Functional Specification Document

1. Introduction

- **Project Title**: AI Powered Student and Alumni Support Network
- **Date**: 11/4/2025
- **Author**: Vasundhara Ravikumar
- **Approvers**: Dr. Storcz Tamás

2. Version history

#	Date	Consultant	Topic
01	2025.09.17	Dr. Tamás Storcz	Thesis Topic Selection : • What makes a topic complex enough to become a thesis? • Discussed the requirements to get custom topic approved
02	2025.10.4	Dr. Tamás Storcz	Discussed the scope and verification of chosen topic such as the features that will make the topic complex. Was given advice on what kind of research to do for specific features and was told to re – consider working on certain features of the project.
03	2025.10.18	Dr. Tamás Storcz	Made some changes to the custom topic chosen. Decided to make the project for academic and practical use purposes only. Added some new features to the academic and practical sections of the app idea and in addition, thought of making the application accessible to university alumni as well.
04	2025.11.4	Dr. Tamás Storcz	Asked questions about the functional specifications document and PowerPoint to be presented. Cleared some doubts on what exactly is meant by “functional” and learned that the PowerPoint and document should focus solely on the user’s point of view when using the application.

3. Purpose

Purpose Statement:

This document defines the functional specifications for the AI-powered student and alumni support network, a platform designed to foster academic collaboration, mentorship, and practical assistance among university students, alumni, and faculty members. The system

integrates artificial intelligence to enhance knowledge sharing, streamline academic communication, and create meaningful professional connections within the university ecosystem.

Scope

In – Scope

Academic Support

- Collaborative note creation and editing with version control for shared learning.
- AI-assisted topic linking across notes and discussion threads to connect related concepts.
- Course-based “Ask a Doubt” threads for peer-to-peer academic help.
- Role based access – Roles include student, alumni, professor (any other university staff) and admin (Controls who has access to what)
- Automatic study group formation based on shared subjects, learning preferences, and activity patterns.
- Centralized repository of academic FAQs and study resources.

Practical Support

- AI-powered alumni clustering system that connects students to mentors based on their queries and interests.
- Geo-based help board showing nearby campus resources (e.g., study spaces, labs, housing options).
- Personalized campus feed highlighting relevant events, workshops, and opportunities.
- Career and internship advice board featuring alumni insights and student experiences.
- AI-driven routing that redirects student questions to the most suitable individuals or communities for response.

Out of Scope

- Development of a full-fledged AI tutor or virtual teacher – Only collaborative notes and AI assisted topic linking between notes and threads are included.
- Integration with external Learning Management Systems (LMS) like Moodle or Canvas.
- Direct integration with official university systems for grades, attendance, or course registration – only authentication occurs using student neptun codes but other than that the system is independent of other university systems.
- Payment or subscription features — the app is intended to be free for students.
- Offline-only functionality — most features, including maps, require internet access.
- Advanced gamification or reward systems beyond peer encouragement cards.
- Full-scale campus event management tools — only a localized feed of opportunities and events is included.

- Real-time navigation beyond campus boundaries is out of scope — the geo-based help board offers *basic directional guidance and live campus navigation* (e.g., to study spots, printers, or help desks) but does not provide full city-level or turn-by-turn mapping.
- Full-scale recruitment/job portal functionalities — only peer advice and guidance boards for internships and jobs.

4. Overview

System Overview: High-level description of the system and its objectives.

The AI-Powered Student and Alumni Support Network is a unified digital platform designed to strengthen academic collaboration, mentorship, and practical engagement within the university community. It connects **students, alumni, and professors** through intelligent recommendation systems and AI-assisted tools that promote shared learning and professional development.

The system operates under two core modules — **Academic Support** and **Practical Support**.

- The **Academic Support** module enhances collaborative learning through shared note creation, AI-assisted topic linking, and course-based Q&A discussions. It allows professors to directly engage with students by reviewing shared notes and contributing to "Ask a Doubt" threads.
- The **Practical Support** module bridges students with alumni mentors and provides AI-curated campus resources, geo-based tools, and career insights.

Overall, the objective is to foster a self-sustaining, AI-enhanced academic and professional ecosystem that promotes continuous knowledge exchange across generations of learners.

Assumptions: List any assumptions made during the creation of the specification.

- Users (students, professors, and alumni) will have access to a stable internet connection for platform use.
- All users possess basic digital literacy skills to navigate web and mobile applications.
- Universities will support professor and alumni participation through verification of their institutional credentials.
- The AI models used for recommendations and topic linking will be trained on anonymized data and university-approved content sources.
- The application will be hosted on a secure cloud infrastructure (e.g., AWS, Azure, or Google Cloud).
- User-generated content (notes, questions, discussions) will follow academic integrity and community guidelines.

