FAST-NUCES

EDU-TRACK

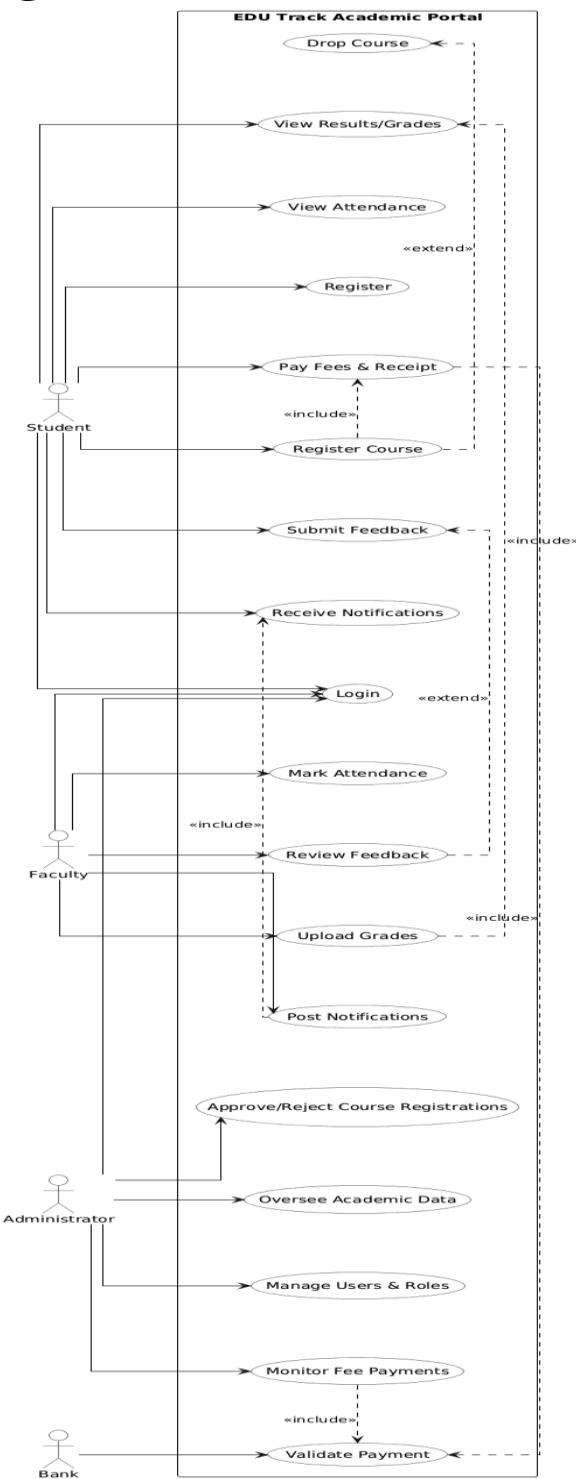# PHASE-01

TEAM -02

Member Name	Roll-No	Primary Responsibility
Asjal Abdullah	22L-6273	Use Case Diagram
M Waleed	22L-7788	Use Case Description
Salman Saeed	22L-7789	Class Analysis Diagram
Hamdan Malik	22L-7773	Documentation

Table of Contents

1) Use Case Diagram

EDU Track Academic Portal

- Drop Course
- View Results/Grades
- View Attendance
- Register
- Pay Fees & Receipt
- Register Course
- Submit Feedback
- Receive Notifications
- Login
- Mark Attendance
- Review Feedback
- Upload Grades
- Post Notifications
- Approve/Reject Course Registrations
- Oversee Academic Data
- Manage Users & Roles
- Monitor Fee Payments
- Validate Payment

Actors: Student, Faculty, Administrator, Bank

«extend» «include»

2) Use Case Description (Use Case -01)

Identifier	UC-01
Name	Register Course
Summary	Allows a student to register for courses offered in the semester.
Priority	High
Actors	Student ,Administrator
Pre-Conditions	Student must be logged into the system. Course registration period must be open.
Post-Conditions	Course registration request is submitted and awaits approval.

