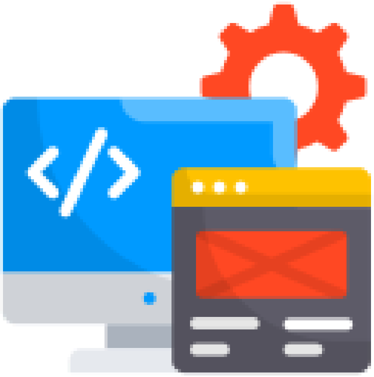
Software Requirements Specification

Student Departmental Query Management System



CS19BTECH11045 - Kommana Vikas
CS19BTECH11042 - Routhu Prashanth
CS19BTECH11005 - M. Shathanand Sai
CS19BTECH11040 - Eshwar T

Table of contents

1. Introduction

a. ***Purpose:***-

The purpose of this document is to make a baseline agreement between client(MVP sir) and development team about the functional and nonfunctional requirements of the Student Departmental query management system.

b. ***Scope***:-

This software system will be an online portal for IITH which manages the academic approvals required by the students. The student can request approvals within the department, like DUGC approval or HOD approval or faculty approval through the system.

c. ***Definitions,Acronyms and Abbreviations***:-

SDQMS- Student departmental query management system

d. ***Remark***:-

This is a working document and it can be subject to change during the course of design and it is incomplete by definition. This will be continuously refined and reviewed. The following is the current definition of the problem to be solved.

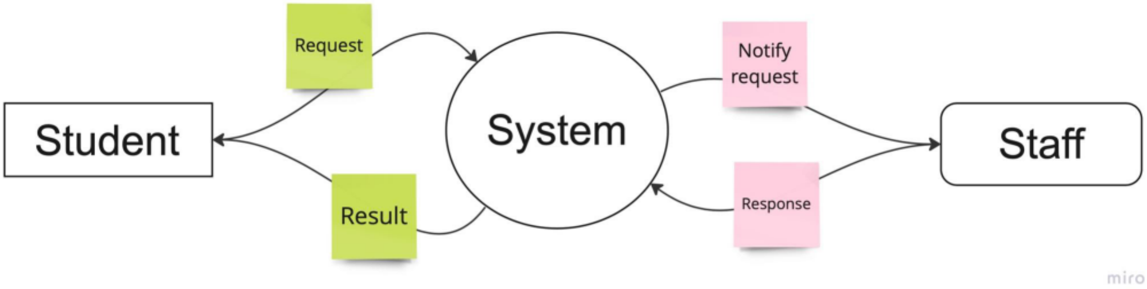
2. Overall description

a. ***Product Perspective***:-

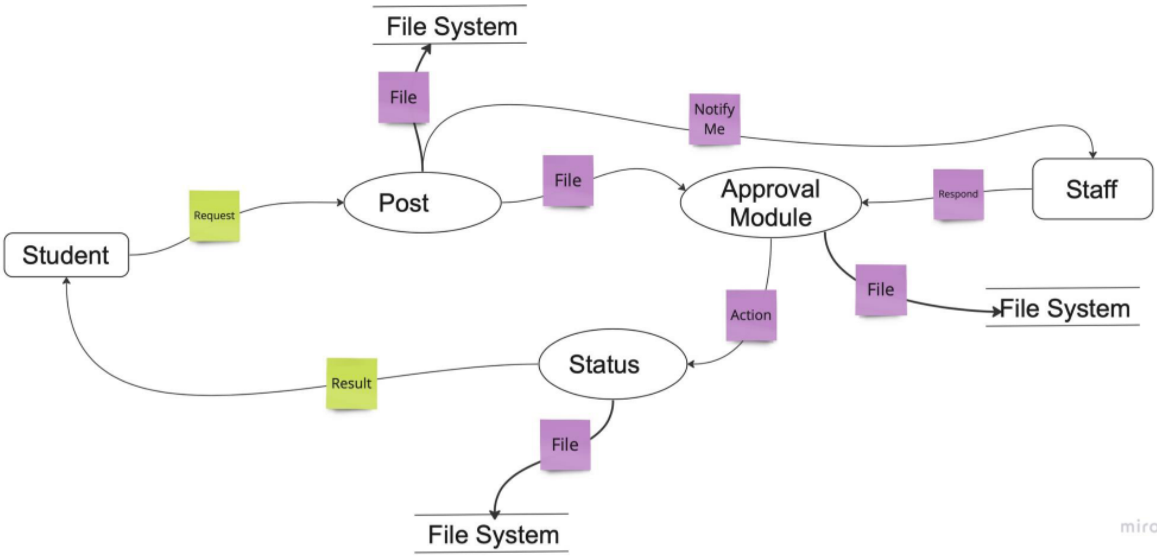
SDQMS is meant to serve as communicative software between students and faculty regarding approvals of various academic requests. It is intended to replace the existing system of communications via email.