Dependencies: Identify any dependencies on other systems, software, or hardware.

- **AI Frameworks:** The system depends on external AI and NLP libraries (e.g., TensorFlow, PyTorch, or OpenAI APIs) for semantic analysis, topic linking, and intelligent routing.
- **Database and Storage:** Relies on PostgreSQL or MongoDB for data management and cloud storage (e.g., AWS S3) for user files and note repositories.

- **Authentication Services:** Integration with secure login and identity verification services (e.g., OAuth 2.0, university email verification).
- **Mapping Services:** Depends on third party APIs (e.g., Google Maps or OpenStreetMap) for geo-based help boards and location visualization.
- **Notification System:** Push notifications and real-time updates rely on Firebase or a similar service.
- **Hosting and Infrastructure:** Deployment depends on containerized environments such as Docker for scalability and ease of updates.

Technical Constraints: Any technical limitations or constraints.

- **Internet Dependency:** Most features require active internet connectivity; offline access is limited.
- **Processing Limitations:** AI-driven features like topic linking and recommendation may experience latency during high-traffic periods due to model processing times.
- **Storage Limits:** Cloud storage quotas may restrict the size and duration of stored multimedia content.
- **Integration Restrictions:** No direct integration with external LMS platforms or university administrative databases.
- **Privacy Compliance:** Must comply with data protection regulations (e.g., GDPR), limiting the use of certain personal data in AI training models.
- **Device Compatibility:** Designed primarily for web and modern mobile browsers; performance on legacy devices may vary.

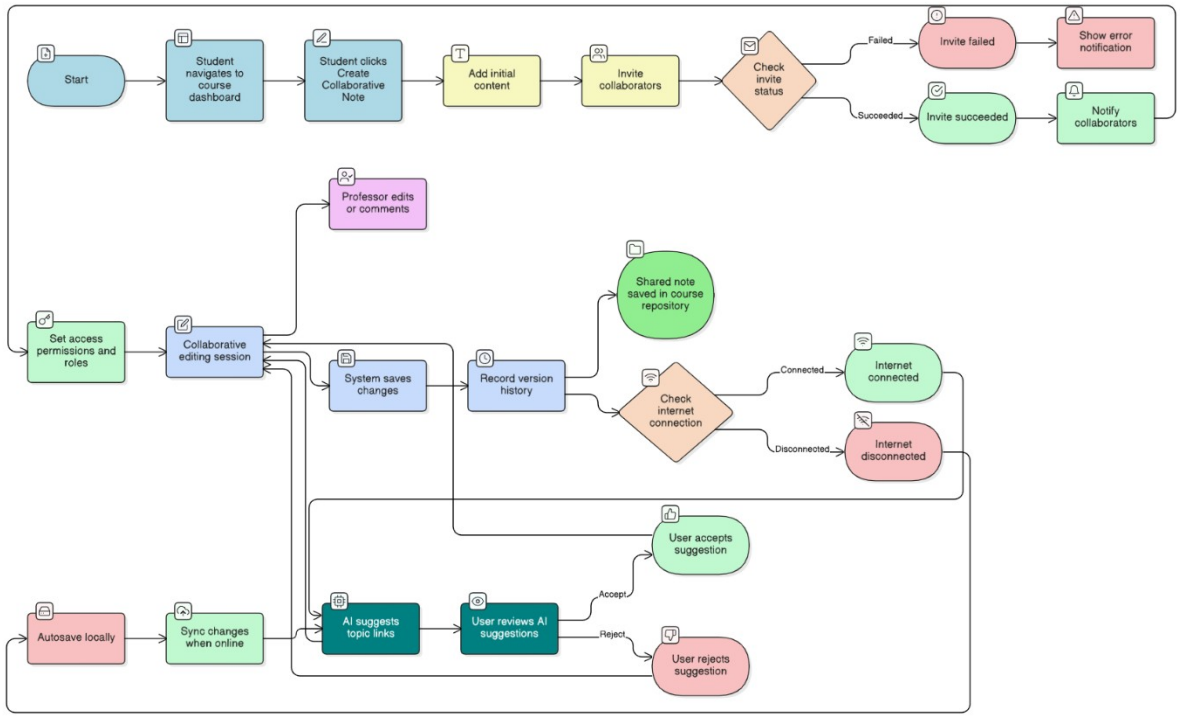
Business Constraints: Business rules or constraints.

