

Software Construction & Development



Submitted by:

Mahmood Khan (56067)

Submitted to:

Mr. Muhammad Islam Abbasi

Dated:

3rd October 2025

RIPHAH International University

Fall 2025

Faculty of Computing

Introduction

RIPHAHLINK is a web-based attendance management system designed to modernize and streamline the attendance process in throughout different campuses of the well-known RIPHAI International University all over Pakistan. Built with security, stability and user experience as its core, RIPHAHLINK would empower the university, teachers, students to engage with a live tracking software enabled to mark the attendance with ease and high reliability with a transparent and efficient process. This system would support real-time check-in's via QR codes, automated reporting and role-based access control. It would be optimized for both desktop and mobile use, ensuring accessibility for all the relevant stakeholders. RIPHAHLINK aims to reduce administrative overhead, improve academic accountability and also serve as a model for digital transformation in the educational market and world.

Stakeholders

This project would aim the following stakeholders:

- University Administration
- Teachers
- Students
- Co-ordination Offices

Functional Requirements

This project shall have the following functional requirements

1. Teacher Module

FR 1.1: Teacher will be able to login through OTP/Password or recover password

FR 1.2: Teacher will be able to create and manage class schedules

FR 1.3: Teacher will be able to mark attendance manually or via QR Code scan

FR 1.4: Teacher will be able to view attendance history & generate downloadable reports

FR 1.5: Teacher will be able to flag students with low attendance.

2. Student Module

FR 2.1: Student will be able to login through password/OTP request

FR 2.2: Student will be able to mark attendance

FR 2.3: Student will be able to view personal attendance dashboard

FR 2.4: Student will be able to receive alerts for attendance thresholds

FR 2.5: Student will be able to submit leave applications to teacher/faculty

3. Admin Module

FR 3.1: Admin will be able to add, edit or remove users & customize permissions

FR 3.2: Admin will be able to monitor attendance trends across departments

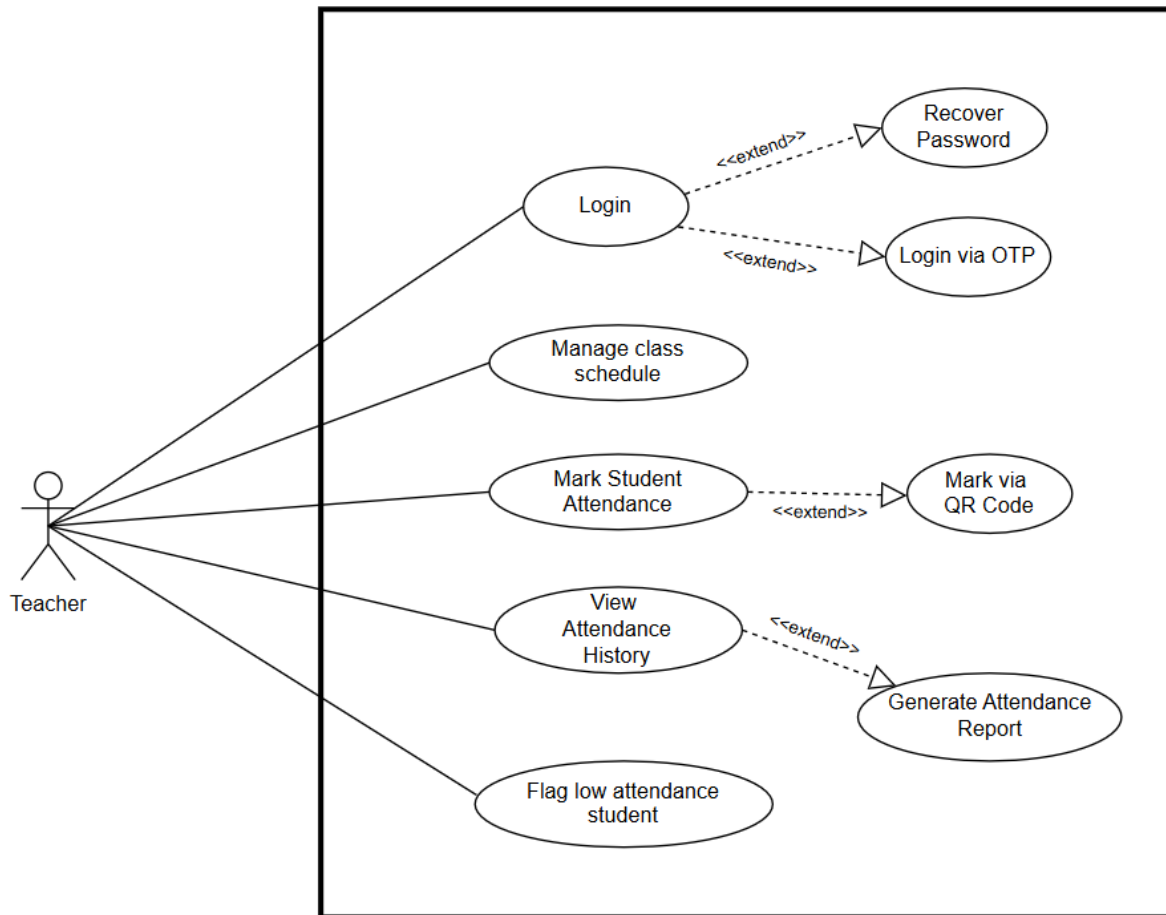
FR 3.3: Admin will be able to export data for audits or compliance.

