Software Requirements Specification

for

College PathFinder

Version 1.0

Prepared by

Group 1: **Group Name: DominatorDevs**

Kumar Shivam	**220562**	**kshivam22@iitk.ac.in**
Manan Kumar	**220607**	**manankumar22@iitk.ac.in**
Ayushmaan Jay Singh	**220276**	**ayushmaanj22@iitk.ac.in**
Anitej Jain	**220153**	**anitej22@iitk.ac.in**
Prathamesh Baviskar	**220285**	**baviskarp22@iitk.ac.in**
Ayush Meena	**220268**	**ayushmeena22@iitk.ac.in**
Mayank Gupta	**220638**	**manyankg22@iitk.ac.in**
Param Soni	**220752**	**paramsoni22@iitk.ac.in**
Astitva Roy	**220248**	**Astitvaroy22@iitk.ac.in**
Ayush Patel	**220269**	**Ayushpatel22@iitk.ac.in**
Ayush Meena	**220268**	**ayushmeena22@iitk.ac.in**

Course:CS253

Mentor: Yemike Abhilash Chandra

Date: 26th January,2024

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Ayushmaan Jay Singh Kumar Shivam Baviskar Prathmesh Param Soni Manan Kumar Anitej Jain Astitva Roy Mayank Gupta Ayush Meena Ayush Patel	This document pertains to the specified version of the project, outlining its features, functionalities, and relevant information.	26/1/23

1 Introduction

1.1 Product Scope

Description:
The College PathFinder is a comprehensive educational search and recommendation platform designed to assist students in finding suitable colleges based on a wide range of criteria. The product encompasses various features to enhance the user experience and streamline the college selection process. It allows users to set a specific All India Rank (AIR) range, helping them narrow their options based on academic performance. College PathFinder ensures that students are presented with colleges that align with their ranking, optimizing the chances of finding a suitable educational institution.

Objectives and Benefits:

- It will help users to choose between their available college options pleasantly.
- It will allow users to combine criteria such as branch and placement range and, based on available facilities of interest, such as sports, swimming pools, or libraries, to find colleges in their desired sweet spot.
- It will allow users to visually explore the college campus, making it easier to make decisions based on college locality.
- It will allow users to communicate directly with the students studying on those campuses.
- It will give users access to the various statistical data related to courses, their past grading schemes, and reviews by the students who have done that course.

1.2 Intended Audience and Document Overview

Intended Audience
This document can benefit different categories of people, including college-going students and college-searching students, but also cater to a wide range of audiences simultaneously, including developers, content creators, stakeholders, project managers, testers, users, and approvers. The intent and purpose of the app for each stakeholder may vary significantly.

Document Overview

- **Product Overview:**
  - Describes the context and origin of the College PathFinder.
  - Outlines major features and objectives.
  - Provides a brief on the intended audience and stakeholders. Catering to a diverse group of individuals, including developers, content creators, stakeholders, project managers, testers, users, and approvers.
- **Product Perspective:**
  - Illustrates a simple diagram highlighting major components, subsystem interconnections, and external interfaces.
  - Offers a high-level view of how the product interacts with its environment. Providing insights tailored for developers, project managers, and other stakeholders.
- **Product Functionality:**

  - Summarizes the primary functions the product must perform, focusing on user roles and functionalities for both college-seeking students and college-attending/graduated persons. This section is intended to address the specific needs and expectations of users, making it relevant for college applicants, current students, and alumni.
- **Design and Implementation Constraints:**
  - Describes items and issues limiting development options.
  - Addresses hardware, software, and connectivity constraints.
  - Provides insights into assumed factors and dependencies., ensuring developers, project managers, and stakeholders are aware of the constraints impacting the project.
- **External Interface Requirements:**
  - Details logical characteristics of user interfaces.
  - Outlines hardware and software interfaces.
  - Describes user login/registration process and other critical interfaces. This section caters to users, emphasizing the user experience and interface aspects crucial for successful interaction.
- **Use Case Model:**
  - Highlights critical use cases with specifications for sign-up and login, forgotten password, and viewing reviews.
  - Identifies authors, purposes, priorities, preconditions, and actors for each use case. This section is aimed at developers, project managers, and testers, providing detailed specifications for critical user interactions.
- **Other Non-functional Requirements:**
  - Articulates performance, safety, and security requirements.
  - Explores software quality attributes, including flexibility, portability, maintainability, usability, and reliability. This section is essential for developers, testers, and stakeholders concerned with the overall quality and performance of the product.
- **Other Requirements (Optional):**
  - Invites the inclusion of any additional requirements not covered elsewhere in the document.
- **Appendices:**
  - Appendix A includes a Data Dictionary to track variables and functional requirements.
  - Appendix B - Group Log records minutes from group meetings, activities, and relevant information.
  - These appendices are valuable for developers, document writers, project managers, and other stakeholders seeking additional detailed information and records.

1.3 Definitions, Acronyms and Abbreviations

Term	Definition
CSS	Cascading style sheet is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.
Database	Collection of all the information monitored by this system.
HTML	The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.
Java	Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
JavaScript	JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.

1.4 Document Conventions

Formatting Conventions -
• Arial font size 11 is used throughout the document for text.
• The document maintains a 1'' margin and is single-spaced throughout
• Section titles use font size 18; subsection titles use font size 14, and sub-subsection titles use font size 12.
• Comments are italicized, and important words are made bold.

1.5 References and Acknowledgments

1. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. Explore the UML sequence diagram - IBM

2 Overall Description

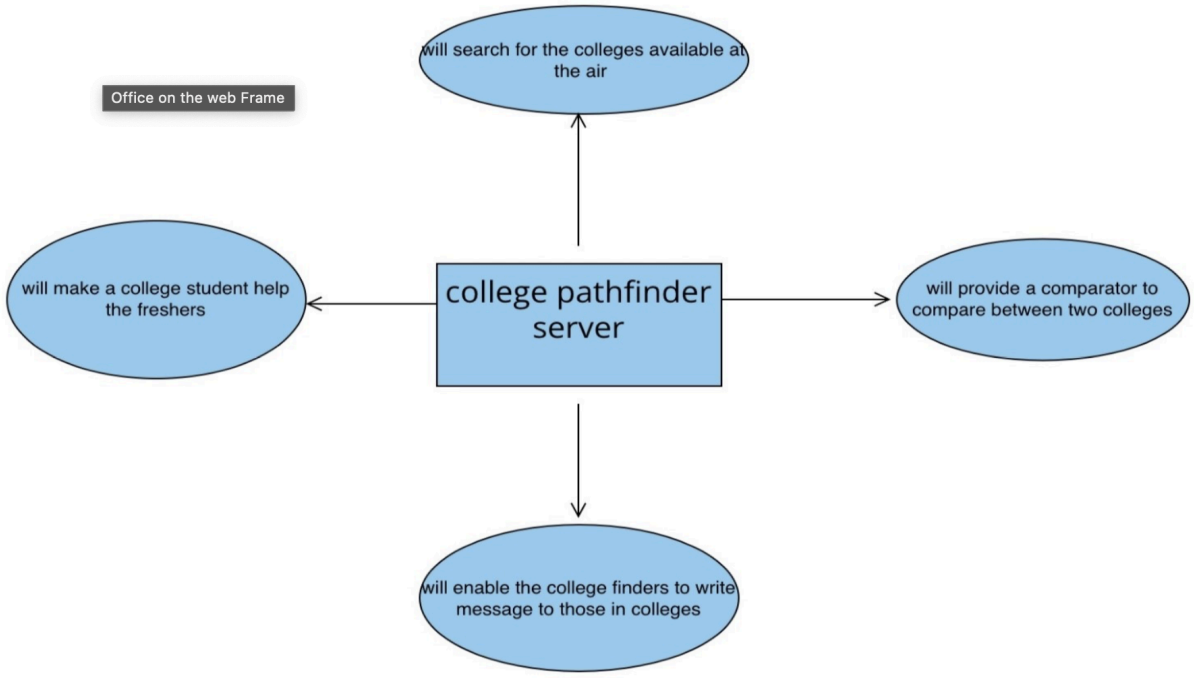
2.1 Product Overview

The College Pathfinder is an online platform designed to cater to the needs of both college-seeking students and college attending/graduated individuals. This platform serves as a bridge between these two user groups, facilitating the exchange of information, reviews, and advice related to college programs.

