

360Campus

Software Requirements Specification (SRS)

Table of Contents

Table of Contents

- Revision History2**
- 1. Introduction.....3**
 - 1.1 Purpose3
 - 1.2 Intended Audience and Reading Suggestions.....3
 - 1.3 Product Scope3
 - 1.4 Definitions, Acronyms, and Abbreviations3
- 2. Overall Description4**
 - 2.1 Product Perspective4
 - 2.2 Product Functions5
 - 2.3 Product Constraints.....6
 - 2.4 User Classes and Characteristics6
 - 2.4 Operating Environment7
 - 2.5 Assumptions and Dependencies7
 - 2.6 Apportioning of Requirements8
- 3. Requirements.....8**
 - 3.1 External Interfaces8
 - 3.1.1 User Interfaces 8
 - 3.1.2 Hardware Interfaces 10
 - 3.1.3 Software Interfaces 10
 - 3.2 Functional Requirements.....12
 - 3.3 Quality of Service.....14
 - 3.3.1 Performance..... 14
 - 3.3.2 Security 14
 - 3.3.3 Reliability..... 15
 - 3.3.4 Availability 15
 - 3.4 Compliance.....16
 - 3.4.1 Report Format 16
 - 3.4.2 Data Naming..... 16
 - 3.4.3 Audit Tracing 17

Revision History

Date	Version	Description	Author
2024-10-11	1.0	Initial draft of SRS	Eman Abbas

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive overview of the **360Campus** system. This integrated platform aims to enhance the university experience for students, instructors, and administrators by addressing various critical aspects of university life.

360Campus is designed to streamline educational resource management, facilitate class discussions, and organize events while also providing project ideas insights and performance analytics.

1.2 Intended Audience and Reading Suggestions

This document is intended for:

- **Developers:** To understand the functional and non-functional requirements for building the system.
- **Project Stakeholders:** To review and approve the defined requirements.
- **Testers:** To use as a basis for testing the system once developed.
- **Instructors and Students:** To provide insight into what the platform will offer.

For optimal understanding, readers should have familiarity with software development concepts and educational technology.

1.3 Product Scope

360Campus is an integrated online platform designed to facilitate various functions within the university, including educational resources, event management, campus mapping, and performance analysis. The platform will serve as a central hub for students, instructors, and administrators to access and manage university-related information and activities. The project will also explore advanced features such as a virtual classroom and AI-driven timetable generation.

1.4 Definitions, Acronyms, and Abbreviations

- **API (Application Programming Interface):** A set of rules and protocols for building and interacting with software applications, allowing different software components to communicate.
- **Django:** A high-level Python web framework that encourages rapid development and clean, pragmatic design.

- **ORM (Object-Relational Mapping):** A programming technique used to convert data between incompatible type systems in object-oriented programming languages, facilitating interaction with databases.
- **SQL (Structured Query Language):** A standardized programming language used for managing and manipulating relational databases.
- **TA (Teaching Assistant):** A person who assists a professor with instructional responsibilities, often helping with grading, classroom management, and student support.
- **UI (User Interface):** The space where interactions between humans and machines occur, encompassing all the controls, buttons, and visual elements.
- **Vue.js:** A progressive JavaScript framework used for building user interfaces and single-page applications.
- **CDN (Content Delivery Network):** A system of distributed servers that deliver web content to users based on their geographic location, enhancing speed and performance.

2. Overall Description

2.1 Product Perspective

The **360Campus** system is a new, self-contained product designed to enhance the university experience for students, instructors, and administrators. It is not a follow-on member of an existing product family, nor is it intended to replace any specific existing systems. Instead, 360Campus aims to fill a gap in the current educational ecosystem by providing an integrated platform that addresses both academic and non-academic needs within the university environment.

Context and Origin: The idea for 360Campus originated from the observation of fragmented systems currently in use at universities, which often require students and staff to navigate multiple platforms for different functions, such as course management, attendance tracking, and event organization. By consolidating these functionalities into a single platform, 360Campus seeks to improve user experience, enhance collaboration, and streamline administrative processes.

System Relationships: While 360Campus is designed as a standalone solution, it may interface with existing university systems, such as student information systems (SIS) and learning management systems (LMS), to import/export relevant data. For example, student enrollment data could be synchronized with the platform to ensure accurate tracking of attendance and performance.

