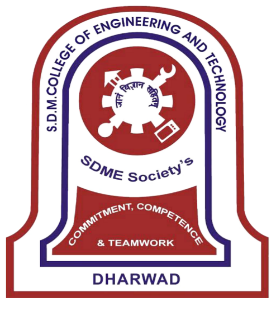
S.D.M.E Society’s

SDM COLLEGE OF ENGINEERING AND TECHNOLOGY, DHAVALGIRI, DHARWAD-580002



(An Autonomous institute affiliated to Visvesvaraya technological university.)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

**Software Requirements Specification Document**

**“MOVIE TICKET BOOKING SYSTEM”**

Under guidance of

**“Prof. Varsha S Jadhav”**

Submitted by

**SHEIKH U 2SD23IS048**

**SRUSHTI G 2SD23IS050**

**SUPRIYA H 2SD23IS052**

**SHREYA I 2SD23IS049**

**3rd Semester B.E Academic Year 2024-25**

**Table of Contents**

**1. Introduction 4**

*1.1 Purpose 4*

*1.2 Scope 4*

*1.3 Definitions, Acronyms, and Abbreviations 4*

*1.4 References 5*

*1.5 Overview 5*

**2. The Overall Description 5**

*2.1 Product Perspective 5*

2.1.1 System Interfaces 6

2.1.2 Interfaces 6

2.1.3 Hardware Interfaces 7

2.1.4 Software Interfaces 7

2.1.5 Communications Interfaces 7

2.1.6 Memory Constraints 7

2.1.7 Operations 8

2.1.8 Site Adaptation Requirements 8-9

*2.2 Product Functions 9-10*

*2.3 User Characteristics 10-11*

*2.4 Constraints 11*

*2.5 Assumptions and Dependencies 12*

*2.6 Apportioning of Requirements*

*12-13*

**3. Specific Requirements 14**

*3.1 External interfaces 14-16*

*3.2 Functions 16-17*

*3.3 Performance Requirements 17-18*

*3.4 Logical Database Requirements 18*

*3.5 Design Constraints 19*

3.5.1 Standards Compliance 19

*3.6 Software System Attributes 19*

3.6.1 Reliability 19-20

3.6.2 Availability 20

3.6.3 Security 20-21

3.6.4 Maintainability 21

3.6.5 Portability 21-22

*3.7 Organizing the Specific Requirements 22*

3.7.1 System Mode 22

3.7.2 User Class 22-23

3.7.3 Objects 23

3.7.4 Feature 23

3.7.5 Stimulus 23-24

3.7.6 Response 24

3.7.7 Functional Hierarchy 24

*3.8 Additional Comments 24-25*

**4. Change Management Process 25-26**

**1. Introduction**

1.1 Purpose

SDM CampusFlow is a dedicated social media platform designed exclusively for students and faculty of SDM College of Engineering and Technology. Its purpose is to create a secure, interactive space where users can connect, share posts, form friendships, and engage in college events. Unlike broad social platforms, CampusFlow is focused on enhancing student life and academic interaction within the campus environment. Through features like a structured friendship system, event management, and direct messaging, CampusFlow fosters collaboration, helps students stay informed about campus happenings, and strengthens the sense of community, all while ensuring user privacy through verified college-only access.

1.2 Scope

SDM CampusFlow is an innovative social media platform tailored specifically for the SDM College of Engineering and Technology community. This platform is designed to facilitate connections among students and faculty, promote academic collaboration, and foster a sense of community within the campus. The scope of CampusFlow includes a range of interactive features: users can post updates, send friend requests, like and comment on posts, and participate in campus events. The platform also enables event organizers to promote academic and social events, ensuring higher engagement. With robust user authentication through University Seat Number (USN) verification, CampusFlow maintains a secure, campus-only environment that prioritizes privacy and community-focused interaction, enhancing the overall college experience.

1.3 Definitions, Acronyms, and Abbreviations

Definitions:

|  |  |
| --- | --- |
| Term | Description |
| CRUD | Operations like Create read update delete |
| DBMS | Database management system |
| UI/UX | Metadata about the interface of the website |
| API | Application programming interface for more functionalities |
| HTTP | Hyper-text transfer protocol to send website over intenet |
| MVC | Model-View-Controller |
| SRS | Software Requirements Specification |
| DFD | Data Flow Diagram |

References:

* ***Django Documentation****. (n.d.). Retrieved from* [*https://docs.djangoproject.com/en/stable/*](https://docs.djangoproject.com/en/stable/)
* **Jinja Documentation**. (n.d.). Retrieved from https://jinja.palletsprojects.com/
* **HTML Living Standard**. (n.d.). Retrieved from https://html.spec.whatwg.org/
* **CSS: Cascading Style Sheets**. (n.d.). Retrieved from <https://www.w3.org/Style/CSS/>
* **JavaScript (MDN Web Docs)**. (n.d.). Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
* **PostgreSQL Documentation**. (n.d.). Retrieved from <https://www.postgresql.org/docs/>
* **Python Official Documentation**. (n.d.). Retrieved from <https://docs.python.org/3/>