Typical Course of Action		
S#	Actor Action	System Response
1	Student selects "Register Course" option	System displays available courses
2	Student chooses desired course(s)	System validates prerequisites and course availability
3	Student confirms registration	System submits registration request for admin approval
4	Administrator reviews request	System updates registration status

Alternate Course of Action		
1	Student selects a course with missing prerequisites	System displays error message
2	Student attempts to register after deadline	System denies registration request

2) Use Case Description (Use Case -02)

Identifier	UC-02
Name	Drop Courses
Summary	Enables a student to drop a registered course within the allowed period.
Priority	Medium
Actors	Student ,Administrator
Pre-Conditions	Student must be logged in for course registration.
Post-Conditions	Course is removed from student's registration list.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Student selects "Drop Course" option	System displays list of registered courses
2	Student selects a course to drop	System validates deadline for dropping
Alternate Course of Action		
1	Student attempts to drop after deadline	System rejects request
2	Student tries to drop last compulsory course	System displays restriction warning

2)Use Case Description (Use Case -3)

Identifier	UC-03
Name	View Results/Grades
Summary	Allow students to view their academic performance in each course.
Priority	High
Actors	Student ,Faculty
Pre-Conditions	Student must be logged in. Grades must be uploaded by faculty.
Post-Conditions	Student view final or interim grades.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Student selects "View Results" option	System displays grade report for registered courses
2	Faculty uploads grades	System updates student records

Alternate Course of Action		
1	Student views results before grades are uploaded	System displays "Grades not available" message

2) Use Case Description (Use Case-04)

Identifier	UC-04
Name	View Attendance
Summary	Student can check their attendance record for each course.
Priority	Medium
Actors	Student ,Faculty
Pre-Conditions	Faculty must have marked attendance. Students must be logged in.
Post-Conditions	Attendance details displayed.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Student selects "View Attendance" option	System retrieves attendance data
2	Faculty marks attendance	System updates attendance records
Alternate Course of Action		
1	Attendance not yet marked	System shows "Attendance data unavailable" message

2) Use Case Description (Use Case-05)

Identifier	UC-05
Name	Pay Fees and Receipt
Summary	Students can pay tution fees and receive payment receipts.
Priority	High
Actors	Students, Bank, Administrator
Pre-Conditions	Students must have pending fee. Bank gateway must be online.
Post-Conditions	Fee payment completed and receipt issued.

Typical Course of Action		
1	Student selects "Pay Fees" option	System generates payment request
2	Student provides payment details	System sends request to bank
3	Bank validates payment	System confirms transaction
4	System issues receipt	Student downloads/prints receipt

Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	Bank rejects payment	System notifies student
2	Network failure occurs	System saves payment as "Pending"

2) Use Case Description (Use Case-06)

Identifier	UC-06
Name	Submit Feedback
Summary	Students provide feedback for courses and instructors.
Priority	Medium
Actors	Students, Faculty
Pre-Conditions	Students must be enrolled in the course. Feedback session must be open.
Post-Conditions	Feedback stored and available for faculty review.

	Typical Course of Action	
S#	**Actor Action**	**System Response**
1	Student selects "Submit Feedback" option	System displays feedback form
2	Student fills and submits feedback	System saves feedback
3	Faculty reviews feedback	System generates feedback reports

	Alternate Course of Action	
1	Feedback session closed	System denies submission

2) Use Case Description (Use Case-07)

Identifier	UC-07
Name	Reveive Notifications
Summary	Students receive notifications about academic events, deadlines, and updates.
Priority	Medium
Actors	Students, Faculty, Administrator
Pre-Conditions	User must be logged in.
Post-Conditions	Notifications delivered to user.

Typical Course of Action		
S#	Actor Action	System Response
1	Faculty/Admin posts notification	System sends notification to concerned students
2	Student views notification	System displays notification details

Alternate Course of Action		
1	Student offline	System queues notification for later delivery

2) Use Case Description (Use Case-08)

Identifier	UC-08
Name	Login
Summary	Authenticates users before accessing the portal.
Priority	High
Actors	Students, Faculty, Administrator
Pre-Conditions	User must have valid credentials.
Post-Conditions	User gains access to system based on role.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User enters credentials	System validates credentials
2	Credentials valid	System grants access

Alternate Course of Action		
1	User enters invalid credentials	System denies access
2	Account locked due to failed attempts	System notifies user

2) Use Case Description (Use Case-09)

Identifier	UC-09
Name	Mark Attendance
Summary	Faculty marks attendance for their classes.
Priority	High
Actors	Faculty
Pre-Conditions	Faculty must be logged in. Class session must exist.
Post-Conditions	Attendance records updated.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Faculty selects "Mark Attendance" option	System displays enrolled students
2	Faculty marks attendance	System updates records

Alternate Course of Action		
1	Faculty forgets to save	System prompts to confirm before exit

2)Use Case Description (Use Case-10)

Identifier	UC-10
Name	Review Feedback
Summary	Faculty reviews student feedback to improve teaching.
Priority	Medium
Actors	faculty
Pre-Conditions	Feedback must be submitted by student
Post-Conditions	Feedback displayed.

