Student Helper Portal

Division: D

Batch: D1

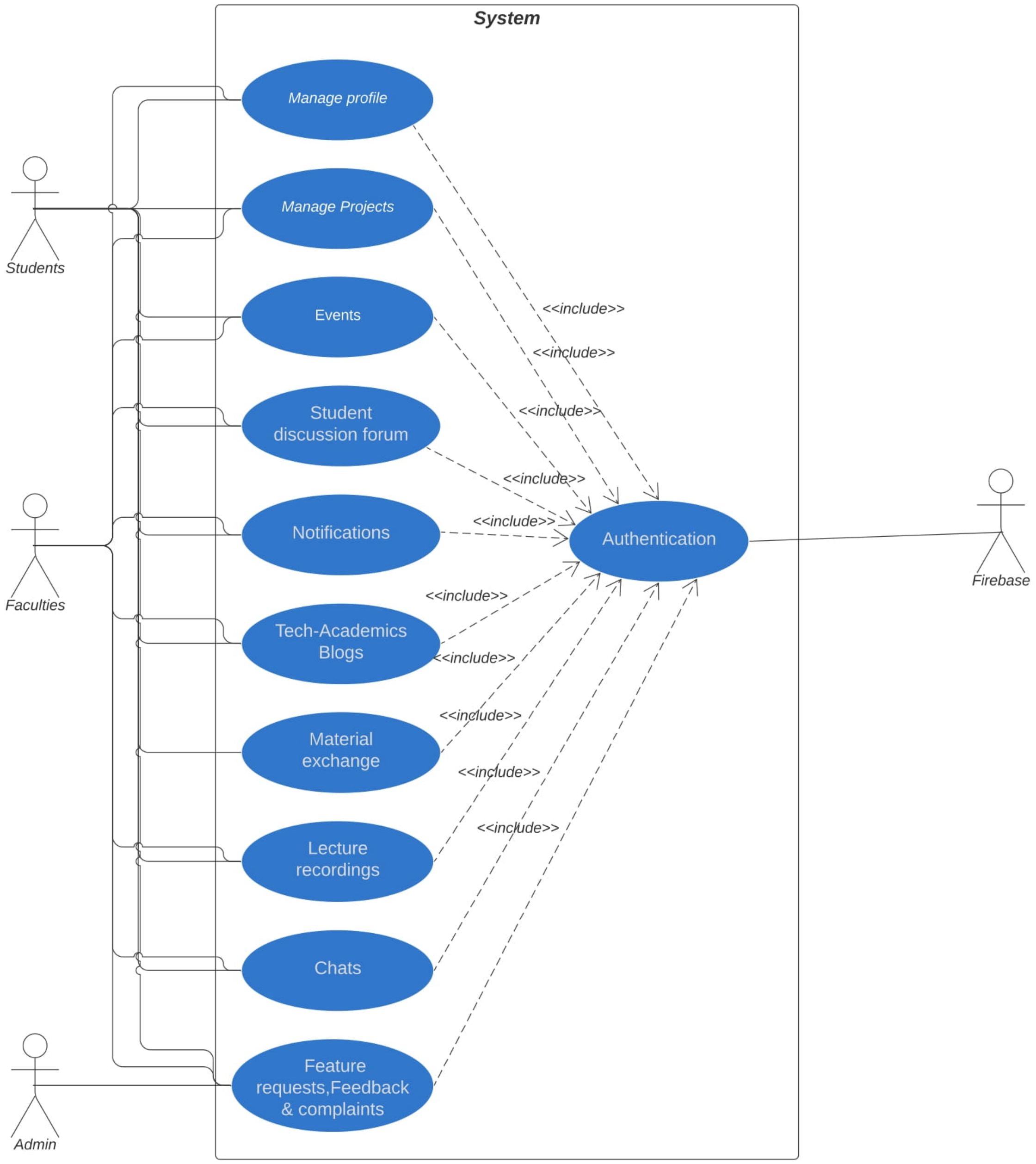
Group Details: Vatsal Sakariya (19BCE234), Harshil Sanghvi (19BCE238), Aayush Shah (19BCE245)

Use Case Diagrams & Use Case Specifications

Note - As discussed with Anuja Mam in lab, we just made 5 use case specifications as it would become too lengthy to create use case specifications for all use case diagrams.

Use case diagram SE practical 4

19BCE234 | 19BCE238 | 19BCE245



System

Name of use case

- System

Brief Description

- Used (Student/Faculty) can use this Student Helper Portal for the seamless teaching learning process and easily connecting with one another.
- User (Admin) will be responsible for monitoring the system and bring solutions to any complaints/feedback/requests.

Actors

- Student
- Faculty
- Admin

Basic flow

- User (Student/Faculty/Admin) need to login either via Github Auth or Google Auth.
- Once user (Student/Faculty) is successfully logged in, then they have access to wide array of functionalities such as:
  - Managing profiles
  - Managing projects
  - Events
  - Student discussion forum
  - Notifications
  - Tech-Academic blogs
  - Lecture recordings
  - Chats
  - Feature request, feedback and complaints
- Apart from this, students will have an exclusive access to section for Material exchange.
- Admin will be majorly accessing only Feature request, feedback and complaints