1.5 Overview

SDM CampusFlow is a social media platform created specifically for the students and faculty of SDM College of Engineering and Technology, aiming to foster community engagement and streamline campus interactions. Built using Django as the backend framework, CampusFlow allows users to connect, share updates, and participate in events in a secure, college-only environment. Features include user authentication via University Seat Number (USN), friend connections (Rapport System), post creation, direct messaging, and a notification system for real-time updates on friend requests, comments, and likes. Additionally, the platform includes an advertisement module where event organizers can promote college events like workshops and hackathons. With its focus on privacy and a user-friendly interface, CampusFlow enhances both social and academic connections within the SDM campus community.

**2. Overall Description**

2.1 Product Perspective

**SDM CampusFlow** is a standalone, web-based social media platform specifically designed to foster connection and communication among the students and faculty of SDM College of Engineering and Technology. Unlike mainstream social media platforms like Facebook or Instagram, which cater to the general public and include various online integrations, CampusFlow is developed exclusively for the college community, emphasizing user privacy, focused interaction, and secure, college-only access. This platform is intended for academic and campus-based networking, allowing users to share posts, form friend connections, join communities, and advertise campus events such as workshops and hackathons.

**Comparison with Existing Products**

1. **Mainstream Social Media Platforms**:

Unlike widely-used platforms with extensive features and public access, SDM CampusFlow operates within the secure college environment, focusing on specific campus needs like verified user registration, event promotion, and exclusive student-faculty interactions.

1. **College Notice Boards and Event Flyers**:

Similar to traditional notice boards, CampusFlow allows for event promotion and information sharing; however, it digitizes and enhances accessibility, enabling students and faculty to stay updated on campus activities without needing physical access to the college.

**Product Scope and Interfaces**

SDM CampusFlow is a self-contained application without external integrations such as payment gateways or third-party APIs. Its design centers on core social functionalities, campus event management, and user-friendly administrative controls, making it ideal for streamlined, secure use within the SDM College community.