Product Interaction Overview:

The platform comprises three major user categories - college-seeking students, college attending/graduated persons, and administrators. College-seeking students can register, search for programs, read reviews, ask questions, seek on-call advice, and compare programs. College attending/graduated individuals can register, provide reviews, answer forum questions, accept on-call session requests, and engage with notifications. Administrators have the authority to log in, manage reported posts, and deregister users.

The platform's functionality involves a dynamic interaction between these user categories, creating a comprehensive ecosystem for information exchange and support. The diagram below illustrates the high-level interaction between the major components of the system:



2.2 Product Functionality

The users can identify either as a college seeking student or a college attending/graduated person.

Functionalities for college seeking students:
1. Register as a college seeking student.
2.Search for the programs available at a college or in a branch with their cutoffs.
3.Search for the reviews of the desired program for academics, non-academics, college life, etc.
4.Ask questions and read the answers on the 'Q&A' forum and like, dislike or report the answers.
5.Ask for on-call advice from the experts.
6.Compare two programs based on the desired criteria like placements, rankings, etc.
7.Bookmark any information for future reference.
8.View the notifications and reply.
Functionalities for college attending/graduated persons:
1.Register as a college attending/graduated person.
2.Give reviews about academics, non-academics, college life, etc. of a program.
3.Answer the questions asked on the forum.
4.Accept the requests from the college seeking students and schedule on-call sessions.
5. View the notifications and reply.
Functionalities for the admins:
1.Login as an administrator
2.Delete the posts reported by the users.
3.Deregister any user.

2.3 Design and Implementation Constraints

Design and Implementation Constraints:

1. Hardware Limitation:
- No specific hardware limitations.
2. Software Specifications:
- Frontend (User Side):
- Implemented using HTML, CSS and JS.
- Compatible with any modern browser supporting JavaScript, website loading, and socket communication.
- Minimal browser requirements: 100MB Hard Disk, 128MB RAM.

- Backend (Admin Side):
- Developed in Java.

- Database (Admin Side):
MySQL
3. Language Requirements:
- Project involves HTML, CSS, JavaScript/TypeScript, Java, Bash, and SQL as programming languages.

4. Communication Protocol:
- Utilizes HTTP and Websockets for data transfer and subcomponent interaction.

5. Interface to Third-Party Applications:
- Integration with Email for relevant functionalities.

6. Security Considerations:
- All data encrypted during communication using secure protocols.

- No local server maintains a copy of the database.
- Protection against SQL injection attacks for accessible API points.

2.4 Assumptions and Dependencies

TO DO: Provide a short list of some major assumptions that might significantly affect your design.> Assumptions and Dependencies:

1. Email Services:
   - Assumption: The project relies on email services for user authentication.
   - Impact: Efficient working of the system is contingent upon the availability and reliability of email services. Any disruptions in email services may affect user login functionality.

2. Web Browser Compatibility:
   - Assumption: The application is designed with modern web browsers in mind.
   - Impact: Users utilizing older versions of web browsers may experience inefficiencies in the interface. The system's efficiency is optimized for compatibility with modern browser technologies.

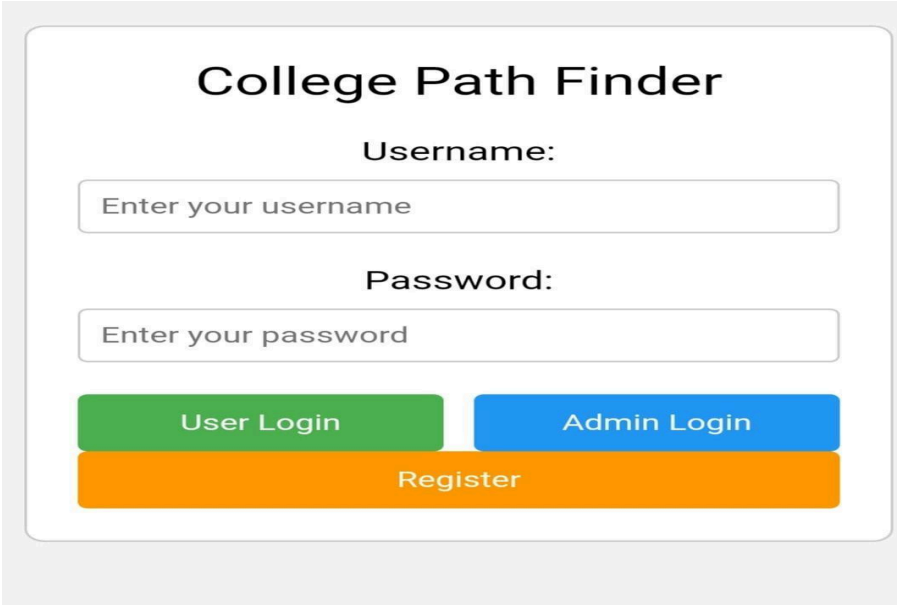
3. Reliable Internet Connection:
   - Assumption: Users are assumed to have a reliable internet connection.
   - Impact: The project's efficiency is dependent on a stable internet connection. Users with unreliable or slow connections may experience delays and performance issues.