Alternate flows

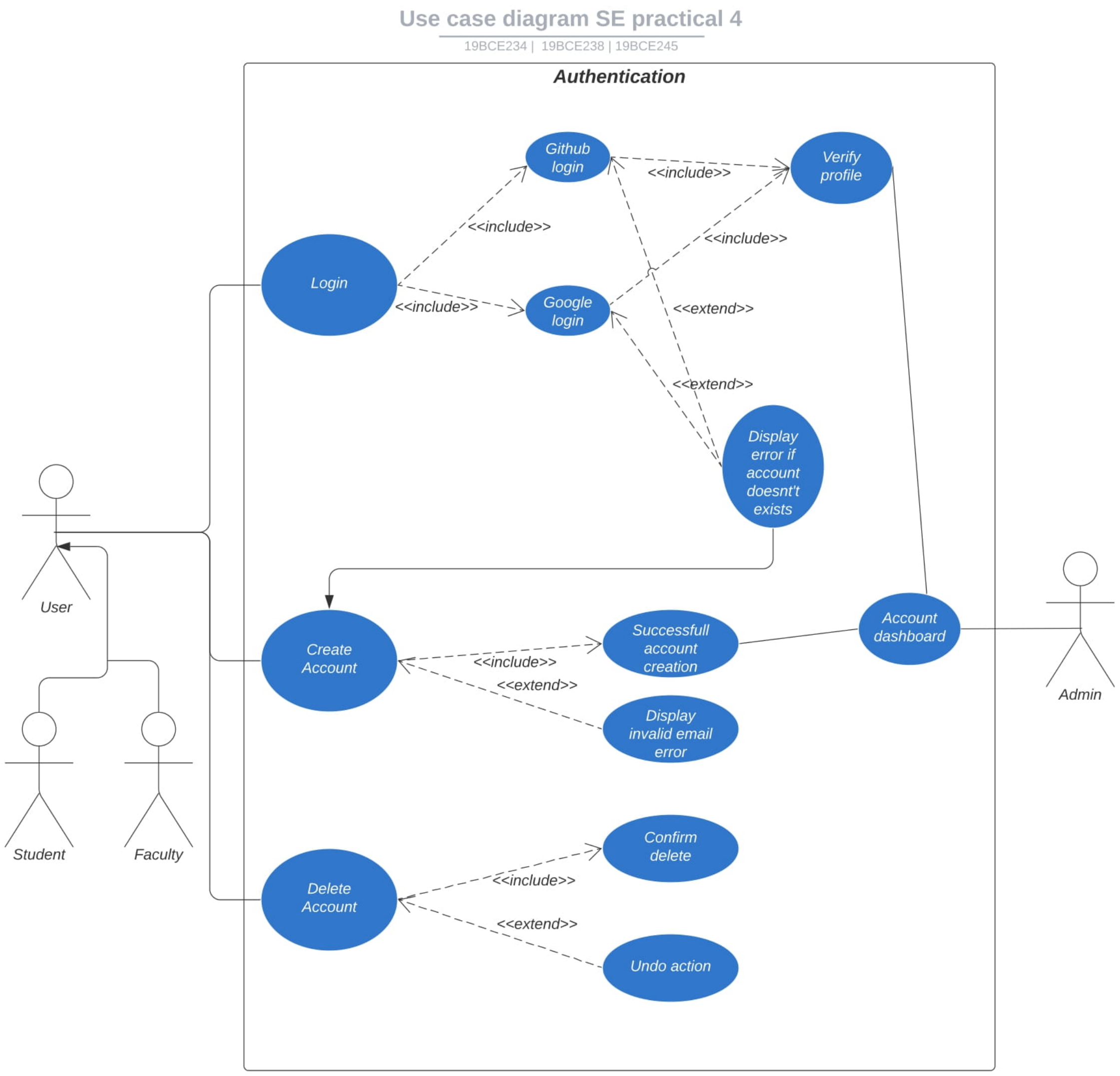
- User (Student/Faculty/Admin) will receive an error on login if his/her account doesn't exist. In such a scenario, user will be asked to create a new account. On successful account creation, user will be redirected to the account dashboard where they can verify their profile.
- In case of any error during any functionality, respective user (Student/Faculty/Admin) will be redirected back to their dashboard.

Pre-condition

- Visitor must be a registered user in the firebase.

Post-condition

- User (Student/Faculty/Admin) will be logged into his account and redirected to the account dashboard.



Authentication

Name of use case

- Authentication

Brief Description

- Used (Student/Faculty/Admin) to authenticate user in the firebase.
- User (Student/Faculty/Admin) can register/login with this functionality.

Actors

- Student
- Faculty
- Admin

Basic flow

- User (Student/Faculty/Admin) need to login either via Github Auth or Google Auth.

Alternate flows

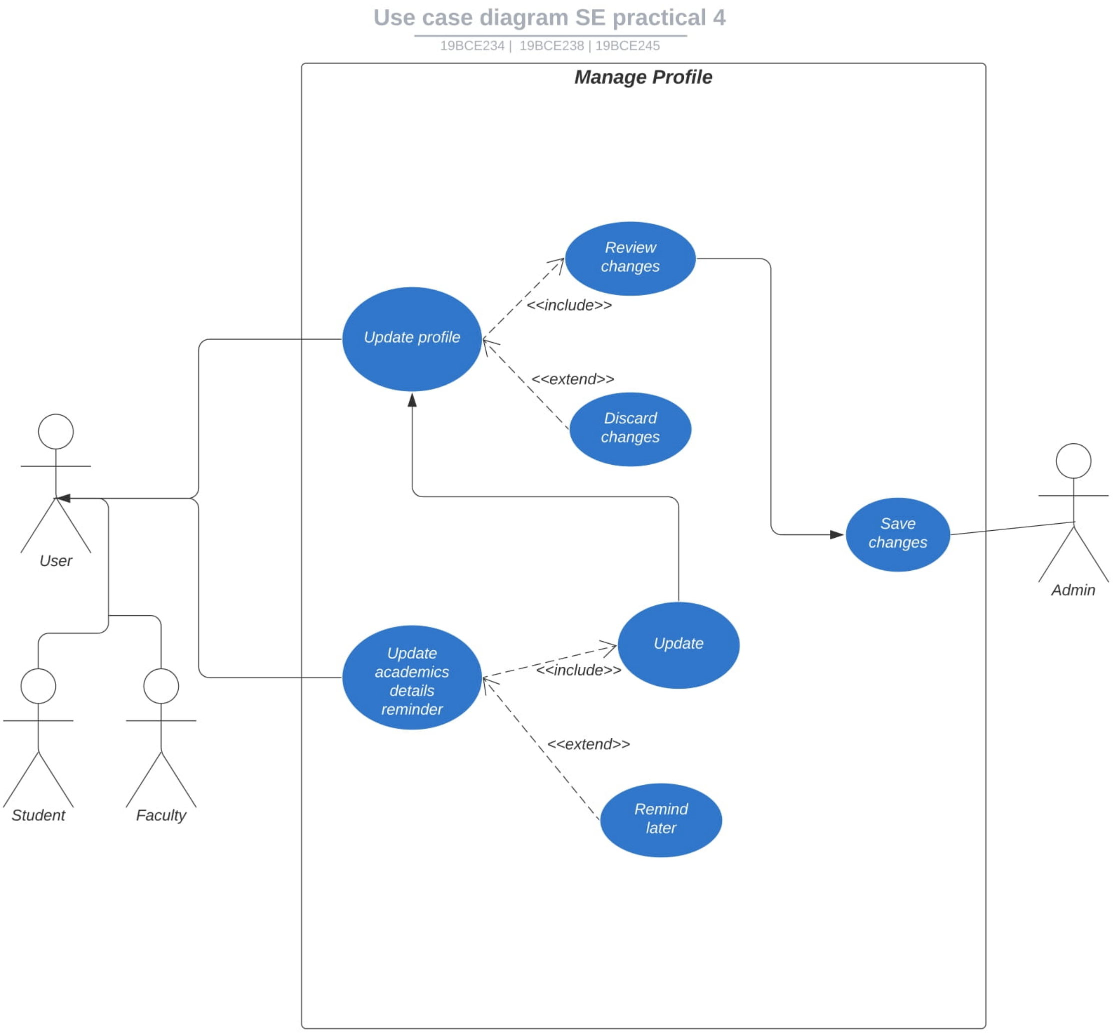
- User (Student/Faculty/Admin) will receive an error on login if his/her account doesn't exist. In such a scenario, user will be asked to create a new account. On successful account creation, user will be redirected to the account dashboard where they can verify their profile.
- In case of any error during any functionality, respective user (Student/Faculty/Admin) will be redirected back to their dashboard.

