Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The UCSH Posting and Chatting System aims to facilitate seamless communication and information sharing within the University of Computer Science, Hpa-An (UCSH). It allows students and faculty members to post academic updates, share content, and engage in real-time discussions. This document specifies the software requirements for the system.

1.2 Document Conventions

This document follows the IEEE 830 standard for Software Requirements Specification (SRS).

1.3 Intended Audience and Reading Suggestions

- **Developers:** To understand system requirements.
- **Project Managers:** To oversee system design and implementation.
- **University Administrators:** To manage platform policies and user roles.
- **End Users (Students & Faculty):** To understand system capabilities.

1.4 Product Scope

The system will provide functionalities including post creation, real-time chat, user profiles, an admin panel, and security features to enhance communication in an academic environment.

1.5 References

- Facebook: Real-time messaging and post interaction functionalities.
- Twitter: Instant updates and threaded discussions.
- Slack: Real-time chat and team collaboration features.
- Discord: Group chat and notification management.
- LinkedIn: Professional networking and content sharing.
- Reddit: Community-driven discussions and upvote-based engagement.
- Telegram: Secure messaging and channel-based communication.
- WhatsApp: Encrypted messaging and group discussions.
- Instagram: Multimedia content sharing and engagement.
- IEEE 830 Software Requirements Specification Standard

2. Overall Description

2.1 Product Perspective

This system is a web-based platform integrating academic communication, announcements, and real-time messaging.

2.2 Product Functions

- **Posting:** Users can create and manage posts.
- **Commenting:** Users can interact with posts via comments.
- **Real-Time Chat:** Users can communicate instantly with real-time updates using `fetch` and `setInterval`.
- **User Profiles:** Customizable user profiles with privacy settings.
- **Admin Panel:** User and content moderation.
- **Media Sharing:** Users can upload and share multimedia content.

2.3 User Characteristics

- **Students:** Engage in discussions, access announcements.
- **Faculty:** Share academic updates and interact with students.
- **Admins:** Manage users, enforce policies, and oversee content.

2.4 Constraints

- Hosted on **InfinityFree** (limitations on resource usage).
- Supports **Burmese and English languages**.
- Data storage managed using **MySQL**.
- Requires a stable internet connection for real-time functionalities.

2.5 Assumptions and Dependencies

- Users have internet access.
- System is primarily accessed via **modern web browsers**.
- Adheres to **university policies and regulations**.

3. Specific Requirements

3.1 Functional Requirements

- **User Registration & Authentication:** Role-based access (Student, Faculty, Admin).
- **Post Management:** Create, edit, delete posts.
- **Commenting System:** Threaded discussion under posts.
- **Real-Time Chat:** One-on-one and group chat updated using `fetch` with `setInterval`.
- **User Profiles:** Customization and privacy controls.
- **Admin Controls:** Role management, content moderation, reporting.
- **Search & Filtering:** Efficient retrieval of posts and messages.
- **Notifications:** Alerts for messages, comments, and friend requests.
- **File Sharing:** Upload/download academic materials.
- **Real-Time Data Fetching:** Implemented using `fetch` with `setInterval` for periodic updates.
- **Media Sharing:** Support for uploading images, videos, and documents.

3.2 External Interface Requirements

- **User Interface:** Responsive, mobile-friendly UI using Tailwind CSS.

- **Hardware Interfaces:** Requires standard web-enabled devices.
- **Software Interfaces:** PHP (Backend), MySQL (Database), Fetch API for real-time data updates.
- **Communications Interfaces:** HTTPS for secure data transmission.

3.3 System Features

- **Security:** Secure authentication with bcrypt hashing.
- **Performance:** New posts and messages should be displayed instantly.
- **Scalability:** Handle increasing users and data volume efficiently.
- **Multimedia Support:** Allow users to share images, videos, and documents.

3.4 Non-Functional Requirements

- **Performance:** Response time <2s for real-time chat.
- **Scalability:** Modular design supporting future enhancements.
- **Usability:** Intuitive design for students and faculty.
- **Availability:** 99.9% uptime required.
- **Security:** Role-based access, HTTPS encryption.
- **Maintainability:** Well-documented code with logging mechanisms.
- **Backup & Recovery:** Regular backups for data protection.
- **Compliance:** Adherence to university data privacy regulations.

4. Appendices & Glossary

- **RBAC:** Role-Based Access Control.
- **HTTPS:** HyperText Transfer Protocol Secure.
- **Fetch API:** JavaScript method for retrieving real-time data updates.
- **Media Sharing:** Functionality allowing users to upload and share multimedia files.