- **Non-Commercial Intent:** The platform is designed as a **non-profit academic initiative** and will not include advertisements, payment systems, or subscription models.
- **Institutional Verification:** Professor and alumni participation requires approval through verified university credentials, ensuring content authenticity and academic relevance.
- **Content Moderation Policies:** User posts and discussions must comply with institutional guidelines and community standards.
- **Data Retention Rules:** All data must be securely stored and retained according to university data management policies.
- **Resource Limitation:** The project operates under limited funding and infrastructure support as part of a university research initiative.
- **Scope of Use:** Initial deployment is limited to one university network, with the potential for later expansion to other institutions

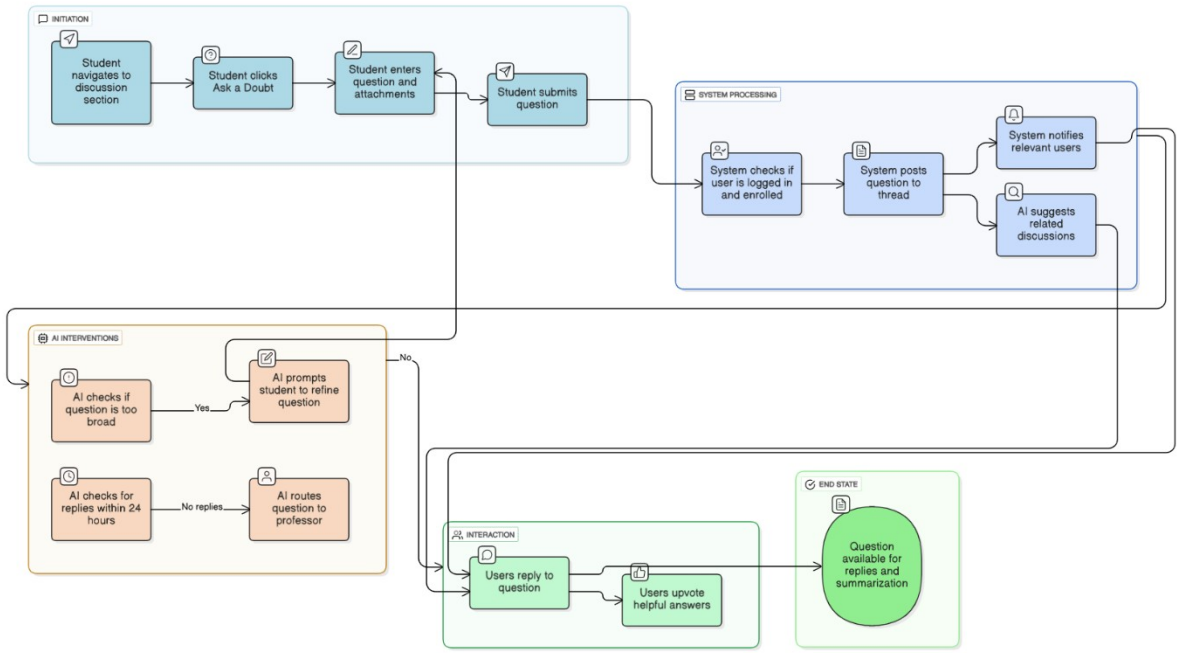
5. Functional Requirements

Use Case Diagrams: Visual representation of the interactions between users and the system.

Use Case : UC - 01



Use Case : UC – 02



Use Cases: Detailed descriptions of how users (actors) interact with a system to achieve a specific goal. It outlines the steps involved, the conditions, and the outcomes.

Use Case 1: Collaborative Note Creation

- **Use Case ID:** UC-01
- **Title:** Create and Edit Collaborative Notes
- **Description:** Students collaboratively create and edit lecture notes with real-time version control.
- **Actors:** Students (primary), Professor (secondary, as reviewer)
- **Preconditions:** User must be logged in and must collaborate while texting either another student and/or professor.
- **Postconditions:** A shared note document is saved in the course's repository with version tracking enabled.
- **Triggers:** Student selects "Create Note" within a course.
- **Main Flow:**
    - Student navigates to the course dashboard.
    - Clicks on "Create Collaborative Note."
    - Adds initial content and invites peers and/or professors to collaborate.
    - System saves changes and records version history.
    - AI suggests topic links based on the content.
- **Alternate Flows:**
    - **1a:** If internet connection fails, the system autosaves locally and syncs later.
    - **2a:** If collaboration invite fails, user receives an error notification.