Pre-condition

- Visitor must be a registered user in the firebase.

Post-condition

- User (Student/Faculty/Admin) will be logged into his account and redirected to the account dashboard.



Manage profile

Name of use case

- Manage profile

Brief Description

- User (Student/Faculty) can manage their profile through this functionality
- It facilitates user (Student/Faculty) to update profile, academic details etc.

Actors

- Student
- Faculty
- Admin

Basic flow

- User (Student/Faculty) can update their profile details and save them after reviewing changes. They will also be prompted to update their academic details and save them. Admin can see the saved changes.

Alternate flows

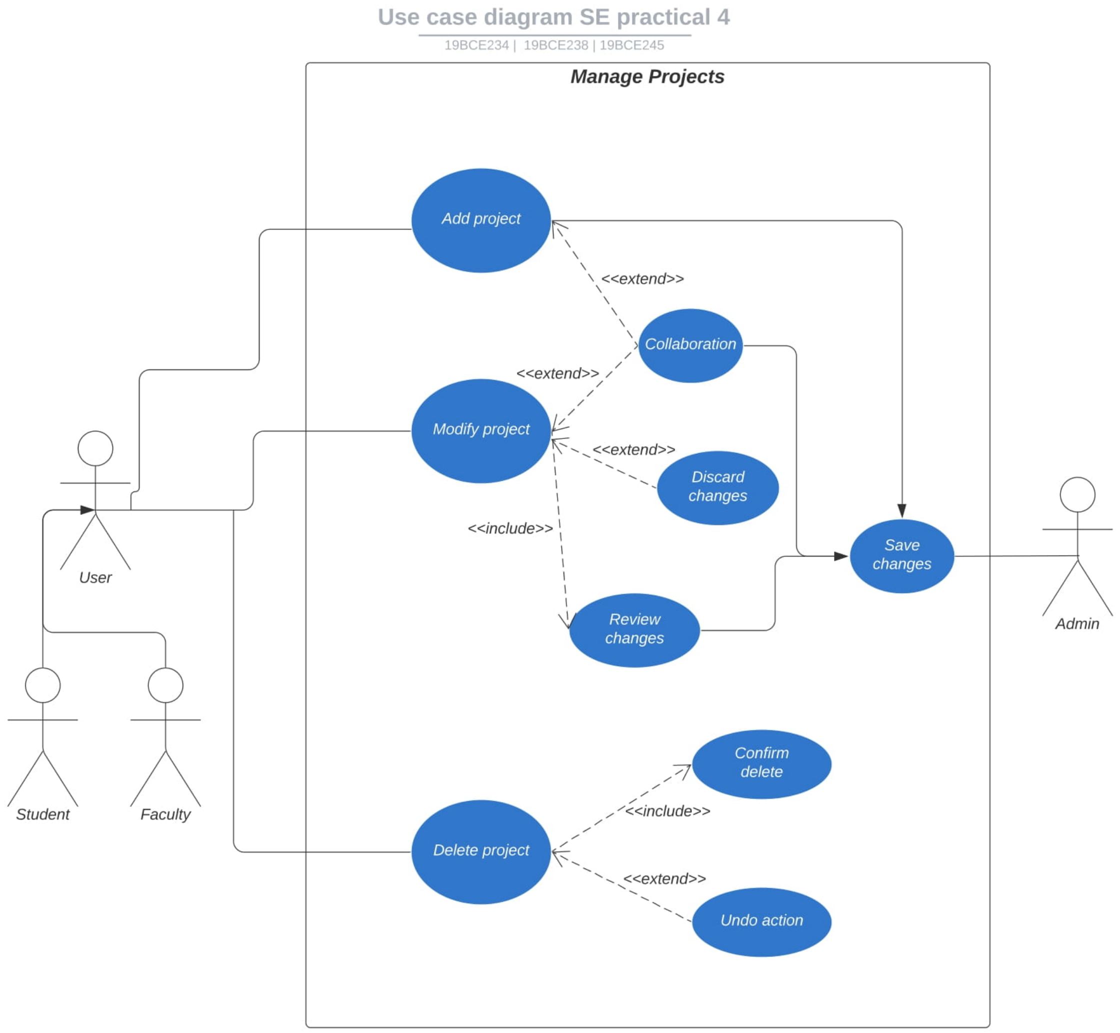
- In case user (Student/Faculty) doesn't wish to update their profile with new changes, they can always have an option to discard those changes.
- In case user (Student/Faculty) doesn't wish to update their academic details when prompted, they can always opt for a reminder later.
- In case of any error during any functionality, respective user (Student/Faculty/Admin) will be redirected back to their dashboard.

Pre-condition

- User (Student/Faculty) must be logged into their accounts where they wish to make required changes.

Post-condition

- On successful updation, User (Student/Faculty) will be able to see their changes reflecting in their dashboard as well as profile section.



Manage projects

Name of use case

- Manage projects

Brief Description

- User (Student/Faculty) can add/modify/delete their projects through this functionality.
- It facilitates user (Student/Faculty) to collaborate with other users as well.

Actors

- Student
- Faculty
- Admin

Basic flow

- User (Student/Faculty) can add a new project and save it.
- User (Student/Faculty) can modify an existing project and save them after reviewing changes.
- User (Student/Faculty) can also collaborate with other users (Student/Faculty) for the projects.
- User (Student/Faculty) can delete a project, which gets deleted from the firebase after getting a confirmation from the user.
- Admin can see the saved changes.