b. Product Functions

i. Context diagram



Next level context diagram:-



c. ***User Characteristics***:-

   i. Students:-

      These are the primary users of the portal. They post requests which require approval on different levels in the academic hierarchy of the department.
      Use case options:- signup,login, mode, resolve, sign,reject,remark, escalate, edit profile,search(tags), search(requests)

   ii. Faculty:-

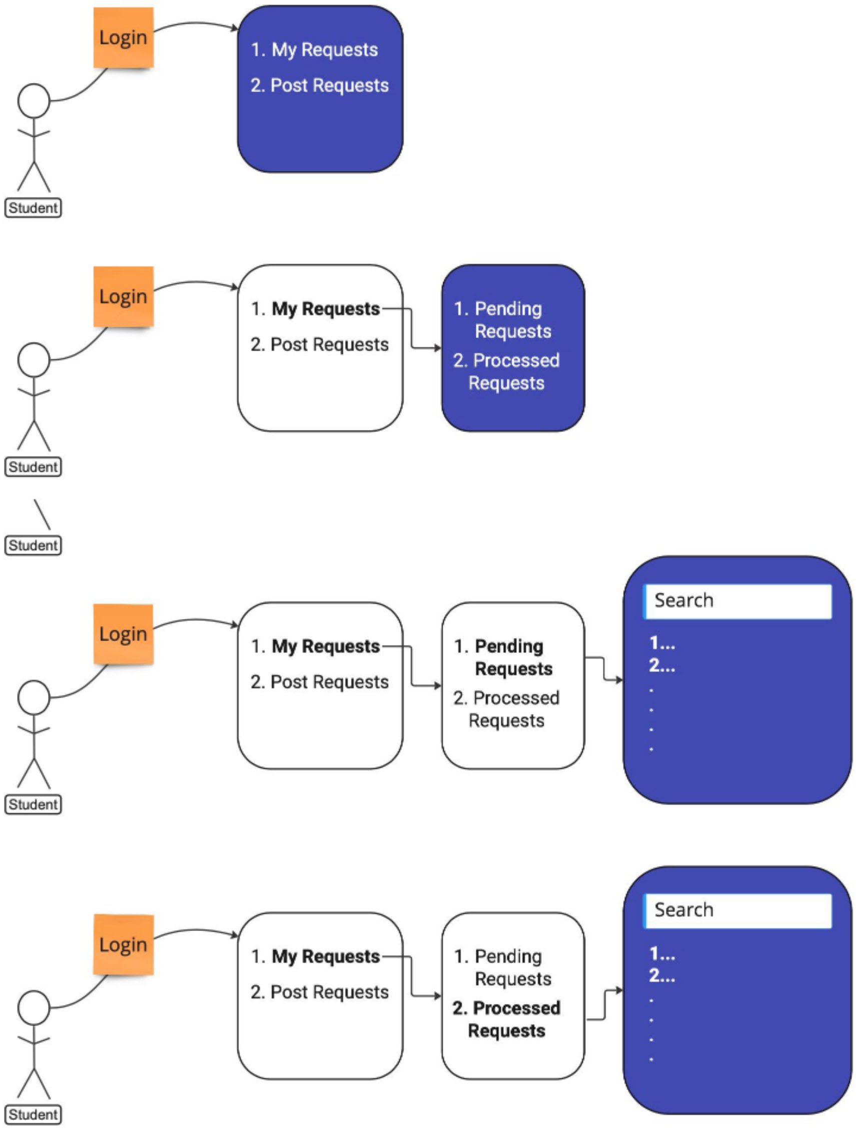
      These are the primary administrators in the system. They resolve requests made by students and take them for further consideration of hierarchy as and when required.
      Use case options:-signup, login, edit profile, pending requests, processed requests, post new request, search(tags), search(requests)