Use Case 2: "Ask a Doubt" Discussion

- **Use Case ID:** UC-02
- **Title:** Post Academic/Practical Doubts and Receive Answers
- **Description:** A student posts a course-related question in a discussion thread; peers, professors, or alumni can respond.
- **Actors:** Student (primary), Professor and Alumni (secondary)
- **Preconditions:** User must be logged in.
- **Postconditions:** Question appears in the thread, available for replies.
- **Triggers:** Student clicks "Ask a Doubt" and submits a query.
- **Main Flow:**
    1. Student navigates to the course discussion section.
    2. Clicks "Ask a Doubt."
    3. Enters a question and optionally attaches notes or screenshots.
    4. System posts the question and notifies relevant users.
    5. AI suggests related existing discussions or materials.
    6. Users reply and upvote helpful answers.
- **Alternate Flows:**
    - **1a:** If the question has already been asked and answered in a different discussion thread, AI analyses the context of the thread and pulls up that thread and reccomends the user to go through it so that repetition is avoided making the system more efficient.

- **2a:** If no responses within 24 hours, AI re – routes the question by analysing contexts to find corresponding answers via alumni, professors or other students.

Use Case 3: Professor Responds to Student Notes

- **Use Case ID:** UC-03
- **Title:** Professor Reviews Collaborative Notes
- **Description:** Professors view and provide feedback on student-created notes.
- **Actors:** Professor (primary), Student (secondary)
- **Preconditions:** Professor must be verified and linked to a faculty/course.
- **Postconditions:** Comments or suggestions are visible to all collaborators.
- **Triggers:** Professor receives a tag or request notification.
- **Main Flow:**
  1. Professor logs in to the platform.
  2. Views notifications for tagged notes or "Ask a Doubt" threads.
  3. Opens the note and adds comments or corrections.
  4. System saves feedback and alerts all collaborators.
- **Alternate Flows:**
  - **1a:** Professor declines to review; request is reassigned.
  - **2a:** If duplicate content exists, AI merges suggestions.

Use Case 4: AI-Powered Alumni Matching

- **Use Case ID:** UC-04
- **Title:** Alumni Matching and Discussion Threads
- **Description:**
  Students can post questions or discussion threads related to academic or career topics. The AI analyzes the keywords and context of each post, then personalizes the student's feed with similar content and connects relevant alumni who can provide insights. Alumni receive notifications for posts that match their expertise or study background and can choose to respond. This minimizes unnecessary messaging and ensures meaningful, focused interactions.
- **Actors:**
- **Student** (primary)
- **Alumnus/Alumna** (secondary)
- **AI Recommendation Engine** (system actor)
- **Preconditions:**
  - Both students and alumni have active profiles with relevant academic or professional fields filled in.
  - The AI recommendation system is trained on profile and interaction data.
  - **Post conditions:**
  - The thread is visible to relevant alumni and students.
  - Personalized feed updates for users engaging with similar content.
  - Responses are logged and can contribute to the AI's learning for better future recommendations.
- **Triggers:**

- A student posts a new thread or question.

Main Flow:

1. The student logs in and navigates to the “Discussion” or “Ask Alumni” section.
2. The student creates a thread by typing a question or discussion topic.
3. The system (AI engine) analyzes the content using keywords and context.
4. The AI personalizes the student’s feed based on interests and previous engagements.
5. The AI identifies relevant alumni profiles based on expertise, program history, and interaction data.
6. Alumni matching those parameters receive notifications about the new thread.
7. Interested alumni view the thread and may respond publicly.
8. The system notifies the student when a response is received.
9. Both participants can continue the discussion in the thread if they wish.

Alternate Flows:

- **4a.** If no immediate match is found, the system still displays the thread publicly and suggests it to users with related interests over time.
- **7a.** If an alumnus chooses not to respond, no notification is sent to the student, preventing spam or unwanted follow-ups.
- **8a.** If multiple alumni respond, the AI may highlight the most relevant or up voted responses first.

User stories: *A user story is a brief, informal description of a feature from the perspective of the end user. It focuses on the value and outcome for the user.*

Academic Support

- As a student, I want the system to automatically link related topics across notes and discussions so that I can easily explore connected ideas and study more efficiently.
- As a student, I want to post course-specific doubts so that I can get quick help from anyone in the university who is familiar with the subject.
- As a professor, I want to view and respond to student discussion threads so that I can guide them academically even outside the classroom environment.
- As a student, I want the AI to suggest study groups based on my courses and learning style so that I can collaborate with others who share similar study goals.

Practical Support

- As a student, I want to post a question or discussion thread so that the AI can connect me with alumni who have relevant experience and are willing to respond.
- As an alumnus, I want to receive questions related to my field of study or career so that I can provide guidance to students without being directly messaged.
- As a student, I want to see a map of nearby campus resources such as study areas and housing options so that I can easily find what I need.
- As a student, I want to see events, workshops, and opportunities that match my interests so that I can participate in relevant campus activities.

- As a student, I want to browse a board of career and internship advice shared by alumni so that I can make informed decisions about my career path.
- As a student, I want my questions to be automatically directed to the right person or community so that I can receive accurate and quick answers.