Alternate flows

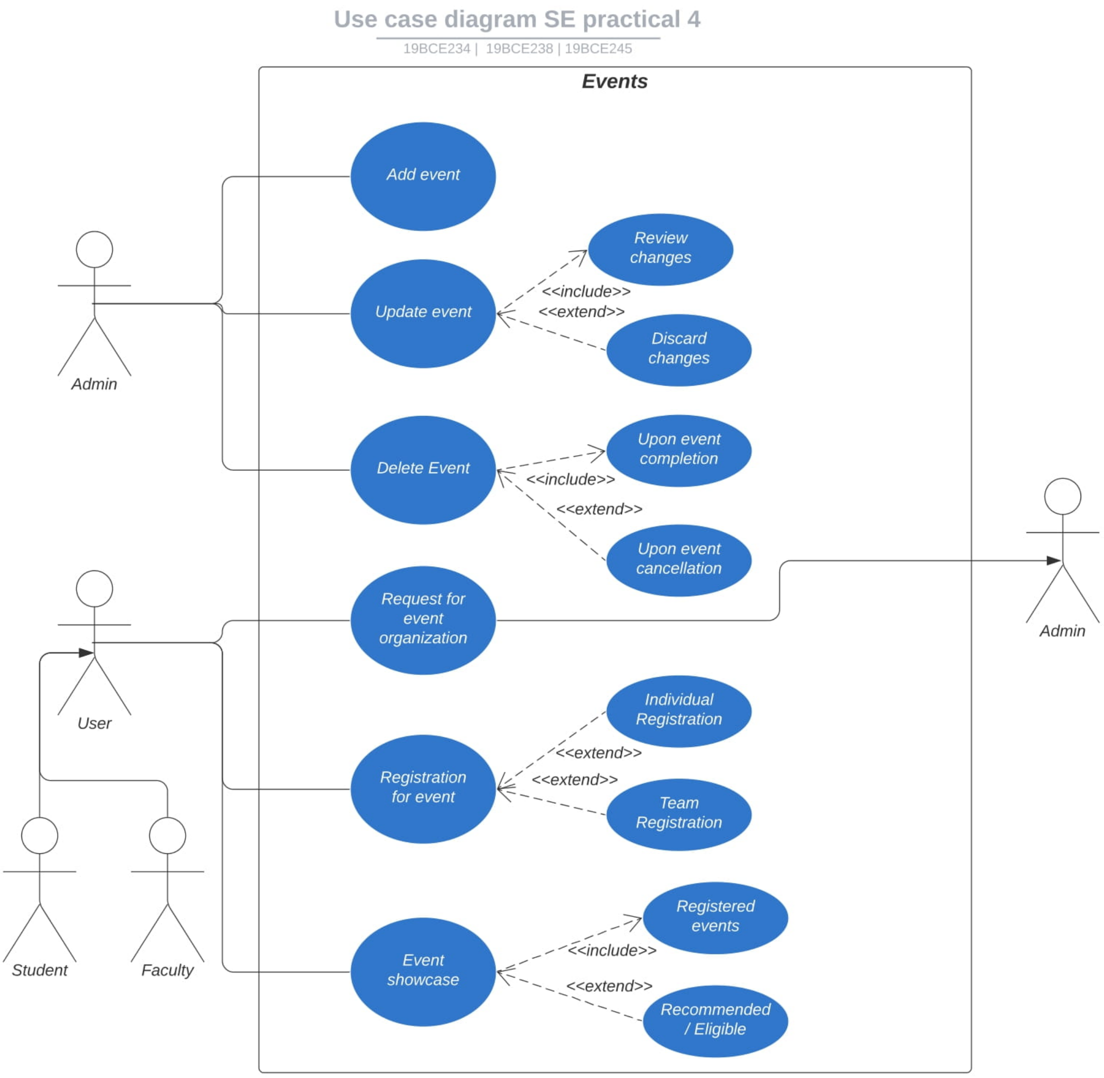
- In case user (Student/Faculty) doesn't wish to modify their projects with new changes, they can always have an option to discard those changes.
- In case user (Student/Faculty) doesn't wish to delete their projects, they can always undo their action.
- In case of any error during any functionality, respective user (Student/Faculty/Admin) will be redirected back to their dashboard.

Pre-condition

- User (Student/Faculty) must be logged into their accounts where they wish to manage their projects.

Post-condition

- On successful addition/modification/deletion, User (Student/Faculty) will be able to see their changes reflecting in their projects section.



Events

Name of use case

- Events

Brief Description

- User (Admin) can add/update/delete the events through this functionality.
- It also faciliates user (Student/Faculty) to request for event organization, register for event as well as events showcase.

Actors

- Student
- Faculty
- Admin

Basic flow

- User (Admin) can add a new event.
- User (Admin) can update an existing event and save them after reviewing changes.
- User (Admin) can delete a event when it is completed, which gets deleted from the firebase after getting a confirmation from the user.
- User (Student/Faculty) can request for event organisation which will be reviewed upon by Admin.
- User (Student/Faculty) can register for event showcased in the account. Registered can be done as an individual as well as a team.
- User (Student/Faculty) can see the events which are registered being showcased. Recommended/Eligible events will also be showcased.

Alternate flows

- In case user (Admin) doesn't wish to update the event with new changes, they can always have an option to discard those changes.
- User (Admin) have an optional functionality to delete the event upon completion. It is not mandatory.
- In case of any error during any functionality, respective user (Student/Faculty/Admin) will be redirected back to their dashboard.