d. ***Assumptions and dependencies:-***

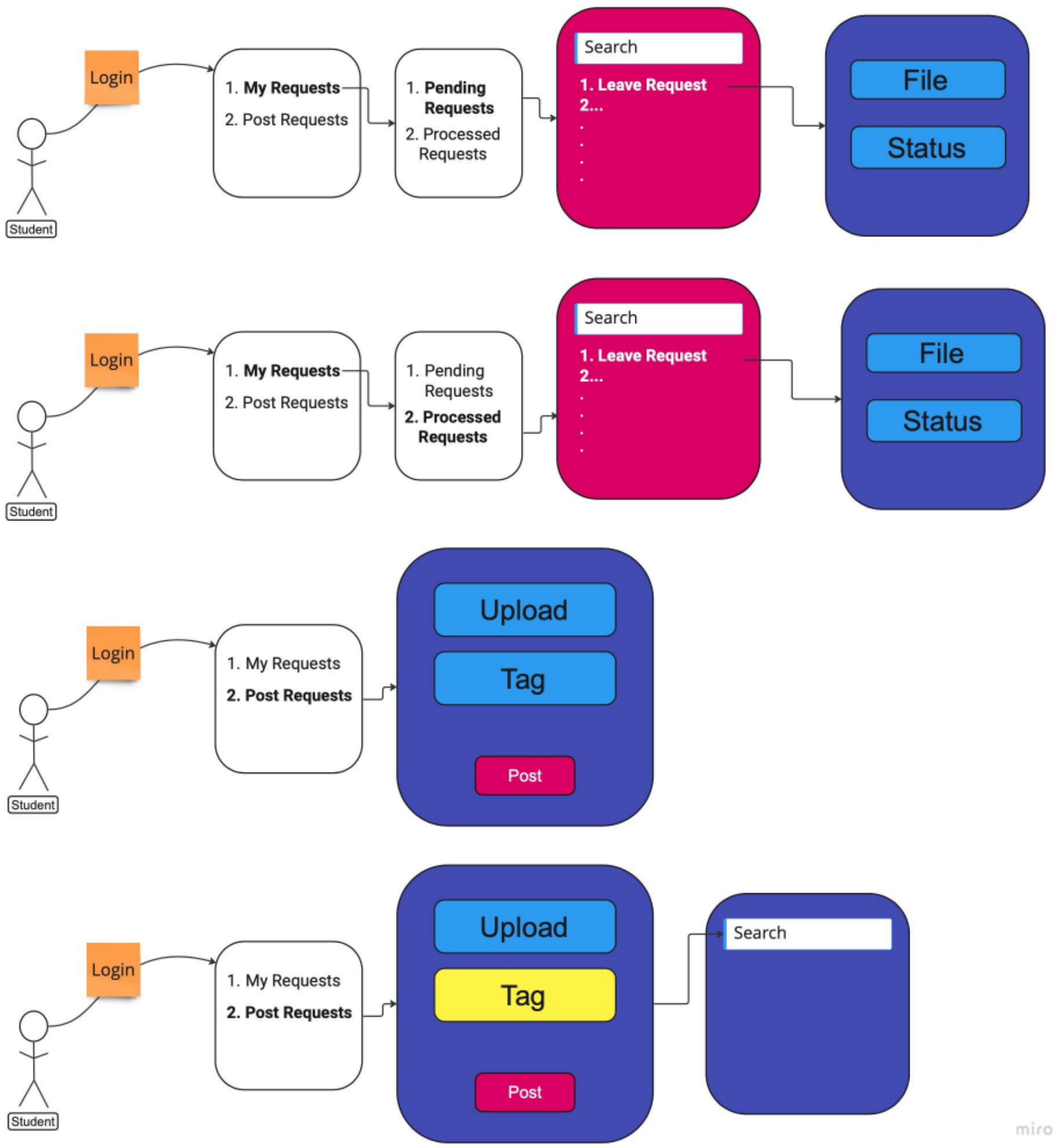
   User should have the hardware has chrome and supports the latest version of it.
   Login system depends on LDAP authentication
   Information provided for profile creation is assumed to be correct and verified