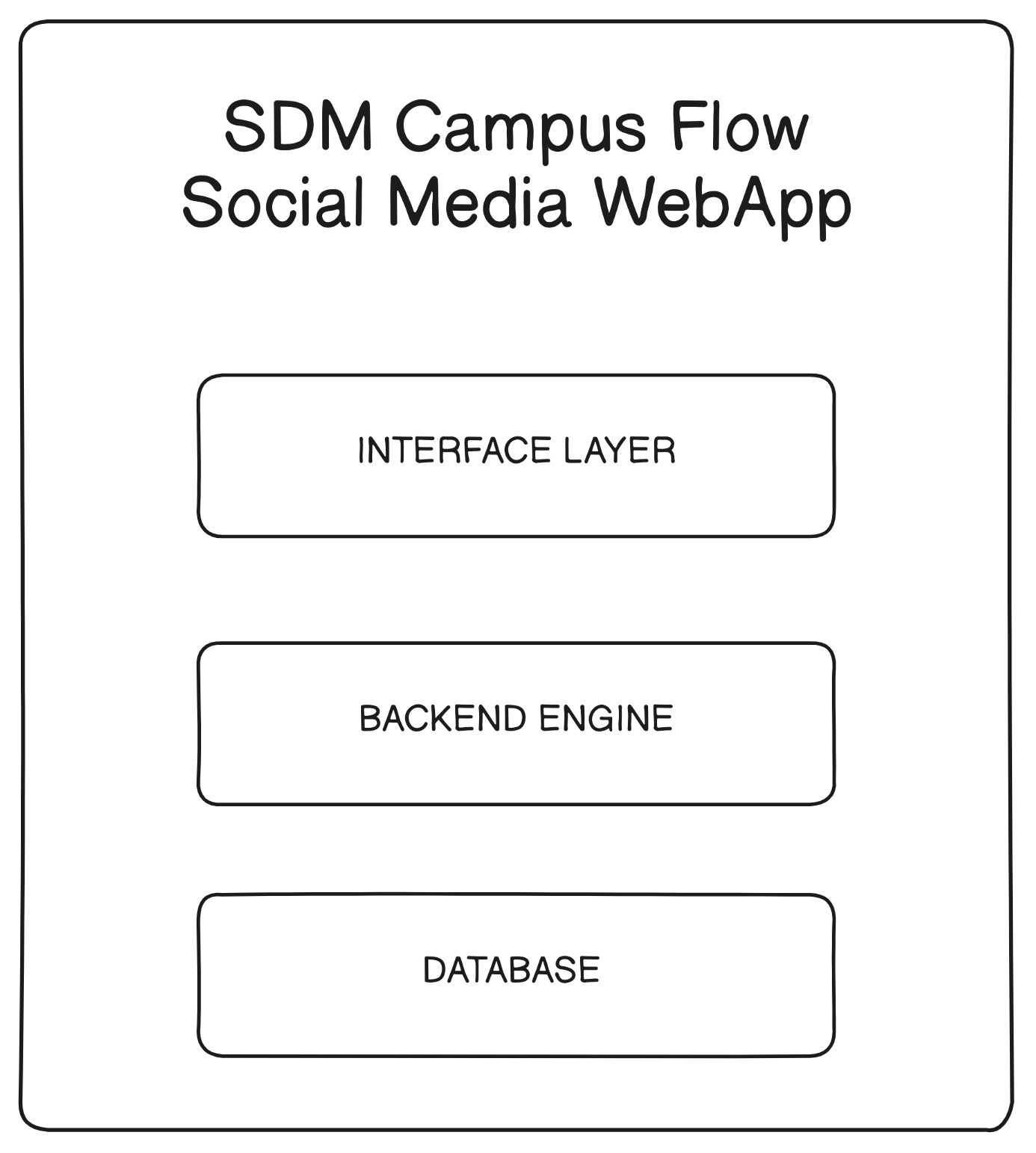


Fig 1.Block Diagram

This diagram would outline the main components of the SDM CampusFlow platform, emphasizing its independent, secure design and lack of external dependencies, ensuring a focused, community-oriented experience.

2.1.1 System Interfaces

The SDM CampusFlow platform includes intuitive interfaces to support seamless interaction for users within the college community. The **User Interface (UI)** allows students and faculty to register, log in, view posts, manage friendships, and interact through likes, comments, and messaging. The **Admin Interface** provides tools for verifying users, managing profiles, and overseeing content. Additionally, the system interacts with the **Database Interface** to securely store and retrieve user data, posts, and notifications, ensuring efficient data handling. These interfaces work together to create a cohesive, user-focused platform.

2.1.2 Interfaces

The **SDM CampusFlow** application will utilize a web-based interface for both students and faculty, designed for easy navigation and interaction.

#### Logical Characteristics of the Interface

1. **G**raphical User Interface (GUI)**:** Users interact through a visually appealing web interface, with clickable buttons and menus for registration, post creation, following users, and managing profiles.
2. **Input/Output**: Users can input data (e.g., text for posts, profile information), and the system provides immediate feedback through notifications, such as confirmations of actions (e.g., "Post successfully created") or alerts for errors (e.g., "Invalid USN format").
3. **Error Handling**: Informative error messages guide users on correcting inputs, with options to retry or navigate back to the previous page easily.

#### Optimizing the User Interface

1. **User-Friendly**: The interface prioritizes simplicity and intuitiveness, enabling users with varying levels of technical expertise to navigate effortlessly. Clear icons and straightforward layouts reduce cognitive load.
2. **Accessibility**: The design includes features for accessibility, such as high-contrast color schemes, screen reader compatibility, and keyboard navigation support, ensuring usability for all users, including those with disabilities.

2.1.3 Hardware Interfaces

The **SDM CampusFlow** application primarily relies on standard computing devices such as laptops, desktops, and tablets, which facilitate access to the web interface. Users can utilize various input devices, including keyboards and mice, for navigation and data entry. The application is optimized for modern web browsers, ensuring compatibility across different operating systems. Additionally, mobile devices like smartphones will provide users with access to CampusFlow through a responsive design, enhancing usability on smaller screens. Network connectivity is essential, as the application relies on internet access for real-time interactions, notifications, and data synchronization among users.

2.1.4 Software Interfaces

The **SDM CampusFlow** application operates on a web-based architecture, utilizing Django as the backend framework for handling requests and data processing. It communicates with a PostgreSQL database for secure data storage and retrieval, ensuring efficient management of user profiles, posts, and interactions. The frontend is developed using Jinja templating, enabling dynamic content rendering and user interaction. The application interfaces with external libraries for features like image uploading and location services. Additionally, RESTful APIs are implemented for seamless data exchange between the frontend and backend, supporting functionalities such as user authentication, following users, and managing posts in real-time.

2.1.5 Communications Interfaces

The **SDM CampusFlow** application employs a client-server communication model to facilitate interactions between users and the backend system. User requests, such as registration, posting updates, and following others, are sent via HTTP/HTTPS protocols to the Django backend, which processes the requests and returns appropriate responses. Real-time notifications, such as friend requests and post interactions, are implemented using WebSocket technology, enhancing user engagement with instant updates. Additionally, email notifications are integrated for account verifications and important announcements, ensuring users remain informed. This multi-faceted communication approach promotes a smooth and interactive user experience across the platform.