Major Components and Interfaces:

The 360Campus system consists of several key components, each addressing specific functionalities:

- **User Interface Module:** Provides access for students, instructors, and administrators to navigate the platform and utilize its features.

- **Content Management System (CMS):** Manages course materials, event information, and resources.
- **Analytics Engine:** Processes data related to performance analysis, attendance tracking, and report generation.
- **Communication Module:** Facilitates class discussions and interactions among users.
- **AI Scheduling System:** Automates timetable generation for classes and events.

2.2 Product Functions

The **360Campus** system is designed to facilitate various functions that enhance the university experience for students, instructors, and administrators. Below is a summary of the major functions:

Major Functions:

- **User Management**
 - User registration and authentication
 - Role management for students, instructors, and administrators
- **Course Management**
 - Access and management of course materials
 - Facilitation of class discussions and communication
- **Event Management**
 - Creation and promotion of university events
 - RSVP and attendance tracking for events
- **Attendance Tracking**
 - Digital attendance forms for classes
 - Monitoring student participation and engagement
- **Performance Analysis**
 - Collection and analysis of class data
 - Performance metrics for students and instructors
- **Graduation Project Tracking**
 - Management of graduation project ideas and descriptions
 - Grades provided by instructors for student reference

- Aiding students in brainstorming and selecting graduation project topics
- **Report Generation**
 - Generation of reports on attendance, performance, and project progress
- **Virtual Classroom**
 - Support for online classes and remote learning
 - Integration of multimedia tools for enhanced learning experiences
- **Timetable Generation**
 - Automated scheduling of classes and events using AI algorithms
 - Optimization of resources and instructor availability

2.3 Product Constraints

The **360Campus** system is subject to several constraints that limit development options:

1. **User Interfaces:**
 - Must be intuitive and accessible for students, instructors, and administrators.
 - Compatibility with mobile devices is required for access on smartphones and tablets.
2. **Quality of Service:**
 - Response time must be under two seconds for user actions.
 - Uptime must be at least 99.5% to ensure reliability during peak periods.
3. **Standards Compliance:**
 - Must comply with data protection regulations (e.g., GDPR, FERPA).
 - Adherence to W3C web standards for compatibility across browsers.
4. **Design and Implementation:**
 - Technology stack may be limited by existing university infrastructure and expertise.
 - Development must stay within budget and time constraints.

2.4 User Classes and Characteristics

The **360Campus** system will serve the following user classes:

1. **Students**

- **Use Frequency:** Daily
- **Functions:** Access materials and courses, track attendance, use map.
- **Technical Expertise:** Moderate
- **Security Level:** Standard user access
- **Importance:** High

2. Instructors

- **Use Frequency:** Daily to weekly
- **Functions:** Upload materials, track attendance, analyze performance.
- **Technical Expertise:** Moderate to high
- **Security Level:** Elevated access
- **Importance:** High

3. Administrators

- **Use Frequency:** Weekly to monthly
- **Functions:** Manage user accounts, oversee courses.
- **Technical Expertise:** High
- **Security Level:** Elevated access
- **Importance:** Medium

2.4 Operating Environment

The platform will operate in a cloud environment, ensuring scalability and accessibility. It will support major web browsers (Chrome, Firefox, Safari) and mobile platforms (iOS, Android). Users will require an internet connection and compatible devices (PCs, tablets, smartphones) to access the system.

2.5 Assumptions and Dependencies

Assumptions:

1. **User Engagement:** It is assumed that students and instructors will actively use the system.
2. **Technology Proficiency:** Users will have a basic understanding of technology and web applications.

3. **Infrastructure Stability:** The university's IT infrastructure will support the application's requirements without significant upgrades.
4. **Data Security Compliance:** It is assumed that the system will comply with data protection regulations (e.g., GDPR).

Dependencies:

1. **Third-Party Tools:** The system may rely on third-party software for features like attendance tracking and analytics.
2. **Existing Systems:** Integration with current university systems (e.g., Student Information System) is essential for data consistency.
3. **Internet Connectivity:** The application's performance is dependent on reliable internet access for users.

