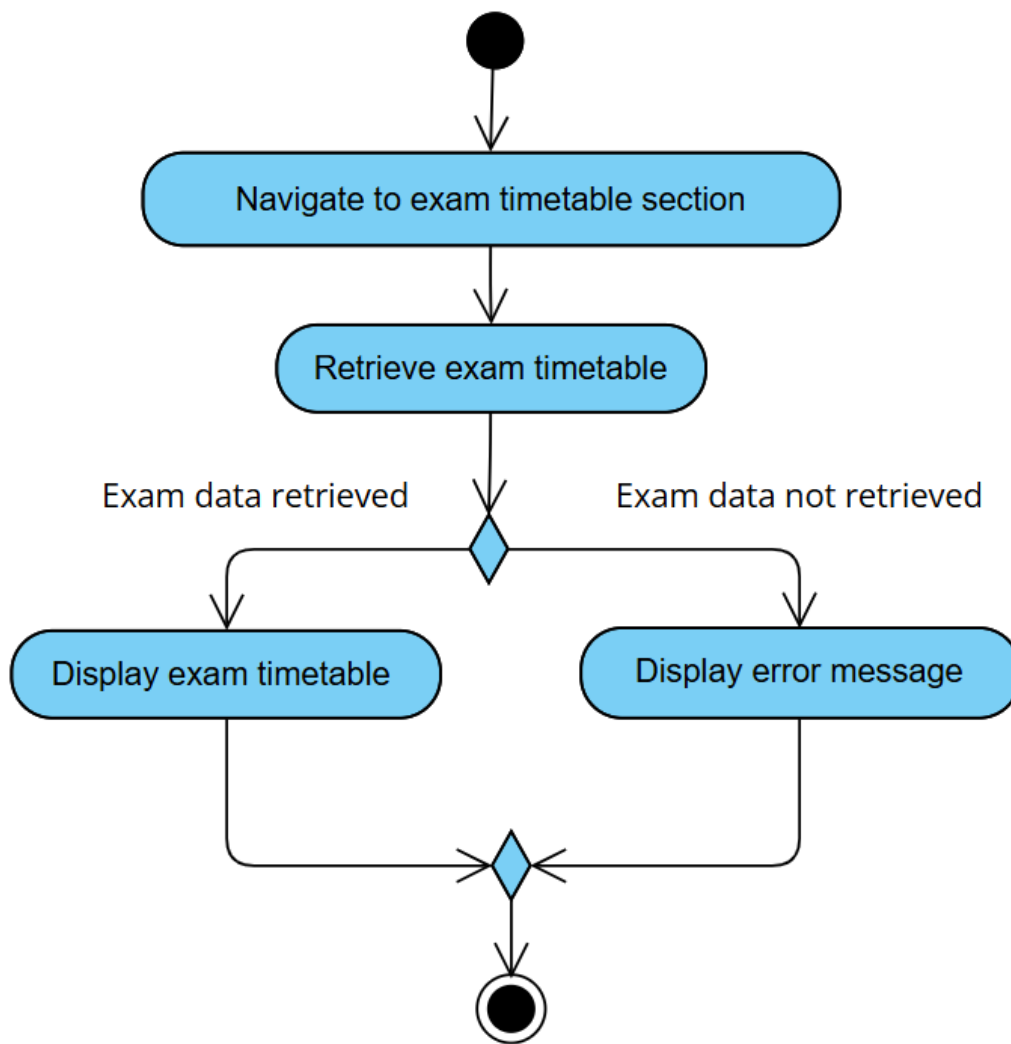


Figure 3.1.15: Activity Diagram for Use Case UC015 View Exam Timetable

3.1.16 F016 View Billing Information

The functional requirement(s) for F016 View Billing Information:

Requirement ID	REQ_F1601	Version	1.0
Description	The student dashboard shall display finance section with billing information, payment status, and fee breakdowns.		
Author	Hesham		

Table 3.1.16 illustrates the use case for the view billing information functionality (UC016), detailing the process as defined by Requirement REQ_F1601, followed by an activity diagram which represents the process flow.

Use Case ID	UC016	Version	1.0
Use Case	View Billing Information		
Purpose	To allow students to access and view their invoices and fee as well as due dates to make those payments		
Actor	Student		
Trigger	1. Student accesses billing section 2. Billing update notification received and clicked		
Precondition	Student is logged in		
Postcondition	Billing information is displayed		
Scenario Name	Step	Action	
Main Flow	1	Student navigates to billing section	
	2	System retrieves current billing data	
	3	System displays comprehensive billing information	
	4	Student views detailed billing information with breakdowns by course and due dates	
Alternative Flow – If Billing Data Cannot Be Retrieved	2.1	System cannot retrieve billing information	
	2.2	System displays error message	
Rules	i. System must maintain consistent header formatting [REQ_I0001] ii. System must provide tooltips for how finances are calculated [REQ_F0701]		
Notes	Billing data is view-only within COMSYS, payments are made externally and as such, not within the confines of the system.		
Author	Hesham		

Table 3.1.16: Use Case UC016 View Billing Information

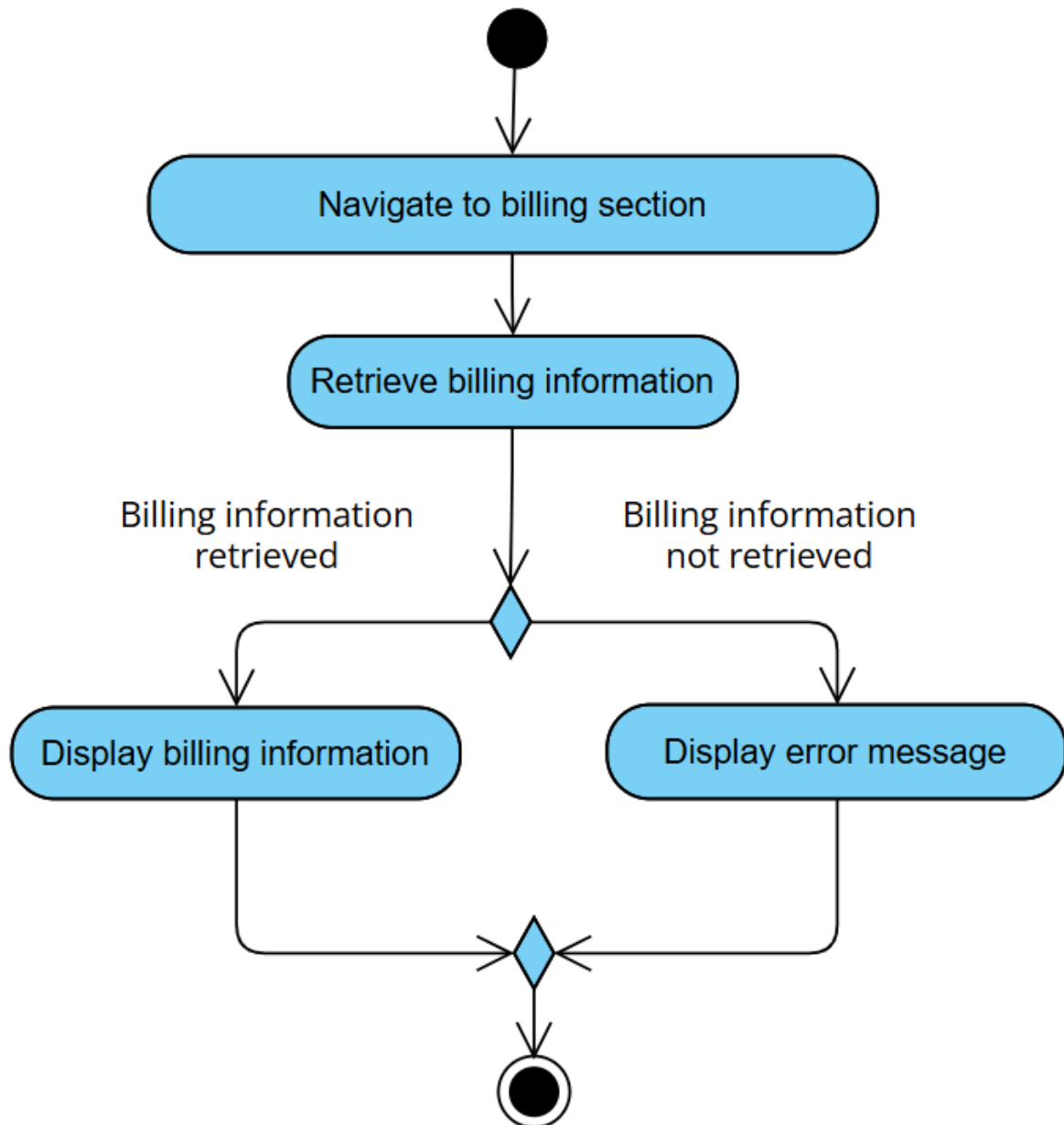


Figure 3.1.16: Activity Diagram for Use Case UC016 View Billing Information

3.1.17 F017 Enrol in Course

The functional requirement(s) for F017 Enrol in Course:

Requirement ID	REQ_F1701	Version	1.0
Description	The student dashboard shall provide a section for managing course registration and enrolment.		
Author	Hesham		

Table 3.1.17 illustrates the use case for the enrol in course functionality (UC017), detailing the process as defined by Requirement REQ_F1701, followed by an activity diagram which represents the process flow.

Use Case ID	UC017	Version	1.0
Use Case	F017 Enrol in Course		
Purpose	To enable students to register for courses during their designated enrolment period		
Actor	Student		
Trigger	1. Student navigates to enrolment section		
Precondition	1. Student is logged in 2. Student has an active enrolment session		
Postcondition	1. Student is enrolled in selected course(s) 2. Student schedule is updated 3. Calendar is synced with new schedule		
Scenario Name	Step	Action	
Main Flow	1	Student navigates to course enrolment section	
	2	System displays available courses for enrolment	
	3	Student selects desired course(s)	
	4	System validates enrolment eligibility	
	5	Student confirms course selection	
	6	System processes enrolment request	
	7	System displays "enrolment successful"	
	8	System calls calendar sync to update calendar	
	9	System asks whether user wants to enrol another course	
	10	User chooses "No"	
Alternative Flow – If Prerequisites Not Met	4.1.1	Student does not meet prerequisite requirements	
	4.1.2	System displays "prerequisite(s) not met!" error	
	4.1.3	Return to main flow step 9	
Alternative Flow – If Schedule Conflict Found	4.2.1	System detected a schedule conflict	
	4.2.2	System displays "schedule conflict found!" error	
	4.2.3	Return to Main Flow step 9	
Alternative Flow – Enrol Another Course	9.1	Student choses "Yes"	
	9.2	Return to Main Flow step 2	

Rules	<ol style="list-style-type: none"> 1. System must maintain consistent header formatting [REQ_I0001] 2. The course enrolment process must be in a single window [REQ_I0007]
Author	Hesham

Table 3.1.17: Use Case UC017 Enrol in Course

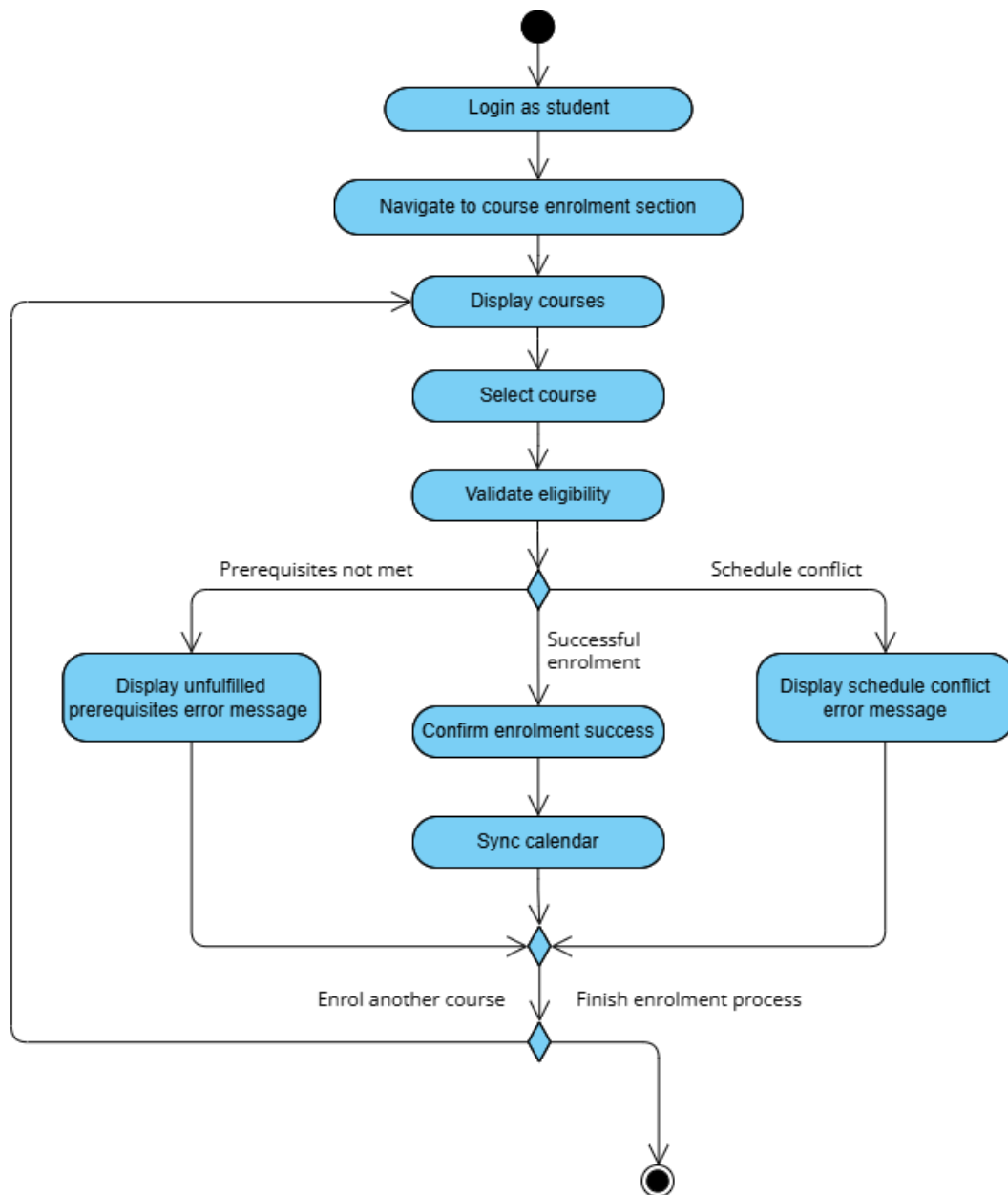


Figure 3.1.17: Activity Diagram for Use Case UC017 Enrol in Course

3.1.18 F018 Search Past Announcement

The functional requirement(s) for F018 Search Past Announcement:

Requirement ID	REQ_F1801	Version	1.0
Description	The student dashboard shall display university announcements relevant to the student.		
Author	Hesham		

Requirement ID	REQ_F1802	Version	1.0
Description	The system shall provide a search functionality for past announcements and communications		
Author	Hesham		

Table 3.1.18 illustrates the use case for the search past announcement functionality (UC018), detailing the process as defined by Requirement REQ_F1801 and REQ_F1802, followed by an activity diagram which represents the process flow.

