Software Requirements Specification

for

FAP web application

Version 1.0 approved

Prepared by Group 4

Summer 2024

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 2

1.1 Purpose 2

1.2 Document Conventions 2

1.3 Project Scope 2

1.4 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 User Classes and Characteristics 2

2.3 Operating Environment 2

2.4 Design and Implementation Constraints 2

2.5 Assumptions and Dependencies 2

3. System Features 2

3.1 System Feature 1 2

3.2 System Feature 2 (and so on) 2

4. Data Requirements 2

4.1 Logical Data Model 2

4.2 Data Dictionary 2

4.3 Reports 2

4.4 Data Acquisition, Integrity, Retention, and Disposal 2

5. External Interface Requirements 2

5.1 User Interfaces 2

5.2 Software Interfaces 2

5.3 Hardware Interfaces 2

5.4 Communications Interfaces 2

6. Quality Attributes 2

6.1 Usability 2

6.2 Performance 2

6.3 Security 2

6.4 Safety 2

6.5 [Others as relevant] 2

7. Internationalization and Localization Requirements 2

8. Other Requirements 2

Appendix A: Glossary 2

Appendix B: Analysis Models 2

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This document is intended to serve as a comprehensive guide for various stakeholders involved in the development, implementation, and usage of the Web FAP.

## Document Conventions

Following these document conventions will help ensure that all academic documents are presented in a professional and standardized manner, facilitating readability and comprehension.

## Project Scope

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Release 1** | **Release 2** | **Release 3** |
| FE-01, Mark Report | View individual grades by term | Advanced filtering and sorting | Detailed subject insights |
| FE-02, Academic Transcript | View average grades and subject status | View status of subjects: "Passed", "Studying", "Not Started", "Not Passed" |  |
| FE-03, Notification System | View newest notification of the university |  |  |
| FE-04, View weekly timetable | To view next week's or previous week's class schedule, select the weekly timetable and select the time you want to view. | “attended”: present, “absent”: absent, “Not yet”: not yet studying or the lecturer has not taken attendance. |  |
| FE-05, View class and lecturer | Choose the subject want to view | Choose “Student group”: student class, you can click on the class to see class members. “Instructor”: name of the instructor who will teach that class |  |
| FE-06, View exam schedule | Choose “View exam schedule” | The exam schedule table have: SUBJECTCODE, SUBJECTNAME, DATE, ROOM NO, TIME, EXAM FORM, EXAM, DATE OF PUBLICATION |  |

## References

COS Vision and Scope

# Overall Description

## Product Perspective

The FPT University Academic Portal is an entirely new product designed to address the needs of educational institutions in managing and displaying students' academic records. This system aims to streamline the process of tracking and reporting students' academic performance, ensuring accurate and up-to-date information is readily accessible to students, faculty, and administrative staff.

## User Classes and Characteristics

The Academic Transcript Management System (ATMS) is designed to be utilized by a diverse range of user classes, each with specific characteristics and requirements. The primary user classes include Students, Parents, Lecturers.

**Students**

Characteristics:

-Require access to their academic transcripts and schedule for reviewing grades, tracking progress, and planning future coursework.

-Varying levels of technical proficiency, necessitating an intuitive and user-friendly interface.

**Parents**

Characteristics:

-Include guardians and family members responsible for supporting and monitoring the student's academic progress.

-Often less technically proficient than students and faculty.

Interested in ensuring their children meet academic requirements and progress appropriately.

**Lecturers**

Characteristics:

-Comprise teachers, professors, and academic advisors.

-Responsible for entering and updating student grades.

-Available to view the class timetable.

-Need tools to monitor student progress and provide academic guidance.

-Generally proficient in using educational software systems.

**Favored User Classes**

While the FAP is designed to serve all user classes effectively, the primary focus is on Students and Lecturers. These classes directly interact with the system's core functionalities on a daily basis. Ensuring a seamless and efficient user experience for these groups is critical for the system's overall success and adoption.

**Summary of User Classes**

Students: Need intuitive access to their academic records, schedule, attendance, mark report and updates.

Parents: Require straightforward access to monitor their children's academic progress.

Lecturers: Require efficient tools for grade entry and student progress tracking.

## Operating Environment

FAP will operate within a diverse and robust technical infrastructure designed to ensure reliable access, security, and performance for all users. Below is a detailed description of the environment in which the software will operate.

**Hardware Platform**

The FAP is designed to be flexible and scalable, capable of running on various hardware configurations.

**Operating Systems and Versions**

The FAP will support multiple operating systems to accommodate the diverse needs of its users.

**Geographical Locations**

The FAP is designed to serve users across various geographical locations, including but not limited to Users: Students, parents, and lecturers located globally, with a concentration in regions where the educational institution operates.