2.6 Apportioning of Requirements

The requirements for the **360Campus** system will be apportioned across different software elements as follows:

Function	Software Element	Implementation Status
Materials	Course Management Module	Defined and ready for implementation
Classes with Discussions	Class Management Module	Defined and ready for implementation
Events	Events Management Module	Defined and ready for implementation
University Map	Mapping Module	Defined and ready for implementation
Graduation Projects	Project Tracking Module	Defined and ready for implementation
Performance Analysis	Analytics Module	Defined and ready for implementation
Attendance Tracking	Class Management Module	Defined and ready for implementation
Virtual Classroom	TBD	May be deferred to future versions
Timetable Generation	TBD	May be deferred to future versions

3. Requirements

3.1 External Interfaces

This subsection defines all the input and output requirements of the 360Campus software system, detailing the various interfaces involved.

3.1.1 User Interfaces

- Home Page
 - Source: University-wide information.

- Features: General announcements, university events, and quick links to key sections (Materials, Events, Help).
 - Standard Buttons: Home, Login/Logout, Help.
- User Dashboard
 - Source: User profile data from the database.
 - Features: Personalized view of registered classes, performance metrics, quick access to materials, events, and discussions.
 - Standard Buttons: Home, Logout, Help, View Profile.
- Materials Section
 - Source: University-wide materials, available to all.
 - Features: Access to academic resources (PDF, Word), search by course or department.
 - Standard Buttons: Search, Download, View Details.
- Classes Section
 - Source: Data from active class registrations.
 - Features: Class-specific materials, discussions, assignments, and attendance tracking.
 - Standard Buttons: View Materials, Submit Assignment, Discussion Board.
- Events Section
 - Source: University events data.
 - Features: Calendar view, RSVP, add to personal calendar.
 - Standard Buttons: Add to Calendar, RSVP, View Details.
- University Map
 - Source: Campus geographical data.
 - Features: Interactive map of the university campus showing key buildings, departments, classrooms, and facilities. Search by location or building name.
 - Standard Buttons: Zoom In/Out, Search, View Location.

3.1.2 Hardware Interfaces

- **Supported Device Types:**

The 360Campus system will be accessible from a wide range of devices, including:

- Desktops and Laptops: Running Windows, macOS, and Linux operating systems.
- Tablets and Smartphones: Running iOS and Android operating systems.

- **Nature of Data and Control Interactions:**

The system will interact with hardware components in the following ways:

- Input Devices: The system will receive input from standard input devices such as keyboards, mice, touchscreens, and styluses.
- Output Devices: The system will output data to monitors, mobile screens, and printers.
- Storage Devices: The system will store and retrieve data from local devices and predominantly through cloud storage services.

- **Communication Protocols:**

- HTTP/HTTPS: Used for secure communication between the front-end user interfaces and back-end servers.
- Wi-Fi/Ethernet: For reliable network connectivity to the system's web-based components.
- Bluetooth: In some cases, for local device communication, e.g., attendance tracking using Bluetooth beacons (if implemented).
- API Calls: Communication between application and external services will be facilitated through RESTful APIs.

- **Mobile Hardware Specifics:**

- GPS Access: The university map will leverage GPS features for precise location mapping on mobile devices.
- Camera Access: The app may use the device camera for features such as QR code scanning during event attendance.

- **Cloud Servers and Database Interaction:**

- The system will interact with cloud-based servers for data storage, processing, and retrieval. The backend databases will handle queries and responses between hardware and software.