3 Specific Requirements

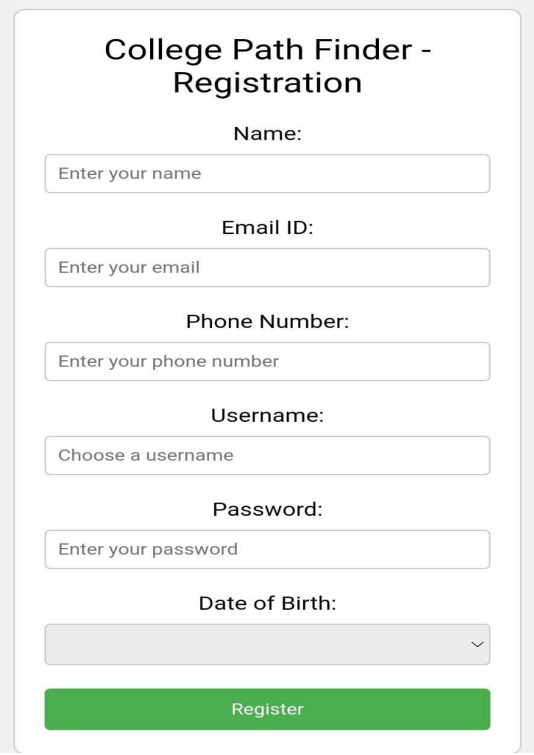
3.1 External Interface Requirements

3.1.1 User Interface

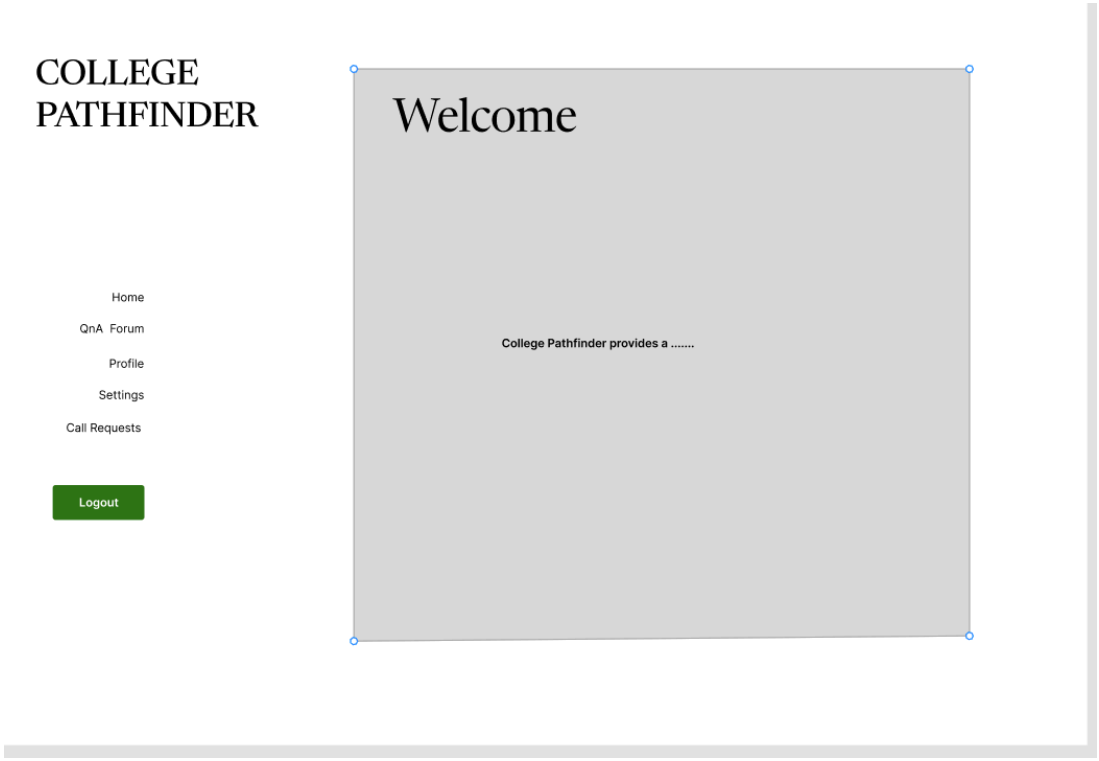
User Login/Registration:-



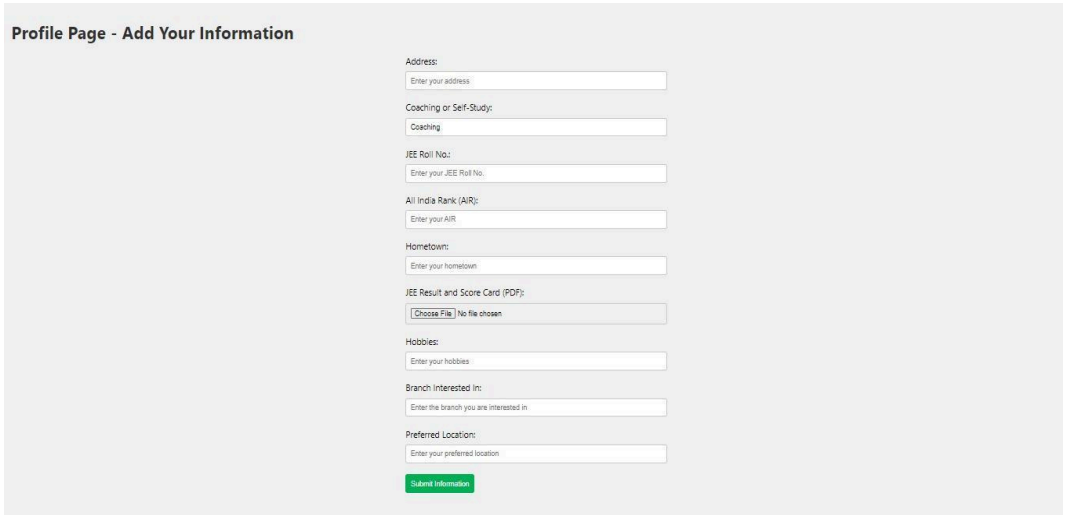
Registration: (this is done separately for both the college going and college searching user)



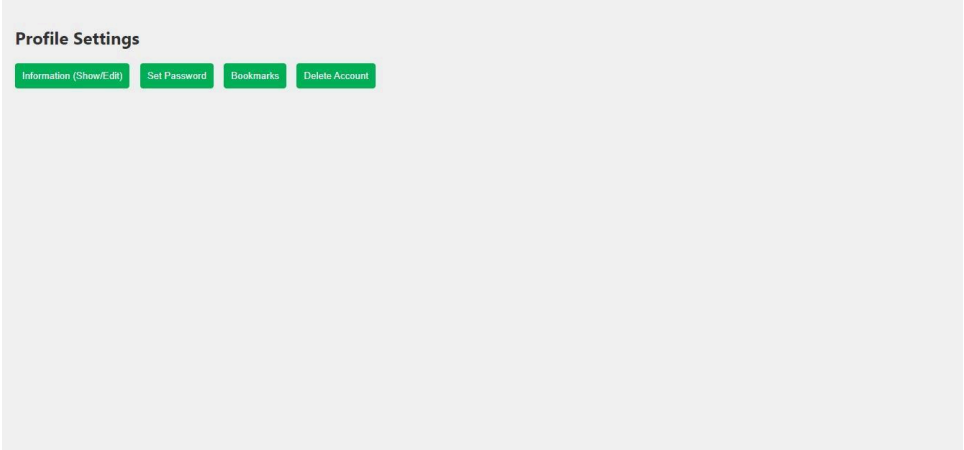
Home Page:



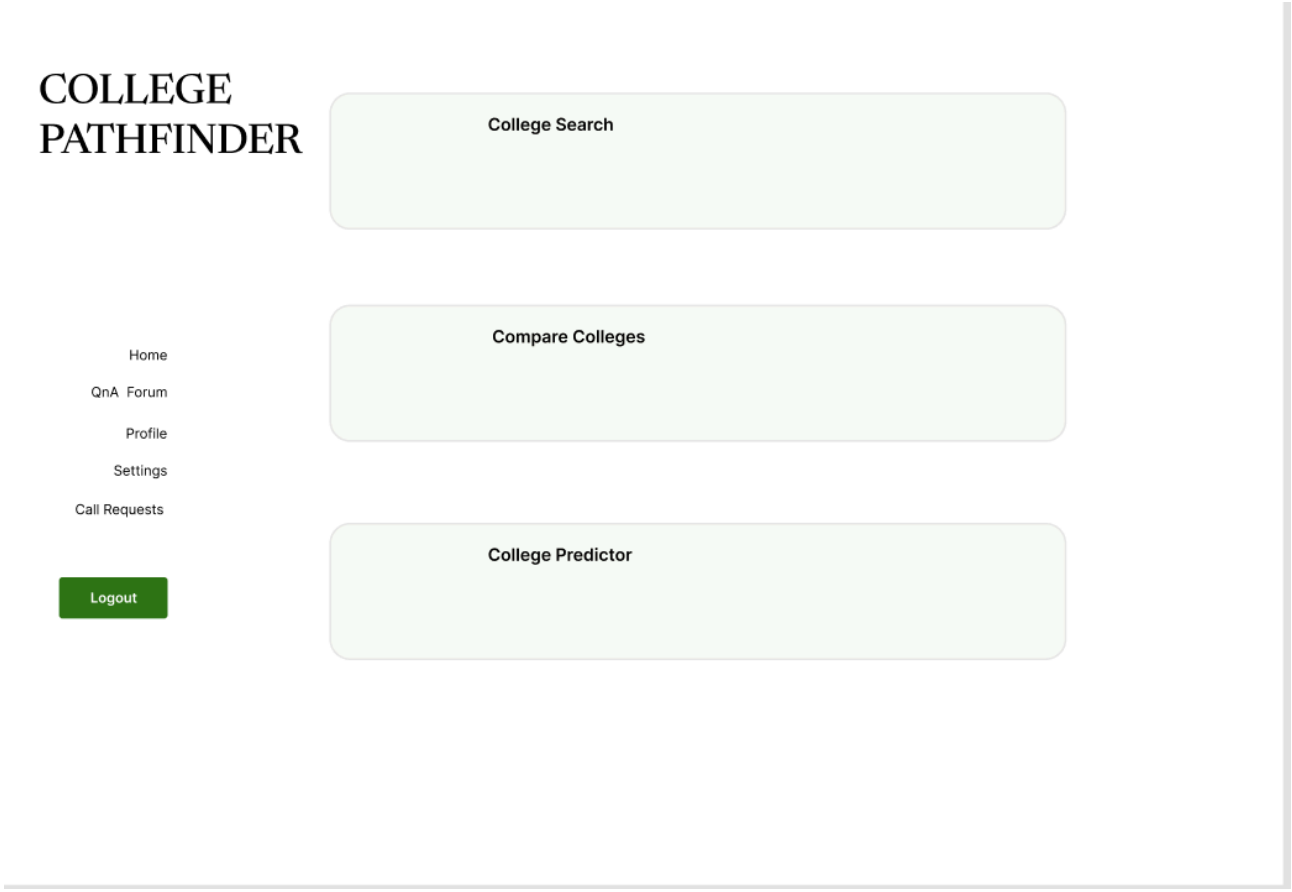
Profile:



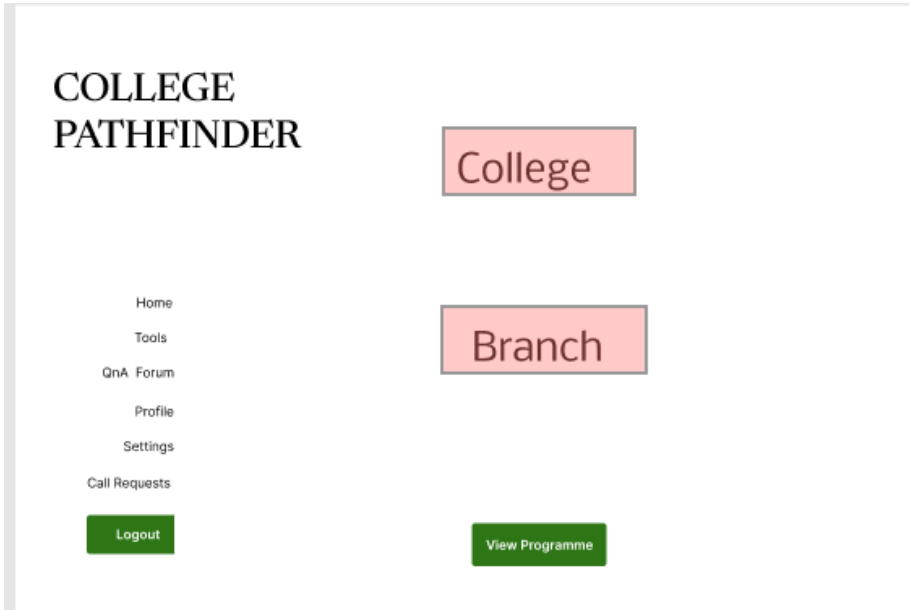
Settings:


Profile Settings
Information (Show/Edit)
Set Password
Bookmarks
Delete Account

College tools:



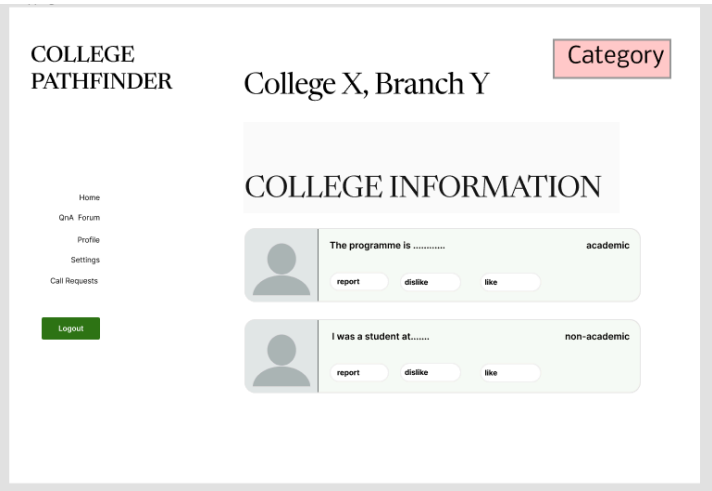
College search:



Comparison tool:

QnA Forum:

College information and reviews:



3.1.2 Hardware Interfaces

- **Supported Device Types:**
  - Desktops, laptops, tablets, and smartphones.
- **Browsers:**
  - Compatibility with major web browsers.
- **Connectivity Hardware:**
  - Modem: Facilitates the connection between the local network and the internet.
  - Network Card: Enables communication between the device and the local network.

3.1.3 Software Interfaces

In the process of system development, we need to design static and dynamic website interfaces, create website functions, establish a database system.

This necessitates a specific set of software requirements, including:

- **Web Browser (Google Chrome is used):**
  - Utilized during the development phase for debugging and mainly testing.
- **Java:**
  - Selected as the server-side programming language for the backend development of the progressive web application.
- **HTML, CSS, JS:**
  - HTML for content structure, CSS for styling, and Java for backend functionality in web development
  - Employed as foundational elements for constructing the responsive web components, forming the frontend.
- **ReactJS and NodeJS:**
  - Implemented to enhance the visual appeal of the frontend interface for users.
  - ReactJS for building dynamic user interfaces, and NodeJS for server-side JavaScript execution in web development.
- **MySQL:**
  - Communication with the server database, managing data storage and retrieval.
- **Visual Studio Code:**
  - Integrated Development Tool for various coding operations during the development process.

3.2 Functional Requirements

Login/Register:

- **New Users:**
  - Users navigate to the "Sign Up" or "Register" option on our homepage.

- Complete the registration form by providing the required information.
- Now users can set up a secure password for their account.

- **Registered Users:**
  - Existing Users should click on the "Login" or "Sign In" button on the homepage.
  - They can enter their registered email address and password in the space provided.
  - Now on clicking "Login" one can access their personalized profile.
- **Administrators:**
  - Administrators should be able to securely log in to the system using unique credentials.
  - Authentication mechanisms should be in place to ensure the confidentiality of administrator accounts.

OBJ

User Registration:

★ College-seeking students

- **Functionality:** The system must allow college-seeking students to register by providing necessary information, including their name, phone number, Email ID, date of birth, entrance exam rank, college & branch preferences.
- **Interfaces:** Registration form with fields for personal information, educational background, and preferences. It should include a user-friendly interface for data entry and validation.

★ College-attending/graduated persons

- **Functionality:** The system must allow college-attending/graduated persons to register by providing necessary information, including their name, phone number, Email ID, date of birth, educational background, account password, and preferences.
- **Interfaces:** Web-based registration form with fields for personal information, educational background, password, etc. It should include a user-friendly interface for data entry and validation.

User Login:

- Registered users must be able to log in using a unique username or email and a secure password.
- The system must authenticate user credentials and provide appropriate error messages for invalid inputs.

User Profile:

- Both user categories could customize their password, modify personal information, toggle notification settings, view saved bookmarks, and deactivate their account.

Program Search:

- **Functionality:** Users should be able to search for academic programs available at the preferred colleges or branches. The system must display program details, including cutoffs, eligibility criteria, location, how to reach, rankings, etc.
- **Interfaces:** A search bar on the platform's interface with filters for program type, colleges, and branches. Program details should be displayed in a structured format on the interface.

Program Comparison:

- **Functionality:** college-seeking students should be able to compare two academic programs based on desired criteria such as placements, rankings, etc.
- **Interfaces:** A side-by-side program comparison interface with checkboxes for selected criteria. Clear visual representation for easy comparison.

