Software Requirements Specification

for

AcademicAssistant: Your Academic Helper

Prepared by: Group-5
Zulker Nain Mahbub (2112287042)
Muhamamd Raiyan Alam (1831100642)

North South University, Computer Science Engineering
(CSE327-SECTION-7)

8th April, 2024

Contents

Chapter 1 Introduction

Introduction: Transforming the NSU Experience with AcademicAssistant
The NSU community thrives on connection, growth, and collaboration. However, existing technologies may only partially cater to these needs.We propose AcademicAssistant, a dynamic online platform that empowers NSU students by fostering a connected, collaborative, and skill-sharing environment. By implementing AcademicAssistant, we aim to create a unified NSU community hub that empowers students to connect, grow, and succeed throughout their academic and professional journeys.

1.1 Purpose

The purpose for AcademicAssistant delves deeper into the functionalities, development phases, and expected benefits of AcademicAssistant. We are confident that AcademicAssistant will offer a valuable resource for the NSU community, empowering individuals and fostering a culture of collaboration and success.

This SRS outlines the development plan for AcademicAssistant, a platform built with user-friendliness that leverages familiar technologies like HTML, CSS and Bootstraps. AcademicAssistant will mainly serve purposes on:

- **Strengthen connections:** Students can engage, discuss, and build meaningful relationships through forums, chat rooms, and an event calendar.
- **Fuel personal and professional growth:** Skill listings, workshops, and collaboration tools equip students with valuable skills and opportunities for advancement.
- **Promote collaboration and teamwork:** AcademicAssistant facilitates collaborative projects, resource sharing, and teamwork fostering a supportive learning environment. By implementing AcademicAssistant, we aim to create a unified NSU community hub that empowers students to connect, grow, and succeed throughout their academic and professional journeys.

1.2 Intended Audience

AcademicAssistant is designed for the University community, including students, faculties and admins. Its intended audience primarily consists of the following:

1) **Students:** AcademicAssistant offers students a platform to connect with peers and find collaborators for projects through chatting system, participate in discussions relevant to their studies and career interests, can create poles to

have votes based on their opinions, and stay updated on on-campus events and resources through Newsfeeds. They can use calender for marking and looking for any events.Students can rank faculties through top faculty options as well. They can keep an eye on notifications based on academic support, networking opportunities, or extracurricular engagement, activities from university. AcademicAssistant is a hub for student interaction and growth.

2) **Faculty:** Faculty members can use AcademicAssistant to broadcast information about workshops, seminars, their timetables of classes or extra classes as well and other academic events to a interested student audience. Faculties can make schedule in the Google Calender which will be handled with the help of API, share resources, and chat with students.They can also engage with students through AcademicAssistant, develop collaboration, and contribute to the vibrant scholarly community by sharing resources, insights, and expertise.Faculties can also communicate with the club in order manage seminars and workshops.
3) **Admin:** Admins manage AcademicAssistant, monitoring activity of registered users, updating users, gathering feedback, providing support, enforcing policies and managing complains. They maintain platform integrity, engagement, and community guidelines adherence, ensuring a safe, functional environment for collaboration within the university community. Admins can also find out wat new features can be added and what to make changed in the Software. Admins play a crucial role in fostering a vibrant online community, facilitating connections, and promoting growth among students and faculty.

1.3 Intended Use

For the advancement scenario of the university, AcademicAssistant emerges as a vital tool to developing an active community where students and teachers connect, collaborate, and develop academically and professionally. Considering the current situation, AcademicAssistant serves as a centralized hub, offering diverse communication channels, and collaboration tools. By providing a unique platform, AcademicAssistant seamless access to resources, empowering students to enhance their academic prospects. Through skill listings, workshops, and collaboration features, AcademicAssistant facilitates and students can make a better collaboration learning, and a easier communication User Stories and USE cases shows how AcademicAssistant helps students to search for contributor, engage in discussions, and see workshop announcements while providing a platform for faculty to reach a larger audience. With its user-friendly interface and comprehensive features, AcademicAssistant aspires to become the go-to platform for university-community students to connect, grow, and succeed in their academic and professional life.

1.4 Product Scope

AcademicAssistant aims to address the evolving needs of the University-Community by providing an exclusive platform for communication, skill development, and collaboration. With existing solutions lacking a centralized hub and robust skill development features. AcademicAssistant is a user-friendly solution built on familiar technologies.
The platform offers news feeds, chat rooms, poll options, requesting for blood, post a listing, offers top faculties through student's points, an event calendar to facilitate communication and community building.
AcademicAssistant features skill listings, workshops, and collaboration tools to empower students in skill acquisition and sharing. Integrated tools for project management, resource sharing, and teamwork foster a culture of innovation and collaborative learning.
Through user stories and use cases, AcademicAssistant can connect students with specific skills, help discuss on various topics, enable faculty to promote workshop announcements and help students showcase their qualifications to potential recruiter. By addressing these needs comprehensively, AcademicAssistant aspires to become the primary platform for NSU students to connect, grow, and succeed academically and professionally.