Use Case ID	UC018	Version	1.0
Use Case	F018 Search Past Announcements		
Purpose	To allow students to search and retrieve historical announcements using various search criteria		
Actor	Student		
Trigger	1. Student navigates to announcement search page		
Precondition	1. Student has an active session 2. Announcements exist for the requesting student		
Postcondition	Search results are displayed		
Scenario Name	Step	Action	
Main Flow	1	Student navigates to announcement search section	
	2	System displays search interface	
	3	Student enters search criteria	
	4	System retrieves matching announcements	
	5	System displays search results	
	6	Student views desired announcement	
Alternative Flow – If No Results Found	4.1	System finds no result for search criteria	
	4.2	System displays “No Results” message	
	4.3	System prompts retry	
	4.4	Return to Main Flow step 2	
Rules	System must maintain announcement read status tracking [REQ_F0001]		
Author	Nickleirsch		

Table 3.1.18: Use Case UC018 Search Past Announcement

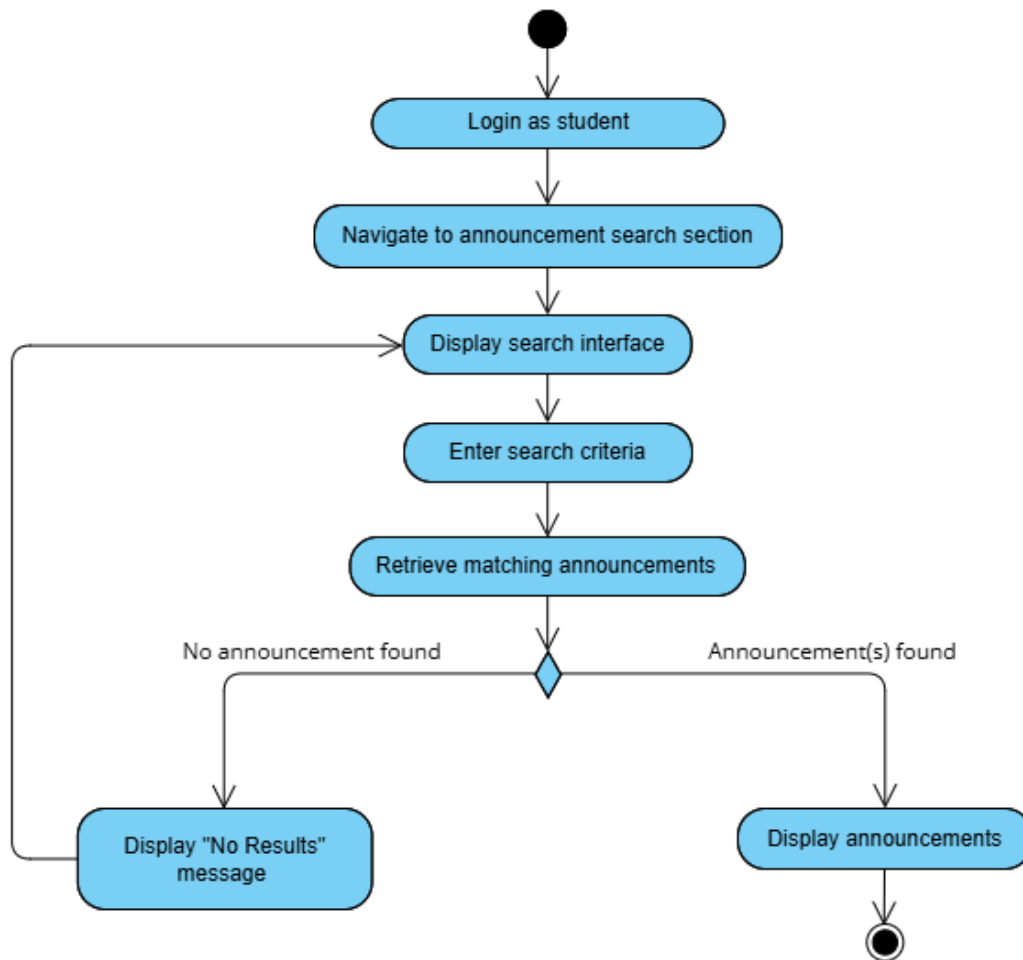


Figure 3.1.18: Activity Diagram for Use Case UC018 Search Past Announcement

3.1.19 F019 Customize Notification Preference

The functional requirement(s) for F019 Customize Notification Preferences:

Requirement ID	REQ_F1901	Version	1.0
Description	The system shall allow student users to enable or disable notifications for specific communication types, such as academic alerts, billing updates, or general announcements.		
Author	Hesham		

Requirement ID	REQ_F1902	Version	1.0
Description	The system shall allow users to customize their notification preferences and channels for different types of communications.		
Author	Hesham		

Requirement ID	REQ_F1903	Version	1.0
Description	The system shall store and apply notification preferences per student in their user profile.		
Author	Nickleirsch		

Table 3.1.19 illustrates the use case for the customize notification preferences functionality (UC019), detailing the process as defined by Requirement REQ_F1901, REQ_F1902 and REQ_F1903, followed by an activity diagram which represents the process flow.

Use Case ID	UC019	Version	1.0
Use Case	F019 Customize Notifications Preference		
Purpose	To allow students to personalize their notification settings across different channels and categories		
Actor	Student		
Trigger	Student accesses notification settings		
Precondition	Student is logged in		
Postcondition	1. Notification preferences are updated 2. New settings are active immediately		
Scenario Name	Step	Action	
Main Flow	1	Student navigates to notification preferences	
	2	System displays current notification settings	
	3	Student modifies preferred channels/categories	
	4	Student confirms modifications	
	5	System saves new preferences	
Rules	1. System must support filtering and muting by category [REQ_F1901] 2. System must respect quiet hours setting [REQ_F2001] 3. The system will store and apply notification preferences ensuring it persists across sessions [REQ_F1903].		

Author	Hesham
--------	--------

Table 3.1.19: Use Case UC019 Customize Notification Preference

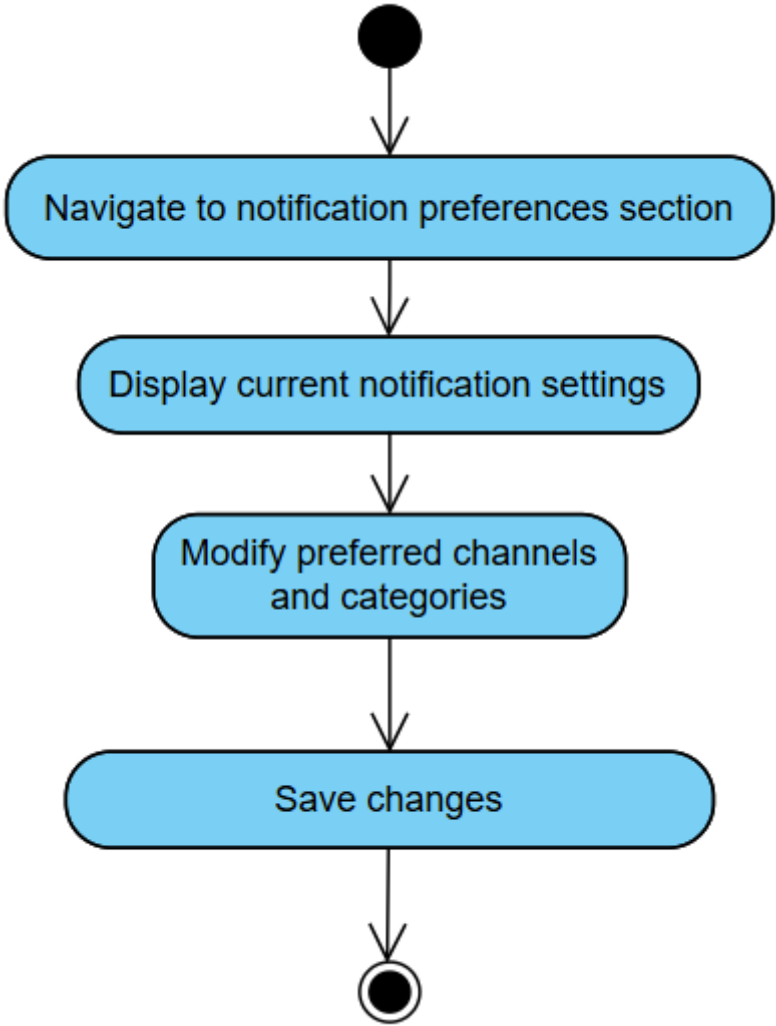


Figure 3.1.19: Activity Diagram for Use Case UC019 Customize Notification Preference

3.1.20 F020 Set 'Quiet Hours'

The functional requirement(s) for F020 Set 'Quiet Hours':

Requirement ID	REQ_F2001	Version	1.0
Description	The system should provide a "quiet hours" feature allowing students to temporarily suspend non-urgent notifications		
Author	Hesham		

Table 3.1.20 illustrates the use case for the set 'quiet hours' functionality (UC020), detailing the process as defined by Requirement REQ_F2001, followed by an activity diagram which represents the process flow.

Use Case ID	UC020	Version	1.0
Use Case	F020 Set 'Quiet Hours'		
Purpose	To allow users to define time periods when notifications should be suppressed		
Actor	Student		
Trigger	1. Student accesses quiet hours settings 2. System prompts student for quiet hours setup		
Precondition	User is logged in		
Postcondition	1. Quiet hours settings are updated 2. Changes are saved and active		
Scenario Name	Step	Action	
Main Flow	1	User navigates to quiet hours settings	
	2	System displays current quiet hours configuration	
	3	User sets quiet hours' time ranges	
	4	User selects days of week for quiet hours	
	5	User configures critical notification exceptions	
	6	User confirms changes	
	7	System saves new quiet hours configuration	
Rules	1. System must support filtering and muting by category [REQ_F1901] 2. The system will store and apply notification preferences ensuring it persists across sessions [REQ_F1903].		
Author	Hesham		

Table 3.1.20: Use Case UC020 Set 'Quiet Hours'

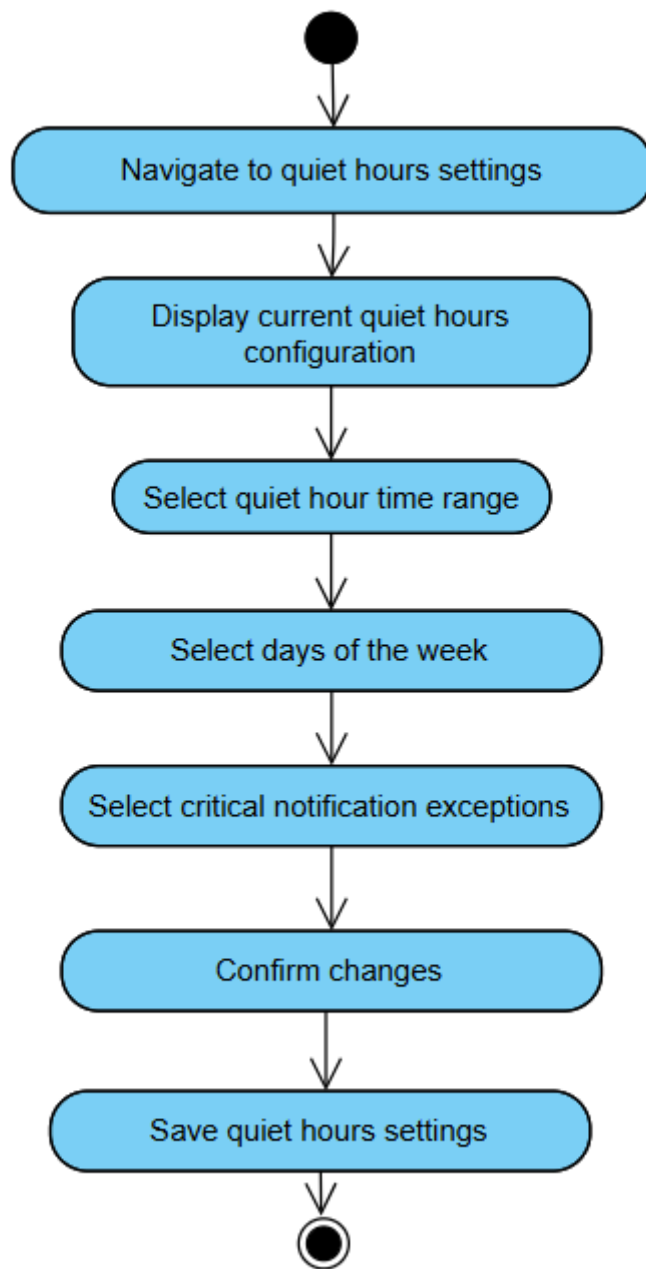


Figure 3.1.20: Activity Diagram for Use Case UC020 Set 'Quiet Hours'

3.1.21 F021 View Child's Information

The functional requirement(s) for F021 View Child's Information:

Requirement ID	REQ_F2101	Version	1.0
Description	The system shall provide a dedicated portal for parents to access their child's grades, attendance, and financial information.		
Author	Lim Xin Yee		

Table 3.1.21 illustrates the use case for the view child's information functionality (UC021), detailing the process as defined by Requirement REQ_F2101, followed by an activity diagram which represents the process flow.

Use Case ID	UC021	Version	1.0
Use Case	F021 View Child’s Information		
Purpose	To allow parents to securely access and view their child's academic, attendance, and financial records through the university's portal.		
Actor	Parent		
Trigger	Parent logs in to the university portal and selects the option to view their child’s information.		
Precondition	1. Parent has a registered and verified user account. 2. The student (child) has granted the necessary consent for data access, in accordance with university privacy policies		
Postcondition	The parent can view up-to-date information regarding their child’s grades, attendance, financial status, and academic details, as presented in a readable format (e.g., tables, summaries, charts).		
Scenario Name	Step	Action	
Main Flow	1	Parent selects navigates to child information section	
	2	System checks that consent has been provided by the student.	
	3	System retrieves child’s academic, attendance, and financial data.	
	4	System displays the information with visual aids	
	5	Parent may choose to export academic data for record-keeping.	
Alternate Flow – Student Consent Not Granted	2.1	Student did not grant consent to the parent.	
	2.2	System displays a prompt stating that access to the requested information is restricted.	
Rules	1. Only parents with valid authentication and authorization (student’s explicit consent) can access child-related data. [REQ F0009, REQ F3001]		

	<ol style="list-style-type: none"> 2. Information displayed must comply with privacy regulation. [REQ_F0002] 3. Parental access is restricted to only the child's academic, financial, and attendance information. [REQ_F2101] 4. Data presented must be in a readable, structured format for comprehension. [REQ_I0004]
Notes	Visual aids refer to tables, charts and summaries as parents are users who need quick understanding
	<p>For a parent to view a student's personal data, the student must first provide explicit consent to the university. This consent must be given by physically submitting a signed consent form to the university administration.</p> <p>Once the form is received, an admin user will configure the parent's access to the student's data in the system, as defined in F00X: Configure Parent Access.</p> <p>This process is required to ensure compliance with applicable privacy laws and protect student data.</p>
Author	Lim Xin Yee and Nickleirsch