Pre-condition

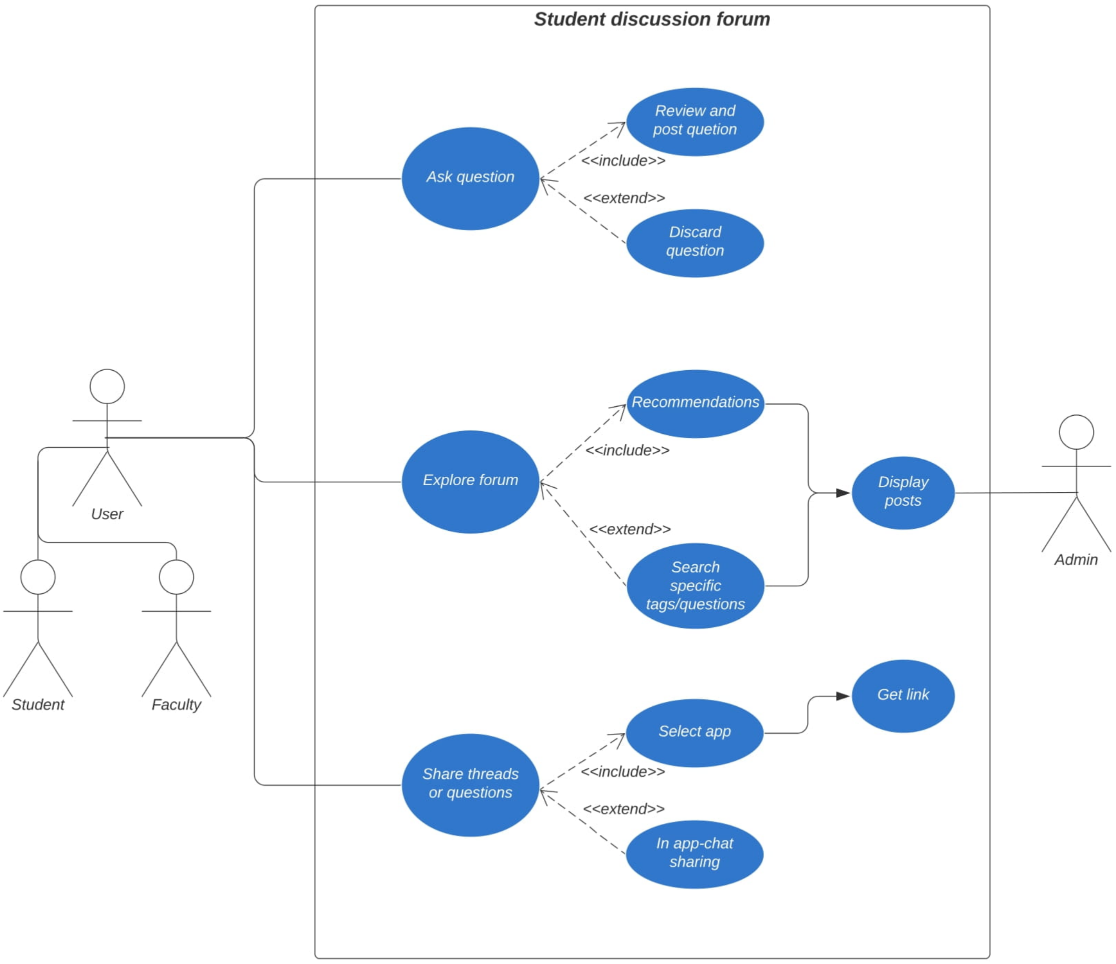
- User (Student/Faculty) must be logged into their accounts where they wish to register/request for events.

Post-condition

- On successful registration, User (Student/Faculty) will be able to see their registrations in events section.
- On successful raise of request for event addition, user (Student/Faculty) will be able to see their requests in events section.