Goals:
Gain insights into student learning through participation in online discussions and collaborative projects with faculties. Facilitate peer review and feedback mechanisms within the platform. Potentially access anonymized student feedback on teaching methods and course content (optional future feature).

Scopes for Students:

Develop and showcase skills: Gain experience, network with peers, and build a strong portfolio.
Enhanced learning: Participate in collaborative projects, access valuable resources, and attend workshops led by faculty and industry professionals.
Improved communication: Connect with peers and faculty, build relationships, and find collaborators for projects.
Streamlined organization: Manage schedules, deadlines, and project tasks efficiently with integraBenefits for Faculty:

Scopes for Faculties:

Enhanced Communication and Engagement:
Better communication with students outside the classroom through forums, chat rooms, and announcements. Improved accessibility for students to ask questions, clarify concepts, and receive timely feedback. Opportunity to host online discussions and Q&A sessions to foster deeper student understanding.
Streamlined Workshop and Event Management:
Utilize the platform to announce and manage workshops, seminars, and office hours, reaching a wider student audience. Potentially integrate with Google Calendar API to simplify scheduling and avoid conflicts. Engage students in collaborative learning activities within the platform's dedicated tools.
Skill Sharing and Collaboration: Showcase expertise and research interests on a dedicated faculty profile (optional future feature). Collaborate on projects with colleagues across departments through resource sharing and discussion tools. Identify and connect with students possessing valuable skills for potential research collaborations.

1.5 Risk Definition

Risk Definition for AcademicAssistant :

1) **Adoption Challenges:** Some students and faculty may only accept AcademicAssistant due to preference for existing platforms or for choosing a better one-stop service, for having a better opportunity or not willing to engage in better community.
2) **Technical Issues:** AcademicAssistant reliance on technology could lead to technical glitches, such as server downtime or software bugs, which could disrupt user experience and undermine trust in the platform.
3) **User Engagement:** Low participation rates in discussions, workshops, and skill-sharing activities may limit AcademicAssistant's ability to cultivate a better and active community, diminishing its overall impact.
4) **Privacy Concerns:** Users may express concerns regarding the privacy and security of their personal information on AcademicAssistant, which could lead to hesitancy in sharing skills or participating in discussions.
5) **Scalability Challenges:** As AcademicAssistant gains popularity and the user base grows, scalability issues may arise, causing performance degradation or system overload during peak usage periods.

Chapter 2: Overall Description

2.1 User Classes and Characteristics

1) **Students:**
    a) **Characteristics:** Programming or Study Interests, Programming or Academic Skills and previous Project or Academic Record.
    b) **Goals:** Collaborate on projects, access resources, engage in discussions, and enhance themselves by deliver.
    c) **Preferences:** Easy navigation, skill search functionality, discussion forums, and networking opportunities.
2) **Faculty Members:**
    a) **Characteristics:** Academic professionals with expertise in specific subjects or fields.
    b) **Goals:** Share workshop announcements, disseminate resources, and facilitate student engagement.
    c) **Preferences:** Efficient event posting, resource sharing, and communication tools.
3) **Admins:**
    a) **Characteristics:** Responsible for overseeing AcademicAssistant platform operations and maintenance.
    b) **Goals:** Ensure platform functionality, security, and adherence to policies.
    c) **Preferences:** Comprehensive user management, moderation tools, and system monitoring capabilities
    .

2.2 User Needs

1) **Students:**
    a) Need to find collaborators for projects.
    b) Need to access resources for skill development.
    c) Need to engage in discussions relevant to their studies or career interests.
    d) Need to enhance their employability and career prospects.
2) **Faculty Members:**
    a) Need to share any event announcements and resources with students efficiently.
    b) Need to facilitate student engagement and participation in academic activities.
    c) Need to ensure all interested students are aware of relevant events and opportunities.

3) **Admins:**
    a) Need to maintain platform functionality, security, and compliance with policies.
    b) Need to manage user accounts, permissions, and content moderation effectively.
    c) Need to monitor platform usage and address any technical or policy-related issues.

2.3 Operating Environment

This section delivers into AcademicAssistant technical aspects, outlines the architecture, front-end plan, back-end development approach, and performance considerations.