Program Reviews:

- **Functionality:** college-seeking students should have the ability to search for and read reviews about academic programs. The system must categorize reviews into academics, non-academics, and college life.
- **Interfaces:** A review section with filters for program categories. Reviews should be displayed with clear categories and a rating system (1 to 5).

- **Q&A Forum:**
  - ★ College-seeking students
    - **Functionality:** college-seeking students should be able to ask questions on the platform. The system must allow users to read and upvote answers. There should be a feature to bookmark questions and answers for future reference.
    - **Interfaces:** A Q&A forum interface with options to ask, answer, upvote/downvote, and bookmark questions and answers.
  - ★ College-attending/graduated persons
    - **Functionality:** College-attending/graduated persons should be able to answer questions and read other answers questions posted on the Q&A forum.
    - **Interfaces:** Similar to the general Q&A forum, with additional features for registered college-attending/graduated persons to answer and without the option of upvoting/downvoting their own answers.

- **On-call Advice:**
  - ★ College-seeking students
    - **Functionality:** college-seeking students should have the option to request on-call advice from experts. The system must facilitate communication between the student and the expert.
    - **Interfaces:** A scheduling interface for users to request on-call advice, with options for communication such as video calls or messaging.
  - ★ College-attending/graduated persons
    - **Functionality:** can accept the request.
    - **Interfaces:** requests interface.

- **Bookmarking:**
  - **Functionality:** The system must allow **college-seeking students** to bookmark information, including programs, reviews, and forum discussions, for future reference.
  - **Interfaces:** A bookmarking feature accessible from the user's profile, allowing them to manage and organize their saved information.

- **Administrators** have the authority to perform the following tasks:
  Manage Reported Posts:
  - Administrators should be able to access a dashboard or control panel that displays reported posts or content flagged by users.
  - They should have the ability to review reported posts, assess their content, and take appropriate actions based on predefined policies (e.g., delete, warn, or escalate).

  Deregister Users:
  - Administrators should have the authority to deactivate or deregister user accounts that violate platform policies or exhibit inappropriate behavior.
  - The system should provide a mechanism for administrators to review user accounts, track user activity, and make informed decisions about deregistration.

3.3 Use Case Model

Use Case #1: Sign Up and Login

Author – Team College PathFinder

Purpose - This use case covers login and signup for admins and students, whether currently attending college, already graduated, or currently enrolled.

Requirements Traceability – Profile page

Priority - High

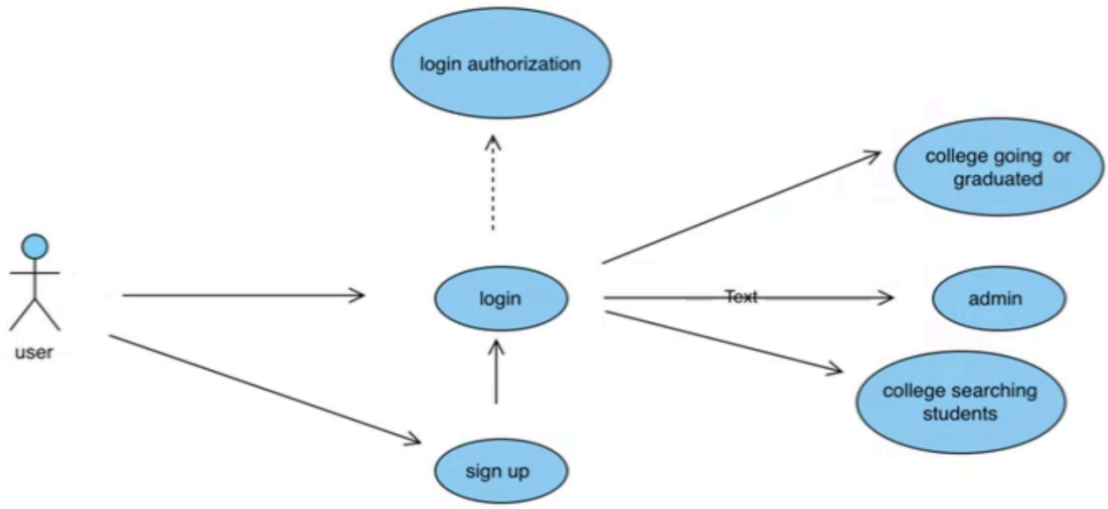
Preconditions – None

Post conditions – User is logged in

Actors – Users of the website including the managing team

Exceptions – User may forget password and go to forgot password option

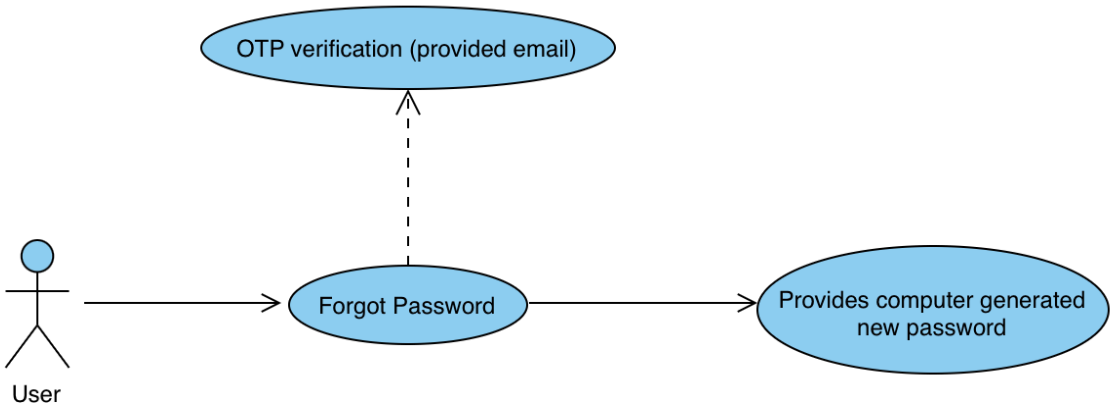
Notes/Issues - Users will have to enter some necessary details (name, email etc.) during signin.



Use Case #2 Forgot Password

Author – Team College PathFinder

Purpose – This use case ensures that users can change their password to securely re-establish access to their account.



Requirements Traceability – Profile page

Priority – Low

Preconditions – User must not be logged in currently and user has signed up already

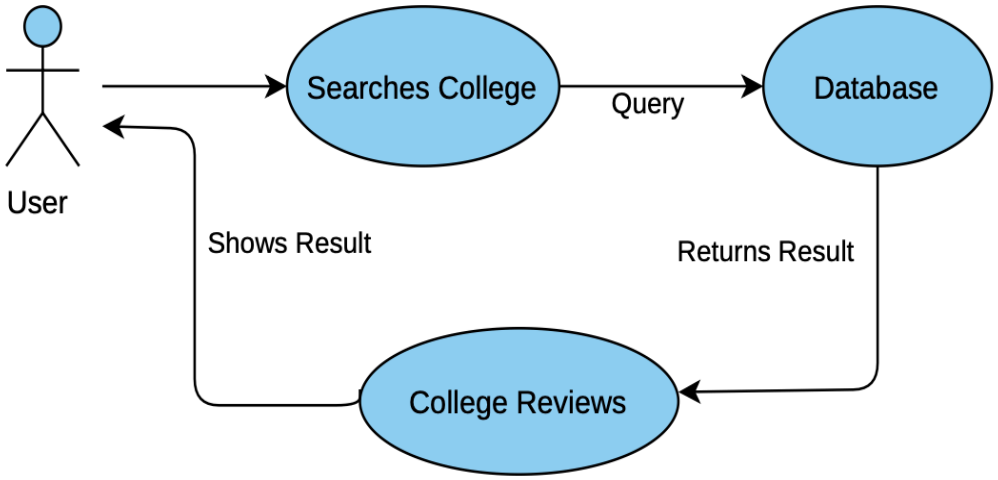
Post conditions – User gets new password

Actors – Users of the website including the managing team

Use Case #3 Viewing Reviews

Author – Team College PathFinder

Purpose - This use case covers the purpose of viewing reviews which gives potential students insights about academic offerings, professors, and campus culture. It enables people to make knowledgeable selections and makes the process of selecting a college that fits their preferences and aspirations easier.