FR 3.4: Admin will be able to configure system settings and institutional policies.

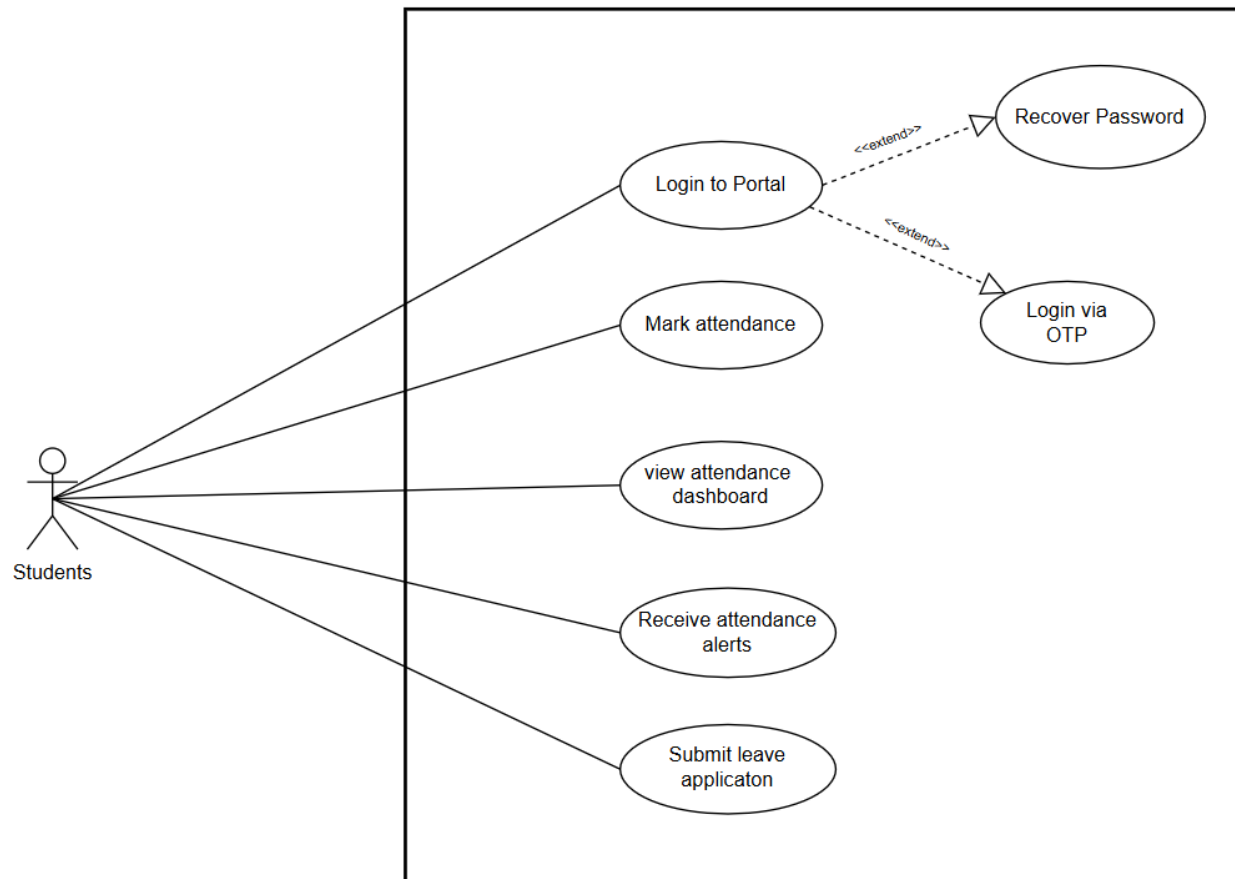
FR 3.5: Admin will be able to upload department wide notifications