6. Non-Functional Requirements

***Performance**: Describe the performance requirements, such as response time and throughput.*

- The system should load primary pages (e.g., home feed, note view, discussion threads) within **3 seconds** under normal network conditions.
- AI-assisted recommendations (e.g., topic linking, alumni matching) should return results within **5 seconds** of a query being made.
- The application should support **simultaneous access by at least 5000 users** without degradation of performance.
- Background AI services such as content analysis and personalized feed generation will run asynchronously to ensure smooth user interaction.
- The system should maintain an uptime of **99%** during academic semesters.

***Security**: Security requirements, including authentication and authorization.*

- All users (students, alumni, and professors) must authenticate through a **secure login system** (e.g., OAuth 2.0 or university SSO integration).
- Passwords and sensitive data will be **encrypted using industry-standard hashing algorithms (e.g., bcrypt)**.
- All communication between client and server will be protected with **HTTPS and SSL/TLS encryption**.
- Access control will ensure **role-based authorization**, restricting actions according to user type (e.g., only professors can verify academic threads).
- AI systems handling personal data will comply with **data protection laws (e.g., GDPR)** to ensure privacy and transparency in data use.
- Users will have full control over what personal information they choose to share publicly.

***Usability**: Usability standards and requirements.*

- The interface will be designed to be **intuitive and accessible** for users across different technical backgrounds.
- Navigation should be consistent and clear, allowing users to complete key tasks (posting a doubt, viewing notes, connecting with alumni) within **three clicks**.
- The platform will follow **WCAG 2.1 accessibility guidelines**, ensuring usability for individuals with visual or motor impairments.
- Support for both **mobile and desktop devices** will be provided via responsive web design or mobile application versions.
- Clear visual cues, tooltips, and contextual help will guide users during their first interactions with the app.

***Reliability**: Requirements for system reliability and availability.*

- The system will use **redundant servers and failover mechanisms** to maintain availability in case of hardware or software failures.
- All user data (notes, threads, profiles) will be **backed up daily**, with an automated recovery system capable of restoring the last known state.
- In case of AI service downtime, users will still be able to manually search, post, and collaborate without loss of core functionality.
- The system should maintain an error rate of **less than 0.5%** for all critical operations (e.g., posting, saving, updating).
- A monitoring and logging service will continuously track uptime and system health metrics.

***Scalability**: Requirements for system scalability.*

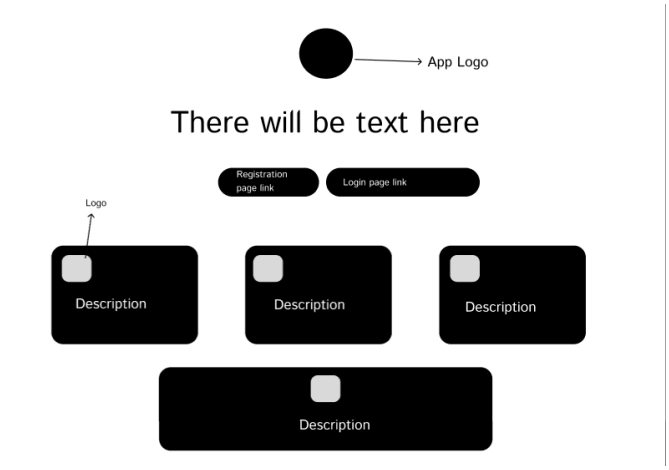
- The application will be built using a **modular, service-oriented architecture**, allowing independent scaling of AI, database, and front-end layers.
- As the number of students and alumni increases, the system should be able to scale horizontally using **cloud infrastructure (e.g., AWS, Google Cloud, or Azure)**.
- Caching mechanisms (e.g., Redis or Memcached) will be employed to optimize response times during high-traffic periods.
- The AI modules will be designed to handle increased input volumes by dynamically allocating computational resources.

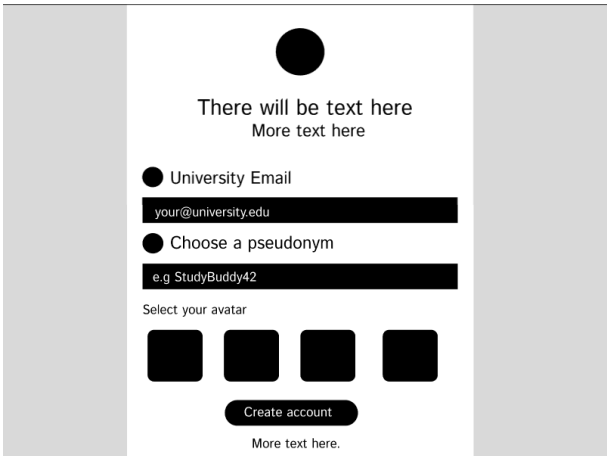
***Compliance**: Legal and regulatory compliance requirements*

