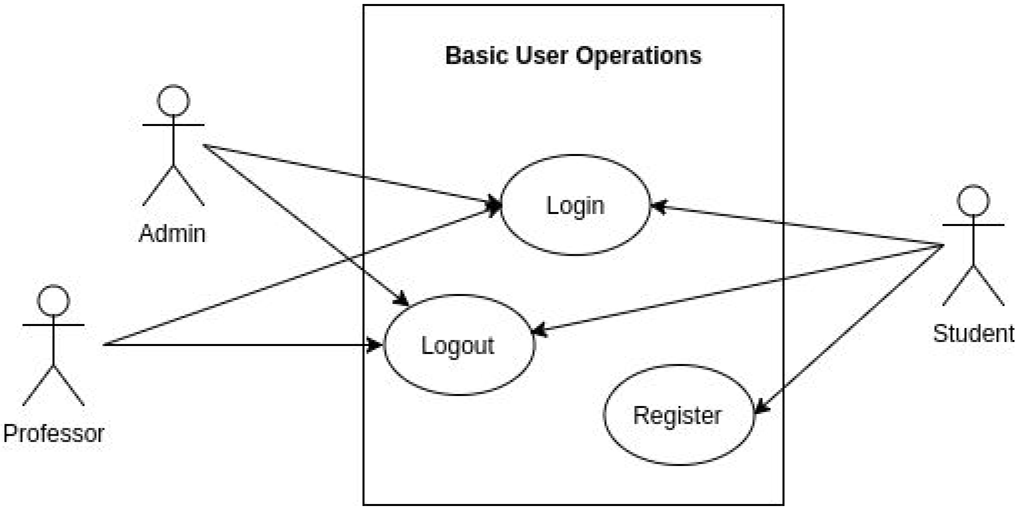
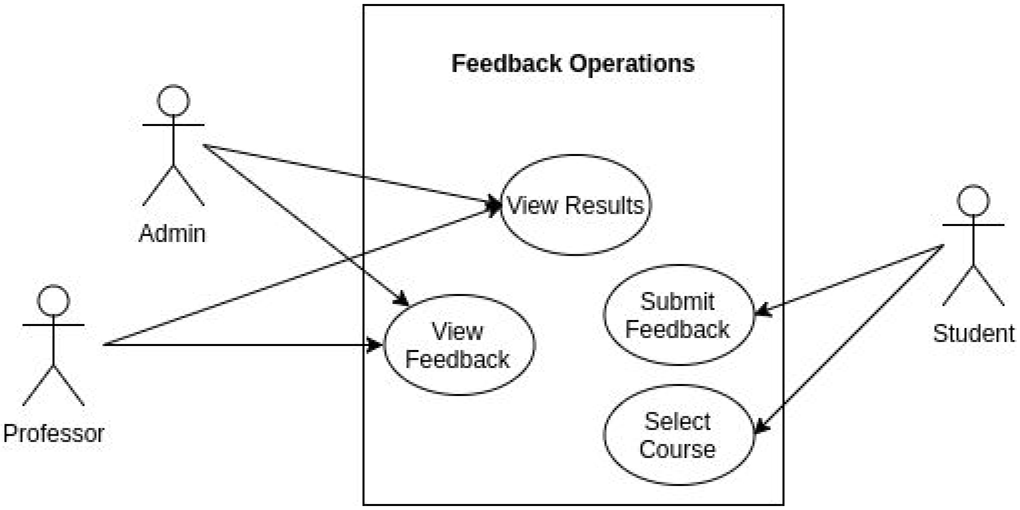
Use Case Document [Group 8]

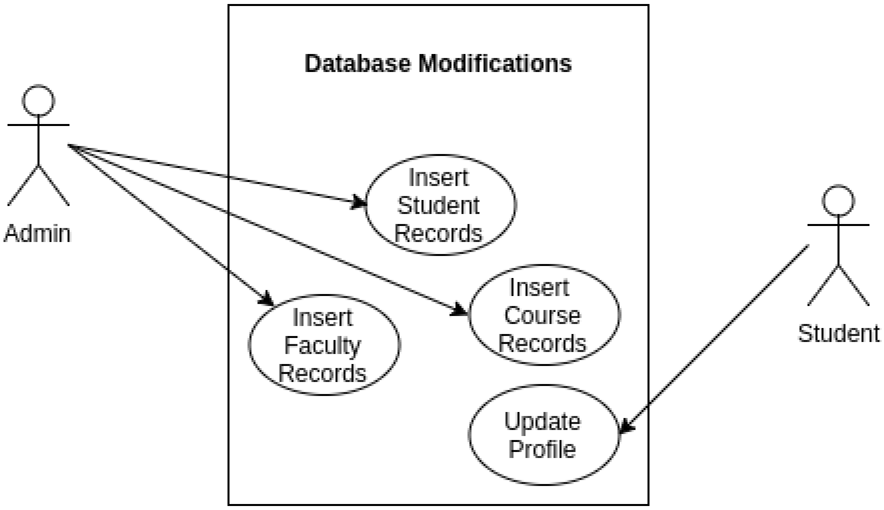
- **List of users, use cases and test cases:**

ACTORS	USE CASES	TEST CASES
Student	Login	Test invalid usernames
		Test invalid password
	Register	Check data validity
	Select Course	Check submissions
		Check registration
	Submit feedback	Check data errors
	Logout	Check access privileges
Administration	Login	Test invalid usernames
		Test invalid password
	Insert/Modify student records	Check data validity
	Insert/Modify faculty records	Check data validity
	Insert/Modify course records	Check data validity
	View feedback	Mock data test
	View feedback result	Analysis checks
	Logout	Check access privileges
Professors	Login	Test invalid usernames
		Test invalid password
	Update profile	Commit checks on changes
	View feedback result	Analysis checks
	Logout	Check access privileges