2.1.6 Memory Constraints

The **SDM CampusFlow** application is designed to operate efficiently within standard memory constraints typical for web applications. The server-side memory usage is optimized to handle concurrent user requests without excessive resource consumption, ensuring responsive performance. Each user session is managed with a limited memory footprint, allowing for scalability as the user base grows. Client-side memory constraints are considered as well; the frontend is optimized to load quickly, minimizing the amount of data cached in the browser. Images and media are compressed to reduce memory usage while maintaining quality, enhancing the overall performance and user experience of the application.

2.1.7 Operations

#### SDM CampusFlow Operations Overview

1. **Modes of Operation:**
   * **User Mode:** Students and faculty can create accounts, post updates, follow other users, and view notifications.
   * **Admin Mode:** Administrators manage user registrations, oversee content moderation, and generate reports on user engagement.
2. **Interactive Operations:**
   * **Real-Time Interaction:** The application supports real-time interactions for posting updates and receiving notifications, requiring active user engagement.
   * **Admin Maintenance:** Administrators can perform system maintenance during low-traffic periods to minimize disruptions.
3. **Data Processing Support Functions:**
   * The system processes user inputs such as registration details, post submissions, and follow requests in real-time, ensuring immediate feedback without batch processing.
4. **Backup and Recovery Operations:**
   * **Backup:** Automated backups of user data and posts are performed daily, with additional manual backups encouraged for critical administrative data.
   * **Recovery:** In the event of system failure, data can be restored from the latest automated backup, facilitating quick recovery with minimal data loss.
5. **Operational Constraints:**
   * **Downtime for Maintenance:** Scheduled maintenance may occur during non-peak hours, with efforts made to minimize downtime and ensure continuity of service for users.

2.1.8 Site Adaptation Requirements

#### Installation and Initialization Requirements for SDM CampusFlow

1. **Data and Initialization Sequences:**
   * **User Profiles and Registration:** Prior to system activation, administrators must input the initial set of user profiles, including student and faculty details, which can be managed through the admin mode during the setup phase.
   * **Content Categories:** Categories for posts (e.g., announcements, events, discussions) must be defined and initialized to organize content effectively and enhance user experience.
   * **Verification Processes:** Procedures for validating user identities using USN must be established and configured to ensure legitimacy before granting access.
2. **Site or Mission-Related Features:**
   * **Campus-Specific Customization:** The software should be tailored to the college’s operational requirements, such as branding elements, specific user groups, and event management features relevant to the campus community.
   * **Log File Management:** Procedures for manual backups of critical logs, such as user activities and system errors, must be established, with guidelines for administrators on performing these backups regularly to prevent data loss.
3. **Required Modifications to Customer Work Area:**
   * **Computer Specifications:** The college must ensure that the computers used for accessing the application meet the minimum memory and processing requirements, with a standard PC featuring at least 4GB of RAM recommended for optimal performance.
   * **Web Access:** The system requires reliable internet connectivity, necessitating a suitable network infrastructure to support smooth operations and user interactions.
   * **No Significant Hardware Modifications:** No major hardware changes (e.g., backup power sources, additional cooling systems) are necessary for installation, but the existing environment should facilitate a stable and efficient operational setup.

2.2 Product Functions

The **SDM CampusFlow** application is a web-based platform designed to enhance communication and interaction among students and faculty. It allows users to create profiles, post updates, and connect with peers, fostering a vibrant campus community. Below are the major functions of the system, highlighting user experience:

1. **User Registration and Profile Management:**
   * Users can register for an account using their USN and personal details. They can manage their profiles by updating information, including profile pictures and bios.
2. **Post Creation and Interaction:**
   * Users can create posts to share updates, events, or questions. They can comment on and like posts, encouraging engagement within the community.
3. **Following and Friend Requests:**
   * Users can follow other students and faculty members to receive updates on their activities. They can also send and manage friend requests to connect with peers.
4. **Search Functionality:**
   * The system provides a search feature that allows users to find specific posts, users, or topics of interest, enhancing navigation and content discovery.
5. **Notification System:**
   * Users receive real-time notifications for friend requests, likes, comments, and other interactions, keeping them informed and engaged with community activities.
6. **Admin Management:**
   * Administrators can oversee user registrations, manage content, and moderate posts to ensure a safe and respectful environment. They can also generate reports on user activity and engagement.
7. **System Logout:**
   * Users can log out of the system at any time, with their session securely terminated to protect personal information. The state of their activity is preserved for future access.

2.3 User Characteristics

The **SDM CampusFlow** application targets a diverse user base, primarily consisting of students and faculty members, along with administrative users. Understanding the characteristics of these users is crucial for designing an effective and engaging platform.