**Hosting Organizations**

The FAP will be hosted by reputable cloud service providers to ensure reliability, scalability, and security.

In summary, the FAP will operate in a versatile and secure environment, supporting multiple hardware platforms and operating systems to deliver a seamless and efficient management experience.

## Design and Implementation Constraints

The development of FAP will be subject to various constraints that could limit the options available to the developers. These constraints include corporate policies, regulatory requirements, hardware limitations, interface requirements, and specific technology and tool choices

# System Features

The University Academic Portal is designed to provide students, faculty, and administrative staff with a comprehensive, user-friendly platform to manage and access academic resources and information.

## User Login

**3.1.1 Description and Priority**

* **Description**: This feature enables existing members to log in. It is a high-priority feature as it is fundamental for user access and experience.
* **Priority**: High

**3.1.2 Stimulus/Response Sequences**

**Sequence 1 (User Login):**

* **Stimulus**: A guest selects the "Login" option.
* **Response**: The system provides a login form for existing members to enter their credentials.

**Sequence 2 (Forgot Password):**

* **Stimulus**: A user clicks the "Forgot Password" link.
* **Response**: The system guides the user through a password recovery process.

**3.1.3 Functional Requirements**

**REQ-2: User Login:**

* The system shall authenticate existing members based on their provided credentials. Access to member-specific features is granted upon successful login.

**REQ-3: Forgot Password:**

* The system shall assist users who have forgotten their passwords in recovering their accounts. A password reset link shall be sent to the user's registered email.

## Check the Latest Announcement

### 3.2.1 Description and Priority

· **Description**: This feature enables users to check the latest announcements related to academic updates, events, deadlines, and other important information. It is a medium-priority feature as it enhances user engagement and ensures they are informed about relevant updates, but it is not as critical as the login functionality.

· **Priority**: Medium

### 3.2.2 Stimulus/Response Sequences

#### Sequence 1 (View Announcements):

* **Stimulus**: A logged-in user selects the "Announcements" option from the main menu.
* **Response**: The system displays a list of the latest announcements sorted by date, with the most recent at the top.

#### Sequence 2 (Announcement Details):

* **Stimulus**: A user clicks on a specific announcement.
* **Response**: The system displays the full details of the selected announcement, including any attached files or links.

### 3.2.3 Functional Requirements

#### REQ-4: View Announcements

* **The system shall display a list of the latest announcements to logged-in users. The announcements shall be sorted by date, with the most recent announcement appearing first.**

#### REQ-5: Announcement Details

* **The system shall allow users to view the full details of a selected announcement. This shall include the announcement's title, date, content, and any attached files or links.**

## View the Weekly Timetable

### Description and Priority

* **Description**: This feature enables users to view their weekly timetable, including class schedules, room assignments, and instructor details. It is a high-priority feature as it is essential for students and lecturers to manage their time and activities efficiently.
* **Priority**: High

### Stimulus/Response Sequences

#### Sequence 1 (View Weekly Timetable):

* **Stimulus**: A logged-in user selects the "Weekly Timetable" option from the main menu.
* **Response**: The system displays the user's timetable for the current week, showing all scheduled classes, room locations, and instructor names.

#### Sequence 2 (Change Week):

* **Stimulus**: A user navigates to the next or previous week using navigation buttons.
* **Response**: The system updates the display to show the timetable for the selected week.

### Functional Requirements

#### REQ-6: View Weekly Timetable

* **The system shall display the user's timetable for the current week. The timetable shall include details such as class names, scheduled times, room locations, and instructor names.**

#### REQ-7: Navigate Weekly Timetable

* **The system shall allow users to navigate to the previous or next week to view the timetable for different weeks. The display shall update to show the timetable for the selected week.**

## View exam schedule

### Description and Priority

* **Description**: This feature enables users to check their exam schedules, including exam dates, times, locations, and subject details. It is a high-priority feature as it is crucial for students to prepare for their exams and manage their study plans effectively.
* **Priority**: High

### Stimulus/Response Sequences

#### Sequence 1 (View Exam Schedule):

* **Stimulus**: A logged-in user selects the "Exam Schedule" option from the main menu.
* **Response**: The system displays a list of upcoming exams for the user, including dates, times, locations, and subjects.

#### Sequence 2 (Exam Details):

* **Stimulus**: A user clicks on a specific exam entry.
* **Response**: The system displays the full details of the selected exam, including any additional information or instructions.

