Use Case Description

1. **Use Case:** Student signs up for Portal.

   Actor: Student.

   Preconditions:

   1. Portal exists

Actor Actions	**System Responses**
Student creates account for accessing application/Portal	Student is saved in the system

2. **Use Case:** User logs in to Portal when already assigned to a course.

   Actor: Admin, Professor, TA and student.

   Preconditions:

   1 User account exists.

Actor Actions	**System Responses**
1 User signs in to the system	1. User is given access to the system.

3. **Use Case:** Student logs in to the system when he/she is not assigned to any course.

   Actor: Student.

   Preconditions:

   1 User exists.

Actor Actions	**System Responses**
Student tries to log in to the system.	System does not allow user to login to the portal and redirects to an error page.

4. **Use Case:** Admin creates accounts for professors.

   Actor: Admin.

Preconditions:

1. Admin should be logged in.

Actor Actions	System Responses
Admin creates profile for new professor.	New professor's information is saved in the system.

5. **Use Case:** Admin deletes professor.

Actor: Admin.

Preconditions:

1. Admin should be logged in.
2. Professor should exist in the system.

Actor Actions	System Responses
Admin deletes existing professor.	Professor is deleted from the system.

6. **Use Case:** Admin edits professor details.

Actor: Admin.

Preconditions:

1. Admin should be logged in.
2. Professor should exist in the system.

Actor Actions	System Responses
Admin enters updated information for the existing professor.	Professor details are updated in the system

7. **Use Case:** Admin assigns courses to professors.

Actor: Admin.

Preconditions:

1. Admin should be logged in.
2. Courses and Professor exist in the system.

Actor Actions	System Responses
Admin selects courses for professor.	Selected courses are assigned to the professor.

8. **Use Case:** Admin adds courses.

   Actor: Admin.

   Preconditions:

   1. Admin should be logged in.

Actor Actions	System Responses
Admin enters details for new course.	A new course is created.

9. **Use Case:** Admin edits course details.

   Actor: Admin.

   Preconditions:

   1. Admin should be logged in.
   2. Courses should exist in the system.

Actor Actions	System Responses
Admin enters updated information for the existing course.	Course details are updated in the system

10. **Use Case:** Admin deletes course details.

    Actor: Admin.

    Preconditions:

    1. Admin should be logged in.
    2. Courses should exist in the system.

Actor Actions	System Responses
Admin deletes existing courses.	Course is deleted from the system.

11. **Use Case:** Admin assigns courses to students.

    Actor: Admin.

    Preconditions:

    1. Admin should be logged in.
    2. Courses should exist in the system.
    3. Student should exist.

Actor Actions	**System Responses**
Admin maps courses to the students,	Student is assigned to the selected courses

12. **Use Case:** TA declares his office hours

    Actor: TA

    Preconditions:

    1. TA is already assigned to a course.

Actor Actions	**System Responses**
TA posts office hours	1. System saves TA office hours.

13. **Use Case:** TA edits his office hours

    Actor: TA

    Preconditions:

    1. TA is already assigned to a course.
    2. TA schedule exists

Actor Actions	**System Responses**
TA updates his schedule/location	TA schedule is updated in the system

14. **Use Case:** TA deletes his office hours

    Actor: TA

    Preconditions:

    1. TA is already assigned to a course.
    2. TA schedule exists

Actor Actions	System Responses
TA deletes his schedule/location	TA schedule is deleted in the system

15. **Use Case:** Professor selects TA from the list of students.

    Actor: Professor

    Preconditions:

    1. Students are available
    2. Professor is logged in.

Actor Actions	System Responses
Professor selects TA from the list of applicants.	System assigns TA to a course.

16. **Use Case:** Professor posts TA job openings.

    Actor: Professor

    Preconditions:

    1. Professor is assigned to a course.
    2. Professor is logged in.

Actor Actions	System Responses
Professor uploads openings for TA.	Job opening is uploaded.

17. **Use Case:** Professor views job applicant's Job-application.
    Actor: Professor

    Preconditions:

    1. Professor is assigned to a course.
    2. Professor is logged in.
    3. Student, TA must exist.
    4. Student and TA must have submitted job application.

Actor Actions	System Responses
Professor views applicant's information	Job opening is seen by the professor.

18. **Use Case:** Student and TA view job openings for courses which are assigned to them.
    Actor: Student and TA.

    Preconditions:

    1. Student, TA is logged in.
    2. Student, TA assigned to a course.
    3. Job opening is already posted by the professor.

Actor Actions	System Responses
Student and TA views job.	System displays all the jobs posted by a professor for the courses which are assigned to the logged in users.

19. **Use Case:** Student and TA apply for job.

    Actor: Student and TA

    Preconditions:

    1. Job opening for actors is available on Portal.
    2. Actors exist and should be logged in.

Actor Actions	System Responses
Student submits his application	1. System stores Student's profile 2. Applicant information is sent to the Professor.

20. **Use Case:** User creates a new thread for asking question.
    Actor: Professor, Student and TA

    Preconditions:

    1. User is logged in to the system.
    2. A course is already assigned to a user.

Actor Actions	System Responses
User submits a new thread for any assigned course.	A new thread is saved in the system.

21. **Use Case:** User views a thread.