1. **Educational Level:**
   * Users span a range of educational backgrounds, from first-year students to faculty with advanced degrees. This diversity necessitates an intuitive and accessible system design that avoids academic jargon, ensuring that all users can navigate the platform easily. The user interface (UI) will use clear, straightforward language to facilitate understanding.
2. **Experience with Technology:**
   * Most users are expected to possess basic to intermediate computer skills, as they are familiar with common web applications. This indicates that the system should prioritize a user-friendly web interface that minimizes the learning curve. The design will focus on straightforward workflows, clear instructions, and helpful prompts to guide users through various functionalities.
3. **Technical Expertise:**
   * Intended users do not require programming knowledge or advanced technical skills. This characteristic informs the design by emphasizing simplicity and ease of use. The system will incorporate effective error handling and user feedback mechanisms to assist users in correcting mistakes, ensuring confidence even among those with limited technical expertise.

### Design Implications

The characteristics of the user base will significantly impact the system's design:

* **User Interface Design:** The UI will be crafted to be intuitive and navigable, with clear labels and options that reflect users' familiarity with basic applications. Visual cues and prompts will enhance usability for those with limited technical experience.
* **Internal System Design:** The design will accommodate user interactions that may deviate from expected patterns, necessitating robust error handling and feedback mechanisms. This approach ensures that the system remains resilient and user-friendly, even in the event of user errors.
* **Training and Support:** While the system aims for self-sufficiency, the variety in user experience may lead to a need for supplementary materials, such as user guides or quick reference tips, to enhance user confidence and satisfaction.

2.4 Constraints

The **SDM CampusFlow** application operates under various constraints that influence its design and development:

1. **Hardware Limitations:** The system must be optimized to run on standard computers available to students and faculty, requiring minimal RAM and storage to ensure compatibility across various devices used on campus.
2. **Multi-User Operation:** The application is designed to support multiple concurrent users, necessitating careful management of data and resources to handle simultaneous interactions without performance degradation.
3. **Regulatory Policies:** While there are no strict regulatory policies specific to social media applications for educational institutions, adherence to general best practices regarding data privacy, particularly in handling personal information, is crucial.
4. **Internet Connectivity:** The application relies on a stable internet connection for functionality, limiting accessibility for users in areas with poor network coverage.
5. **Simplicity of Features:** The focus on creating an intuitive user experience constrains the complexity of features, leading to a more streamlined system architecture that prioritizes essential functionalities.
6. **Reliability Expectations:** The system must ensure robust user management and data integrity to prevent issues such as duplicate accounts and lost posts, placing a premium on stability and reliability during development.
7. **Basic Security Measures:** While advanced security features like end-to-end encryption may not be necessary, basic security protocols must be implemented to prevent unauthorized access and protect user data, particularly for administrative functionalities.

These constraints define the operational environment and functionality of the application, guiding the design and development process while limiting the scope of implementation options.

2.5 Assumptions and Dependencies

The development of the **SDM CampusFlow** application relies on several assumptions and external dependencies. These factors, while not directly constraining the design, can impact the requirements in the SRS if they change or are not met.

1. **Operating System Compatibility:** It is assumed that the target users will have access to commonly used operating systems (Windows, macOS, Linux). Should the application need to support additional or less common OS environments, the SRS would require updates to reflect these changes.
2. **Web Browser Access:** The system assumes that users will access the application through modern web browsers. If users rely on outdated or incompatible browsers, the SRS may need to incorporate support for those environments.
3. **Internet Connectivity:** It is assumed that users will have reliable internet access to use the application effectively. Changes in connectivity assumptions might necessitate the addition of offline capabilities or local data storage options in the SRS.
4. **User Load:** The initial design assumes a moderate number of concurrent users. If user demand increases significantly, the SRS may require updates to ensure the system can handle higher loads efficiently.
5. **User Technical Skills:** The design assumes that users possess basic to intermediate technical skills. If the user base's proficiency level changes significantly, the SRS might need to enhance user interface guidelines or support materials.
6. **Data Storage Solutions:** It is assumed that the application will use a relational database for data management. If future requirements dictate a shift to more complex data storage solutions (e.g., NoSQL databases), the SRS will need adjustments to accommodate these needs.
7. **Feature Expansion:** The initial feature set is based on current user requirements. Any future demands for additional functionalities (e.g., enhanced analytics, notifications, or external integrations) would necessitate updates to the SRS.
8. **Compliance with Institutional Policies:** The application is developed under the assumption that it will comply with all relevant institutional policies regarding data usage and privacy. Should policies change, the SRS may require modifications to ensure compliance.
9. **Third-Party Services:** The system may depend on third-party services for functionalities such as notifications or authentication. Changes in these services could impact the SRS and necessitate adjustments to integration specifications.
10. **User Feedback Loop:** It is assumed that user feedback will be gathered post-launch to refine and improve the application. Should this feedback mechanism change or be eliminated, the SRS may need revisions to incorporate alternate methods for user input and iteration.

2.6 Apportioning of Requirements

Due to time constraints, resource limitations, and customer prioritization, several features of the **SDM CampusFlow** application may be postponed for future iterations. While these features are beneficial, they are not critical for the initial release and can be planned for later versions.