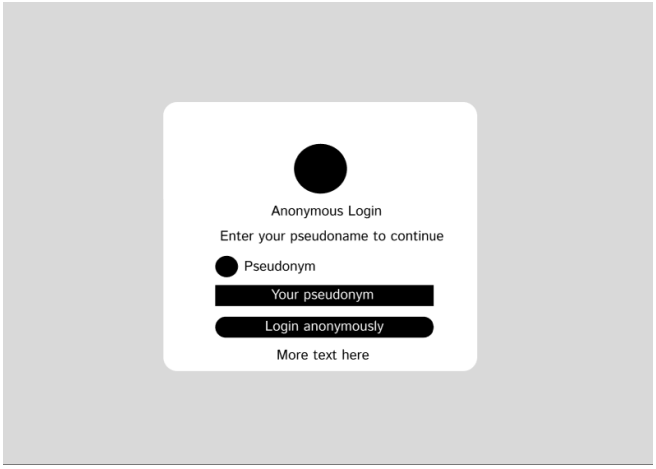
- The system must comply with **General Data Protection Regulation (GDPR)** standards for handling and processing personal data.
- All content uploaded by users must adhere to **university communication policies** and community guidelines.
- AI-generated suggestions or responses will be **transparent and explainable**, avoiding bias in matching or content recommendation.
- Data collection and storage will follow **institutional review board (IRB)** or **university ethical standards** for research-based systems.
- The platform will provide clear **terms of service and privacy policy** for all users upon registration.

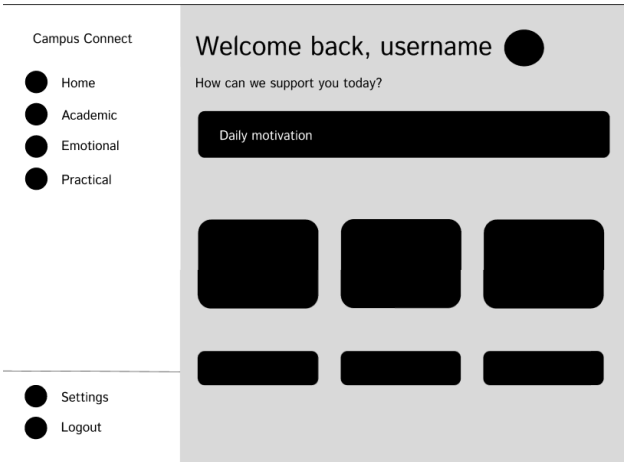
7. User Interface

***Wireframes/Mockups**: Visual representations of the user interface.*









***User Interaction**: Description of how users will interact with the system.*

The interaction design emphasizes **simplicity and engagement**, ensuring smooth navigation and logical task flow for all user types.

- **Students:**
  - Create and edit collaborative notes with peers.
  - Post academic or career-related threads.
  - Join or leave study groups.
  - View AI-suggested connections and relevant posts in their feed.
  - Browse nearby campus resources via the Geo Help Board.
- **Professors:**
  - Review and respond to academic discussion threads or student-submitted notes.
  - Access analytics about popular course topics to identify common learning difficulties.
  - Optionally endorse accurate peer responses, increasing thread credibility.
- **Alumni:**

    - Receive filtered student queries that match their domain of experience.
    - Respond to relevant posts without being overwhelmed by direct messages.
    - Share guidance and resources through the career and internship boards.
- **AI System:**
    - Analyses content, keywords, and engagement data to personalize user feeds.
    - Suggests study groups, related notes, or suitable alumni connections.
    - Ensures posts are routed to the most relevant communities or individuals.

User interactions follow an **event-driven flow**, meaning every action (posting, responding, editing, etc.) triggers AI-assisted updates and database synchronization in real time.

***Navigation**: Explanation of the navigation flow.*

The navigation is designed for **clarity, consistency, and minimal cognitive load**.
Each section is accessible through a **persistent navigation bar** or a **sidebar menu**, depending on the device.

Navigation Flow Summary:

1. **Login/Signup → Dashboard:**
   After authentication, users land on a personalized dashboard displaying their feed and quick access to features (Notes, Discussions, Alumni, Map, Profile).
2. **Dashboard → Feature Modules:**
   Users can access:
    - **Notes** for academic collaboration.
    - **Ask a Doubt** for question threads.
    - **Alumni Hub** for practical support.
    - **Geo Help Board** for campus guidance.
    - **Profile Settings** for personal customization.
3. **Feature Module → Detailed View:**
   Selecting any item (e.g., a post, note, or thread) opens a **detail view** with full comments, AI recommendations, and related content.
4. **Global Navigation Controls:**
    - A **Search bar** enables content, user, or topic discovery across modules.
    - A **Notification panel** alerts users about replies, endorsements, or study group invites.
    - A **Floating action button (FAB)** allows quick posting from anywhere in the app.