Requirements Traceability – Reviews section

Priority – High

Preconditions – User must be logged on the website

Post conditions – a person will be able to get list of the reviews

Actors – Users of the website including the managing team

3.3.4 Use Case #4 College Compare

Author – Team College PathFinder

Purpose – This use case assists prospective college students in comparing two colleges

directly, considering key factors to resolve various dilemmas and make more informed decisions

Requirements Traceability – College Compare section

Priority – Medium

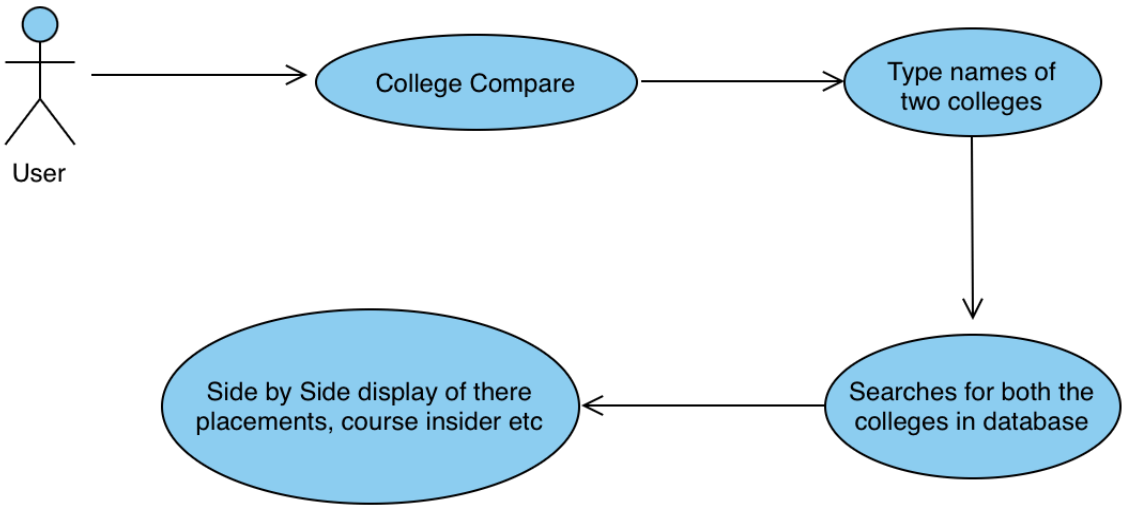
Preconditions - User must be logged on the website

Post conditions - will get the list side by side about the college

Actors – Users of the website including the managing team

Exceptions – Data of all the colleges might not be available

Notes/Issues - College names must be different



3.3.5 Use Case #5 Posting Question on Forum

Author – Team College PathFinder

Purpose - This use case covers the purpose of seeking help and sharing experiences

by posting questions on a forum, creating a collaborative space for diverse perspectives and valuable insights. Any user can like, dislike or even report question.

Requirements Traceability – Forums section

Priority – Medium

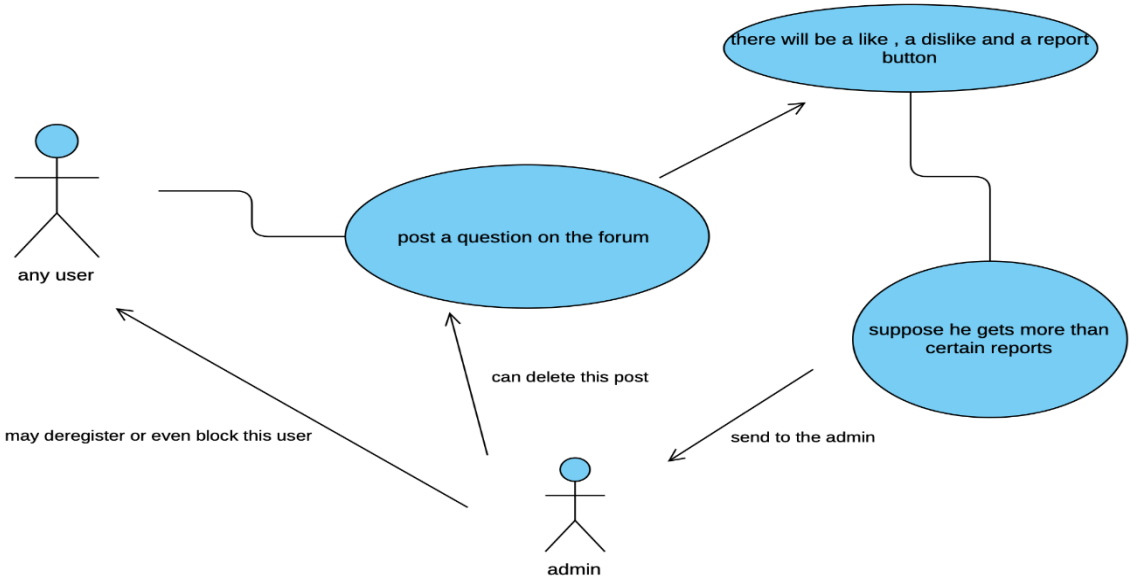
Preconditions – User must be logged on the website

Post conditions – New question is added

Actors – Users of the website including the managing team

Exceptions - not as such

Notes/Issues - If the number of reports on any question is greater than some threshold number admins can delete that post or even block that user



3.3.6 Use Case #6 Bookmarks

Author – Team College PathFinder

Purpose – This use case provides users option to bookmark the messages, programs and

reviews for future references

Requirements Traceability – Bookmarks

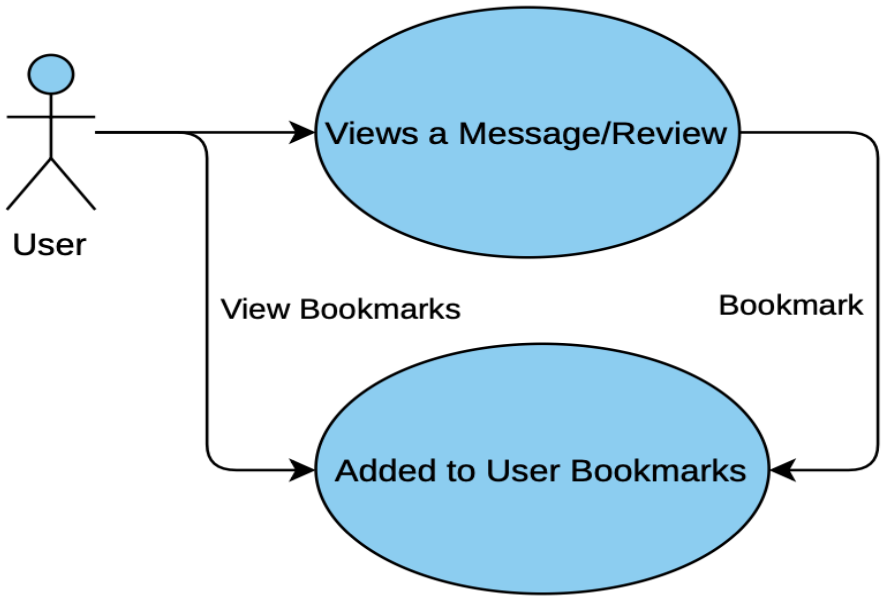
Priority – Low

Preconditions – User must be logged on the website and should select content for bookmarking