1. **Advanced Search Functionality**
   * **Current Version:** Basic user search by name or USN.
   * **Future Version:** Enhanced search capabilities, including filtering by interests, courses, or recent activity.
2. **Messaging System**
   * **Current Version:** No direct messaging feature available.
   * **Future Version:** Implementation of a real-time messaging system for users to communicate.
3. **Event Management Features**
   * **Current Version:** Basic event posting functionality.
   * **Future Version:** Advanced features for event management, including RSVPs, reminders, and attendee tracking.
4. **Mobile Application Support**
   * **Current Version:** Web-based application only.
   * **Future Version:** Development of a mobile application for enhanced accessibility and user engagement.
5. **Analytics and Reporting**
   * **Current Version:** Limited analytics on user engagement.
   * **Future Version:** Comprehensive reporting tools for administrators to track usage patterns and engagement metrics.
6. **Dynamic User Profiles**
   * **Current Version:** Static user profiles with limited information.
   * **Future Version:** Development of dynamic profiles that users can customize, including interests, activities, and achievements.
7. **Notification System**
   * **Current Version:** No notification feature for user interactions.
   * **Future Version:** Addition of notifications for likes, comments, follows, and event reminders.
8. **Integration with Third-Party Services**
   * **Current Version:** No external integrations.
   * **Future Version:** Integration with third-party services for features like event scheduling, document sharing, or authentication.
9. **Accessibility Features**
   * **Current Version:** Basic user interface without specialized accessibility options.
   * **Future Version:** Implementation of enhanced accessibility features to support users with disabilities.
10. **User Feedback Mechanism**
    * **Current Version:** No formal feedback collection system in place.
    * **Future Version:** Development of a feedback mechanism for continuous improvement based on user suggestions.

These deferred features will be prioritized by the customer based on their relevance and importance, enabling an iterative development process that focuses on delivering core functionalities first while planning for advanced features in subsequent releases.

**3. Specific Requirements**

This section outlines the detailed, verifiable requirements for the **SDM CampusFlow** application, ensuring that designers and developers can build a system that meets the needs of college students and faculty. The requirements are divided into functional and non-functional categories, with each requirement being specific, unambiguous, and traceable.

### 3.1 External Interfaces

This section details the technical description of all inputs and outputs for the **SDM CampusFlow** application, necessary for developers to understand how the system will interact with external entities.

#### 3.1.1 User Registration Input

* **Name of Item**: User Registration Input
* **Description of Purpose**: Captures new user information for account creation.
* **Source of Input**: User input via a web-based registration form.
* **Valid Range, Accuracy, and/or Tolerance**:
  + Username: 3-20 alphanumeric characters.
  + Password: Minimum of 8 characters, must include at least one number and one special character.
  + Email: Must follow valid email format (e.g., example@domain.com).
* **Units of Measure**: N/A
* **Timing**: Immediate input during the registration process.
* **Relationships to Other Inputs/Outputs**: Successful registration creates a new user entry in the system database.
* **Data Formats**:
  + USN: String
  + Password: String (hashed for security)
  + Email: String
* **Command Formats**: register [USN] [password] [email]
* **End Messages**:
  + "Registration successful."
  + "Error: Invalid input. Please try again."

#### 3.1.2 User Login Input

* **Name of Item**: User Login Input
* **Description of Purpose**: Authenticates an existing user and grants access to the system.
* **Source of Input**: User input via the web login form.
* **Valid Range, Accuracy, and/or Tolerance**:
  + Username: 3-20 alphanumeric characters.
  + Password: Minimum of 8 characters.
* **Units of Measure**: N/A
* **Timing**: User submits credentials during the login process.
* **Relationships to Other Inputs/Outputs**: Successful login provides access to social features, including posting, commenting, and following other users.
* **Data Formats**:
  + Username: String
  + Password: String
* **Command Formats**: login [username] [password]
* **End Messages**:
  + "Login successful."
  + "Error: Invalid credentials."

#### 3.1.3 User Profile Update Input

* **Name of Item**: User Profile Update Input
* **Description of Purpose**: Allows users to update their profile information.
* **Source of Input**: User input via the profile settings page.
* **Valid Range, Accuracy, and/or Tolerance**:
  + Phone Number: Must follow the standard format (e.g., +91XXXXXXXXXX).
  + Bio: Maximum of 250 characters.
* **Units of Measure**: N/A
* **Timing**: Immediate input during profile updates.
* **Relationships to Other Inputs/Outputs**: Updated profile data is stored in the database and reflected in the user’s public profile.
* **Data Formats**:
  + Phone Number: String
  + Bio: String
* **Command Formats**: update\_profile [phone number] [bio]
* **End Messages**:
  + "Profile updated successfully."
  + "Error: Invalid input. Please try again."