1) **Architecture:**
    a) AcademicAssistant will adopt a Model-View-Controller (MVC) architecture, leveraging the MERN stack for robust functionality.
    b) **Front-end (View):** Utilizing Bootstrap for the user interface, ensuring a seamless experience across various devices and browsers.
    c) **Back-end (Model and Controller):** Employing MongoDB for database management, Express.js for server-side framework, and Node.js for server-side runtime environment. This setup enables efficient data handling and server-side logic implementation.
    d) This architecture allows for potential future integrations with other university community systems or external services.
2) **Front End Plan:**
    a) **Responsive design:** The platform will seamlessly adapt to different screen sizes, providing an optimal user experience on desktops, tablets, and mobile devices.
    b) **Intuitive navigation:** A user-friendly navigation bar will enable easy access to different sections of the platform, enhancing usability.
    c) **Search functionality:** Users can efficiently search for forums, events, skill listings, and other users based on relevant criteria, enhancing discoverability.
    d) **User profiles:** Personalized profiles will empower students to showcase their skills, interests, and contact information, fostering networking and collaboration.

3) **Back-end Development:**
    a) **Secure user authentication and authorization**: Implementing a strong authentication and authorization mechanisms to ensure data privacy and access control, enhancing platform security.
    b) **Database management:** Following to efficient database management practices to maintain data integrity and optimize data retrieve for smooth user experience.
    c) **API development:** Developing APIs as needed to promote smooth integration with other systems, enhancing the platform's extensibility and interoperability.

4) **Performance Plan:**
    a) **Code optimization:** Clean and efficient coding practices will be followed to minimize resource usage and improve loading times.
    b) **Caching mechanisms:** Implementing caching mechanisms will improve response times by storing frequently accessed data.
    c) **Server monitoring and load balancing:** The platform will be monitored to identify and address potential performance bottlenecks. Load balancing may be implemented to distribute traffic efficiently if necessary.
    d) **Google Page Evaluation:** While under development, the website will prioritize Core Web Vitals, mobile-friendliness, and security for an optimal user experience.

2.4 Constraints

Limited feature set: Initial functionalities will focus on core communication and skill sharing features, with additional features like collaborative tools planned for future iterations

Scalability: As the user base grows, the platform might require further optimization to ensure smooth operation and efficient performance.

Integration with Existing Systems: Seamless integration with NSU's existing student information systems or university portals may necessitate further development and collaboration with relevant stakeholders.

2.5 Assumptions

1) Users can read and write English.
2) Users have devices that support internet service.
3) Users should have an internet connection .
4) Users are familiar with web browsing and can interact with website

Chapter 3 Requirements

3.1 Functional Requirements:

Functional requirements outline the specific functionalities and features that AcademicAssistant must possess to meet the needs of its users and achieve its objectives. These requirements are essential for the platform's core functionality and user interaction.

i) **User Authentication:** AcademicAssistant should provide secure user authentication mechanisms to ensure that only authorized users can access the platform's features and functionalities.

ii) **Skill Listings:** The platform should allow students to create profiles and list their skills, interests, and expertise. Users or another student should be able to search for one student based on specific skills to find potential collaborators for a project so that they can make a project and show the project in their CV.

iii) **Event Calendar:** The platform should have an event calendar where a faculty members can post announcements about workshops, seminars, and other events relevant to students' academic pursuits. Clubs and Faculty can also make announcements and make an event about a workshop or events.

iv) **Messaging or Chatting System:** AcademicAssistant should provide a messaging or chat system that allows coursemates to communicate with each other publicly in a group where there will be only coursemates in that group. In the public group made only for student will able them to discuss over any topic regarding study.

Examples of functional requirements for AcademicAssistant include:

1) **As a Student,**
I want to can search for a project collaborator
so that, I want to collaborate on a project

Confirmation:
Search for

- a person available with required
- Programming Language,
- Programming framework, and/or
- Previous Similar Programming Experience.

2) **As a Student,**
I want to collaborate on a project
So that, I can embrace my skills

Confirmation:

- I can fill up a form, that is available in the system, and
- offer that the person wants to collaborate.

3) **As a Student,**
I want to keep record of my course timing and communicate with my coursemate
So that, I dont miss track and class

Confirmation:

- I will keep the record in the Google Calender.
- Person with same course and section can communicate with each other in publicly with all the coursemate of a group.

4) **As a faculty,**
I wants to take a seminar or workshop
So that, People can adjust in the industry after graduation

Confirmation:

- The faculty can set en event in google calender,
- Announce prerequisites and necessary things

5)**As a Student,**
I want to, rate my faculty
So that, other student can get a review

Confirmation:

- Give a respected faculty star rating
- Provide a text review.