3.1.3 Software Interfaces

- **Database Systems:**

- **Name:** MySQL
- **Purpose:** The database will store all essential data, including user profiles, materials, class data, events, graduation projects, and performance analytics.
- **Data Interaction:** The system will send SQL queries to the database for data retrieval (e.g., fetching class materials, user-specific data like registered courses) and data storage (e.g., student performance, attendance tracking).
- **Shared Data:** The database will share user profile data, attendance records, and project information across different components (e.g., course management and performance analytics).
- **Operating Systems:**
 - **Name:** Windows, macOS, Linux (for desktops); iOS, Android (for mobile devices)
 - **Purpose:** The system will be platform-independent and web-based, allowing access through any modern web browser on any operating system.
 - **Interaction:** No direct dependency on specific operating systems, but the front-end web application should be optimized for performance across all operating systems. Compatibility with browser versions (e.g., Chrome, Safari, Firefox) will be critical.
- **Libraries and Frameworks:**
 - **Frontend:**
 - **Vue.js:** For building dynamic user interfaces for the 360Campus platform.
 - **Vuetify:** A Material Design component framework that helps maintain consistent UI/UX.
 - **Purpose:** The frontend will send HTTP requests to the backend API for retrieving or posting data. Vue.js will handle local logic like data presentation and form submission.
 - **Backend:**
 - **Django REST Framework:** Used for creating APIs to manage communication between the frontend and backend.
 - **Purpose:** Manage routing, handle requests (e.g., fetching class materials, attendance reports), and provide the necessary backend logic to store, update, or retrieve data.
- **External Tools and APIs:**
 - **Google Maps API** (version v3.45 or higher):

- **Purpose:** The Google Maps API will be used to power the campus map feature. It will allow students to navigate the campus in real-time using mobile devices.
- **Data Interaction:** The API will send location data to the frontend, enabling real-time maps. The app will not store location data on the server, except for logged-in users who may save specific map preferences.
- **Microsoft Azure Cloud Services:**
 - **Purpose:** Used for cloud storage and data processing.
 - **Data Interaction:** Secure API calls will be made between the system and Azure for scalable cloud hosting of files (e.g., project proposals, class materials).
- **Integrated Commercial Components:**
 - **Zoom or Microsoft Teams Integration:**
 - **Purpose:** These services will be used for the virtual classroom feature (if implemented).
 - **Data Interaction:** APIs from Zoom or Microsoft Teams will facilitate the creation of virtual class sessions, sharing session links with students, and handling class attendance automatically.
 - **Forms (Google Forms or Microsoft Forms):**
 - **Purpose:** For attendance tracking using form submission during class times.
 - **Data Interaction:** Form results will be fetched via an API, which will then be processed by the backend to update attendance records for students.
- **Data Sharing:**
 - **Shared Across Components:** User data (profiles, roles), class materials, performance analytics, and event details will be accessible across the course management, performance analytics, and reporting modules.
 - **Implementation Constraints:** Data consistency must be ensured across various services and APIs. As multiple components interact with shared data (e.g., profile info or class registration), mechanisms like caching and validation will be crucial.

3.2 Functional Requirements

This section specifies the functional requirements of the software, detailing the expected interactions and behaviors within the system for various user roles.

- **User Authentication:**
 - All users (students, professors, TAs, and admins) must log in to access their profiles and specific functionalities based on their roles.
- **Dashboard Access:**
 - **Students:** Access a personalized dashboard displaying registered courses, materials, events, and performance analytics.
 - **Professors:** View their courses, manage class materials, and access performance analytics for their students.
 - **TAs:** Assist professors by accessing course materials, student submissions, and analytics.
 - **Admins:** Manage user accounts, oversee the overall system, and access all data and analytics.
- **Course Management:**
 - **Students:** View and manage their registered courses, including class schedules and materials.
 - **Professors:** Create, update, and manage course content and schedules.
 - **TAs:** Help manage course content and assist with grading and student queries.
- **Material Upload and Access:**
 - **Professors:** Upload and organize course materials for all faculties, which will be accessible to students and TAs.
 - **TAs:** Access uploaded materials to assist students and support professors.
 - **Students:** View and download materials relevant to their registered courses.
- **Event Management:**
 - **Students:** View upcoming events, seminars, and workshops, and register for participation.
 - **Professors and TAs:** Create and manage events related to their courses.
 - **Admins:** Oversee all events and manage registrations.
- **Graduation Project Tracking:**
 - **Students:** View graduation project ideas, descriptions, and grades uploaded by faculty members.
 - **Professors:** Add and update project details and assess student submissions.

- **TAs:** Assist professors in managing and tracking student project submissions.
- **Performance Analytics:**
 - **Students:** Access analytics regarding their academic performance, including attendance and grades.
 - **Professors:** View analytics for their classes to monitor student performance and identify areas for improvement.
 - **TAs:** Help professors analyze student performance data and assist in providing feedback to students.
- **Administrative Functions:**
 - **Admins:** Manage user roles, monitor system usage, ensure data security, and generate system-wide reports.