Use Case Diagram

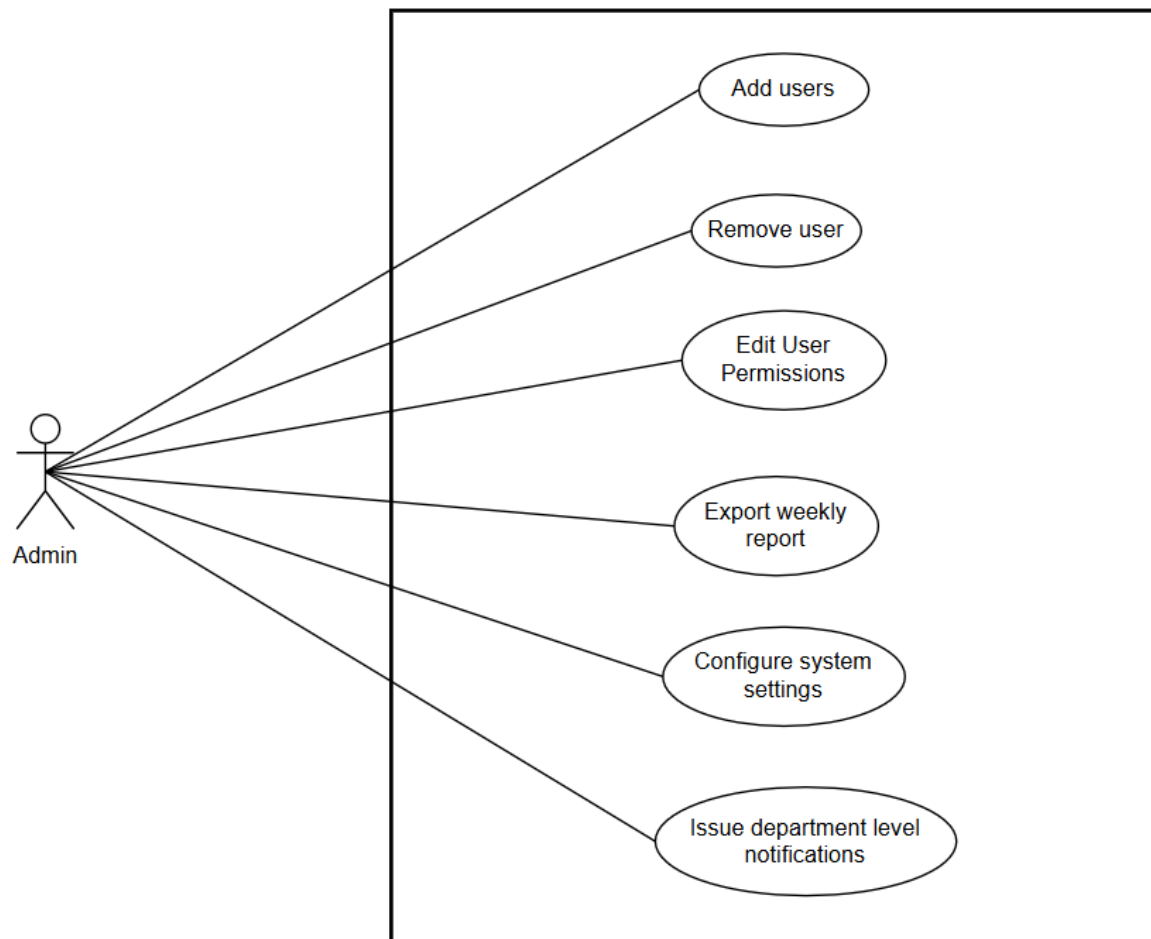
Teacher Use Case



Student Use Case



Admin Use Case



Fully Dressed Use-Cases

Use Case No:	001	
Use Case Name	Login	
Actor	Application User	
Type	Primary	
Description	Application User signs into application	
Pre-conditions	User has active internet connection and must be a registered student of the university.	
Post-conditions	User has successfully been logged into account and is directed to the dashboard of the RIPHAH-Link application	
Normal Flows	Actor 1: User enters their credentials into website	System 2. System verifies user credentials 3. System will redirect user to the main dashboard.
Extension	1.1: User will have to re-enter credentials if password is incorrect.	