Table 3.1.21: Use Case UC021 View Child's Information

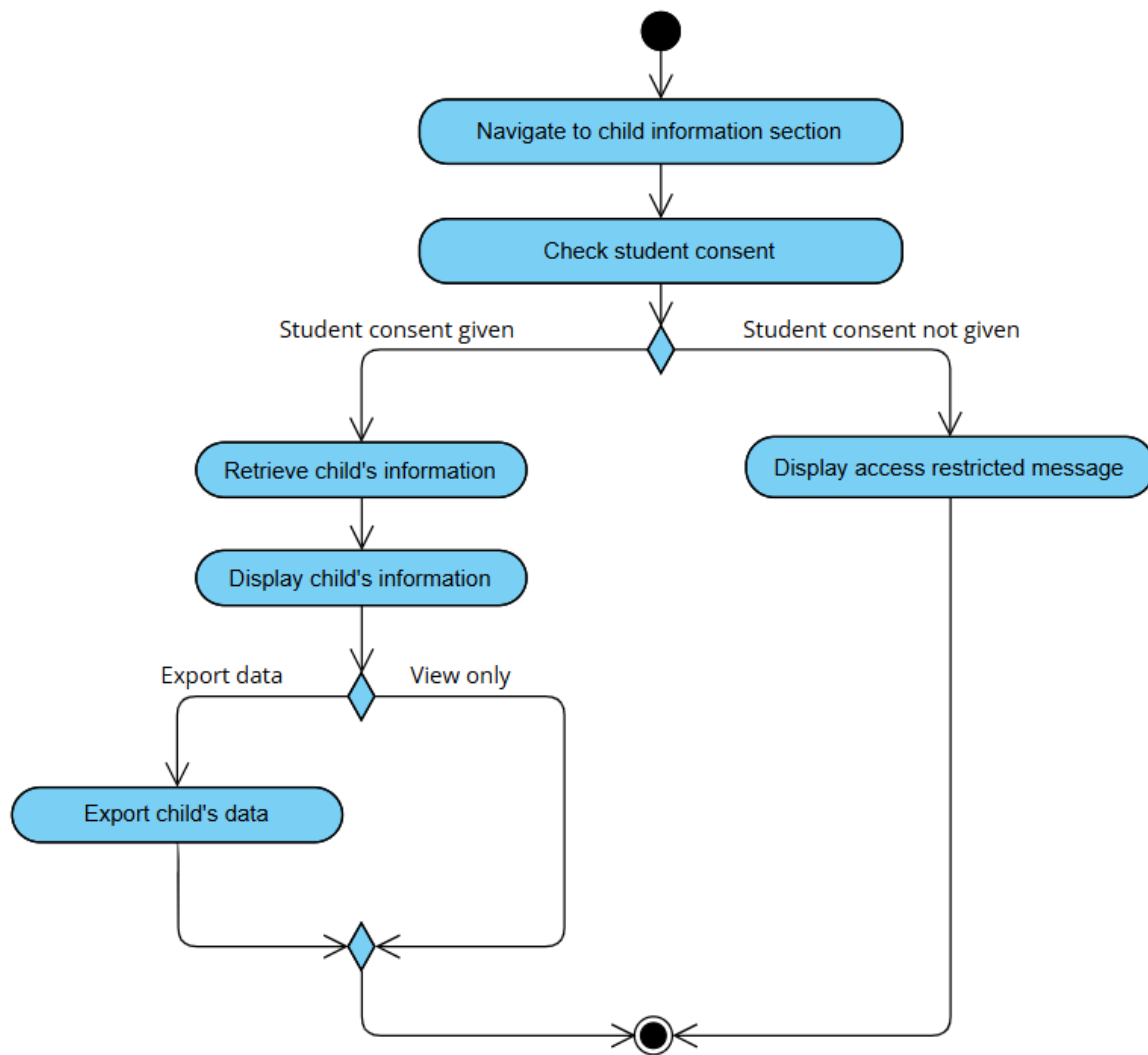


Figure 3.1.21: Activity Diagram for Use Case UC021 View Child's Information

3.1.22 F022 View University Contact Directory

The functional requirement(s) for F022 View University Contact Directory:

Requirement ID	REQ_F2201	Version	1.0
Description	The system shall include a contact directory interface, maintained by admins, for parents to access with filters for faculty and department		
Author	Nickleirsch		

Table 3.1.22 illustrates the use case for the view university contact directory functionality (UC022), detailing the process as defined by Requirement REQ_F2201, followed by an activity diagram which represents the process flow.

Use Case ID	UC022	Version	1.0
Use Case	F022 View University Contact Directory		
Purpose	To allow parents to access and view university contact details, including departments and relevant staff.		
Actor	Parent		
Trigger	The parent selects the "Contact Directory" option from the university portal after logging in.		
Precondition	Parent is logged in		
Postcondition	<div>1. The system displays a filtered contact directory containing permitted university contact details.</div> <div>2. Parent may optionally initiate communication through live chat or contact form.</div>		
Scenario Name	Step	Action	
	1	Parent navigates to the contact directory section	
	2	Parent chooses faculty or department to contact	
	3	The system loads the contact directory interface with filters (if any are applied).	
	4	Parent selects a contact entry to view full details.	
	5	Parent can choose to initiate communication via live chat or contact form.	
Alternate Flow – Parent Initiates Live Chat	5.1.1	Parent clicks on “Live Chat” from a contact entry.	
	5.1.2	Proceed to F011 Chat	
Alternate Flow – Parent Initiates Contact Form	5.2.1	Parent clicks on “Contact form” from a contact entry.	
	5.2.2	Proceed to F023 Schedule Meeting with University Staff	
Rules	<div>1. The contact directory interface shall include filters for faculty and department. [REQ_F2201]</div> <div>2. The system shall enable communication between parents and authorized university staff through chat, or a secure contact form embedded within the directory interface. [REQ_F2201, REQ_F2301]</div>		

Notes	<ol style="list-style-type: none"> 1. Faculty refers to educational divisions within the university (e.g. Faculty of Multimedia, Faculty of Engineering) 2. Department refers to the administrative divisions of the university (e.g. Finance, Student Affairs)
Author	Lim Xin Yee

Table 3.1.22: Use Case UC022 View University Contact Directory

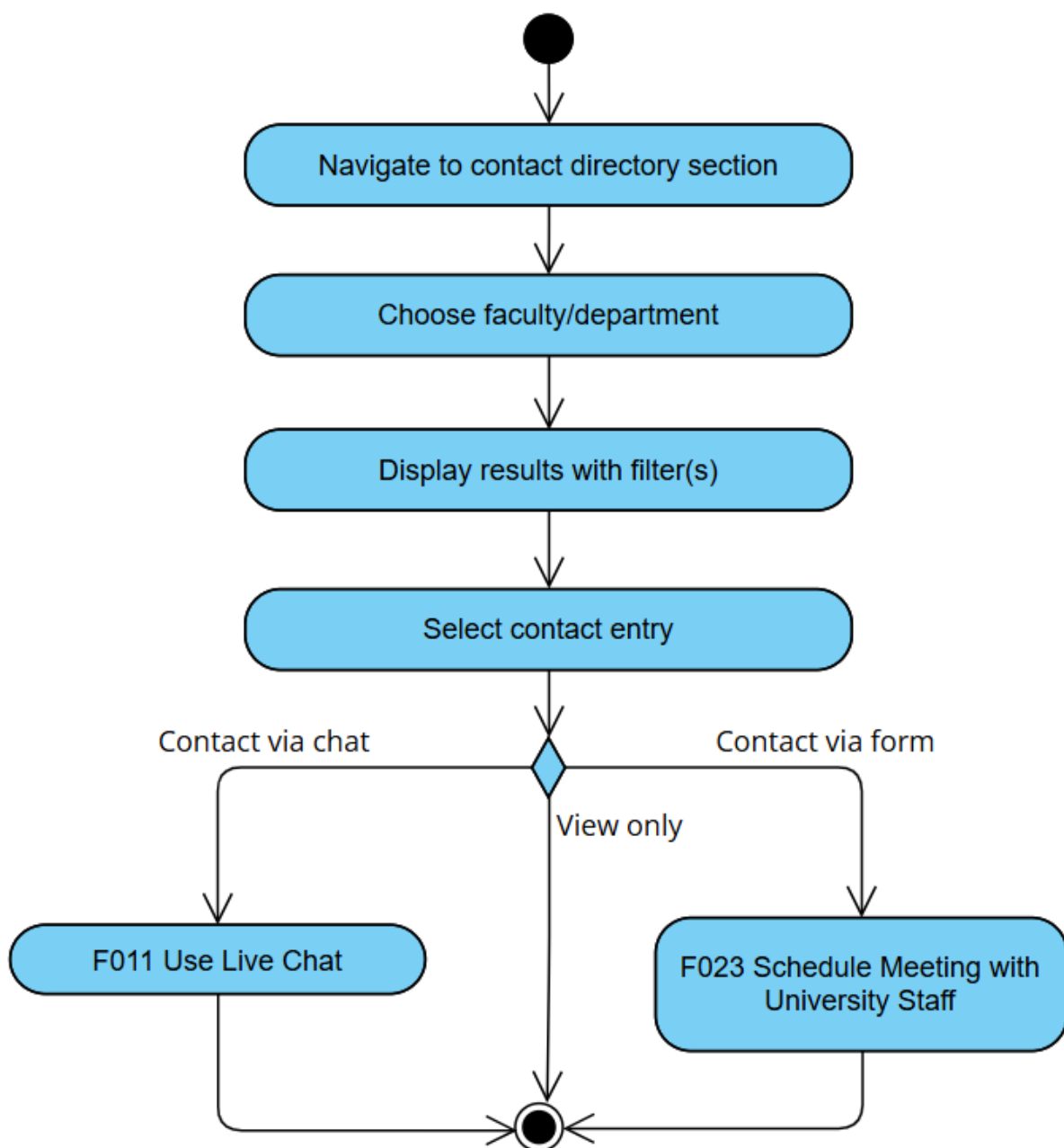


Figure 3.1.22: Activity Diagram for Use Case UC022 View University Contact Directory

3.1.23 F023 Schedule Meeting with University Staff

Requirement ID	REQ_F2301	Version	1.0
Description	The system shall enable communication between university administrators and parents via chat or contact form.		
Author	Lim Xin Yee		

Table 3.1.22 illustrates the use case for the schedule meeting with university staff functionality (UC023), detailing the process as defined by Requirement REQ_F2301, followed by an activity diagram which represents the process flow.

Use Case ID	UC023	Version	1.0
Use Case	F023 Schedule Meeting with University Staff		
Purpose	To allow parents to request and schedule meetings with relevant university staff using a secure contact form integrated within the parent portal.		
Actor	Parent		
Trigger	Parent submits a meeting request through the contact form available on the university portal.		
Precondition	<div>1. Parent is logged in</div> <div>2. The contact form is properly configured and operational.</div>		
Postcondition	<div>1. A meeting request is logged and sent to the selected staff member.</div> <div>2. The requested meeting is scheduled or followed up via further communication.</div>		
Scenario Name	Step	Action	
Main Flow	1	Parent navigates to contact directory section	
	2	Parent selects a university staff member from the contact list.	
	3	Parent fills out the contact form, including preferred meeting date, time, and purpose.	
	4	Parent submits the form.	
	5	System sends the meeting request to the selected staff member.	
	6	The meeting request is sent to the staff member.	
Alternate Flow – Invalid Form Submission	4.1	System detects missing/invalid required fields.	
	4.2	System prompts the parent to complete all necessary fields.	
Rules	<div>1. The contact directory interface shall include filters for faculty and department. [REQ_F2201]</div> <div>2. The system shall enable communication between parents and authorized university staff through chat, or a secure contact form embedded within the directory interface. [REQ_F2201, REQ_F2301]</div>		

	3. System will notify the parent of the meeting status (confirmed, rescheduled, or declined). [REQ_0601]
Author	Lim Xin Yee

Table 3.1.23: Use Case UC023 Schedule Meeting with University Staff

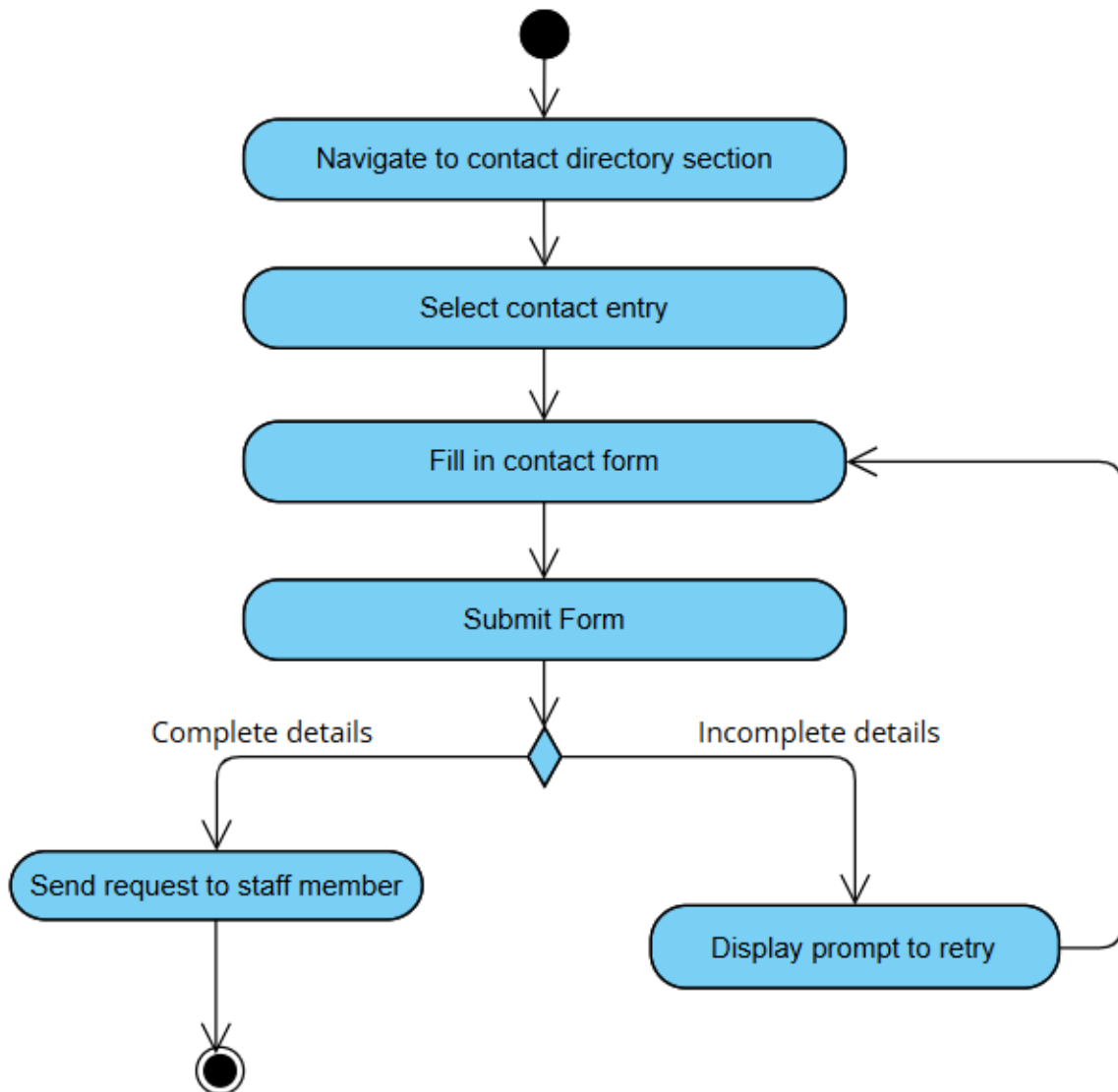


Figure 3.1.23: Activity Diagram for Use Case UC023 Schedule Meeting with University Staff

3.1.24 F024 Manage Academic Resource

The functional requirement(s) for F024 Manage Academic Resource:

Requirement ID	REQ_F2401	Version	1.0
Description	Lecturers shall be able to upload and update materials in a centralized location		
Author	Nickleirsch		

Requirement ID	REQ_F2402	Version	1.0
Description	Lecturers shall be able to generate a single unique link to academic resources usable across platforms.		
Author	Nickleirsch		

Table 3.1.24 illustrates the use case for the manage academic resource functionality (UC024), detailing the process as defined by Requirement REQ_F2401, and REQ_F2402 followed by an activity diagram which represents the process flow.