3.3 Quality of Service

This section outlines the quality-related property requirements that the software must meet to ensure effective performance, security, reliability, and availability.

3.3.1 Performance

- Response Time:
 - The system should respond to user requests within 2 seconds for most operations, such as logging in, fetching course materials, or submitting assignments.
- Concurrent Users:
 - The system must support at least 100 concurrent users without a degradation in performance. This is critical for peak usage times, such as during course registration or exam periods.
- Data Retrieval:
 - SQL queries for retrieving data (e.g., class materials, user profiles) should execute within 1 second.
- Load Handling:
 - The system should handle increased loads gracefully, maintaining performance within acceptable limits during periods of high activity (e.g., midterms or finals).

3.3.2 Security

- User Authentication:

- Implement secure login mechanisms (e.g., OAuth, JWT) to ensure only authorized users can access the system.
- Data Protection:
 - Sensitive user data (e.g., passwords, personal information) must be encrypted using industry-standard algorithms (e.g., AES-256).
- Access Control:
 - Role-based access controls must be enforced, ensuring users can only access features and data relevant to their roles (students, professors, TAs, and admins).
- Compliance:
 - Adhere to relevant security regulations (e.g., GDPR, FERPA) to protect user privacy and data rights.
- Audit Logs:
 - Maintain logs of user activities for security audits and tracking unauthorized access attempts.

3.3.3 Reliability

- Error Handling:
 - The system must gracefully handle errors and exceptions, providing meaningful feedback to users without crashing.
- Data Integrity:
 - Implement checks to ensure data integrity during transactions (e.g., ACID properties in database transactions).
- Testing:
 - Conduct thorough testing (unit, integration, system testing) to ensure all components function correctly and reliably before deployment.
- Backup Mechanisms:
 - Regular automated backups of user data and system configurations must be conducted to prevent data loss.

3.3.4 Availability

- Uptime Requirement:

- The system should guarantee 99.9% uptime, ensuring users can access the system whenever needed.
- **Redundancy:**
 - Implement redundancy measures, such as load balancers and failover systems, to ensure continuous service availability in case of hardware or software failures.
- **Scheduled Maintenance:**
 - Schedule maintenance should be communicated to users in advance, with minimal disruption to service (preferably during off-peak hours).
- **Recovery Procedures:**
 - Establish recovery procedures to restore service quickly after a failure, including defined recovery time objectives (RTO) and recovery point objectives (RPO).

3.4 Compliance

This section outlines the compliance requirements for the system to adhere to relevant standards, regulations, and best practices in data handling, audit tracing, and reporting.

3.4.1 Report Format

- **Standardized Reporting:**
 - All reports generated by the system (e.g., performance analytics, attendance, grades) must follow a standardized format to ensure consistency.
 - Reports should include essential information such as the date, author, and version control, and must be exportable in commonly used formats (e.g., PDF, CSV).
 - Reports related to academic performance and user activity must follow university guidelines for academic records.

3.4.2 Data Naming

- **Uniform Data Naming Conventions:**
 - All data fields must adhere to standardized naming conventions to maintain clarity, consistency, and interoperability across different system modules.
 - For example, student ID fields must be named consistently as `student_id` across the entire system, and event-related data must use `event_date`, `event_title`, and similar logical names.

- Data naming must align with industry standards for database management and comply with university policies for data handling.
- financial regulations, such as local tax laws or financial auditing practices.

3.4.3 Audit Tracing

- **Activity Logging:**

- All critical user actions (e.g., login attempts, course material updates, changes to student performance data) must be logged to an audit trail, capturing information such as the action performed, the user performing it, timestamp, and the before-and-after state of the data.
- Logs should be protected from unauthorized access and stored securely for auditing purposes.
- The audit trail must comply with any local or international regulations governing user data (e.g., GDPR, FERPA) and provide mechanisms to trace activity over time.

- **Change Tracking:**

- All modifications to sensitive data (e.g., grades, course materials, and attendance records) must be recorded in an audit trace with details of the changes made, including before-and-after values where applicable.
- These records must be available for audit for a specified retention period (e.g., 5 years) to comply with institutional and legal standards.