Use Case No:	002	
Use Case Name	Recover Password	
Actor	Application User	
Type	Primary	
Description	Application user chooses to recover password	
Pre-conditions	User must have selected the recover password option in order to move ahead with the process	
Post-conditions	User successfully resets password and can login to portal again	
Normal Flows	<p>Actor</p> <p>1. Actor chooses to recover password by clicking button</p> <p>3. Actor resets password successfully</p> <p>5. Actor logs in with new password</p>	<p>System</p> <p>2. System processes recover password request</p> <p>4. System stores new password in database</p> <p>6. System redirects user to main dashboard</p>
Extension	<p>3.1: User enters existing password</p> <p>4.1: Systems displays "choose a password that isn't your current password" to user</p> <p>5.1: User hits back and logs in with password that they entered to change it.</p> <p>6.1: System closes request and redirects user to main dashboard</p>	

Use Case No:	003	
Use Case Name	Login via OTP	
Actor	Application User	
Type	Primary	
Description	User chooses to login via OTP	
Pre-conditions	User must have an active internet connection along with the email already logged in	
Post-conditions	User successfully logs into web application	
Normal Flows	<p>Actor</p> <p>1. Actor chooses to login to account via OTP</p> <p>3. User successfully gets OTP and logs in</p>	<p>System</p> <p>2. System sends OTP via Email or Phone</p> <p>4. System redirects user to dashboard</p>
Extension	<p>2.1: Systems fails to find phone number and redirects user to enter again</p> <p>3.1 User fails to enter OTP before expiry time set of the OTP</p> <p>3.2: User is sent OTP again via Phone or email</p>	

Use Case No:	017	
Use Case Name	Add user	
Actor	Application Admin	
Type	Primary	
Description	Admin chooses to add user	
Pre-conditions	Admin must enter correct credentials for user	
Post-conditions	Admin successfully adds user into application and database	
Normal Flows	<p>Actor</p> <p>1. Admin chooses to add user into portal</p> <p>4. Admin enters secret code</p> <p>7. Admin successfully adds user into database and portal</p>	<p>System</p> <p>2. System processes admin request</p> <p>3. System redirects admin to enter secret code to enter user</p> <p>5. System verifies secret code</p> <p>6. System verifies successfully and adds user</p>
Extension	<p>4.1: Admin enters wrong secret code 3 times</p> <p>4.2: Admin is locked out of system for 5 hours</p> <p>6.1: If system fails to verify, it sends admin back to home page and stops admin from adding user</p>	

Use Case No:	018	
Use Case Name	Remove user	
Actor	Application Admin	
Type	Primary	
Description	Admin chooses to remove user	
Pre-conditions	Admin must provide reason for removing user and wait for further approval	
Post-conditions	Admin is granted approval & successfully removes user from the application	
Normal Flows	<p>Actor</p> <p>1. Admin chooses to remove user from portal</p> <p>5. Admin successfully removes user from portal</p>	<p>System</p> <p>2. System processes admin request</p> <p>3. System sends request to Student Regulatory Body Department</p> <p>4. System gets approval & grants admin permission</p>
Extension	<p>3.1: System fails to get approval from Student Body and redirects admin to request for approval again</p>	

Use Case No:	019	
Use Case Name	Edit User Permissions	
Actor	Application Admin	
Type	Primary	
Description	Admin chooses to edit permissions for application user	
Pre-conditions	Admin must check user level before editing their permissions	
Post-conditions	Admin successfully changes user permission	
Normal Flows	<p>Actor</p> <p>1. Admin chooses specific user to edit their permissions</p> <p>4. Admin gets approval based on level of the user</p> <p>6. Admin see's live update of the user that was just added</p>	<p>System</p> <p>2. System processes admin request</p> <p>3. System checks users level and responds</p> <p>5. System changes user permissions successfully</p>
Extension	4.1: If admin tries giving user admin privileges, he will redirected to admin page and given warning	