#### 3.1.4 Post Creation Input

* **Name of Item**: Post Creation Input
* **Description of Purpose**: Allows users to create new posts within the application.
* **Source of Input**: User input via the post creation form.
* **Valid Range, Accuracy, and/or Tolerance**:
  + Content: Maximum of 500 characters.
* **Units of Measure**: N/A
* **Timing**: Immediate input during post creation.
* **Relationships to Other Inputs/Outputs**: Successful post creation updates the feed and stores the post in the database.
* **Data Formats**:
  + Content: String
* **Command Formats**: create\_post [content]
* **End Messages**:
  + "Post created successfully."
  + "Error: Post content exceeds the limit."

### 3.2 Functions

This section outlines the functional requirements of the **SDM CampusFlow** application, specifying the core functionality and operations the system must perform.

#### 3.2.1 User Registration

* The system shall allow users to register by providing a username, password, and email, validating inputs for uniqueness and format.
* The system shall hash and securely store user passwords and confirm successful registration with a message.

#### 3.2.2 User Login

* The system shall authenticate users by verifying their username and password, granting access upon successful login.
* The system shall deny access after three failed attempts, displaying appropriate error messages.

#### 3.2.3 Profile Management

* The system shall allow users to update their profile information, including phone number and bio, with validations on input format and length.
* The system shall display an updated profile immediately after successful changes.

#### 3.2.4 Post Management

* The system shall enable users to create, edit, and delete posts, with content validation to enforce character limits.
* The system shall display posts in a feed format, sorted by timestamp.

#### 3.2.5 Following System

* The system shall allow users to follow and unfollow other users, maintaining a many-to-many relationship in the database.
* The system shall display follower and following counts on user profiles.

### 3.3 Performance Requirements

This section outlines the performance requirements for the **SDM CampusFlow** application.

#### 3.3.1 Static Numerical Requirements

* The system shall support 100 concurrent users.
* The system shall accommodate up to 10,000 registered users.
* The system shall handle up to 1,000 posts per day.

#### 3.3.2 Dynamic Numerical Requirements

* 95% of user interactions shall be processed in less than 2 seconds.
* The system shall handle up to 200 transactions per minute during peak hours.

### 3.4 Logical Database Requirements

This section specifies the logical requirements for the database of the **SDM CampusFlow** application.

#### Types of Information

* **User Information**: User ID, Username, Hashed Password, Email, Phone Number, Bio
* **Posts**: Post ID, User ID (FK), Content, Timestamp, Like Count
* **Relationships**: Follower ID, Followed ID

#### Integrity Constraints

* User ID must be unique.
* Email addresses must be unique across users.
* Each post must be associated with a valid user ID.

### 3.5 Design Constraints

This section outlines the design constraints for the **SDM CampusFlow** application.

#### 3.5.1 Standards Compliance

* The application shall adhere to web accessibility standards to ensure usability for all students.
* All user interfaces shall follow a consistent design language for coherence.

### 3.6 Software System Attributes

The **SDM CampusFlow** application must meet quality standards for usability, reliability, performance, maintainability, portability, security, and scalability.

#### 3.6.1 Reliability

* The software shall achieve an MTBF of at least 300 hours.
* Robust error handling will ensure unexpected inputs are managed without crashes.

#### 3.6.2 Availability

* The system shall operate with a minimum availability of 98% each month.

#### 3.6.3 Security

* The system shall enforce strong password policies and secure password storage practices.

#### 3.6.4 Maintainability

* The software shall be modularized to facilitate easy updates and maintenance.

#### 3.6.5 Portability

* The application shall be developed using standard web technologies to ensure compatibility across major browsers.

This structured outline provides clear, specific requirements that will guide the development of the **SDM CampusFlow** application, ensuring it meets user needs and maintains high standards of performance and security.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Characteristic** | **H/M/L** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 1 | Correctness | H |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Efficiency | M |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Flexibility | M |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Integrity/Security | H |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Interoperability | M |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Maintainability | H |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Portability | H |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Reliability | H |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Re usability | M |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Testability | H |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Usability | M |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | Availability | H |  |  |  |  |  |  |  |  |  |  |  |  |

3.7 Organizing the Specific Requirements

#### 3.7.1 System Mode

The **SDM CampusFlow** system operates in distinct modes, each catering to the needs of different user groups. The primary modes include:

* **User Mode**: Designed for students and faculty who want to interact on the platform. In this mode, users can register, create profiles, follow others, post updates, and interact through comments and likes. The interface should prioritize intuitive navigation and accessibility, allowing users to engage with content seamlessly.
* **Admin Mode**: This mode is for administrators who manage the platform’s backend. Tasks include verifying user accounts, moderating content, managing event listings, and viewing analytics on user engagement. The admin interface may require advanced features for data management and user oversight.