The navigation prioritizes **seamless context switching**, allowing users to move between academic and practical features without losing their workflow.

8. Data Requirements

From functional, not from technical or database design point of view.

***Data Model**: Entity-Relationship Diagrams (ERD) or other data models.*

- User Profile 1 --- creates --- Thread Post

- Thread Post 1 --- receives --- Reply
- Thread Post 1 --- generates --- Engagement Data
- Engagement Data --- analyzed by --- AI Engine
- AI Engine --- extracts --- AI Keywords
- AI Keywords --- link relevant --- User Profile

***Data Dictionary**: Detailed descriptions of data elements, including types and constraints.*

Data Element	Description	Used By	Functional Purpose
User Profile	Represents both students and alumni, storing details like name, area of study, interests, and expertise.	Student, Alumni, AI Engine	Enables personalization and targeted content delivery.
Thread Post	Discussion or question posted by users.	Student, Alumni, Professors	Encourages academic and career collaboration.
Reply	Response by alumni or other students to a thread.	Alumni, Student, Professors	Enables interaction and mentorship.
Engagement Data	Tracks interactions such as likes, views, and comments.	AI Engine	Helps the AI refine recommendations and content relevance.
AI Keywords	Keywords extracted from posts and replies.	AI Engine	Used for content analysis and alumni matching.

Data Flow: Description of how data will flow through the system.

1. User Interaction:

- A student logs in and creates a **Thread Post** (e.g., "How do I prepare for a data structures interview?").
- The system stores this as a new **Thread Post** entity.

2. AI Processing:

- The **AI Engine** analyzes the post content to extract **AI Keywords** like *"data structures"*, *"interview prep"*.
- Based on these keywords and past **Engagement Data**, the AI identifies **relevant alumni/Professors/Students** who have expertise in similar topics.

3. User Notification:

- Those users receive notifications or see the thread appear in their personalized feed.
- They may choose to respond, creating new **Replies** linked to that post.

4. Engagement and Learning:

- Other users can view, like, or comment on the replies.
- These actions update the **Engagement Data**, which the **AI Engine** continuously analyzes to further personalize feeds.

5. Continuous Personalization:

- Over time, the **AI Engine** uses historical engagement to refine each user's experience — showing more relevant threads, alumni advice, or topics of interest.

9. Integration

Integration Points: Description of how the system will integrate with other systems or components.

The *AI-powered Student and Alumni Support Network* integrates various system components and, where appropriate, connects to external services to enhance functionality and user experience. The integration design ensures that all modules — academic collaboration, AI recommendations, alumni interaction, and campus navigation — operate seamlessly as part of one cohesive ecosystem.

Internal Integration Points

These define how **modules within the application** communicate with one another:

- **User Management ↔ AI Personalization Engine:**
  User profiles, interests, and activity data are sent to the AI engine to generate

personalized recommendations, such as suggested study groups, relevant alumni threads, or academic resources.

- **Collaborative Notes ↔ Discussion Threads:**
  Notes and discussion modules are integrated so that when a user highlights or references a concept in a note, related discussion threads appear automatically through AI-assisted linking.
- **Discussion Threads ↔ Professor & Alumni Roles:**
  The academic and practical support systems are interconnected, allowing professors and alumni to access and respond to student-generated threads relevant to their expertise areas.
- **Resource Repository ↔ Course Database:**
  Study materials, FAQs, and past exams are categorized by course and accessible from both notes and thread pages, ensuring academic content remains contextually available.
- **Geo Help Board ↔ User Location Services:**
  The geo-based help board integrates with device location data to show nearby campus resources (e.g., study spaces, labs, housing). Data is retrieved on-demand and not stored persistently.
- **Event Feed ↔ User Interests:**
  Events and opportunities are filtered and recommended based on users' academic background, department, and activity patterns, integrating tightly with the AI recommendation logic.