6) **As s student or a faculty**,
I want to discuss or request my other student
So that, I inform people different affairs in the university and stay connected with various topic.

Confirmation:

- I will post a status to the whole community
- People can reply to that status

7) **As a student,**
I want to get various announcement
So that, I don't miss different events

Confirmation:

- Admin can post important event
- I don't miss an event

8) **As a student,**
I want to be able to share files with my coursemates or collaborators for projects.
So that, we can collaborate projects

Confirmation:

- Upload and share files within specific groups or projects.

9) **As a student or faculty member,**
I want to be able to create an announcements and manage tasks related to projects, assignments, or events.
So that, I can either add course to organise my Class and create Workshop event

Confirmation:

- Users can create tasks, assign them to individuals or groups, set deadlines, and track progress.

10) **As an admin,**
I want to manage user complains and monitor user activity
So that, the platform stays free from unrest and maintain clean environment

Confirmation:

- Admin can review and delete them
- Block people from using a platform

11) **As an admin,**
I can give access to the students about what they post on the newsfeed.
So that, student can discuss over any affairs or necessary points in the community.

Confirmation:

- Admin can see all post and delete them

3.2 Non Functional Requirements:

Non-functional requirements define the quality attributes and constraints that AcademicAssistant must adhere to in terms of performance, security, usability, and other aspects. These requirements ensure that the platform operates effectively and provides a satisfactory user experience. Examples of non-functional requirements for AcademicAssistant include:

1) **Performance:** AcademicAssistant should be able to handle a large number of concurrent users without experiencing significant performance degradation. Response times for user interactions should be fast and consistent across different devices and network conditions.
2) **Security:** The platform should implement robust security measures to protect user data and prevent unauthorized access. This includes encryption of sensitive information, secure user authentication mechanisms, and regular security audits and updates.
3) **Usability:** AcademicAssistant should have an intuitive and user-friendly interface that is easy to navigate and understand. Users should be able to access features and functionalities with minimal effort and confusion.
4) **Reliability:** The platform should be highly reliable, with minimal downtime and disruptions. It should be able to recover quickly from failures and errors to ensure continuous availability to users.
5) **Scalability:** AcademicAssistant should be designed to scale seamlessly as the user base grows. It should be able to accommodate increasing numbers of users and data volumes without compromising performance or functionality.

Appendix A:

Glossary

This glossary defines key terms used throughout this Software Requirements Specification (SRS) document for AcademicAssistant.

- **API (Application Programming Interface):** A set of protocols, definitions, and tools that allow for communication between different software applications.
- **Authentication:** The process of verifying a user's identity before granting access to a system.
- **Authorization:** The process of determining the level of access a user has to specific features or functionalities within a system.
- **Cache:** A temporary storage location that holds frequently accessed data to improve retrieval speed.
- **Collaboration:** The act of working together to achieve a common goal.
- **Communication Channels:** Methods used to exchange information between users, such as forums, chat rooms, and messaging systems.
- **Database:** A collection of structured data organized electronically for efficient access and retrieval.
- **Event Calendar:** A calendar application used to schedule, track, and share events.
- **Faculty:** University instructors who teach courses and conduct research.
- **Front-End (View):** The user interface portion of a software application that users interact with directly.
- **MVC (Model-View-Controller):** A software architectural pattern that separates an application into three parts: the Model (data), the View (presentation), and the Controller (user interaction).
- **Node.js:** An open-source, cross-platform JavaScript runtime environment used to build server-side applications.
- **Performance:** The ability of a system to respond to user requests quickly and efficiently.
- **Potential Employers:** Organizations or individuals seeking to recruit NSU students for internships or job opportunities.
- **Responsive Design:** A design approach that ensures a website or application adapts its layout to different screen sizes and devices.
- **Scalability:** The ability of a system to handle increasing amounts of data or users without compromising performance.

- **Security:** Measures taken to protect a system from unauthorized access, use, disclosure, disruption, modification, or destruction.
- **Server-Side (Model and Controller):** The back-end portion of a software application that handles data processing and business logic.
- **Skill Listings:** A feature that allows users to showcase their skills and expertise on a platform.
- **Students:** Individuals enrolled in academic programs at NSU.
- **User Interface (UI):** The elements on a computer screen that users interact with to control a software application.
- **User Management:** The process of adding, modifying, and deleting user accounts within a system.
- **Workshop:** An educational event or program that focuses on a particular skill or topic.