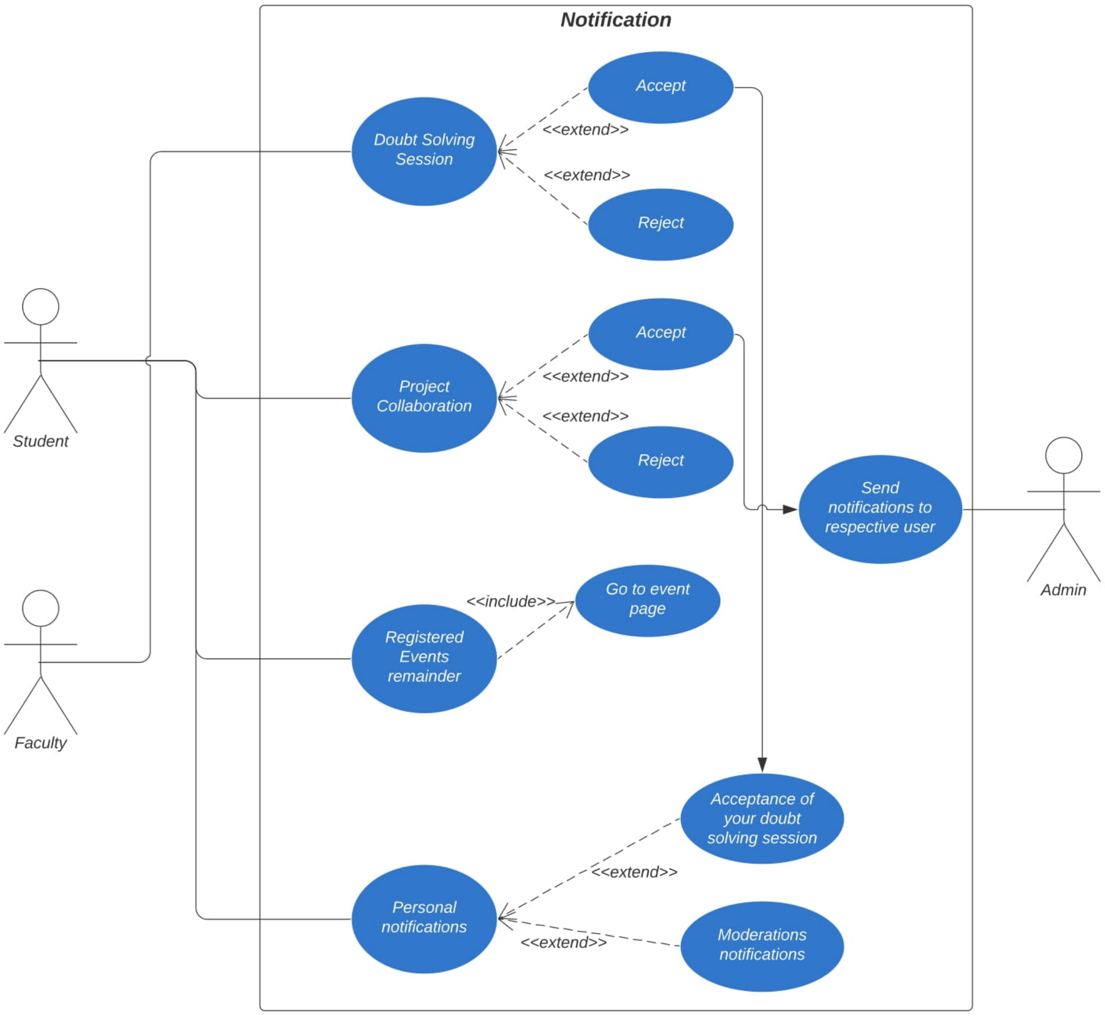
Use case diagram SE practical 4

19BCE234 | 19BCE238 | 19BCE245



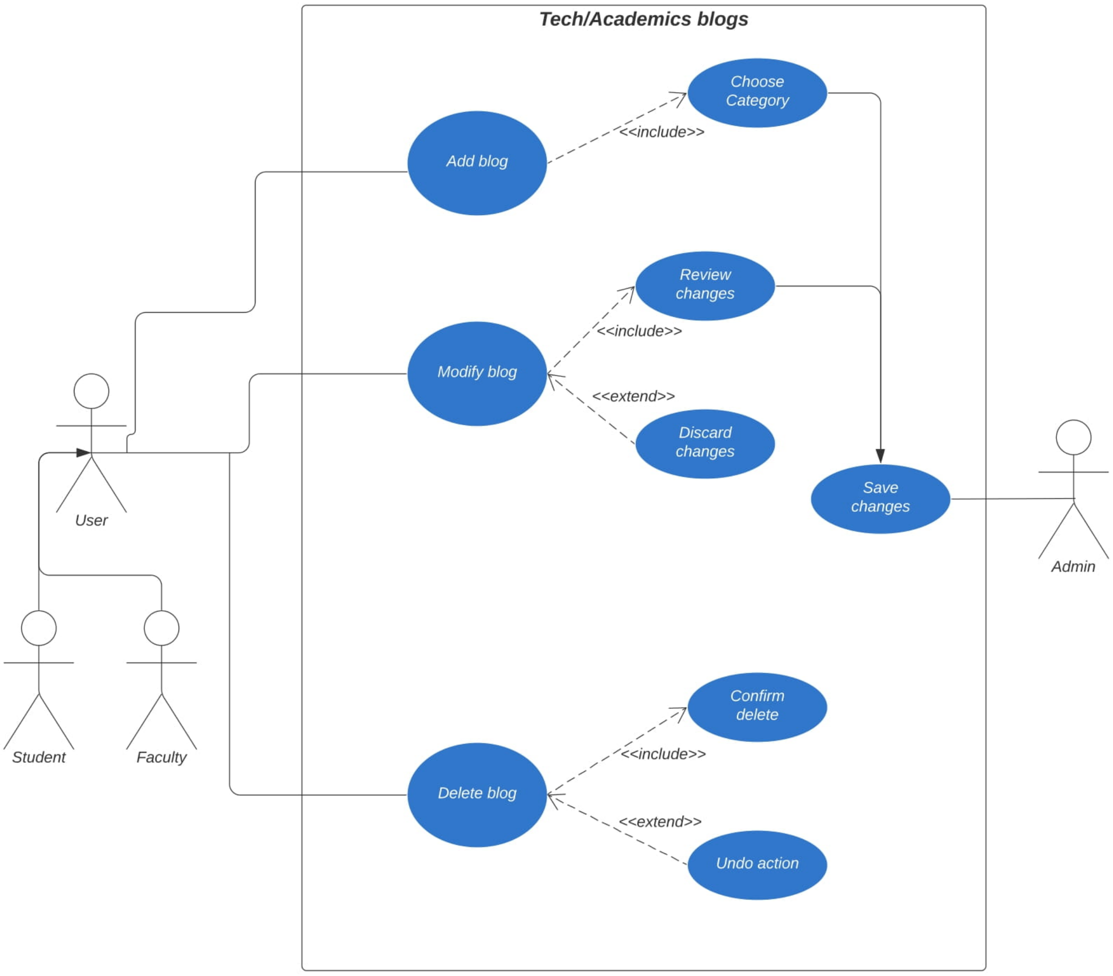
Use case diagram SE practical 4

19BCE234 | 19BCE238 | 19BCE245



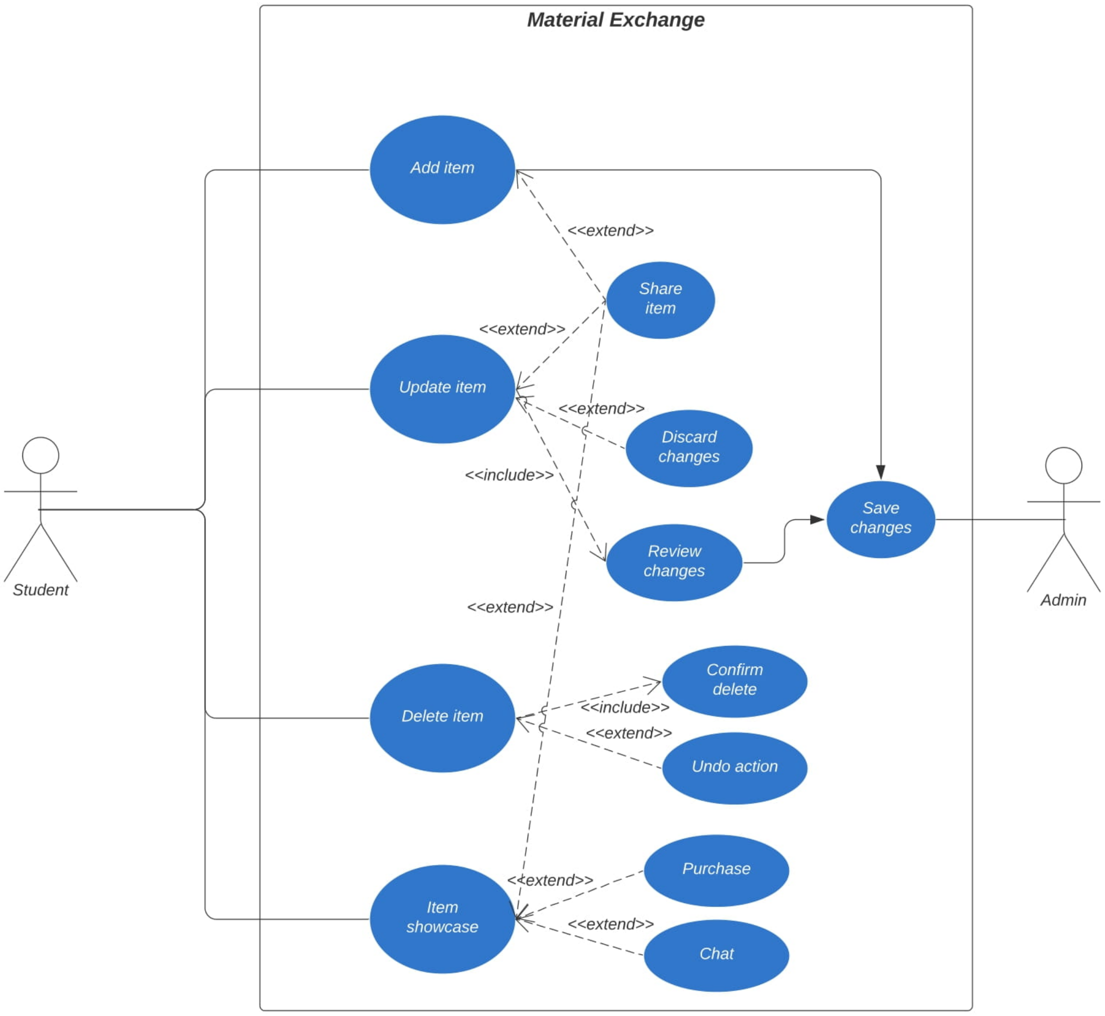
Use case diagram SE practical 4

19BCE234 | 19BCE238 | 19BCE245



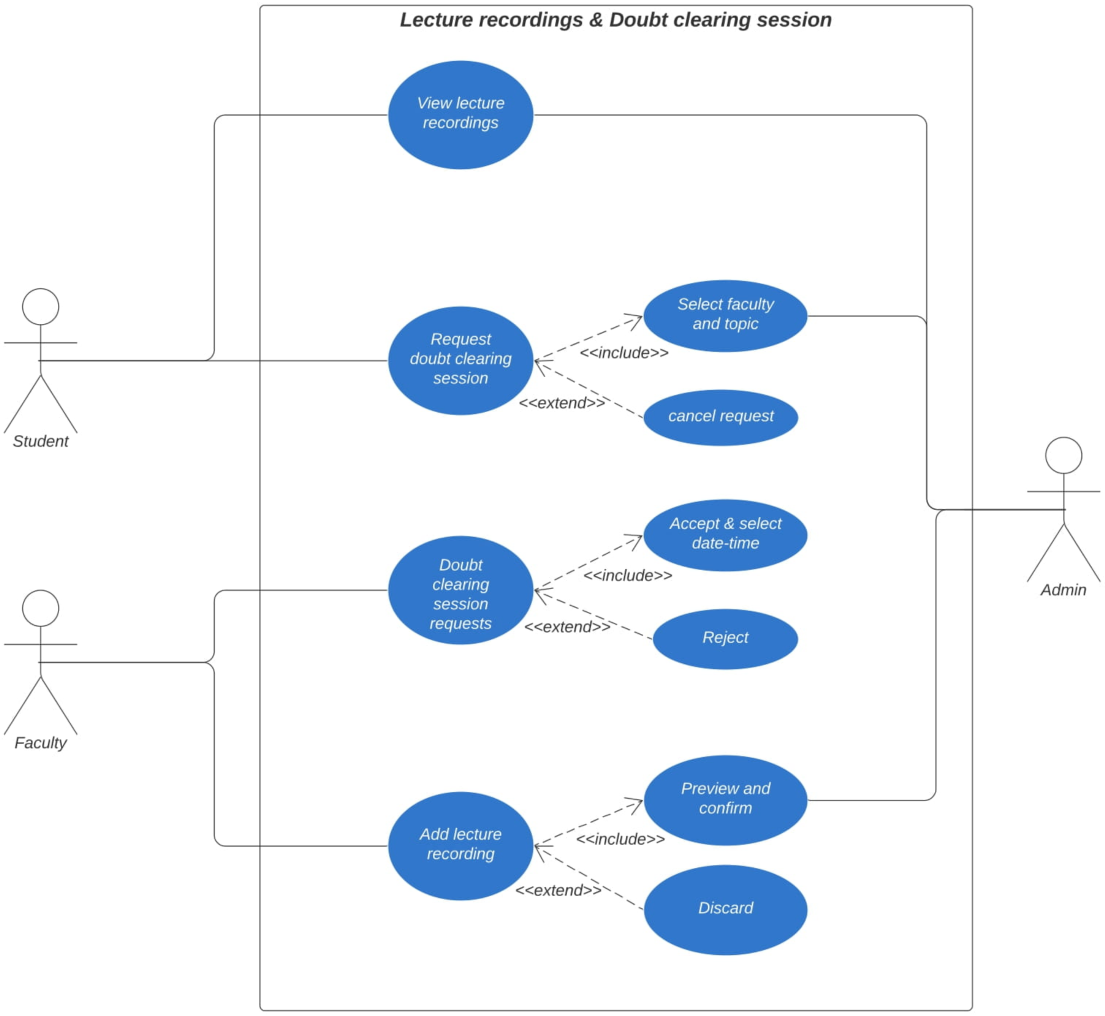