Actor: Professor, Student and TA

Preconditions:

1. User is logged in to the system.
2. A course is already assigned to a user.
3. A thread already exists.

Actor Actions	System Responses
User can view an existing thread.	System renders a thread for the logged in user.

22. **Use Case:** User marks a new thread as favorite.

Actor: Professor, Student and TA

Preconditions:

1. User is logged in to the system.
2. Thread exists.
3. A course is already assigned to a user.

Actor Actions	System Responses
User sets any thread as favorite.	A thread is marked as a favorite for a user in the system

23. **Use Case:** User un-favorites a thread.

Actor: Professor, Student and TA

Preconditions:

1. User is logged in to the system.
2. Favorite thread exists.
3. A course is already assigned to a user.

Actor Actions	System Responses
User marks a thread as un-favorite.	A thread is no more marked as a favorite for a user in the system

24. **Use Case:** User creates a tag.
    Actor: Professor and TA

Preconditions:

1. Actor is logged in to the system.
2. A course is already assigned to a user.

Actor Actions	System Responses
User creates a new tag.	A tag is created in the system so that user can select from it while posting a thread.

25. **Use Case:** User assigns a tag to a thread while posting it.
    Actor: Professor, Student and TA

Preconditions:

1. User is logged in to the system.
2. A course is already assigned to a user.

Actor Actions	System Responses
User associates tag/tags to a thread.	A tag is assigned to a newly created thread in the system.

26. **Use Case:** User searches for a thread using a keyword.
    Actor: Professor, Student and TA

Preconditions:

1. A user is logged in to the system.
2. A course is already assigned to a user.
3. A thread exists in the system.

Actor Actions	System Responses
User looks for existing thread providing keyword.	System renders all threads containing given keyword in thread's title or content.

27. **Use Case:** User posts replies to a thread.
    Actor: Professor, Student and TA

Preconditions:

1. User is logged in to the system.

2. A course is already assigned to a user.
3. A thread exists in the system.

Actor Actions	**System Responses**
User adds a new post for an existing thread.	A post is saved in the system for the selected thread.

28. **Use Case:** User likes a post.

Actor: Professor, Student and TA

Preconditions:

1. User is logged in to the system.
2. A course is already assigned to a user.
3. A post exists for a thread in the system.

Actor Actions	**System Responses**
User like a post.	A post is marked as liked post for the user.

29. **Use Case:** User un-likes a post.

Actor: Professor, Student and TA

Preconditions:

1. User is logged in to the system.
2. A course is already assigned to a user.
3. A liked post exists for a thread in the system.

Actor Actions	**System Responses**
User unlike a post.	A post is marked as un-liked post for the user.

30. **Use Case:** User views posts.

Actor: Professor, Student and TA

Preconditions:

1. User is logged in to the system.
2. A course is already assigned to a user.
3. A post exists for a thread in the system.

Actor Actions	System Responses
User can view all posts which exist for a thread.	System renders posts for a thread.

31. **Use Case:** User (Professor, students or TA) views TA information.

Actor: TA, Student and Professor.

Preconditions:

1. TA is already assigned to a course.

Actor Actions	System Responses
1. User views TA information.	1. TA details displayed.

32. **Use Case:** User logs out from his account.

Actor: Admin, Professor, TA and Student.

Preconditions:

1. User must be logged in to the system.

Actor Actions	System Responses
Users are taken to the log in page.	A logged in user in logged out of the system.

33. **Use Case:** User views profile of other users who are present in his assigned courses.

Actor: Professor, TA and Student.

Preconditions:

1. User must be logged in to the system.
2. Other users must exist in the course to which a user is assigned.

Actor Actions	System Responses
A user can browse his classmates, TAs and professor's profiles.	System allows a logged in user to view profile of other users.

34. **Use Case:** User follows TAs, students and professor of his assigned courses.

Actor: Professor, TA and Student.

Preconditions:

1. User must be logged in to the system.
2. Other users must exist in the course to which a user is assigned.

Actor Actions	System Responses
User adds TAs, students and professor of his assigned courses to his 'following' list.	System adds user in the 'following' list of the logged in user.

35. **Use Case:** User un-follows TAs, students and professor of his assigned courses from his 'following' users list.

Actor: Professor, TA and Student.

Preconditions:

1. User must be logged in to the system.
2. Other users must exist in the course to which a user is assigned.
3. Other users must exist in the 'following' list of the logged in user.

Actor Actions	System Responses
User removes TAs, students and professor of his assigned courses to his 'following' list.	System removes user in the 'following' list of the logged in user.

36. **Use Case:** User views his ‘following’ and ‘follower’ users list on the profile page.

Actor: Professor, TA and Student.

Preconditions:

1. User must be logged in to the system.
2. Other users must exist in the course to which a user is assigned.
3. Other users must exist in the ‘following’ and ‘follower’ list of the logged in user.

Actor Actions	**System Responses**
User views his ‘following’ and ‘follower’ users list on the profile page.	System displays the list of ‘following’ and ‘followed’ list of the logged in user.

37. **Use Case:** User deletes account.

Actor: Professor, TA and Student.

Preconditions:

1. User account exist.

Actor Actions	**System Responses**
1. Professor and TA create tags.	1 Tag is saved in the system.