#### 3.7.2 User Class

**SDM CampusFlow** supports various user classes, each with distinct roles and permissions. These user classes include:

* **Students**: Regular users who can create accounts, post updates, interact with other users, and access content specific to their academic needs. Requirements for this class should emphasize an easy-to-use interface and straightforward interaction mechanisms.
* **Faculty**: Similar to students but with additional capabilities to post official announcements and events. Their permissions should include content management and oversight to ensure academic integrity.
* **Administrators**: Have full access to all functionalities within the system. They can verify users, manage posts, handle user reports, and analyze system usage. Requirements for this class should focus on security, data integrity, and comprehensive management tools.

#### 3.7.3 Objects

Key objects within the **SDM CampusFlow** system include:

* **Users**: Represents students, faculty, and administrators. Attributes may include user ID, name, email, role, and profile information. Functions related to this object include account creation, logging in, and profile management.
* **Posts**: Represents user-generated content on the platform. Attributes may include post ID, user ID, content, timestamps, and engagement metrics (likes/comments). Functions associated with posts include creating new posts, editing, deleting, and viewing engagement statistics.
* **Events**: Represents college events like workshops and hackathons. Attributes may include event ID, title, description, date, and location. Functions related to events include creating new events, updating details, and displaying upcoming events.
* **Connections**: Represents the relationships between users, such as follows or friendships. Attributes may include connection ID, follower ID, followed ID, and timestamps. Functions related to connections include sending, accepting, and managing follow requests.

#### 3.7.4 Features

**SDM CampusFlow** includes key features that provide externally desired services. Features can be outlined using stimulus-response pairs. Important features include:

* **User Registration**: Users can create accounts using their University Seat Numbers (USNs). The system should confirm registration and send a verification email.
* **Posting Updates**: Users can create and share posts with their peers. The system should respond by displaying the new post and updating the user's feed in real-time.
* **Following Users**: Users can follow each other to see updates. The system should confirm follow requests and allow users to view their connections.
* **Event Management**: Users can view, create, and RSVP to college events. The system should notify users of upcoming events and allow easy event creation.

#### 3.7.5 Stimulus

The **SDM CampusFlow** system functions based on user stimuli, which initiate specific actions. Examples of stimuli include:

* **Registering an Account**: When a user submits their details for registration, the system checks for validity and creates an account.
* **Creating a Post**: When a user submits a new post, the system processes the input and displays the post on the user's feed.
* **Following Another User**: Upon sending a follow request, the system processes the request and updates the following status.

#### 3.7.6 Response

The system can also be organized by the responses generated from user interactions. For example:

* **Registration Confirmation**: After successful registration, the system sends a confirmation email and provides the user with a welcome message.
* **Post Submission Feedback**: Once a post is submitted, the system displays a success message along with the newly created post in the user's feed.
* **Error Notifications**: If a user tries to register with an existing USN, the system should provide immediate feedback indicating that the registration failed due to duplication.

#### 3.7.7 Functional Hierarchy

The overall functionality of the **SDM CampusFlow** system can be structured into a hierarchy based on common inputs, outputs, or internal data access. This hierarchy may include:

* **User Management**: Covers functionalities for user registration, login, profile management, and connection handling.
* **Content Management**: Includes creating, editing, and deleting posts and managing comments and likes.
* **Event Management**: Encompasses features for creating, updating, and RSVPing to events. Utilizing functional hierarchies clarifies the relationships between various functions and ensures comprehensive coverage of the system’s requirements.

### 3.8 Additional Comments

1. **Combining Approaches**:
   * **Mode and User Class**: Outline requirements based on user roles and operational modes, highlighting differences in user management and interaction for students, faculty, and admins.
   * **Feature and Functional Hierarchy**: Detail stimulus-response sequences for key features alongside a functional hierarchy to provide a clear understanding of user interactions and system responses.
2. **Notations and Tools**:
   * **Finite State Machines**: Useful for visualizing the state transitions based on user inputs in different operational modes.
   * **Object-Oriented Analysis**: UML diagrams can organize requirements around objects and interactions, enhancing clarity.
   * **Data Flow Diagrams (DFDs)**: Illustrate data flows between processes, inputs, and outputs, clarifying overall system operations.
   * **Stimulus-Response Sequences**: Help clarify how the system reacts to specific inputs, particularly in user-facing applications.
3. **Requirement Specification**:
   * Functional requirements can be structured in various ways, such as using native language descriptions, pseudocode, or formal specification methods.
   * A **Four Subsections Approach** can be adopted for requirements:
     + **Introduction**: Outlines the purpose and importance of the requirement.
     + **Inputs**: Specifies required inputs, formats, and valid ranges.
     + **Processing**: Details how inputs are processed to generate outputs, including algorithms.
     + **Outputs**: Lists outputs produced, including formats and expected results.

Using multiple organizational techniques enriches the Software Requirements Specification (SRS), catering to diverse stakeholder needs and ensuring clarity in essential functions. This approach enhances communication and collaboration, contributing to the success of the software development process.