e. ***Prototype***:- main functionality is shown(edit profile is omitted as it is standard)

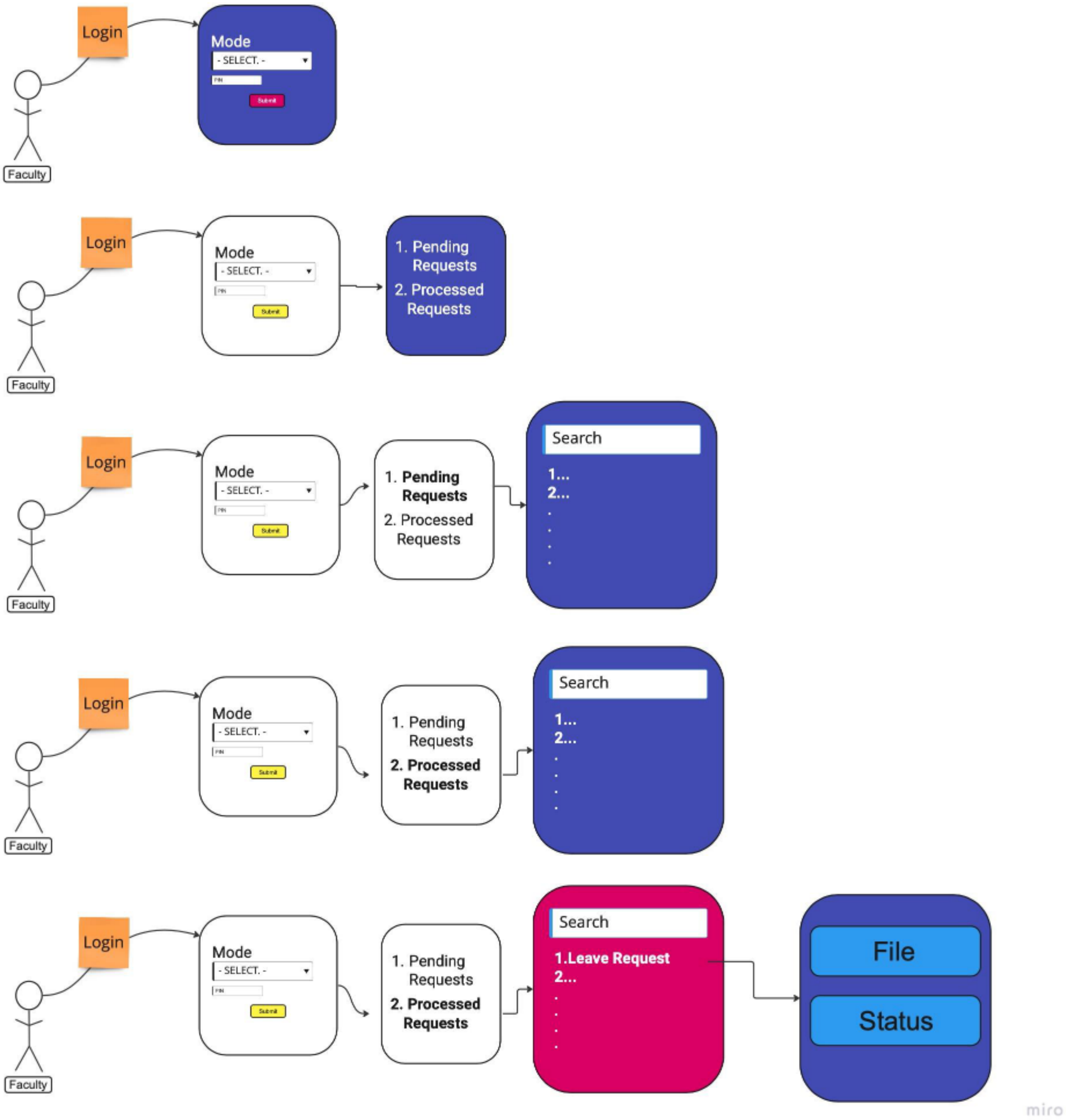
   i. Student side

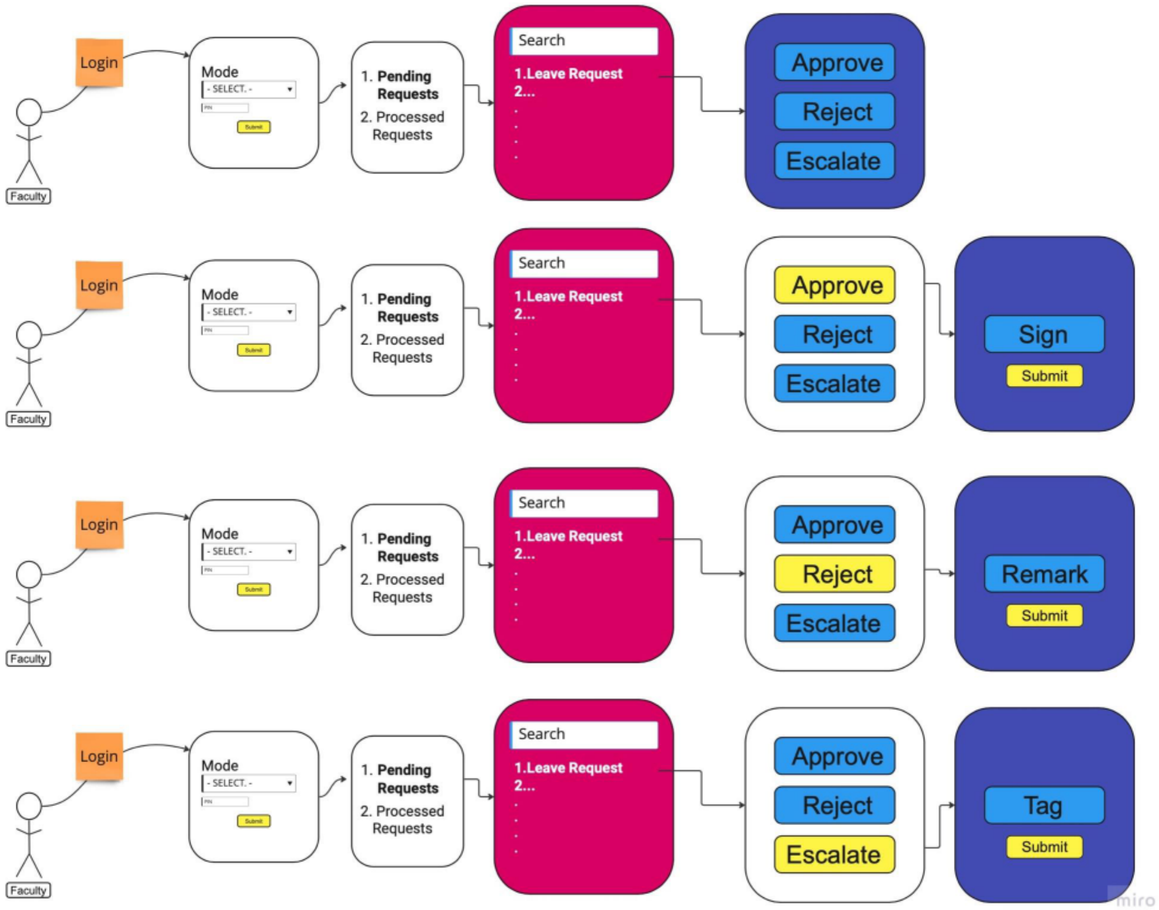


Login
Student
1. My Requests
2. Post Requests
Login
Student
1. My Requests
2. Post Requests
1. Pending Requests
2. Processed Requests
Student
Login
Student
1. My Requests
2. Post Requests
1. Pending Requests
2. Processed Requests
Search
1...
2...
.
.
.
.
Login
Student
1. My Requests
2. Post Requests
1. Pending Requests
2. Processed Requests
Search
1...
2...
.
.
.
.