Use case diagram SE practical 4

19BCE234 | 19BCE238 | 19BCE245



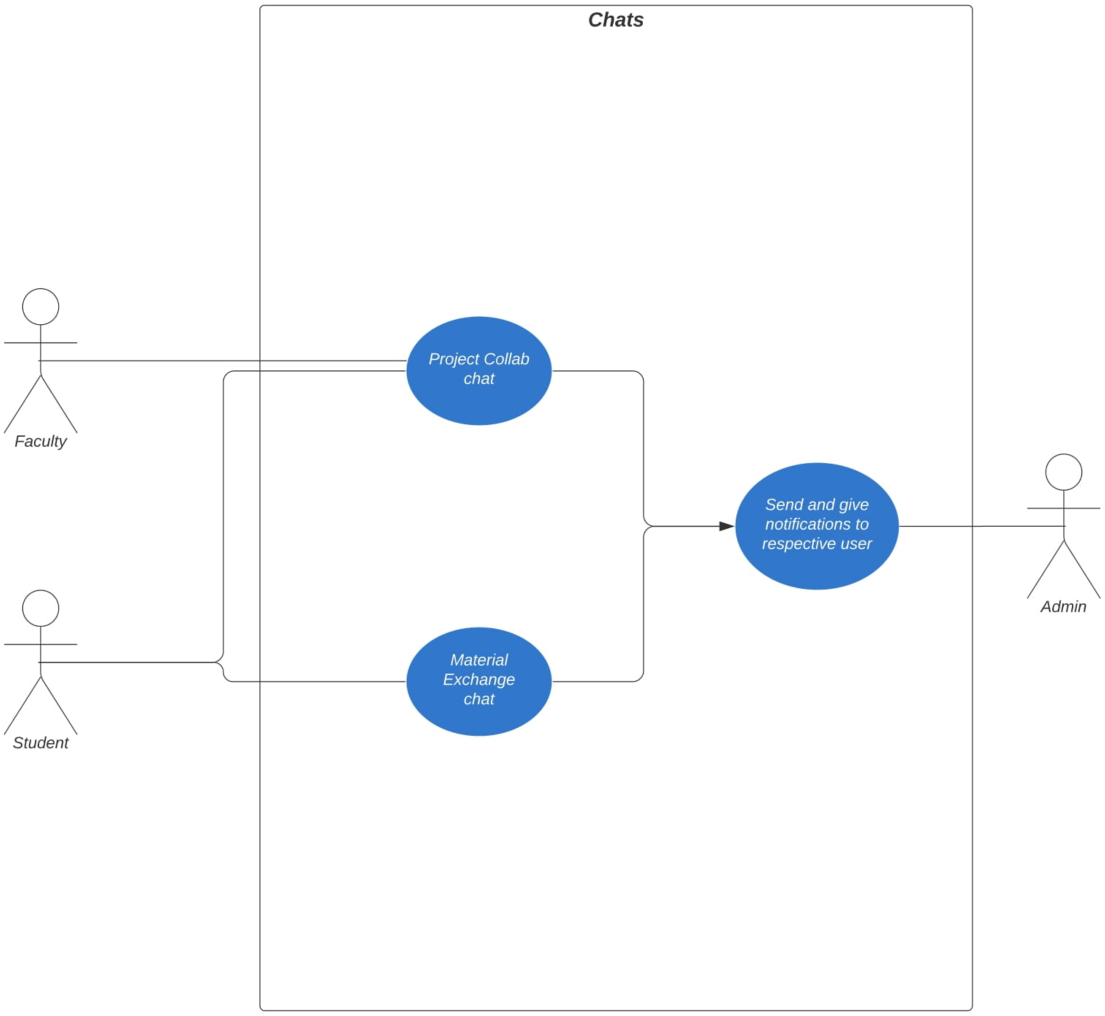
Use case diagram SE practical 4

19BCE234 | 19BCE238 | 19BCE245



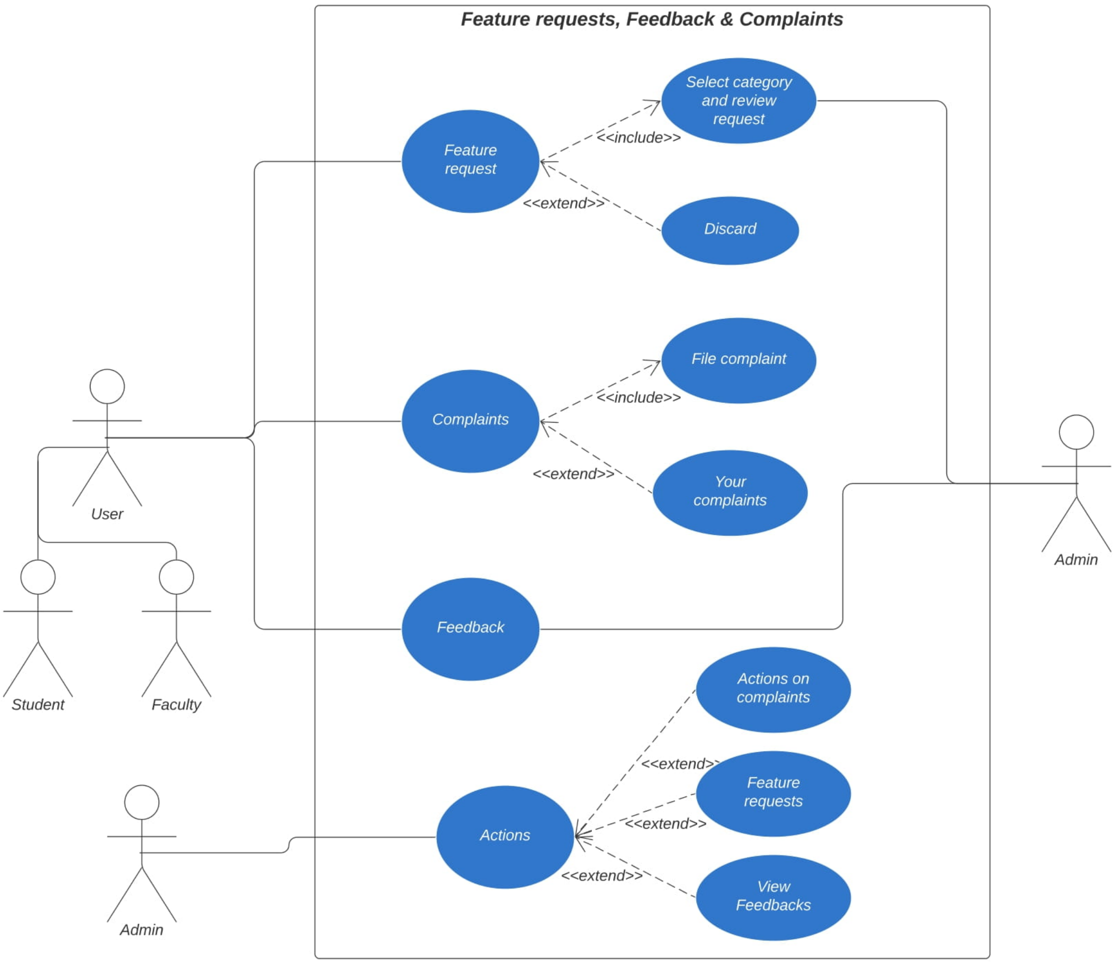
Use case diagram SE practical 4

19BCE234 | 19BCE238 | 19BCE245



Use case diagram SE practical 4

19BCE234 | 19BCE238 | 19BCE245


Feature requests, Feedback & Complaints
Select category and review request
Feature request
<<include>>
<<extend>>
Discard
File complaint
Complaints
<<include>>
<<extend>>
Your complaints
User
Student
Faculty
Admin
Feedback
Actions on complaints
<<extend>>
Feature requests
<<extend>>
Actions
<<extend>>
View Feedbacks
Admin