Post conditions – Bookmarked contents are shown in the user bookmarks

Actors – Users of the website including the managing team

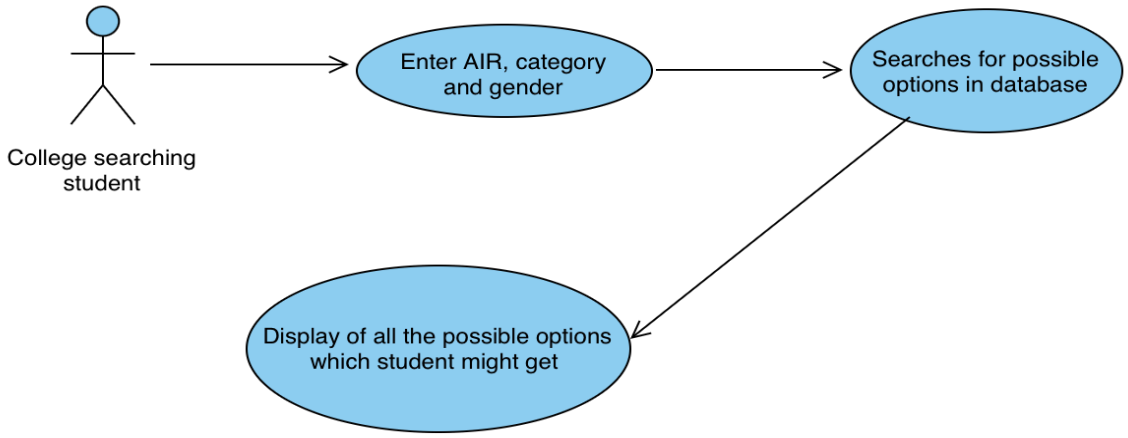
Exceptions -



3.3.7 Use Case #7 Options Filter

Author – Team College PathFinder

Purpose – This use case provides user the possible college options on the basis of AIR, category and gender based on previous occurrences.



Requirements Traceability – Filters

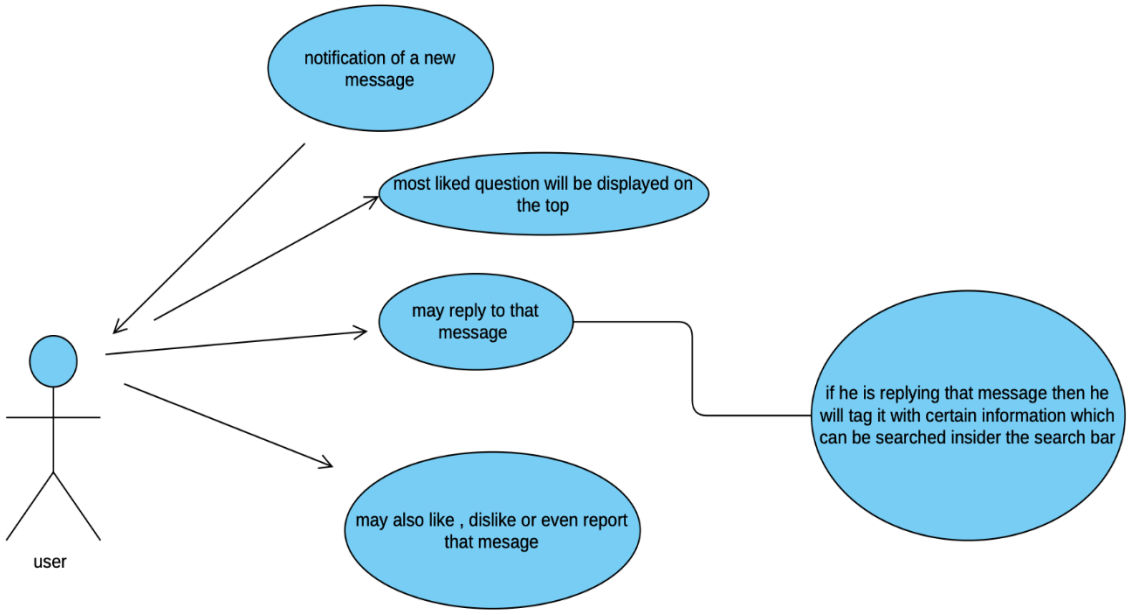
Priority – Medium

Preconditions –User must be logged on the website

Post conditions -

Actors – Users of the website including the managing team.

3.3.8 Use case #8 Replying to the messages



Author – any user of the college pathfinder

Purpose – To reply the message issued

Requirements Traceability – profile page

Priority –high

Preconditions – the person who is replying to the message should be logged in

Postconditions – The person's reply to the message will be saved and tagged with proper tags

Actors – The user who has logged in

Exceptions – if the person is reported and expelled by the admin, he may not be able to reply

Notes/Issues - Any relevant notes or issues that need to be resolved

3.3.9 Use case #9 Updating profile

Author – college going students

Purpose – the college going students can write about the college they are going

Requirements Traceability – profile page

Priority –high

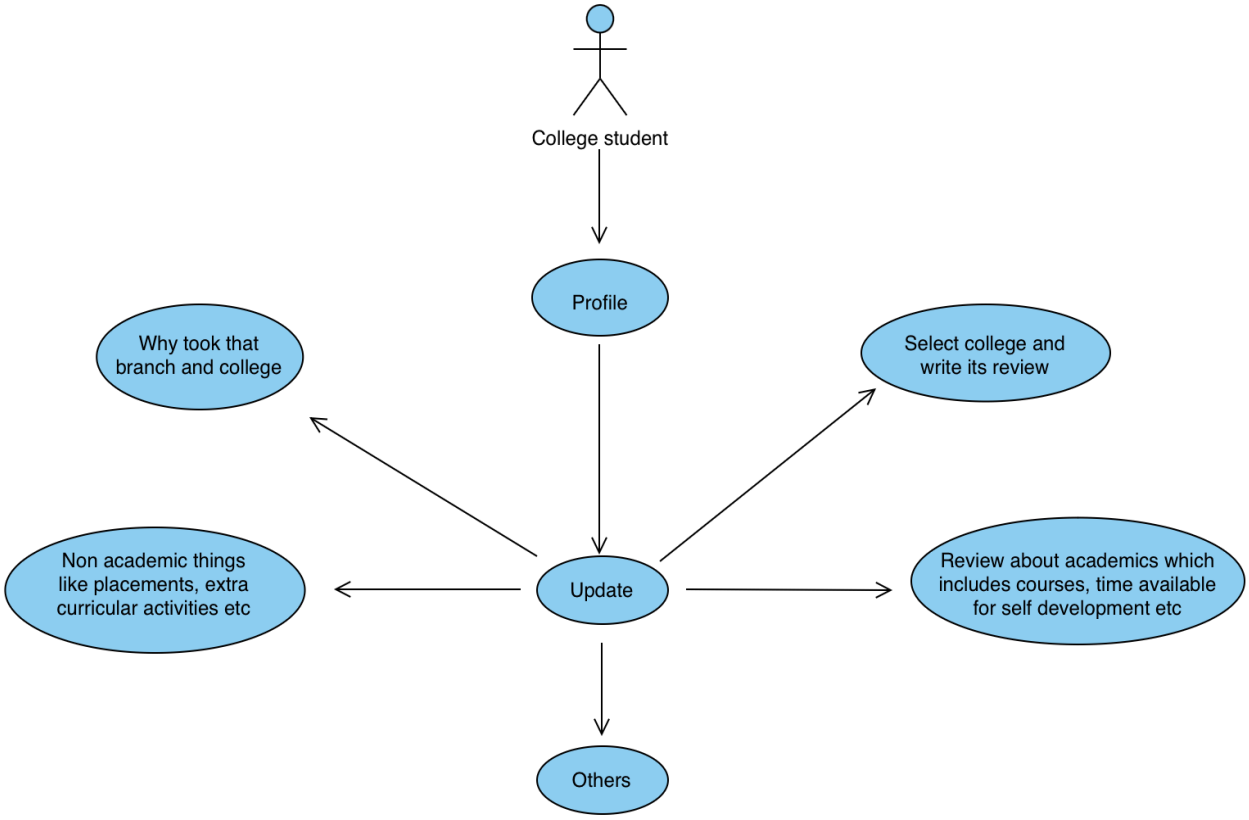
Preconditions – those who are adding the reviews should be logged in as college going students

Post conditions – every information given by the person will be completed with a tag

Actors – The user who has logged in

Exceptions – if the person is reported and expelled by the admin, he may not be able to reply

Notes/Issues - Any relevant notes or issues that need to be resolved



4 Other Non-functional Requirements

4.1 Performance Requirements

Responsiveness:
The system should exhibit unparalleled responsiveness, ensuring swift handling of user requests, with a target response time of 2 seconds for standard operations.