External Integration Points

While the application primarily functions as a standalone ecosystem, limited integration with trusted external services enhances functionality:

- **AI and NLP Services:**
  Third-party Natural Language Processing APIs (e.g., OpenAI API, Google Cloud Natural Language) may be used for topic extraction, keyword identification, and thread categorization.
- **Map and Location Services:**
  Integration with **Google Maps API** (or a similar open-source alternative) enables geo-tagged campus resource display and simple route visualization.
- **Email / Notification Services:**
  Integration with **SMTP or push notification services** allows users to receive alerts about thread replies, study group invitations, or new events.
- **Authentication Services (Optional):**
  University Single Sign-On (SSO) or OAuth 2.0 may be integrated for secure identity verification, particularly when collaborating with institutional networks.

APIs: Details of any APIs used or provided by the system.

The system will both **consume external APIs** and **expose internal APIs** to ensure modularity and scalability.

External APIs (Consumed)

API/Service	Purpose	Integration Type
Google Maps API	Displays campus locations, study spaces, and housing areas	REST
NLP/AI API	Topic linking, keyword extraction, and content analysis for personalization	REST
Email/Notification API	Sends verification emails, alerts, and activity notifications	REST
University SSO	Secure authentication for professors and students	Oauth2.0

Internal APIs (Provided by the System)

API Name	Purpose	Accessible By
/api/users	Handles user registration, login, and profile management	Frontend Modules
/api/notes	Manages creation, editing, and version history of collaborative notes	Node Module, AI engine
/api/threads	Handles creation, retrieval, and AI categorization of discussion threads	Academic and Practical Modules
/api/reccomendations	Provides AI-driven suggestions based on behavior and keywords	Threads Feed/Resource Library

/api/resources	Retrieves study materials, FAQs, and past exam data	Repository module
/api/geo	Fetches nearby campus resources using location input	Geo help board
/api/events	Returns personalized campus events and opportunities	Event feed
/api/analytics	Tracks user engagement metrics for AI learning	Internal Only

10. Testing requirements

Explain the predefined (technology independent) requirements of testing. Specify the testing criteria and methodologies for validating the functional requirements.

The main objectives of testing this system are to:

- Verify that all functional requirements are implemented correctly.
- Ensure that the application performs reliably under different user scenarios.
- Validate that user experience remains consistent and intuitive across all roles (students, professors, alumni, and admins).
- Confirm that the AI-based recommendation and matching mechanisms operate as intended and do not generate irrelevant or biased results.
- Identify and resolve any usability or accessibility issues before deployment.

Testing Criteria

Criterion	Description	Validation Goal
Functional Accuracy	Each feature (e.g., thread creation, reply submission, AI keyword extraction) should perform its intended task correctly.	Confirm that every use case works according to specification.
Data Integrity	All data entered or generated (e.g., user profiles, posts, replies) must remain	Ensure no data loss, duplication, or corruption

	consistent across the system.	occurs.
Usability	The system should be intuitive and easy to navigate for users of different backgrounds.	Validate through user feedback sessions and task completion tests.
Performance	Response times for user actions (posting, viewing feeds, or receiving recommendations) must remain within acceptable limits.	Measure latency and throughput during simulated load tests.
Security and Privacy	User data, including profiles and posts, must be protected through secure access control and anonymization where applicable.	Test for proper authentication, role-based access, and data protection.
Reliability	The application should continue to function without failure over extended periods of typical use.	Conduct endurance and stress testing.
AI Relevance	Recommendations and alumni matches should align with the context of user posts and preferences.	Evaluate AI output quality against sample test cases.

Testing Methodologies

1. **Requirement Based Testing** - Each functional requirement will be mapped to one or more test cases. This ensures all system functionalities are explicitly validated.
2. **Use Case Testing -** All identified use cases (e.g., creating a thread, posting a reply, forming a study group) will be simulated end-to-end to verify system behavior and expected outputs.

3. **Black – Box Testing -** Focuses on validating the system's input-output behavior without considering internal code logic. Example: verifying that when a student posts a question, relevant alumni receive notifications.
4. **User Acceptance Testing -** Conducted with a small group of students, professors, and alumni to ensure that the platform meets user expectations in real-world scenarios.
5. **Usability Testing -** Checks navigation flow, clarity of layout, and overall user satisfaction through task-based trials and feedback forms.
6. **Integration Testing -** Verifies the communication between the main components — e.g., between the AI engine, user interface, and feed personalization logic.
7. **Performance Testing -** Assesses system responsiveness under various loads (e.g., simultaneous thread posts or multiple feed requests).
8. **Security Testing -** Ensures proper authentication, authorization, and data protection protocols are in place. Includes login validation, session management checks, and access control tests.

Acceptance Criteria

The system will be considered **ready for deployment** once:

- All critical and high-severity defects have been resolved.
- 100% of functional test cases pass successfully.
- User satisfaction scores from UAT exceed 85%.
- The system demonstrates stable performance with under 3 seconds of average response time under moderate load.
- No security vulnerabilities are left unaddressed.