Use Case ID	UC024	Version	1.0
Use Case	F024 Manage Academic Resource		
Purpose	Enable lecturers to update academic materials in a centralized folder and share a unique link to it.		
Actor	Lecturer		
Trigger	Lecturer selects the option to update course material.		
Precondition	Lecturer is logged in		
Postcondition	Academic resource is successfully uploaded		
Scenario Name	Step	Action	
	1	Lecturer navigates to the course management section.	
	2	Lecturer selects the relevant course.	
	3	Lecturer selects “Update Academic Resource.”	
	4	Lecturer chooses one or more files to upload.	
	5	System checks file type and size	
	6	System stores the material in a centralized location.	
	7	System displays a confirmation of successful upload along with unique link to the resource.	
Alternate Flow – Uploading Unsupported File Type or Size	5.1.1	The lecturer tries to upload an unsupported file format or size	
	5.1.2	The system displays a clear error, prompting a retry.	
Rules	1. Only authenticated lecturers can upload materials. [REQ_F0009] 2. Upload location must be centralized and accessible via a unique link. [REQ_F2401, REQ_F2402]		
Author	Nickleirsch		

Table 3.1.24: Use Case UC024 Manage Academic Resource

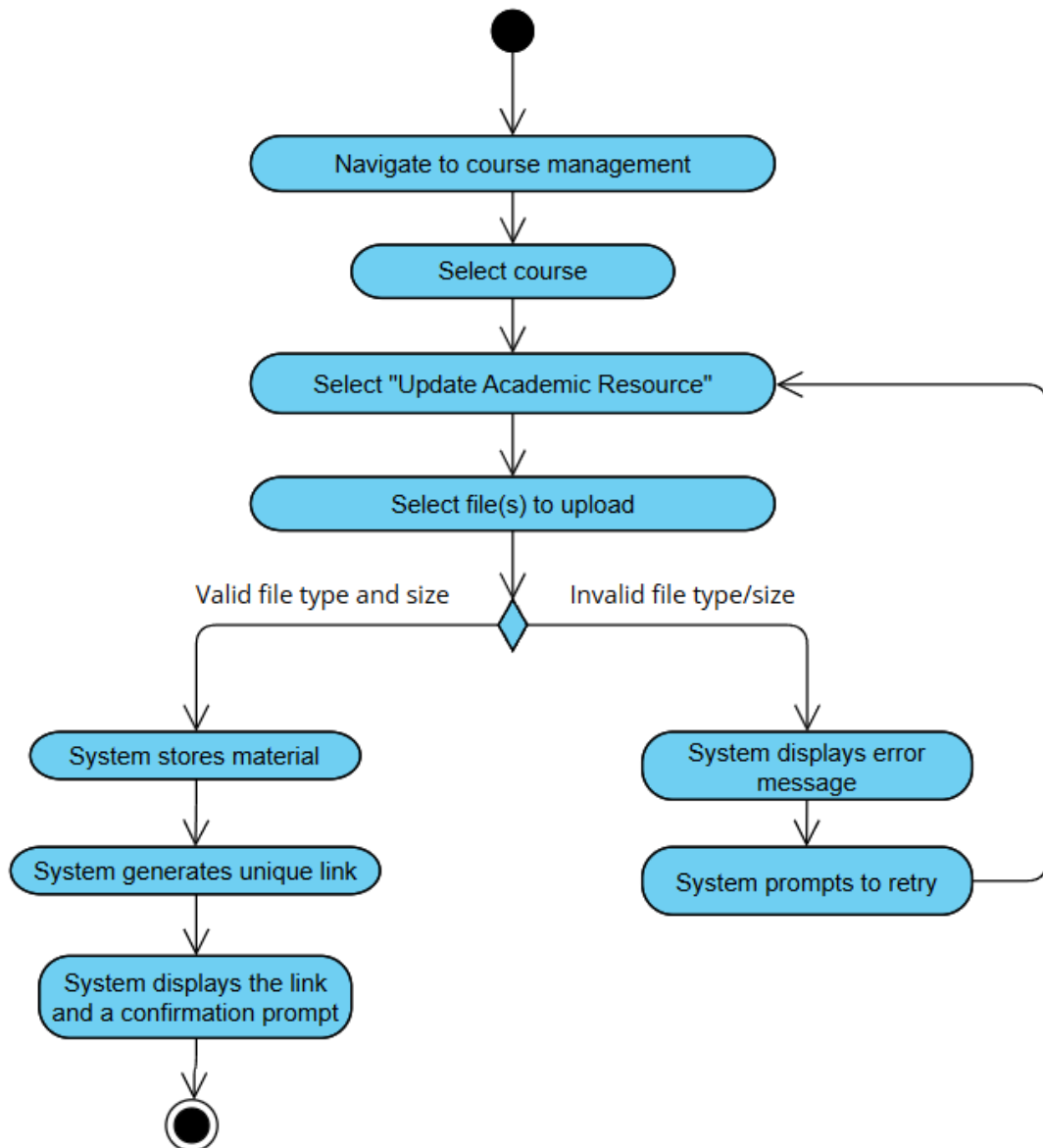


Figure 3.1.24: Activity Diagram for Use Case UC024 Manage Academic Resource

3.1.25 F025 View Announcement Read Status

The functional requirement(s) for F025 View Announcement Read Status:

Requirement ID	REQ_F2501	Version	1.0
Description	The system shall allow lecturers and admin to view the read status of announcements they have made		
Author	Nickleirsch		

Table 3.1.25 illustrates the use case for the view announcement read status functionality (UC025), detailing the process as defined by Requirement REQ_F2501 followed by an activity diagram which represents the process flow.

Use Case ID	UC025	Version	1.0
Use Case	F025 View Announcement Read Status		
Purpose	To allow lecturers to view which students have read a particular announcement.		
Actor	Lecturer		
Trigger	Lecturer wants to check which recipients have read a specific announcement.		
Precondition	Lecturer is logged in		
Postcondition	Lecturer can see which recipients have read or have not read the announcement.		
Scenario Name	Step	Action	
	1	Lecturer navigates to the announcement history.	
	2	Lecturer selects an announcement they have sent.	
	3	System displays a list of recipients with their read status, with real-time updates	
Alternate Flow – No Announcements Exist	1.1	If no announcements have been made yet, the system prompts the user to make an announcement	
Rules	1. Only announcement authors can view read statuses. [REQ_F2501] 2. Read status is updated in real time. [REQ_F0001]		
Notes	Read receipts are only supported for portal announcements		
Author	Nickleirsch		

Table 3.1.25: Use Case UC025 View Announcement Read Status

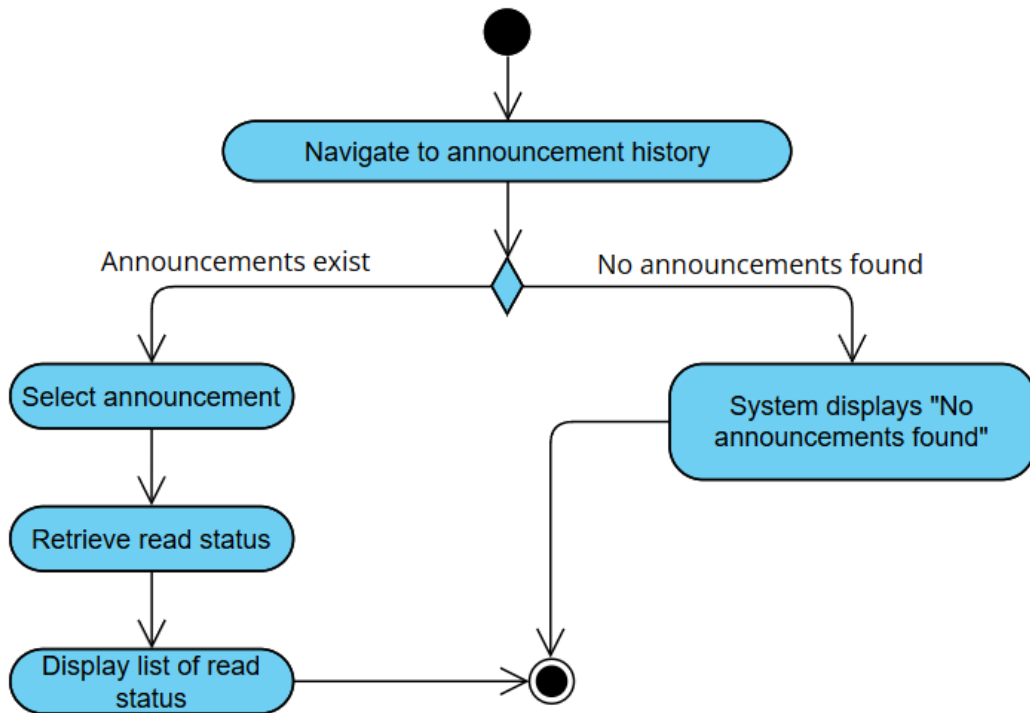


Figure 3.1.25: Activity Diagram for Use Case UC025 View Announcement Read Status

3.1.26 F026 Update Student Academic Data

The functional requirement(s) for F026 Update Student Academic Data:

Requirement ID	REQ_F2601	Version	1.0
Description	The system shall allow exporting and importing student academic data in Excel or CSV format.		
Author	Nickleirsch		

Table 3.1.26 illustrates the use case for the update student academic data functionality (UC026), detailing the process as defined by Requirement REQ_F2601 followed by an activity diagram which represents the process flow.

Use Case ID	UC026	Version	1.0
Use Case	F026 Update Student Academic Data		
Purpose	Allow lecturers to import or export student academic data for analysis or backup.		
Actor	Lecturer		
Trigger	Lecturer selects import or export option.		
Precondition	Lecture is logged in		
Postcondition	Lecturer can see which recipients have read or have not read the announcement.		
Scenario Name	Step	Action	
	1	Lecturer accesses “Academic Data” section.	
	2	Lecturer chooses to import	
	3	Lecturer selects file for upload	
	4	System validates file format and content.	
	5	System applies changes and displays confirmation prompt.	
Alternate Flow – Export Data	2.1.1	Lecturer chooses to export.	
	2.1.2	The system retrieves the file in selected format and provides download link	
Alternate Flow – Invalid File Format	4.1	The lecturer has uploaded an invalid file	
	4.2	The lecturer is prompted to retry	
Rules	The system shall verify the file type before accepting uploads [REQ_F0007]		
Author	Nickleirsch		

Table 3.1.26: Use Case UC026 Update Student Academic Data

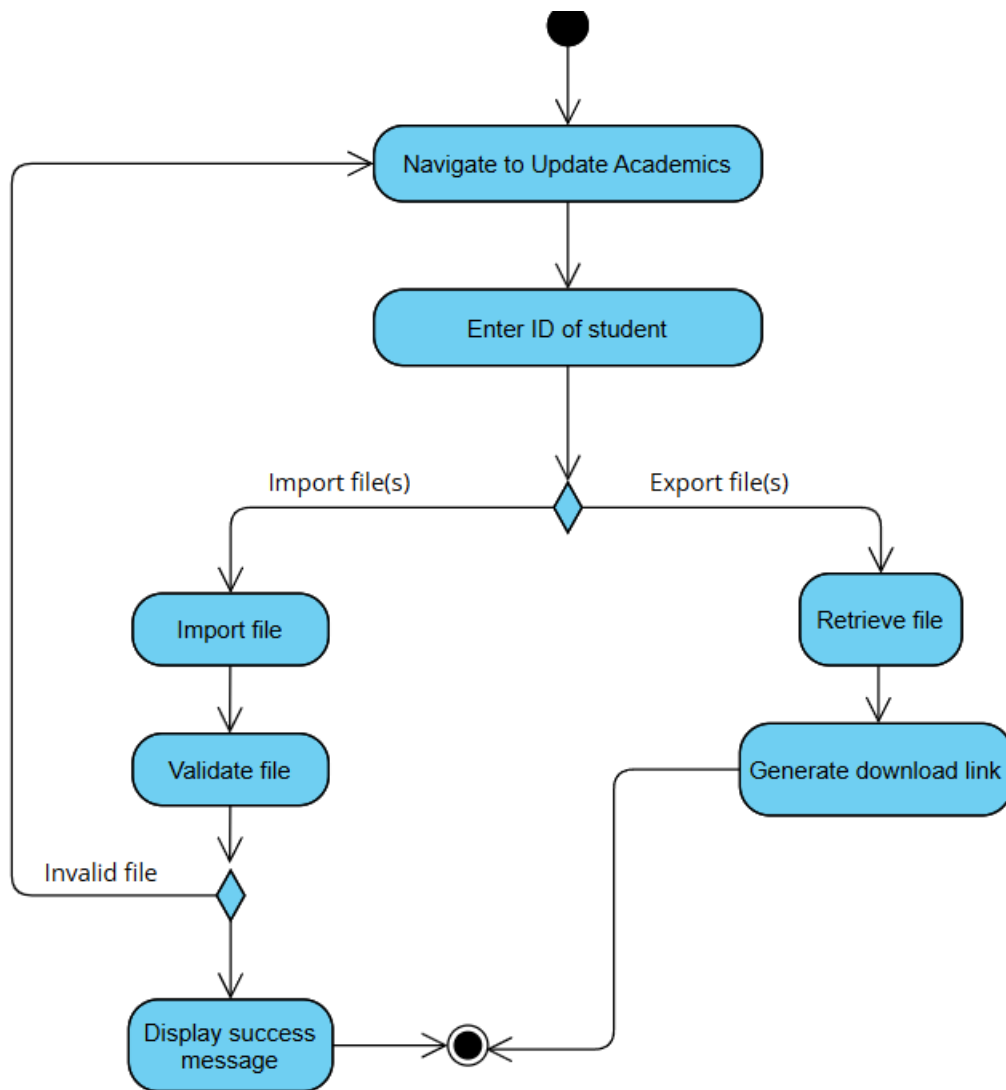


Figure 3.1.26: Activity Diagram for Use Case UC026 Update Student Academic Data

3.1.27 F027 Manage Communication Template

The functional requirement(s) for F026 Manage Communication Template

Requirement ID	REQ_F2701	Version	1.0
Description	The system shall allow creation, customization, and management of communication templates by admins.		
Author	Danesh Veran		

Table 3.1.27 illustrates the use case for the manage communication templates functionality (UC027), detailing the process as defined by Requirement REQ_F2701 followed by an activity diagram which represents the process flow.

Use Case ID	UC027	Version	1.0
Use Case	F027 Manage Communication Template		
Purpose	Allow administrators to create, modify, view, and delete reusable communication templates for SMS, email, and portal notifications		
Actor	Admin		
Trigger	Admin navigates to communication template management section		
Precondition	Admin is logged in		
Postcondition	Communication template is created, updated, or deleted. Changes are logged.		
Scenario Name	Step	Action	
Main Flow	1	Admin navigates to communication template management section	
	2	System displays existing templates and options (Create, Edit, Delete)	
	3	Admin selects create new template	
	4	System presents a form for template details (Name, Type [SMS/Email/Portal], Subject [if applicable], Body content with placeholders)	
	5	Admin enters template details and content	
	6	Admin saves the template	
	7	System validates and stores the new template, making it available for use	
Alternate Flow – Modify Template	2.1	Admin selects an existing template and chooses to edit	
	2.2	System loads the template details for modification	
	2.3	Admin modifies the template content or details and saves	
Alternate Flow – Delete Template	2.1	Admin selects an existing template and chooses to delete	
	2.2	System prompts for confirmation.	