Login
Student
1. My Requests
2. Post Requests
1. Pending Requests
2. Processed Requests
Search
1. Leave Request
2...
File
Status
Login
Student
1. My Requests
2. Post Requests
1. Pending Requests
2. Processed Requests
Search
1. Leave Request
2...
File
Status
Login
Student
1. My Requests
2. Post Requests
Upload
Tag
Post
Login
Student
1. My Requests
2. Post Requests
Upload
Tag
Post
Search
miro

ii. Faculty side




Login
Faculty
Mode
- SELECT. -
PIN
Submit
1. Pending Requests
2. Processed Requests
Search
1.Leave Request
2...
Approve
Reject
Escalate
Login
Faculty
Mode
- SELECT. -
PIN
Submit
1. Pending Requests
2. Processed Requests
Search
1.Leave Request
2...
Approve
Reject
Escalate
Sign
Submit
Login
Faculty
Mode
- SELECT. -
PIN
Submit
1. Pending Requests
2. Processed Requests
Search
1.Leave Request
2...
Approve
Reject
Escalate
Remark
Submit
Login
Faculty
Mode
- SELECT. -
PIN
Submit
1. Pending Requests
2. Processed Requests
Search
1.Leave Request
2...
Approve
Reject
Escalate
Tag
Submit
miro

3. Specific Requirements

a. ***External Interface***:-

The external interface looks quite similar to the prototyped version above and the user can conveniently move across portals and use features seamlessly.

b. Functional requirements

i. Use case Scenario

1. User signup

Purpose	To register user in the portal
Primary actor	faculty,student
Input data	Details of the respective user
Pre-conditions	Internet connectivity, valid email address
Post-conditions	Account is created for respective user and details are stored in database of the system
Basic flow	1.User visits the website 2.clicks on signup 3.Enters details Details for faculty include mode and tenure of the mode and pin for each mode 4.Click on register

	5. Corresponding portal is logged in
Alternate flow	1.Email address already in use 2.Not a valid email address

2. User login

Purpose	User will login into system with existing profile
Primary actor	Faculty, student
Input data	Username and password
Output data	Corresponding data in the portal opened
Invariants	User Information
Pre-conditions	User is a registered one, enters correct password, connected to internet
Post-conditions	Appropriate portal opens for the user of entered profile and users computer is supplied with apt cookie
Basic flow	Web page looks up profile data and returns the matched

	cookie. Its updated to match new user data
Alternative flows	Error message of invalid login credentials

3. Post request

Purpose	User wants to request an approval from department
Primary actor	Student
Input data	File, associated faculty member
Output data	File available to the faculty
Invariants	file
Pre-conditions	User is logged in, file exists with user
Post-conditions	Faculty will be able to view and respond to the request
Basic flow	The user uploads file using the upload button and tags faculty members who he want to request by their id by clicking on tag and clicks on the post button. The file then gets uploaded to the

	server. Visible to faculty tagged in pending requests

4. Mode

Purpose	A user wants to operate in a different mode. For example faculty may be HOD, or Dugc member or instructor
Primary actor	Faculty
Input data	Password for he mode (authentication code)
Output data	Corresponding data in the portal opened
Pre-conditions	Authentication code is given to faculty for respective positions.
Post-conditions	Appropriate portal opens for the user of entered profile and users computer is supplied with apt cookie
Basic flow	Web page looks up profile data and returns the matched cookie. Its updated to match new user data