Data Upgrades:
Rapid data upgrades are a priority, with the system facilitating real-time updates to ensure seamless integration with evolving client requirements.

Stability and Maintainability:
The product should prioritize stability and maintainability, with a commitment to eliminating bugs and performance issues to ensure uninterrupted and continuous functionality.

Scalability:
The infrastructure, both hardware and software, should effortlessly accommodate a higher volume of clients than the anticipated average, reflecting the scalable nature of the product platform.

Accessibility:
The product should be accessible across multiple platforms, with adaptive capabilities to dynamically conform to clients' specific hardware and software configurations, ensuring compatibility and responsiveness even on low-spec hardware.

Dynamism:
The system should embody dynamism, allowing seamless incorporation of updates and changes without causing significant system downtime.

Agility:
The product's agility should extend to scalability, poised to cater to the ever-expanding community of clients actively.

Continuous Improvement:
The product should demonstrate a commitment to continuous improvement, adaptability, and uninterrupted service for a growing and diverse user base.

4.2 Safety and Security Requirements

1. Passwords will be saved encrypted in the database to ensure the user's privacy.

2. The user's IP address will be logged.

3. The system will be protected against vulnerabilities such as SQL injection attacks.

4. All users must create a login account to access any facility. This ensures the authenticity of users.

4.3 Software Quality Attributes

4.3.1 Flexibility

The application will be crafted with a focus on flexibility, enabling the seamless integration of new requirements across any module of the system. Embracing a modular format, the design will facilitate not only the incorporation of fresh elements but also the ease of making changes, additions, or deletions within the system without imposing a substantial impact on existing functionality. This modular approach ensures that the system remains adaptable and scalable, accommodating the evolving needs of users and allowing for efficient enhancements and modifications in response to dynamic requirements

4.3.2 Portability

The application will be easily portable on any Windows-based system. The website frontend is designed using React JS, ensuring a responsive and progressive web app that can seamlessly run on various platforms.

4.3.3 Maintainability

Efforts will be made to ensure that the architecture, design, implementation, and documentation of the product are structured to minimize maintenance costs.

Specifically:

- **Security Defect Resolution**: Any security defects discovered should be addressed promptly, with the maximum person-time required for resolution (including regression testing and documentation update) capped at two persons per day. If this limit is exceeded, appropriate action such as taking the software system offline or disabling the offending feature will be taken, prioritizing emergency situations.

- **Enhancement Implementation**: Minor enhancements to the software system, including testing and documentation updates, should be completed within an average person-time of one person per week. This ensures that routine improvements can be efficiently integrated into the system without significant disruption to operations.

4.3.4 Usability

The frontend of the software application is meticulously crafted to ensure user-friendliness, enabling users to utilize the system effectively.

4.3.5 Security

The application prioritizes security measures to safeguard user data and system integrity. Key aspects include:

- **Access Control**: Only privileged users have permissions for system modifications, such as upgrades and deletions.
- **Authentication**: Users can log in using their email and a custom password (created by them). Passwords are stored in encrypted format in the database to enhance security.
- **Password Recovery**: If users forget their passwords, they will receive a one-time password (OTP) via their registered email address. This OTP can then be used to generate a new password securely.

4.3.6 Reliability and Availability

The system prioritizes reliability and availability through the following measures:

- **Database Backup**: Regular backups of user data, including passwords, recently uploaded posts, achievements, and user activities, will be performed to ensure data integrity and facilitate recovery in case of unforeseen events.
- **Continuous Operation**: The system is designed to be continuously operational and ready to perform its functions whenever users require access. This ensures that users can rely on the system's availability to carry out their tasks without disruption.

5 Other Requirements

- **Internationalization:**
  - Support multiple languages and adaptable date, time, and currency formats.
- **Legal Compliance:**
  - Adhere to data protection and privacy laws and comply with educational regulations.
- **Reuse Objectives:**
  - Implement modular code practices for ease of reuse and future integration.

Appendix A – Data Dictionary

A.1 User Class

Element Name	Description	Operations	Requirements
User ID	System defines a unique ID for each user	Identifies unique user	Unique string
Password	It takes the password as arguments then checks the condition of login and returns a unique id	Secure user login	String
User Profile Data	Maintain the data of each user	Store user info	Strings and integers
Email Address	It takes an Email address as an argument and associates it with a unique user ID	Profile Verification and completion	Any valid email address
Q&A Queries	Users post their queries under Q&A Section	Displays post to other users	Abstract
Reviews	Insights and opinions shared by users about colleges, academics or experiences.	Review gets posted under review section	Abstract

A.2 User Interface Class

Element Name	Description	Operations	Requirements
Registration	It takes the name, surname, username, password, email as arguments then checks the condition of registration and returns the status of registration.	First time registration and data input	string, int
Login/Logout	It takes user ID or username and password as arguments then checks the condition of login and returns the status of login.	Enables login/logout functionality for user	User Id/ Username, password
Follow	It takes the User ID as an argument and returns the status of following the user.	Adds the user in the following list	User ID

React	It takes user ID and an emoji (reaction for a post)	Emoji along with user ID gets associated with the post	Emoji
Share	It takes in a post's URL and copies it to the local clipboard	URL gets copied	URL
Review	It takes in user ID and user's comment and adds it to the comments section of a post	Comment along with user ID gets associated with the post	Text

A.3 User Admin Interface Class

Element Name	Description	Operations	Requirements
Profile information	It takes the user Id and its data as arguments and returns profile information for public users.	Provides Admin contact info	User ID, user data
Block User	It takes in the user ID and blocks the user associated with that ID	User can no longer post queries	User ID
Add College Item	It takes in admin's credentials and new college data.	Adds college item in database	Abstract
Update College Item	It takes in admin's credentials and change in existing college data.	College data gets updated	Abstract

Appendix B - Group Log

SL. no.	Date	Timings	Venue	Description
1	7/01/2024	10 pm - 11:30 pm	Zoom	Brain-stormed various possible prospective ideas for the project.
2	11/01/2024	10 pm - 11:30 pm	Zoom	Finalized the idea for the project and discussed various aspects of it.
3	13/01/2024	5:30 pm - 6:30 pm	Zoom	Studied the SRS template given and distributed the work amongst the team members.
4	15/01/2024	5:15 pm - 5:40 pm	Call	First meet with the Teaching Assistant Mr. Abhilash C Discussed some doubts regarding the SRS documentation.
5	16/01/2024	12:40 pm - 1:20 pm	Meet	Redistributed the pending work after having clarified the doubts with the TA.
6	18/01/2024	9 pm - 12 am	RM building	Explored some more functionalities for the product and progressed with the SRS document.
7	20/01/2024	5 pm - 8 pm	RM building	Explored various implementation tools and software for the project.
8	24/01/2024	5 pm - 8 pm	RM building	Progressed with SRS documentation work.
9	25/01/2024	5 pm - 8 pm	RM building	Progressed with SRS documentation work.
10	26/01/2024	5: 15 pm - 6 pm	Meet	Second Meet with the TA to discuss the drafted SRS document.