	2.3	Admin confirms. System removes the template.
Rules	1. The system shall allow creation, customization, and management of communication templates for admins. [REQ_F2701] 2. Templates can be categorized or tagged by administrators for better organization and retrieval. [REQ_F2701]	
Author	Danesh Veran	

Table 3.1.27: Use Case UC027 Manage Communication Template

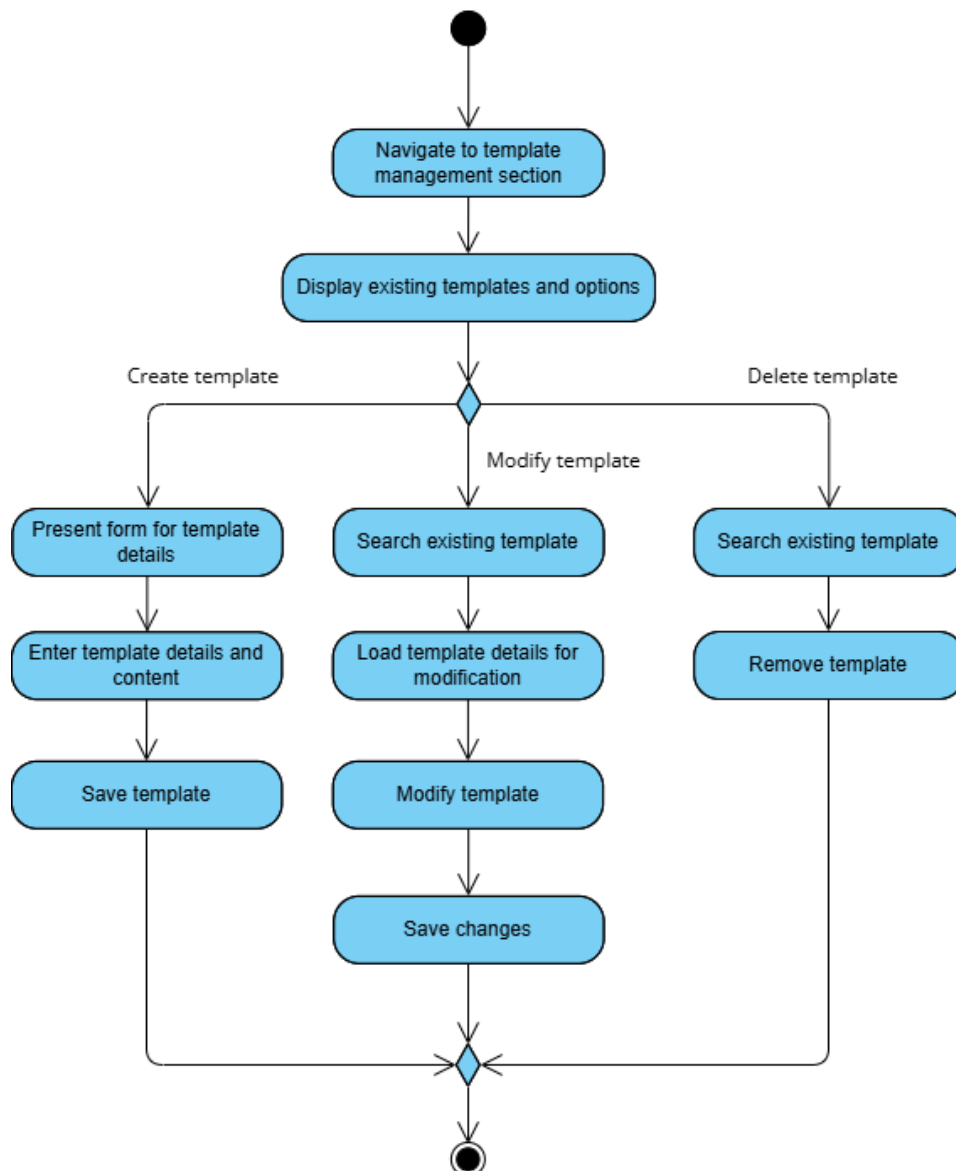


Figure 3.1.27: Activity Diagram for Use Case UC027 Manage Communication Template

3.1.28 F028 Manage University Contact Directory

The functional requirement(s) for F028 Manage University Contact Directory:

Requirement ID	REQ_F2201	Version	1.0
Description	The system shall include a contact directory interface, maintained by admins, for parents to access with filters for faculty and department		
Author	Danesh Veran		

Table 3.1.28 illustrates the use case for the manage university contact directory functionality (UC028), detailing the process as defined by Requirement REQ_F2801 followed by an activity diagram which represents the process flow.

Use Case ID	UC028	Version	1.0
Use Case	F028 Manage University Contact Directory		
Purpose	Allow administrators to create, update, and manage entries in the university-wide contact directory accessible to relevant stakeholders.		
Actor	Admin		
Trigger	Admin navigates to contact directory section		
Precondition	Admin is logged in		
Postcondition	Contact directory is updated with new information		
Scenario Name	Step	Action	
Main Flow	1	Admin navigates to “Contact Directory Management”	
	2	System displays existing directory structure and entries with options (Add, Edit, Delete)	
	3	Admin selects add a new entry	
	4	System presents a form for contact details (Name, Department, Role, Email, Phone, Office Hours, communication channels available)	
	5	Admin enters the required information	
	6	Admin saves the new entry	
	7	System validates and adds the entry to the directory	
Alternate Flow – Modify Entry	3.1	Admin selects an existing entry and chooses to edit	
	3.2	System loads the entry details for modification	
	3.3	Admin modifies the details and saves	
Alternate Flow – Delete Entry	4.1	Admin selects an entry and chooses to delete	
	4.2	System prompts for confirmation.	
	4.3	Admin confirms. System removes the entry.	
Rules	1. Filters (department, role) must be configurable for the directory display [REQ_F2201] 2. Contact information must use descriptive names [REQ_I0003]		

	3. The system shall enable communication between parents and authorized university staff through chat, or a secure contact form embedded within the directory interface, where applicable. [REQ_F2201]
Author	Danesh Veran

Table 3.1.28: Use Case UC028 Manage University Contact Directory

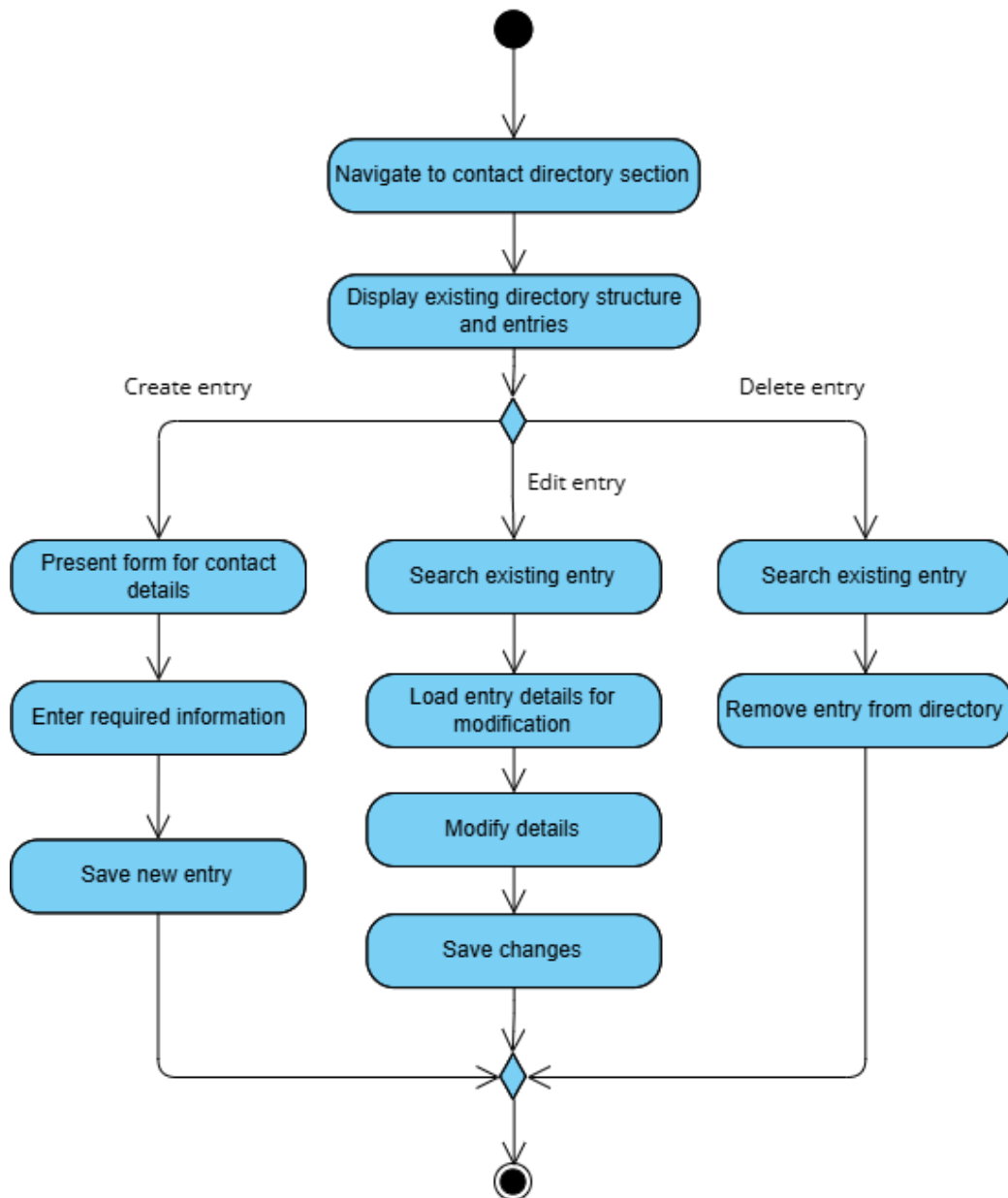


Figure 3.1.28: Activity Diagram for Use Case UC028 Manage University Contact Directory

3.1.29 F029 View System Audit Log

The functional requirement(s) for F029 View System Audit Log:

Requirement ID	REQ_F2901	Version	1.0
Description	The system shall maintain an audit log that records all user and system activities, including but not limited to logins, data modifications, access to sensitive records, and administrative actions.		
Author	Nickleirsch		

Requirement ID	REQ_F2902	Version	1.0
Description	Each audit log entry shall include the timestamp, user ID, action performed, and affected resources.		
Author	Nickleirsch		

Table 3.1.29 illustrates the use case for the view system audit log functionality (UC029), detailing the process as defined by Requirement REQ_F2901 followed by an activity diagram which represents the process flow.

Use Case ID	UC029	Version	1.0
Use Case	F029 View System Audit Log		
Purpose	Allow administrators to review system activity logs for security, troubleshooting, and compliance purposes.		
Actor	Admin		
Trigger	Admin navigates to audit logs section		
Precondition	Admin is logged in		
Postcondition	Admin has viewed relevant audit log entries		
Scenario Name	Step	Action	
Main Flow	1	Admin navigates to audit log section	
	2	System displays options to filter logs	
	3	Admin enters query	
	4	System retrieves and displays matching audit log entries (e.g., timestamp, user, action, details)	
	5	Admin reviews the log entries	
	6	(Optional) Admin exports selected log entries	
Alternate Flow – No Matching Logs	4.1	If no logs match the filter criteria, system displays "No matching entries found."	
Rules	1. Access to audit logs must be restricted to authorized administrative personnel [REQ_F0009] 2. Sensitive information within logs must be appropriately masked or access controlled [REQ_F0003]		
Author	Danesh Veran		

Table 3.1.29: Use Case UC029 View System Audit Log

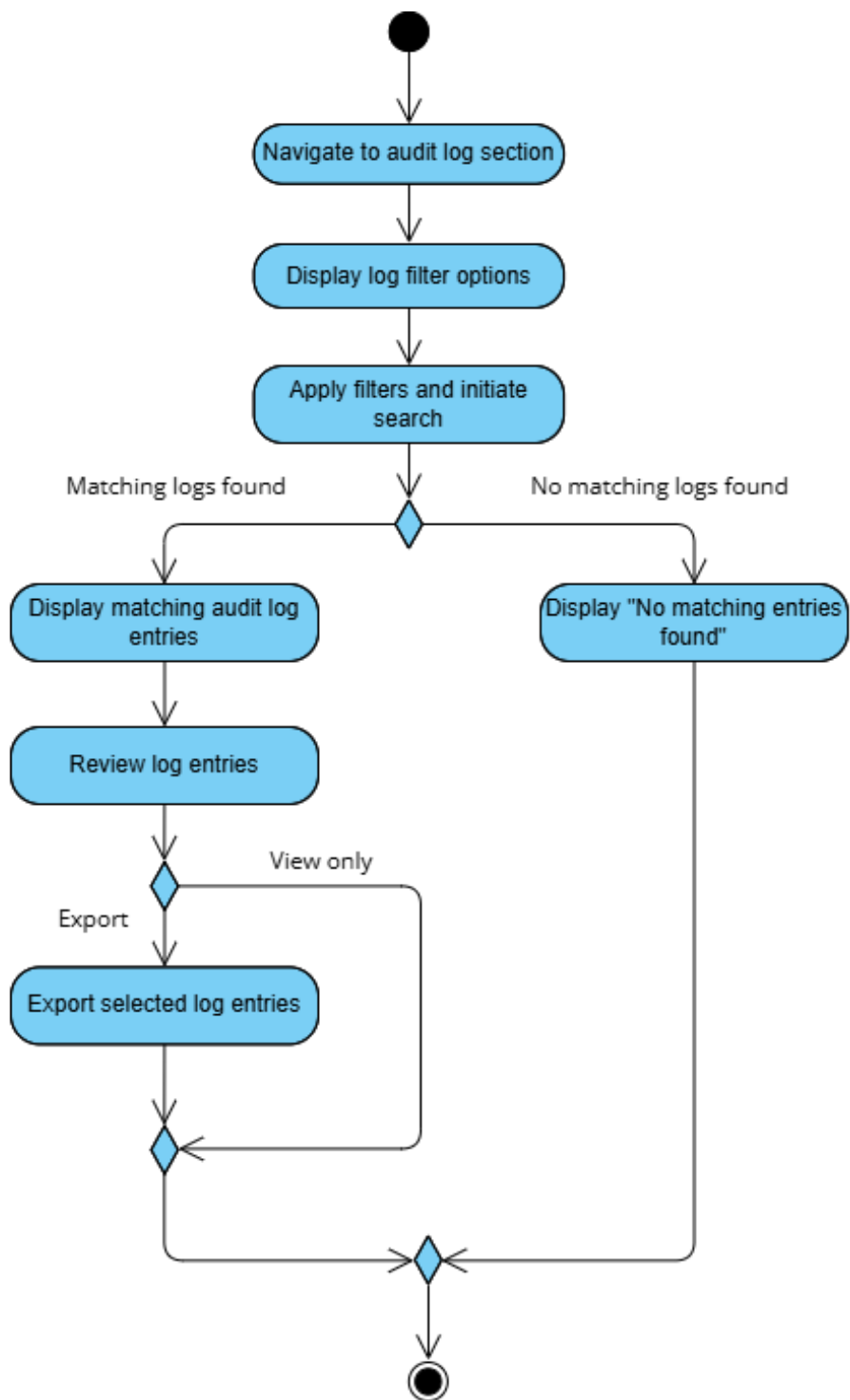


Figure 3.1.29: Activity Diagram for Use Case UC029 View System Audit Log

3.1.30 F030 Configure Parent Access

The functional requirement(s) for F030 Configure Parent Access:

Requirement ID	REQ_F3001	Version	1.0
Description	Parental access and notifications shall comply with university privacy policies and require explicit student consent.		
Author	Danesh Veran		

Table 3.1.30 illustrates the use case for the configure parent access functionality (UC030), detailing the process as defined by Requirement REQ_F3001 followed by an activity diagram which represents the process flow.

Use Case ID	UC030	Version	1.0
Use Case	F030 Configure Parent Access		
Purpose	Allow administrators to manage university-level settings for parental access to student information, including default consent mechanisms and information visibility rules.		
Actor	Admin		
Trigger	Admin navigates to parental access section		
Precondition	Admin is logged in		
Postcondition	<div>1. System-wide settings for parent access and student consent are updated.</div> <div>2. Changes are logged.</div>		
Scenario Name	Step	Action	
Main Flow	1	Admin navigates to parental access section	
	2	System displays current configurations for parental access	
	3	Admin modifies parental access to view child’s information	
	4	Admin saves the configuration changes	
Rules	<div>1. All configurations must comply with university privacy policies and explicit student consent requirements [REQ_F0002]</div> <div>2. The system shall provide a dedicated portal for parents to access their child’s grades, attendance, and financial information, subject to consent. [REQ_F3001, REQ_F2101]</div>		
Notes	The student’s consent must be a physical letter that is sent in by the student personally		
Author	Danesh Veran		

Table 3.1.30: Use Case UC030 Configure Parent Access

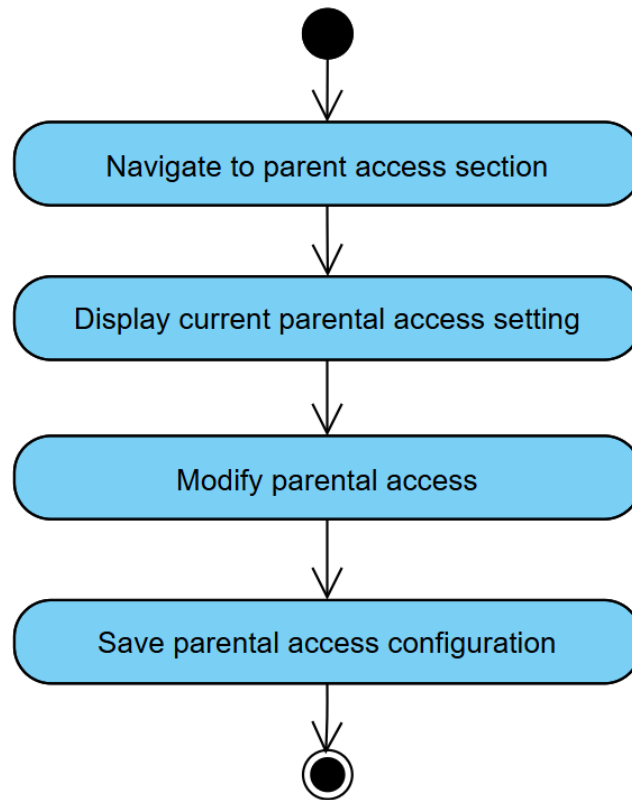


Figure 3.1.30: Activity Diagram for Use Case UC030 Configure Parent Access

3.1.31 F031 Authenticate User

The functional requirement(s) for F031 Authenticate User:

Requirement ID	REQ_F3101	Version	1.0
Description	The system shall support single sign-on authentication for accessing all university services.		
Author	Lim Xin Yee		

Table 3.1.31 illustrates the use case for the authenticate user functionality (UC031), detailing the process as defined by Requirement REQ_F3101 followed by an activity diagram which represents the process flow.