- Use case summary diagrams:







- Informal Use Case Definitions:

- **Register** - Enable new students to be able to register to the feedback system.
- **Login** - Enable existing users to be able to sign in to their accounts to submit feedback.
- **Logout** - Enable logged in users to sign out of their accounts once they have finished.
- **Select Course** - Enable students to select courses that are relevant to them in order to submit feedback
- **Submit Feedback** - Enable a student to give feedback for the course that he/she has selected.
- **Insert/Modify Student Records** - Enable the admin to be able add new students to the feedback system
- **Insert/Modify Faculty Records** - Enable the admin to be able add new faculty to the feedback system
- **Insert/Modify Course Records -** Enable the admin to be able add new courses to the feedback system
- **View Feedback** - Enable admin to be able to view the feedback given by students.
- **View Feedback Result -** Enable the user to view the analytical results derived from the feedback.
- **Update Profile -** Enable faculty to change their profiles information.

- Formal Use Case Definitions:

1. Register

Definition	**Success Scenario**	**Failure Extension**
New users must be able to fill out a form giving useful information to be able to register for the feedback system. Users that are not registered must not be allowed to utilise the system. Registered users may **login** to use the system	- All data has been correctly validated and the user has successfully registered. - User may proceed to **login** and utilise the system.	- Data is not valid: User will be redirected to the same form to modify the data until valid data is entered.

2. Login

Definition	**Success Scenario**	**Failure Extension**
A user must be able enter their username and password to be able to access the feedback system.	- A valid username and a valid password for that username is entered. - User may utilise the system	- User will be redirected to the login page until valid username and password are entered

3. Logout

Definition	**Success Scenario**	**Failure Extension**
User must be able to sign out of the system after which he/she may not utilise the system from their current source without logging in again.	- User logs out and is not able to utilise the system any longer unless he logs in again.	- This use case must not fail under any conditions, a logged out user must remain logged out till logged in again.

4. Select Course

Definition	**Success Scenario**	**Failure Extension**
Student must be able to select a course from the list of courses he/she is enrolled in to be able to submit feedback for the course	- Student selects the course and is redirected to the feedback form for that course. - Courses relevant to student must only be available.	- Check if wrong feedback form displayed - Check if irrelevant course displayed

5. Submit Feedback

Definition	**Success Scenario**	**Failure Extension**
Display forms for a particular course to enable a student to give feedback for it.	- All data has been correctly entered and received at the server side	- In case data is not properly received, data checks would prompt the user to fill the form again

6. Insert/Modify Student/Faculty/Course Records

Definition	**Success Scenario**	**Failure Extension**
Enable admin to enter or to modify records of the students/faculty/courses	- Data successfully entered into the database - Updates to the database are committed and are persistent	- Perform data transactions only if all the updates are successfully committed. - Check key constraints between the three types of data

7. View Feedback

Definition	**Success Scenario**	**Failure Extension**
Enable admin to view feedback submitted by students in a readable and easily comprehensible format.	- New feedback entries are instantaneously available for the admin to view.	- Check response of the system for mock data to resolve any issues

8. View Feedback Result

Definition	**Success Scenario**	**Failure Extension**
Enable professors and admin to view the analysed results on the feedback data	- New feedback entries are analysed and correct results displayed on the visualisation page	- Check response of the system for results of mock data.

9. Update Profile

Definition	**Success Scenario**	**Failure Extension**
Enable professors to update info on their profiles and change their passwords	- Persistent changes that are instantaneously visible on user profiles	- Check response of the system for changes in the profile.

- Test Cases:

1. Login :: Test Invalid Username

Input	Output
- Randomly generated invalid usernames	- Fail
- Correct usernames	- Pass
- Correct usernames with case change	- Fail
- Correct usernames with modifications	- Fail

2. Login :: Test Invalid Password

Input	Output
- Randomly generated invalid passwords	- Fail
- Correct password	- Pass
- Correct password with case change	- Fail
- Correct password with modifications	- Fail

3. Register :: Check Data Validity

Input	Output
- Invalid email ids [strings that are not email ids]	- Fail
- Invalid email ids [Non IIITS email ids]	- Fail
- Invalid name [Name not in student DB]	- Fail
- Invalid passwords [Passwords that don’t adhere to Django password validation norms]	- Fail
- Valid passwords [Passwords that adhere to Django password validation	- Pass

norms]	

4. Select Course :: Check Submission

Input	Output
- Courses that have already been checked [Feedback submitted]	- Fail
- Unchecked courses	- Pass

5. Select Courses :: Check Registration

Input	Output
- Courses not registered by student	- Fail
- Registered courses	- Pass

6. Submit Feedback :: Check Data Errors

Input	Output
- Forms with blank fields	- Fail
- Forms with invalid data [Invalid descriptive feedback]	- Fail
- Complete forms with valid descriptive data	- Pass

7. Logout :: Check Access Privileges

Input	Output
- Requests for pages with login credentials requirements	- Fail
- Request for non-secure, non-authenticated pages	- Pass

8. Insert/Modify Students/Faculty/Professor records :: Check Data Validity

Input	Output
- Data for users already in database	- Fail
- Data for new users not in database	- Pass
- Data violating database schema	- Fail
- Data violating database constraints	- Fail
- Valid data [not violating schema or constraints]	- Pass

9. View Feedback :: Mock Data Check

Input	Output
- Insert mock data	- Data viewable on view page

10. View Results :: Analysis Checks

Input	Output
- Valid data test case for which analysis results are computed and verified	- If results match: Pass - Else: Fail

11. Update Profile :: Commit Checks

Input	Output
- Enter mock information in the update profile page	- If changes persist: Pass - Else: Fail