Typical Course of Action		
S#	Actor Action	System Response
1	Faculty selects "Review Feedback"	System retrieves feedback records
2	Faculty views reports	System displays summarized feedback
Alternate Course of Action		
1	No feedback submitted	System displays "No feedback available"

2)Use Case Description (Use Case-11)

Identifier	UC-11
Name	Upload Grades
Summary	faculty uploads student grades in system.
Priority	High
Actors	faculty
Pre-Conditions	Faculy must be logged in. Students must be enrolled.
Post-Conditions	Grades stored and visible to students.
	Typical Course of Action

S#	Actor Action	System Response
1	Faculty selects "Upload Grades"	System displays enrolled students
2	Faculty enters grades	System validates and stores data
3	System updates results	Students can view grades
	Alternate Course of Action	
1	Faculty uploads incomplete grades	System prompts for completion

2)Use Case Description (Use Case-12)

Identifier	UC-12
Name	Post Notifications
Summary	Faculty or admin posts announcements for users.
Priority	Medium
Actors	Faculty, Administrator
Pre-Conditions	User must be logged in with posting rights.
Post-Conditions	Notification sent to concerned users.

Typical Course of Action		
S#	Actor Action	System Response
1	Faculty/Admin selects "Post Notification"	System provides notification form
2	User submits notification	System distributes notification

Alternate Course of Action		
1	Notification form incomplete	System prompts correction

2) Use Case Description (Use Case-13)

Identifier	UC-13
Name	Approve/Reject Course Registrations
Summary	Administrator approves or rejects student course registration requests.
Priority	High
Actors	Administrator approves or rejects student course registration requests.
Pre-Conditions	Students must submit course registration requests.
Post-Conditions	Registration status updated.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Admin selects "Course Requests"	System lists pending requests
2	Admin reviews each request	System displays student details
3	Admin approves/rejects request	System updates status and notifies student

Alternate Course of Action		
1	Admin delays action	System marks request as "Pending"

2) Use Case Description (Use Case-14)

Identifier	UC-14
Name	Oversee Academic Data
Summary	Administrator monitors and manages institutional academic data.
Priority	Medium
Actors	Administrator monitors and manages institutional academic data.
Pre-Conditions	Admin must be logged in
Post-Conditions	Academic data updated.

Typical Course of Action		
S#	Actor Action	System Response
1	Admin selects "Oversee Academic Data"	System displays academic records
2	Admin reviews data	System updates changes if made
Alternate Course of Action		
1	Data unavailable	System shows error message

2) Use Case Description (Use Case-15)

Identifier	UC-15
Name	Manages Users and Roles
Summary	Administrator manages accounts and assign roles.
Priority	High
Actors	Administrator manages accounts and assign roles.
Pre-Conditions	Admin must be logged in
Post-Conditions	Users added/modified/removed with assigned roles.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Admin selects "Manage Users"	System displays user list
2	Admin adds/updates/deletes user	System updates database
3	Admin assigns roles	System saves permissions

Alternate Course of Action		
1	Invalid user details	System prompts for correction

2) Use Case Description (Use Case-16)

Identifier	UC-16
Name	Monitor Fee Payments
Summary	Administrator monitors fee payment statuses.
Priority	Medium
Actors	Administrator ,Bank
Pre-Conditions	Bank transactions recorded.
Post-Conditions	Updated payment status report.

Typical Course of Action		
S#	Actor Action	System Response
1	Admin selects "Monitor Fee Payments"	System displays student payment records
2	System verifies payments with bank	Updated status shown

Alternate Course of Action		
1	Bank validation fails	System flags transaction for review

2) Use Case Description (Use Case-17)

Identifier	UC-17
Name	Validates Payments
Summary	Bank validates student payment transactions.
Priority	High
Actors	Bank, System
Pre-Conditions	Student submits payment details.
Post-Conditions	Payment approved or rejected.

Typical Course of Action		
S#	**Actor Action**	**System Response**
1	System sends payment request	Bank validates payment
2	Bank approves transaction	System records payment success

Alternate Course of Action		
1	Bank rejects payment	System notifies student
2	Payment timed out	System marks as pending

3) Analysis Class Diagram