# SECURE SOFTWARE ENGINEERING

# SRS DOCUMENT For UniVoice

Table of Contents :

4. External Interface Requirements

   a. User Interfaces
   b. Hardware Interfaces
   c. Software Interfaces

5. Data Model

   a. User Data
   b. News Data
   c. Admin Data
   d. Archived Radio

6. Other Non-Functional Requirements

   a. Performance Requirements
   b. Scalability
   c. Security Requirements
   d. Usability

## 1.<u>Introduction:</u>

### a. Purpose:

**UniVoice** is a **College News Management System** designed to facilitate the publication, management, and dissemination of news and radio broadcasts within a college environment.

### b. Scope:

The system enables users to post and manage news articles, integrate radio streaming for live and recorded broadcasts, and ensure real-time communication through role-based access, notifications, and search functionalities.

### c. Acronyms and Definitions:

- **User:** Any individual accessing the system (Student, Faculty, Admin).
- **Admin:** A privileged user who manages news content and user roles.
- **SRS:** Software Requirements Specification.
- Radio Streaming: The functionality that allows users to broadcast and listen to live or recorded audio content.

### d. References:

- IEEE Std 830-2019: IEEE Recommended Practice for Software Requirements Specifications.

## 2. General Description:

### a. Product Perspective:

The College News Management System will provide a web-based platform for publishing and managing news updates

within a college. It will replace traditional notice boards and ensure real-time communication.

**b. Product Functions:**

- News posting and categorization.
- User role management (Admin, Faculty, Students).
- Search and filtering options.
- Live and recorded radio streaming for announcements, interviews, and discussions.

**c. User Characteristics:**

- **Students:** Read and listen to radio broadcasts.
- **Faculty:** Post news, host discussions, and manage audio content.
- **Admins:** Manage news, users, system settings, and oversee radio streaming operations.

**d. Assumptions and Dependencies:**

- The system will run on college-provided servers.
- Users will access the system through web browsers.
- Database connectivity is essential for storing news articles.

- The radio streaming feature will require third-party integration with audio streaming platforms.

## 3. Specific Requirements

### a. System Features:

- Role-based access control (Admin, Faculty, Students).
- News submission, approval, and publishing workflow.
- Multimedia support (Images, Videos, PDFs).
- Search and filtering functionalities.

### b. News Management:

- Users can create, edit, and delete news articles.
- Admin approval required for news publication.
- Categorization of news into sections (Academics, Events, Sports, Notices).

### c. User Roles and Management:

- Admins can create and manage user accounts.
- Faculty can post and edit their own news articles.
- Students can read the news updates.

### d. Radio Streaming Integration:

- **Live streaming** support for college announcements and discussions.
- **Recorded audio** storage for past broadcasts.
- **User access control** to determine who can broadcast and listen.
- **Integration with third-party streaming services** for scalability.

## 4. External Interface Requirements:

### a. User Interfaces:

- Responsive web-based interface for all users.
- Admin panel for managing news and users.

### b. Hardware Interfaces:

- Compatible with standard desktops, tablets, and smartphones.
- Requires a web server and database server.

### c. Software Interfaces:

- Integration with college authentication system (LDAP/SSO).

- Supports database systems like MySQL or PostgreSQL.

## 5. Data Model:

### a. User Data:

- Username, password, role (Admin, Faculty, Student).

### b. News Data:

- Title, description, article content, images, videos, submission date, category, author, approval status.

### c. Admin Data:

- Pending news status, admin ID on approval/rejection, date of update.

### d. Archived Radio:

- Audio files, date of post, metadata (title, speaker, keywords).

## 6. Other Nonfunctional Requirements:

### a. Performance Requirements:

- The system should be able to support high number of concurrent users.

### b. Scalability:

- Must handle increasing content and user growth.

**c. Security Requirements:**

- Encrypted user data , secure login , Token based sessions.
- Data encryption for audio streaming to prevent unauthorized access.

**d.Usability:**

- Intuitive UI/UX for students and faculty.
- Dark mode option for accessibility.