Use Case ID	UC031	Version	1.0
Use Case	F031 Authenticate User		
Purpose	To ensure that only authorized users can securely access the university portal and its associated services by verifying their credentials through a centralized authentication mechanism		
Actor	Campus Management System		
Trigger	User credentials are received by the authentication service		
Precondition	The Campus Management System is up and running		
Postcondition	<div>1. The user is granted access to the portal with permissions appropriate to their role.</div> <div>2. A secure session is initiated with session timeout policies applied.</div>		
Scenario Name	Step	Action	
Main Flow	1	System receives authentication credentials from the interface	
	2	System transmits encrypted credentials to for validation	
	3	System receives valid credential response	
	4	System creates encrypted session token	
	5	System returns session token with access permissions	
Alternate Flow – Invalid Credentials	3.1	System receives invalid credential response	
	3.2	The user is prompted to try again.	
	3.3	Redirect to login	
Rules	<div>1. Role-based access control must be enforced upon authentication. [REQ_F0009]</div> <div>2. Authentication must use university SSO system. [REQ_F3101]</div> <div>3. Authentication process must comply with FERPA, GDPR, and university privacy policies. [REQ_F0002]</div> <div>4. All credentials and session data must be encrypted in transit and at rest. [REQ_F0003]</div> <div>5. Logging of authentication events [REQ_F2901]</div>		

Table 3.1.31: Use Case UC031 Authenticate User

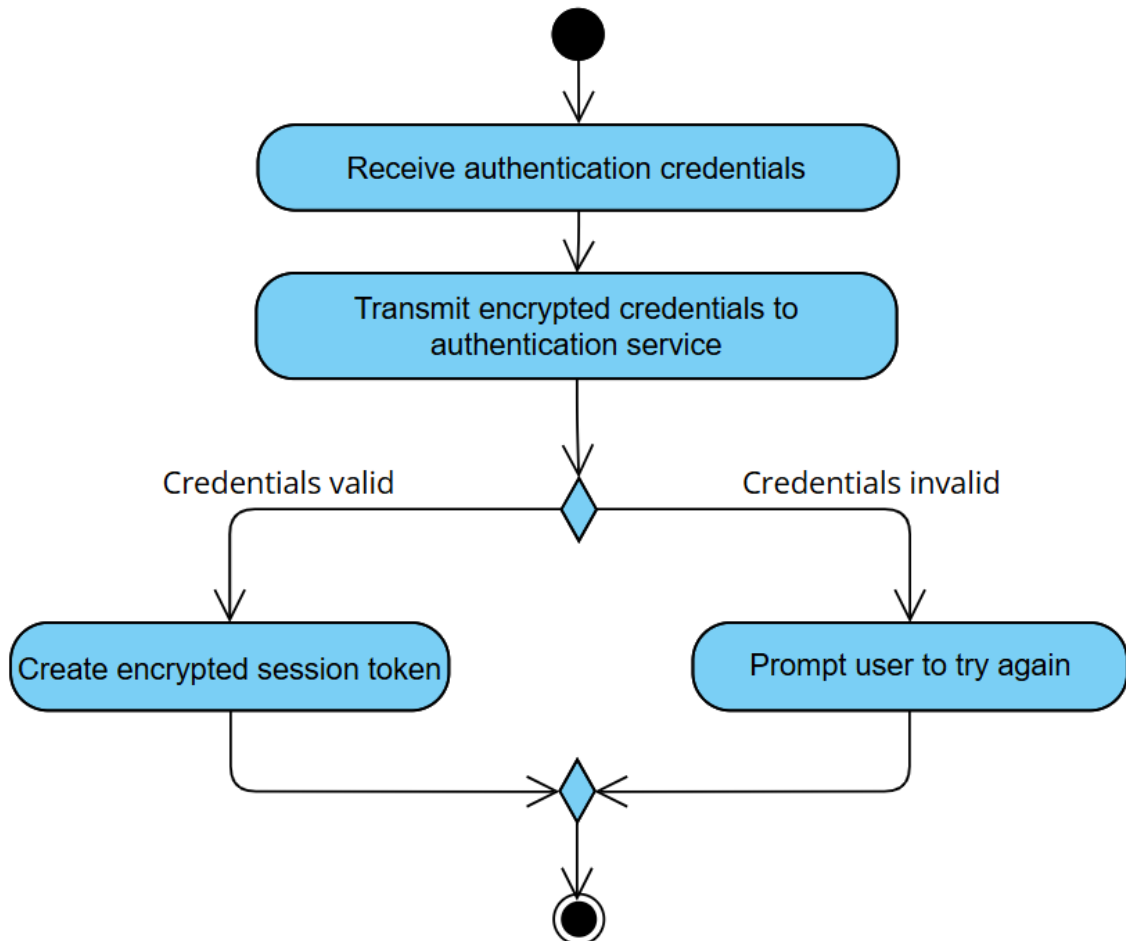


Figure 3.1.31: Activity Diagram for Use Case UC031 Authenticate User

3.1.32 F032 Send SMS Notification

The functional requirement(s) for F032 Send SMS Notification:

Requirement ID	REQ_F3201	Version	1.0
Description	The system shall support sending notifications via SMS to students and parents where mobile numbers are available, based on channel preference.		
Author	Nickleirsch		

Requirement ID	REQ_F3202	Version	1.0
Description	The system shall provide urgent/critical alerts (low attendance, overdue fees) via SMS		
Author	Nickleirsch		

Table 3.1.32 illustrates the use case for the send SMS notification functionality (UC032), detailing the process as defined by Requirement REQ_F3201 and REQ_F3202 followed by an activity diagram which represents the process flow.

Use Case ID	UC032	Version	1.0
Use Case	F032 Send SMS Notification		
Purpose	Deliver urgent or scheduled SMS notifications to users as directed by the university system.		
Actor	SMS Gateway		
Trigger	SMS Gateway receives a request from the university portal to send an SMS notification.		
Precondition	<div><div>1.</div><div>University portal has validated the message, recipient(s), and preferences.</div></div> <div><div>2.</div><div>SMS Gateway is operational and authenticated.</div></div>		
Postcondition	<div><div>1.</div><div>SMS is delivered to intended recipients, with delivery status communicated back to the university portal.</div></div>		
Scenario Name	Step	Action	
Main Flow	1	SMS Gateway receives a notification payload (recipient, message, metadata)	
	2	SMS Gateway validates payload integrity.	
	3	SMS Gateway attempts delivery to the recipient(s).	
	4	SMS Gateway receives delivery status from carrier.	
	5	SMS Gateway logs the status and notifies the university portal.	
Alternate Flow – Delivery Fails	3.1	SMS Gateway logs the failure and notifies the portal.	
Rules	<div><div>1.</div><div>Urgent/critical alerts must be delivered within 1 minute [REQ_P0004]</div></div> <div><div>2.</div><div>SMS Gateway must respect recipient opt-in/out [REQ_F1903]</div></div>		

	3. All SMS content must comply with privacy and consent regulations [REQ_F0002, REQ_F3001] 4. Templates and scheduling must be supported [REQ_F0902, REQ_F0904, REQ_F2701]
Author	Nickleirsch

Table 3.1.32: Use Case UC032 Send SMS Notification

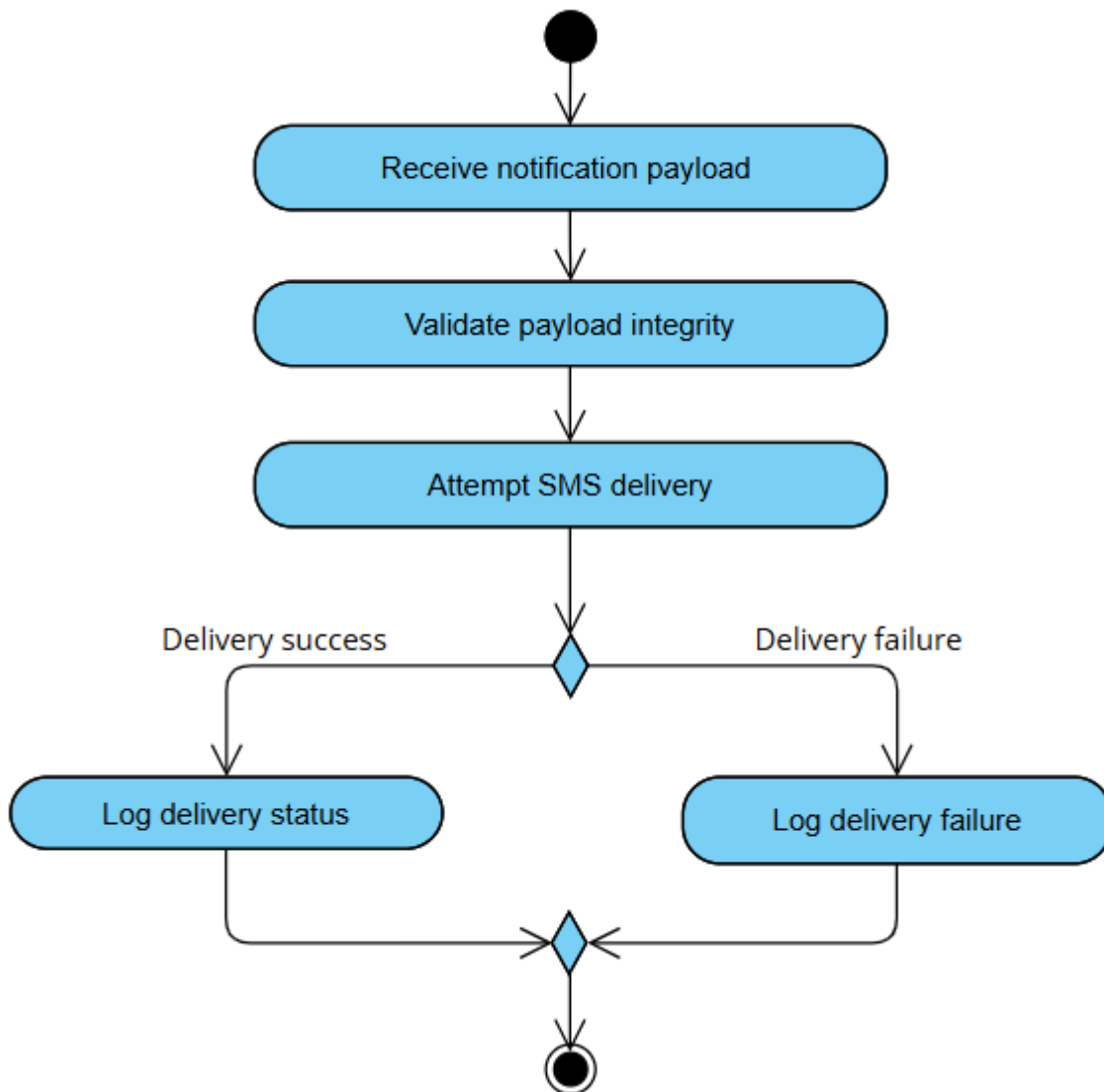


Figure 3.1.32: Activity Diagram for Use Case UC032 Send SMS Notification

3.1.33 F033 Sync with External Calendar

The functional requirement(s) for F033 Sync with External Calendar:

Requirement ID	REQ_F3301	Version	1.0
Description	The system shall support integration with common calendar applications (Google Calendar, Apple Calendar) for academic schedules.		
Author	Nickleirsch		

Table 3.1.33 illustrates the use case for the sync with external calendar functionality (UC033), detailing the process as defined by Requirement REQ_F3301 followed by an activity diagram which represents the process flow.

Use Case ID	UC033	Version	1.0
Use Case	F033 Sync with External Calendar		
Purpose	Enable synchronization of academic schedules and events between the university system and external calendar applications (e.g., Google Calendar, Apple Calendar).		
Actor	Calendar API		
Trigger	The user selects "Sync Calendar" in their calendar settings		
Precondition	<div>1. Calendar API is authenticated with external calendar provider.</div> <div>2. The user has a valid account with an external calendar provider.</div>		
Postcondition	<div>1. Academic schedules and events are synchronized with the selected external calendar application.</div> <div>2. Any subsequent changes in the university calendar are updated in the user's external calendar.</div>		
Scenario Name	Step	Action	
Main Flow	1	Calendar API receives a sync trigger	
	2	Calendar API requests updated academic events (.ics file).	
	3	The system transmits the user's academic schedule and events to the calendar API	
	4	Calendar API updates events in the external calendar	
	5	The system confirms the successful sync	
Alternate Flow – Synchronization Fails	4.1.1	The synchronization fails due to network or API errors	
	4.1.2	The system logs the error for later retrial	
Rules	<div>1. Sync must occur within 2 minutes of any calendar change. [REQ_P0005]</div> <div>2. The system must support integration with at least Google Calendar and Apple Calendar. [REQ_F3301]</div>		

	3. Sync failures and actions are logged [REQ_F2901, REQ_F2902]
Author	Nickleirsch

Table 3.1.33: Use Case UC033 Sync with External Calendar

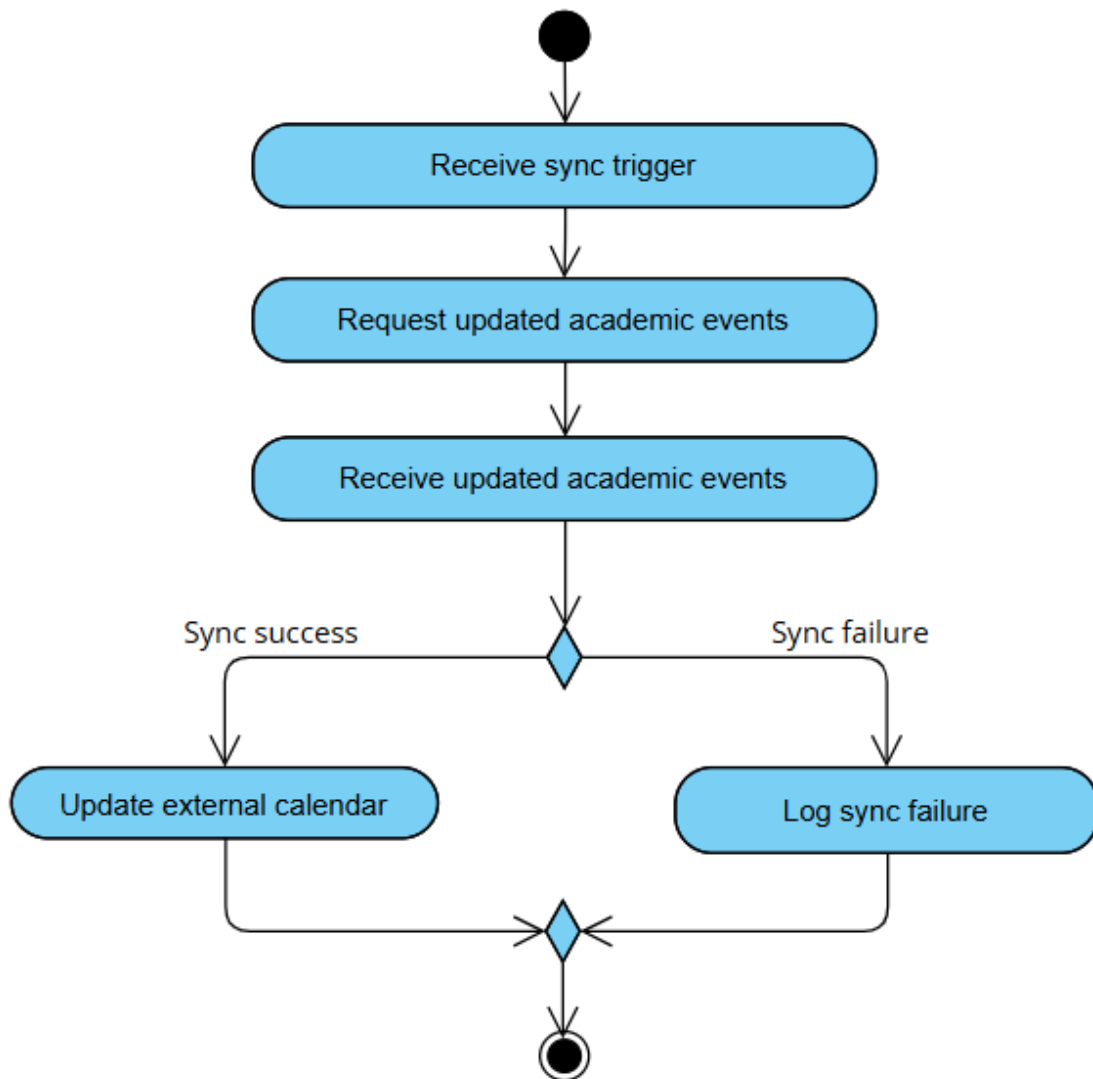


Figure 3.1.33: Activity Diagram for Use Case UC033 Sync with External Calendar

3.2 Performance Requirements

The following the performance requirements for COMSYS:

Requirement ID	REQ_P0001	Version	1.0
Description	The system shall load any page within 3 seconds under normal conditions.		
Author	Nickleirsch		

Requirement ID	REQ_P0002	Version	1.0
Description	The system shall process course enrolment requests within 5 seconds.		
Author	Nickleirsch		

Requirement ID	REQ_P0003	Version	1.0
Description	The system shall synchronize updated data across interfaces within 5 seconds.		
Author	Nickleirsch		

Requirement ID	REQ_P0004	Version	1.0
Description	Critical notifications shall be delivered within 1 minute of their creation.		
Author	Nickleirsch		

Requirement ID	REQ_P0005	Version	1.0
Description	Calendar synchronization shall occur within 2 minutes of changes being made.		
Author	Nickleirsch		

Requirement ID	REQ_P0006	Version	1.0
Description	Notifications shall be guaranteed to reach all selected channels without loss, with 99% reliability for scheduled/automated notifications.		
Author	Nickleirsch		

Requirement ID	REQ_P0007	Version	1.0
Description	The system shall maintain 99.9% uptime during academic terms and 99% during breaks and holidays.		
Author	Nickleirsch		

3.3 Usability Requirements

The following the usability requirements for COMSYS:

Requirement ID	REQ_U0001	Version	1.0
Description	The system shall limit navigation depth to maximum five levels for any feature.		
Author	Nickleirsch		

Requirement ID	REQ_U0002	Version	1.0
Description	The system must meet WCAG 2.1 guidelines for accessibility, ensuring usability for all users, including those with disabilities.		
Author	Nickleirsch		

Requirement ID	REQ_U0003	Version	1.0
Description	The system shall provide context-sensitive help or tooltips for at least 90% of user interface elements.		
Author	Nickleirsch		

3.4 Interface Requirements

The following the overall interface requirements for COMSYS:

Requirement ID	REQ_I0001	Version	1.0
Description	The system shall provide consistent header formatting across all tables and views.		
Author	Nickleirsch		

Requirement ID	REQ_I0002	Version	1.0
Description	The interface shall be responsive and adapt to different screen sizes.		
Author	Nickleirsch		

Requirement ID	REQ_I0003	Version	1.0
Description	The system shall use descriptive course/event names, not codes, throughout the interface.		
Author	Nickleirsch		

Requirement ID	REQ_I0004	Version	1.0
Description	Academic, financial, and attendance data shall be presented with charts, tables, and summaries for quick understanding.		
Author	Nickleirsch		

Requirement ID	REQ_I0005	Version	1.0
Description	The system shall provide visual indicators for navigation paths.		
Author	Nickleirsch		

Requirement ID	REQ_I0006	Version	1.0
Description	Academic, financial, and attendance data shall be presented with charts, tables, and summaries for quick understanding.		
Author	Nickleirsch		

Requirement ID	REQ_I0007	Version	1.0
Description	The system shall provide a single-window course enrolment process		
Author	Nickleirsch		

3.4.1 System Interfaces

The following system interfaces represent the key integration points through which COMSYS interacts with systems and services to deliver its capabilities:

1. Campus Management System

Purpose: Synchronizes and retrieves information such as academic, billing, schedule data and authentication system. [F012-F016, F021]

Interface: Communication with the CMS occurs via secured RESTful API endpoints, using JSON as the standard data exchange format. The portal both queries the CMS for information (e.g., student grades, attendance, billing) and sends updates (e.g., course registrations, academic records) as needed.

Functionality: Enables up-to-date student data, supports dashboard content, manages course registration, grades and attendance. This is achieved by probing the system for information and sending information to be updated.

2. Calendar Applications (Google Calendar, Apple Calendar):

Purpose: Synchronizes academic schedules and user reminders. [F010]

Interface: REST API

Functionality: The system will regularly sync and request updates to reflect with the external calendars. Allows users to sync academic events and deadlines with personal calendars.

3. Single Sign-On Authentication System

Purpose: Provides unified authentication and secure access to the portal. [F031]

Interface: REST API

Functionality: Enables secure user authentication, automatic redirection to role-specific dashboards, session timeout handling, and proper termination of sessions. This interface ensures that only authorized users can access sensitive academic and administrative data.

3.4.2 User interfaces

The COMSYS platform will provide a unified web portal with a responsive, accessible design, delivering tailored experiences for each user role via dedicated dashboards and intuitive interaction elements.

General Web Portal Features

1. **Responsive Design:**
The interface will automatically adapt to various screen sizes and devices (desktop, tablet, mobile) to ensure usability for all users.
2. **WCAG 2.1 Compliance:**
All interface components, including navigation, forms, and content, will follow WCAG 2.1 guidelines to ensure accessibility for users with disabilities.
3. **Fixed Top Navigation Bar:**
Provides quick access to primary features (e.g., dashboard, messages, resources, settings) and persists across all pages.
4. **Consistent Layout:**
All pages will use a consistent structure with clear headings, logical grouping of related functions, and standardized buttons and icons.
5. **Role-Based Dashboards:**
Upon login, users are directed to dashboards tailored to their roles (Student, Parent, Lecturer, Admin), displaying relevant information and actions.

Student Portal

1. **Customizable Dashboard:**
Students can personalize their dashboard to display key academic information (grades, timetable, notifications, financial status).
2. **Quick Access Widgets:**
Tiles/buttons for common actions such as course enrollment, grade review, and messaging.
3. **Navigation Panel:**
Collapsible side or top menu for accessing modules like academic records, resource library, and support.
4. **Data Entry Fields:**
Clear forms for updating personal information, submitting requests, and uploading documents.

Parent Portal

1. **Controlled Access:**
Parents view authorized student data (academic progress, attendance, notifications) based on role permissions and privacy settings.
2. **Communication Tools:**
Buttons to initiate messages with lecturers or administrators.
3. **Information Panels:**
Read-only panels summarizing student status, announcements, and alerts.

Lecturer Portal

1. **Course Management Dashboard:**
Overview of teaching schedules, course rosters, and grading tasks.
2. **Interactive Gradebook:**
Data entry fields for grades and attendance, with validation to prevent errors.
3. **Messaging and Announcements:**
Quick links to send messages or notifications to students and parents.
4. **Resource Uploads:**
Drag-and-drop and file picker for uploading materials and assignments.

Admin Portal

1. **Comprehensive Control Panel:**
Access to user management, system settings, analytics, and audit logs.
2. **Bulk Operations:**
Buttons and selection tools for managing multiple records (e.g., user accounts, notifications) efficiently.
3. **Search and Filter:**
Search bars and filter options for all data tables.
4. **Real-time Monitoring:**
Dashboard widgets showing system status, recent activity, and alerts.

3.4.3 Software interfaces

The following software interfaces represent the software interfaces which COMSYS interacts with:

3.4.3.1 Operating Systems

1. Microsoft Windows

Name	Microsoft Windows
Mnemonic	Win
Version	Current supported versions
Source	Microsoft
Purpose	Supported environment for portal administrative tools and desktop user clients; ensures compatibility with institutional PCs

2. macOS

Name	macOS
Mnemonic	macOS
Version	Current supported versions
Source	Apple
Purpose	Supported platform for the portal's macOS desktop interface; ensures compatibility with Mac environments used by staff and faculty.

3. GNU/Linux

Name	GNU/Linux
Mnemonic	Linux
Version	Current supported versions
Source	GNU/Linux foundation
Purpose	Platform for server-side components or Linux-based desktop use.

4. SMS Gateway

Name	SMS Gateway
Mnemonic	SMS_Gateway
Version	Current supported versions
Source	GNU/Linux foundation
Purpose	Send critical and regular notifications to users' mobile devices
Message Format	SMPP Submit_SM PDUs with fields: source_addr (sender ID), dest_addr (phone number), short_message (up to 160-char text). Delivery receipts via Submit_SM_RESP.

3.4.3.2 Client Web Browsers

Message content general guidelines:

HTTP/HTTPS with HTML/CSS/JavaScript content (UTF-8 text); complies with W3C standards.

1. Google Chrome

Name	Google Chrome
Mnemonic	Chrome
Version	Chromium-based browser
Source	Google
Purpose	Primary client browser for accessing the web portals

2. Mozilla Firefox

Name	Mozilla Firefox
Mnemonic	Firefox
Version	Gecko-based browser
Source	Mozilla Foundation
Purpose	Primary client browser for accessing the web portals

3. Safari

Name	Safari
Mnemonic	Safari
Source	Apple
Purpose	Primary client browser for accessing the web portals

3.4.3.3 RESTful APIs:

- i. Used for data exchange with the Campus Management System (retrieving/updating academic, billing, schedule data).
- ii. Used for integration with the SMS Gateway to send notifications.
- iii. Used for syncing with external calendar applications (Google Calendar, Apple Calendar).

3.4.4 Communication interfaces

The following system interfaces represent the key communication interfaces through which COMSYS interacts with systems and services to deliver its capabilities:

No.	Interface	Purpose
1	HTTPS	Used for all browser-based access to ensure secure communication between users (students, parents, lecturers, admins) and the portal. All web and API traffic is encrypted for confidentiality and integrity.
2	SMTP	Used for sending email notifications and alerts to users. Ensures secure delivery of emails via the university's or a third-party email server.
3	WebSocket	Used for real-time communication features such as live chat and push notifications within the portal. Provides bidirectional, low-latency data exchange between server and clients.
4	OAuth2	Used for secure Single Sign-On (SSO) authentication and authorization. Ensures centralized identity management and secure token/session handling.
5	TLS/SSL encryption	All communications are encrypted using TLS/SSL to protect data privacy and prevent unauthorized access

3.5 Logical Database Requirements

Specification of the flow of data and database requirements:

COMSYS operates primarily as an intermediary system, minimizing direct data storage and instead focusing on efficient data retrieval and caching from the existing Campus Management System. While COMSYS maintains its own database for user preferences, notification settings, and communication templates, it fetches core academic data (grades, attendance, billing) in real-time from the Campus Management System through secure APIs. The system should employ a caching mechanism that temporarily stores frequently accessed data to reduce system load and improve response times, with cache invalidation triggered by updates in the source system. User authentication is synchronized with the main campus system, while COMSYS independently manages communication logs, notification preferences, and delivery status tracking. This approach ensures data consistency while adding new communication capabilities without duplicating sensitive academic records.

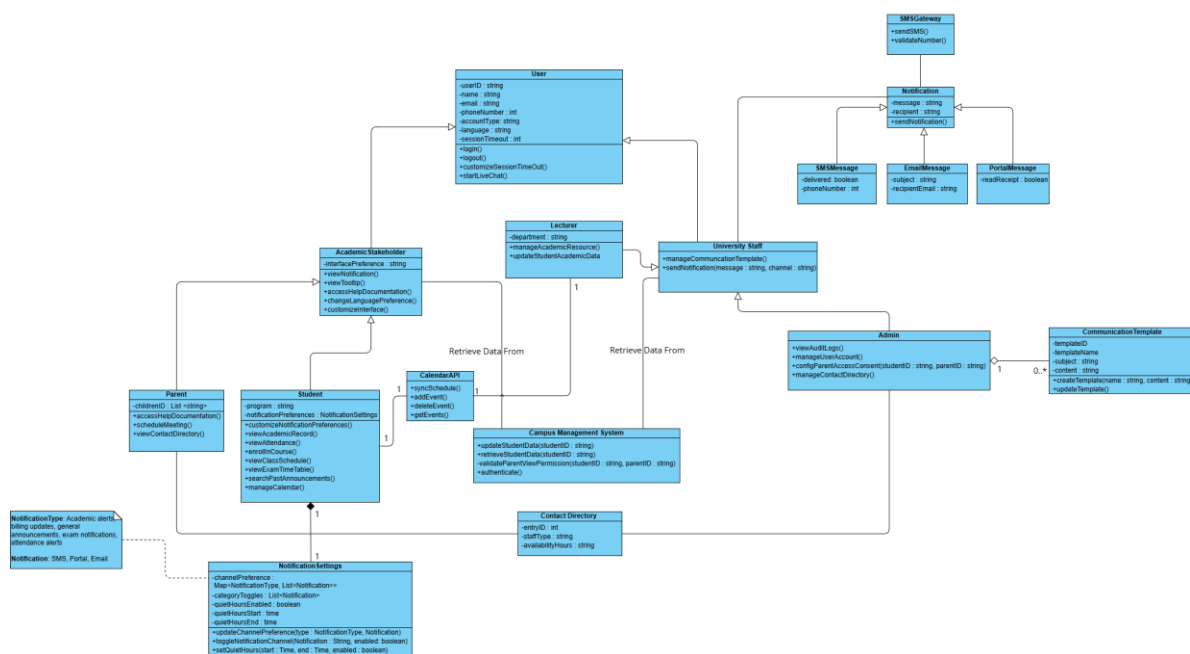


Figure 3.5 Class Diagram; available also at [classdiag.png](#)

In-depth explanation of class diagram:

Class	Attribute/Method	Data Type	Description
User: An abstract base class that defines common attributes and methods for all system users. It handles basic user along with session management and language preferences.	userID	String	Unique identifier for user
	Name	string	User's full name
	email	string	User's email address
	phoneNumber	int	Contact number
	accountType	string	Type of user account
	language	string	Preferred language
	sessionTimeout	int	Session timeout in minutes
NotificationSettings: Manages how users receive notifications by controlling quiet hours, delivery channels (SMS, email, portal), and notification categories. It stores user preferences and determines when and how notifications should be delivered.	quietHoursStart	time	Start of quiet period
	quietHoursEnd	time	End of quiet period
	quietHoursEnabled	boolean	Quiet hours toggle
	categoryToggles	List<Notification Type>	Enabled notification types
	channelPreferences	Map<Notification Type, Channel>	Preferred channels per type
CommunicationTemplate : Stores and manages message templates used for different types of communications. It supports multiple channels (SMS, email, portal) and allows for standardized message creation with customizable content.	templateID	string	Unique template identifier
	templateName	string	Name of template
	subject	string	Email subject line
	content	string	Template content
	templateType	string	Type of template
Notification: Represents a single notification in the	message	string	Notification content

system; an abstract class that is inherited by all notification channel types.	recipient	string	Recipient identifier
	delivered	boolean	Delivery status
	readReceipt	boolean	Read status
ContactDirectory: Stores and manages staff contact information and their availability hours.	entryID	int	Unique contact identifier
	staffType	string	Type of staff member
	availabilityHours	string	Available hours
CalendarAPI: Handles all calendar-related operations including adding, removing, and retrieving events. It manages schedule synchronization and helps coordinate activities across the system.	addEvent()	method	Adds new calendar event
	deleteEvent()	method	Removes calendar event
	getEvents()	method	Retrieves calendar events
SMSGateway: Manages SMS message sending and phone number validation. It tracks message delivery status and ensures proper handling of mobile communications.	sendSMS()	method	Sends SMS message
	validateNumber()	method	Validates phone number
CampusManagementSystem: Manages student data operations and access control. It handles data updates, retrieval, and validates parent permissions for accessing student information.	updateStudentData()	method	Updates the student's data
	retrieveStudentData()	method	Retrieves the student's data
	validateParentViewPermission()	method	Checks if parent has consent to access student information

3.6 Design Constraints

The design of COMSYS is subject to several constraints arising from external standards, regulations, and technical limitations:

1. **Branding and UI Compliance:**
The user interface must comply with the university's official branding guidelines, including colour schemes, logo usage, and typography standards.
2. **Regulatory Compliance:**
All features and data flows must comply with relevant privacy and data protection regulations, including FERPA and GDPR, especially around parental access, data sharing, and consent management.
3. **Integration Requirements:**
The system must integrate with existing university infrastructure, including Single Sign-On (SSO), Academic Database, Financial Database, and external calendar services (Google Calendar, Apple Calendar).
4. **Technology Stack:**
The software should be developed using technologies compatible with both Linux and Windows server environments.
5. **Accessibility Standards:**
The system must meet WCAG 2.1 guidelines for accessibility, ensuring usability for all users, including those with disabilities.
6. **Notification Delivery:**
SMS, email, and push notifications must be routed through approved university and third-party gateways, respecting service limits and anti-spam policies.
7. **Authentication:**
All access must be authenticated via the university's SSO system; no local username/password logins are permitted.