Alternate flow	Error message of invalid password

5. Pending requests

Purpose	The user will be able to see the requests he has made or received by him which are pending
Primary actor	Student, faculty
Input data	No input data (Click on the button pending requests)
Output data	All the pending requests are visible on the portal
Pre-conditions	User must be logged in
Post-conditions	User is able to view each request by clicking view request
Basic flow	User clicks on the button pending requests. The requests are displayed in the order with latest one being at the top
Alternate flow	If there are no pending requests then no pending requests message is

	displayed

6. Processed requests

Purpose	User wants to see completed requests
Primary actor	Student,faculty
Input data	No input data(user clicks on Processed requests)
Output data	List of requests that were previously resolved sorted in timeline
Pre conditions	User is logged in
Post conditions	The user can view requests shown in the resulting portal
Basic flow	Web page looks up previous requests data and returns the matched cookie.

7. View request

Purpose	To see the details of all types of requests
Primary actor	Student, faculty
Input data	No input data(user clicks on request)

Output data	Request is opened and details of the request are shown
Invariants	Viewing doesn’t modify the request
Pre-conditions	User is logged in and requests are not empty
Post-conditions	For students, the details of the request are shown. For faculty, resolve option is shown if its a pending request
Basic flow	User clicks on the request to be viewed and data of the request is fetched appropriately

8. Status of request

Purpose	To know the status of the request whether approved or not
Primary actor	User, faculty
Action	Clicks on status
Outcome	Status is displayed
Pre condition	Should be in the portal of requests
Post condition	The status is displayed

9. Resolve request

Purpose	Purpose is to resolve pending requests
Primary actor	Faculty in different modes
Input data	Signature if approved/escalated, else remark is written for disapproval
Output data	Signal which sends update to the student portal/ Further escalation
Pre-conditions	User should be logged in and the request should be a pending one
Post-conditions	The request is pushed into processed requests
Basic flow	The user clicks on resolve request in which he gets 3 options (approve, reject, escalate) and he chooses one which changes the state of request from pending to processed in the

	current mode

10. Escalate request

Purpose	To forward the request to dugc or hod as per hierarchy
Primary actor	Faculty in any of instructor,dugc,hod mode
Input data	No input data(except sign)
Output data	No output data explicitly (a notification is sent to person tagged)
Pre-conditions	The given request is in pending state
Post-conditions	The file appears in the pending files of the person being escalated to
Basic flow	The faculty in one mode will click on escalate following which he tags whom he wants to forward it to

11. Search person(appears as search bar in interface)

Purpose	To search for user while tagging or escalating
Primary actor	Student, faculty(in all modes)
Input data	Persons name/id to be searched for
Output data	Results in pointer to that person if he exists
Pre-conditions	User is logged in and using the corresponding search bar to search people
Post-conditions	If found then the corresponding person can be tagged
Basic flow	User enters the id or name to be searched and clicks on search icon. If found the user can be tagged to the corresponding post

12. Edit profile

Purpose	To change details such as password for profile(all users) or signature,pin(faculty) or other details
Primary actor	Faculty, student
Input data	The updated details
Pre-conditions	Profile should exist and should be logged in
Post-conditions	Updates the database with the new data
Basic flow	The user clicks on edit profile and makes changes and submits them

13. Search Request(appears as search bar in interface)

Purpose	To search for a particular request
Primary actor	student,faculty(in any mode)
Input data	post name to be searched for
Output data	The corresponding post if it exists

Pre-conditions	User is in My requests portal
Post-conditions	User can view the request if present
Basic flow	The user types the request to be searched in the search bar and clicks on the search icon. The results are fetched appropriately

c. ***General system requirements***:-

i. Should be able to run in all updated versions of browsers
ii. Should be fault tolerant with regard to data when the system fails.
iii. All data will be saved in a database which allows concurrent and consistent access.

d. ***Performance requirements***:-

The system will support at least 100 concurrent users(just a rough number for the sense of reliability). The response time of the system will typically be less than 5 seconds. System should be available at all times. When the user is uploading the response time will be a bit slow.