3.7 Software System Attributes

Attribute Category	Requirement	Factors needed	Priority
Reliability	REQ_F3301: Notifications shall be guaranteed to reach all selected channels without duplication or loss, with 99% reliability for scheduled/automated notifications.	<ol style="list-style-type: none"> 1. Implement message queuing system with retry mechanisms 2. Monitor notification delivery rates 3. Regular testing of all notification channels 	High
	REQ_P0003: The system shall synchronize updated data across interfaces within 5 seconds of the change being committed.	<ol style="list-style-type: none"> 1. Implement real-time data synchronization protocols 2. Create efficient database indexing strategy 3. Use optimized query caching 4. Implement event-driven architecture for updates 5. Regular performance benchmarking 	Medium
	REQ_F0007: The system shall verify file type and size before accepting uploads and display a clear error if requirements are not met.	<ol style="list-style-type: none"> 1. Client and server-side validation of file properties 2. Standardized error handling mechanisms 3. Comprehensive file type whitelist 4. Automated file scanning process 5. User feedback on upload progress 	High

Availability	REQ_P0007: The system shall maintain 99.9% uptime during academic terms and 99% during breaks and holidays.	<ol style="list-style-type: none"> 1. Implement redundant server infrastructure 2. Setup automatic failover mechanisms 3. Regular preventative maintenance scheduling 4. Implement real-time health monitoring 5. Geographic distribution of deployment of system components 	High
Security	REQ_F0009: Access to information and features shall be based on user roles (student, parent, lecturer, admin).	<ol style="list-style-type: none"> 1. Role-based access control implementation 2. Segregation of duties for critical functions 	High
	REQ_F3001: Parental access and notifications shall comply with university privacy policies and require explicit student consent.	Consent management system: The student will be required to mail a signed consent to allow/revoke access of the system to their parent.	Medium
	REQ_F2901: The system shall maintain an audit log that records all user and system activities, including but not limited to logins, data modifications, access to sensitive records, and administrative actions.	<ol style="list-style-type: none"> 1. Tamper-evident logging mechanism 2. Separate storage for security logs 	Medium
Maintainability	REQ_F2701: The system shall allow creation, customization, and management of communication templates by admin.	<ol style="list-style-type: none"> 1. Template management system 2. Template versioning capability 	Low

Portability	REQ_I0002: The interface shall be responsive and adapt to different screen sizes.	<ol style="list-style-type: none"> 1. Responsive design framework implementation 2. Device-specific testing procedures 	Medium
	REQ_F3301: The system shall support integration with common calendar applications (Google Calendar, Apple Calendar) for academic schedules.	<ol style="list-style-type: none"> 1. Standard calendar API implementations 2. iCalendar format support 3. Synchronization conflict resolution 	Medium

3.8 Supporting Information

This section provides supplementary details to help readers and implementers of the SRS.

a) Sample Input/Output Formats:

1. Academic Data Import:

- a. Accepted formats: CSV, Excel (.xlsx)
- b. Sample CSV Header: StudentID, CourseCode, Grade, Attendance, Semester

2. Notification Export:

- a. Exported as CSV: Recipient, Channel, NotificationType, DeliveryStatus, Timestamp

3. Parent Portal Access:

Sample JSON output:

```
{
  "studentName": "Jane Doe",
  "attendance": "95%",
  "billingStatus": "Paid",
  "latestGrades": [
    {"course": "Math101", "grade": "A"},
    {"course": "CompSci201", "grade": "B+"}
  ]
}
```

4. Sample Calendar API Input/Output: iCalendar (.ics) Format

The COMSYS system supports calendar data exchange using the iCalendar (.ics) file format, a widely used standard for representing and sharing scheduling information across platforms (e.g., Google Calendar, Microsoft Outlook).

Example: Exported Calendar Event (.ics)

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//COMSYS University Portal//EN
CALSCALE:GREGORIAN
METHOD:PUBLISH
BEGIN:VEVENT
UID:20250524T133411Z-001@comsys.university.edu
DTSTAMP:20250524T133411Z
DTSTART:20250601T090000Z
DTEND:20250601T100000Z
SUMMARY:Sample Event - Course Registration Deadline
DESCRIPTION:Last day to register for summer courses. Please ensure your enrollment is complete.
```

LOCATION:Online Portal
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

Explanation:

The .ics file format enables COMSYS to import and export calendar events, supporting interoperability with external calendar applications.

This allows users to:

- a. Import university calendar events into their personal calendars.
- b. Export academic deadlines, schedules, or notifications as downloadable .ics files.

Note:

This sample is illustrative. Actual exported fields and their mapping will be defined by the COMSYS Calendar API implementation and requirements.

b) Supporting or Background Information:

1. Requirements Elicitation:

Requirements were gathered through interviews, questionnaires, and observation sessions with stakeholder groups: students, parents, lecturers, administrators, and IT staff.

2. Pain Points Addressed:

- I. Fragmented communication channels
- II. Lack of centralized academic and administrative access
- III. Missed or delayed notifications
- IV. Inefficient workflows

3. Standards & Best Practices:

- I. Follows ISO/IEC/IEEE 29148:2018 for requirements engineering.
- II. WCAG 2.1 for accessibility.

c) Problem Description:

The portal is intended to solve the problem of fragmented academic and administrative systems, inconsistent and unreliable communications, and lack of timely access to important academic, billing, and scheduling information for all stakeholders.

d) Special Packaging Instructions:

1. All deployable code and configuration files must be securely packaged and digitally signed.
2. Media exported for deployment must be encrypted and stored according to university IT security protocols.
3. No sensitive data should be included in deployment or export packages.
4. All third-party component licenses must be included in the deployment package.

Note:

All supporting information provided here is intended for implementation guidance and stakeholder understanding. Sample data formats are illustrative and are not to be considered mandatory requirements unless otherwise specified in Section 3 (Requirements).

4 Verification

4.1 Verification Approach

COMSYS will be verified through a structured combination of manual and automated testing processes to ensure that all functional, performance, usability, and security requirements are met.

How:

1. **Unit Testing:** Individual software modules will be tested for correctness and robustness using automated unit tests.
2. **Integration Testing:** Interactions between modules (e.g., notification system and academic database) will be verified through integration tests.
3. **System Testing:** The complete system will undergo end-to-end testing for all specified use cases and workflows, including regression testing.
4. **User Acceptance Testing (UAT):** Representative end users (students, parents, lecturers, admins) will perform acceptance testing to confirm that the system satisfies real-world requirements.
5. **Performance Testing:** The system will be subjected to load and stress tests to verify response time, reliability, and synchronization speeds.
6. **Security Testing:** Security audits and penetration testing will be conducted to confirm compliance with privacy policies and data protection regulations (e.g., FERPA, GDPR).
7. **API Testing:** Verify all integrations with external systems (Campus Management System, SMS Gateway) through automated API tests
8. **Accessibility Testing:** Ensure compliance with WCAG guidelines for users with disabilities.

Who:

1. The **product development team** will conduct unit and integration testing.
2. The **QA (Quality Assurance) department** will oversee system, regression, and performance testing.
3. **Security testing** will be performed by IT security specialists or external auditors.
4. **User Acceptance Testing** will involve selected representatives from each user group (students, parents, lecturers, admins)

When:

1. Verification will occur at key milestones:
 - a. After completion of individual features and modules (unit testing).
 - b. At the end of each development sprint (integration and system testing).
 - c. Prior to each major system release (performance, security, and acceptance testing).
 - d. After significant updates or bug fixes (regression testing).

Where:

1. All testing will take place in a dedicated QA/testing environment that accurately reflects the production environment.
2. Staging environment will be conducted for integration testing.
3. Sandbox environment will be conducted for security testing.
4. Cloud-based testing platforms for cross-browser and device testing.

4.2 Verification Criteria

The software will be verified against the following criteria:

Performance

1. The system shall load any page within 3 seconds under normal load conditions.
2. Calendar synchronization shall occur within 2 minutes of changes.
3. Critical notifications shall be delivered within 1 minute of their creation.
4. Course enrolment requests shall be processed within 5 seconds.
5. Academic data synchronization across user interfaces shall occur within 5 seconds.
6. The system shall maintain 99.9% uptime during academic terms and 99% during breaks and holidays

Functionality

1. All notifications must be delivered to the selected channels (email, SMS, portal, push) as configured by users, with no duplication or data loss (99% reliability for scheduled/automated notifications).
2. Only authorized users can access, upload, or modify academic data and materials as specified by role-based access controls.
3. The system must allow users to customize notification preferences and filter/mute categories.
4. Read status of announcements must be accurately tracked and displayed in real time to authorized users.

Usability & Accessibility

1. Navigation depth shall not exceed five levels for any feature.
2. Tooltips, help guides, and visual indicators shall be present for complex features.
3. The interface shall be responsive and accessible across supported device types.
4. The system shall support multilingual interface options as specified.

Security & Compliance

1. All access and data transfers must be authenticated via SSO and encrypted in transit and at rest.
2. Parental access and notifications must comply with privacy and consent requirements.
3. The system shall log and audit all critical actions for traceability.

Successful verification will be achieved when the system consistently meets or exceeds these criteria during QA and user acceptance testing.

5 Appendices

5.1 Assumptions and Dependencies

1. Browser Compatibility:

- i. Latest versions of Chrome, Firefox, and Safari will maintain support for WebSocket and current web standards
- ii. Browsers will continue to support TLS/SSL encryption protocols

2. Network Infrastructure:

- i. Reliable internet connectivity with sufficient bandwidth to handle concurrent user sessions

3. External Systems Integration:

- i. Continuous availability of the Campus Management System's RESTful API
- ii. SMS Gateway service reliability for critical notifications
- iii. Calendar Applications (Google Calendar, Apple Calendar) API stability

4. Data Assumptions:

- i. Academic calendar structure remains consistent
- ii. Student ID and course formats remain consistent

5. Authentication and Authorization

- i. The university's Single Sign-On (SSO) service will remain available and maintain current authentication protocols.
- ii. User roles and permissions will be centrally managed and updated by the institution.

6. Data Privacy and Security

- i. University data privacy policies and regulations (e.g., FERPA, GDPR) will remain unchanged during implementation and operation.
- ii. Secure storage and transmission of sensitive data is ensured by university infrastructure.

7. User Base

- i. The number of concurrent users will not exceed projected peak loads defined in performance requirements.
- ii. All users will have access to university-issued email accounts for notifications and password recovery.

8. Maintenance and Support

- i. Regular maintenance windows will be scheduled and communicated in advance.
- ii. IT support staff will be available for troubleshooting and incident response.

9. Third-Party Components

- i. All third-party libraries and frameworks used will remain actively maintained and compatible with the system's technical stack.
- ii. Licensing for any third-party services or components will remain valid and up to date.

5.2 Acronyms and Abbreviations

1. **API** (Application Programming Interface): Set of protocols and tools for building and integrating application software.
2. **CMS** (Campus Management System): The university's core administrative data system.
3. **COMSYS** (Communication and Services Portal): The centralized web platform described in this SRS.
4. **FERPA** (Family Educational Rights and Privacy Act): U.S. law governing the privacy of student education records.
5. **GDPR** (General Data Protection Regulation): European Union regulation on data protection and privacy.
6. **HTML** (Hypertext Markup Language): Standard language for documents designed to be displayed in a web browser.
7. **HTTP/HTTPS** (Hypertext Transfer Protocol [Secure]): Protocols for transferring data over the web (secure variant uses encryption).
8. **iCalendar (.ics)** (Internet Calendaring and Scheduling Core): File format standard for exchanging calendar information.
9. **JSON** (JavaScript Object Notation): Lightweight data-interchange format.
10. **OS** (Operating System): System software that manages hardware and software resources.
11. **RBAC** (Role-Based Access Control): Security paradigm based on user roles.
12. **REST** (Representational State Transfer): Architectural style for designing networked applications.
13. **SRS** (Software Requirements Specification): This document, detailing system requirements and constraints.
14. **SSO** (Single Sign-On): A unified authentication process for multiple applications.
15. **SMS** (Short Message Service): Text messaging service component of most telephone, internet, and mobile device systems.
16. **SMTP** (Simple Mail Transfer Protocol): Protocol for sending email messages.
17. **UI** (User Interface): The point of interaction between the user and the system.
18. **WCAG** (Web Content Accessibility Guidelines): International standard for web accessibility.
19. **XML** (Extensible Markup Language): Markup language for encoding documents in a format that is both human-readable and machine-readable.

5.3 Glossary

This glossary provides in-depth explanations of domain-specific terms and their significance within the context of COMSYS.

1. **Academic Calendar:**
A schedule maintained by the university that includes term dates, exam periods, holidays, and other significant academic events. COMSYS uses this for syncing and managing deadlines and reminders across user roles.
2. **Audit Log:**
A tamper-evident record of all actions and events within the system, including logins, data changes, and administrative operations. Used for security, compliance, and troubleshooting.
3. **Calendar API:**
A set of RESTful endpoints in COMSYS that allows integration and synchronization with external calendar applications (e.g., Google Calendar, Apple Calendar). Supports importing/exporting events in standardized formats like iCalendar (.ics).
4. **Chat Service:**
A real-time messaging functionality within COMSYS that enables direct communication between students, lecturers, parents, and administrators.
5. **Data Caching:**
Temporary storage of frequently accessed or recently fetched data to improve system speed and reduce repeated queries to external systems.
6. **Encryption (TLS/SSL):**
Security protocols that ensure data transmitted between users and the portal is protected from interception and unauthorized access.
7. **External System:**
Any system outside of COMSYS to which it connects for data or service integration (e.g., CMS, SMS Gateway, external calendar service).
8. **Multilingual Support:**
The capability of COMSYS to present its user interface and notifications in multiple languages, facilitating accessibility and user preference.
9. **Notification Channel:**
The medium through which notifications are delivered to users, such as email, SMS, or portal-based in-app alerts.
10. **Parent Portal:**
A dedicated interface within COMSYS that allows authorized parents or guardians to view student-related information and receive notifications, subject to consent and privacy policies.

11. Performance Requirement:

A quantifiable target for system responsiveness, throughput, reliability, or other operational metrics (e.g., page load time, notification delivery speed).

12. Portal Integration:

The process and capability of COMSYS to connect with and exchange data with other university platforms, ensuring a seamless user experience.

13. Role:

A specific category assigned to a COMSYS user (student, parent, lecturer, admin) determining their permissions, accessible features, and data visibility.

14. Session Timeout:

The period of inactivity after which a user is automatically logged out to maintain security.

15. WebSocket:

A communication protocol used in COMSYS for real-time features like live chat and instant notifications, enabling bidirectional, low-latency data exchange.