### Functional Requirements

#### REQ-8: View Exam Schedule

* **The system shall display a list of upcoming exams for the logged-in user. The list shall include the exam dates, times, locations, and subjects.**

#### REQ-9: Exam Details

* **The system shall allow users to view the full details of a selected exam. This shall include the exam's date, time, location, subject, and any additional instructions or information.**

## View Exam Marks

### Description and Priority

* **Description**: This feature enables users to view their exam marks for each subject, including detailed component marks for each term and the overall average marks. It also provides the status of each subject (e.g., "Passed," "Studying," "Not started," "Not passed"). This is a high-priority feature as it is essential for students to track their academic performance and progress.
* **Priority**: High

### Stimulus/Response Sequences

#### Sequence 1 (View Exam Marks Report):

* **Stimulus**: A logged-in user selects the "Exam Marks Report" option from the main menu.
* **Response**: The system displays a report of the user's exam marks for each subject, including detailed component marks for each term and the overall average marks.

#### Sequence 2 (View Academic Transcript):

* **Stimulus**: A user selects the "Academic Transcript" option from the main menu.
* **Response**: The system displays the user's academic transcript, showing the overall average marks for each subject and the status of each subject (e.g., "Passed," "Studying," "Not started," "Not passed").

### Functional Requirements

#### REQ-10: View Exam Marks Report

* **The system shall display a report of the user's exam marks for each subject. This report shall include detailed component marks for each term and the overall average marks.**

#### REQ-11: View Academic Transcript

* **The system shall display the user's academic transcript, showing the overall average marks for each subject and the status of each subject. The status options shall include "Passed," "Studying," "Not started," and "Not passed."**

# Data Requirements

## Logical Data Model

The logical data model for the FAP web application includes the following key entities and their relationships:

* **Student**
  + Attributes: StudentID, Name, Email, Password, Program, YearOfStudy, DateOfBirth, ContactInfo
* **Course**
  + Attributes: CourseID, Name, Description, Credits, Schedule
* **Enrollment**
  + Attributes: EnrollmentID, StudentID, CourseID, EnrollmentDate, Status
* **Attendance**
  + Attributes: AttendanceID, StudentID, CourseID, Date, Status
* **FinancialInfo**
  + Attributes: FinancialID, StudentID, AmountDue, DueDate, PaymentStatus
* **Notification**
  + Attributes: NotificationID, Title, Content, DatePosted, IsRead

*Relationships:*

* A Student can enroll in multiple Courses.
* A Course can have multiple Students.
* Each Enrollment links a Student to a Course.
* A Student has multiple Attendance records.
* A Student has one FinancialInfo record.
* A Student can receive multiple Notifications.

## Data Dictionary

The data dictionary defines the data elements used in the FAP web application:

**Student**

* **StudentID**: Unique identifier for students.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **Name**: Student's full name.
  + Data Type: String
  + Length: 100
  + Format: Text
  + Allowed Values: Any valid text
* **Email**: Student's email address.
  + Data Type: String
  + Length: 255
  + Format: Email format
  + Allowed Values: Valid email addresses
* **Password**: Student's account password.
  + Data Type: String
  + Length: 255
  + Format: Encrypted
  + Allowed Values: Encrypted text
* **Program**: Study program of the student.
  + Data Type: String
  + Length: 100
  + Format: Text
  + Allowed Values: Predefined programs
* **YearOfStudy**: Current year of study.
  + Data Type: Integer
  + Length: 1
  + Format: Numeric
  + Allowed Values: 1-4

**Course**

* **CourseID**: Unique identifier for courses.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **Name**: Name of the course.
  + Data Type: String
  + Length: 255
  + Format: Text
  + Allowed Values: Any valid text
* **Description**: Description of the course.
  + Data Type: String
  + Length: 500
  + Format: Text
  + Allowed Values: Any valid text
* **Credits**: Number of credits for the course.
  + Data Type: Integer
  + Length: 2
  + Format: Numeric
  + Allowed Values: Positive integers
* **Schedule**: Schedule of the course.
  + Data Type: String
  + Length: 255
  + Format: Text
  + Allowed Values: Valid text

**Enrollment**

* **EnrollmentID**: Unique identifier for enrollments.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **StudentID**: Identifier for the student enrolled.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **CourseID**: Identifier for the course enrolled.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **EnrollmentDate**: Date of enrollment.
  + Data Type: Date
  + Length: N/A
  + Format: YYYY-MM-DD
  + Allowed Values: Valid dates
* **Status**: Status of the enrollment (e.g., active, completed).
  + Data Type: String
  + Length: 50
  + Format: Text
  + Allowed Values: Predefined statuses

**Attendance**

* **AttendanceID**: Unique identifier for attendance records.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **StudentID**: Identifier for the student.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **CourseID**: Identifier for the course.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **Date**: Date of attendance.
  + Data Type: Date
  + Length: N/A
  + Format: YYYY-MM-DD
  + Allowed Values: Valid dates
* **Status**: Attendance status (e.g., present, absent).
  + Data Type: String
  + Length: 50
  + Format: Text
  + Allowed Values: Predefined statuses

**FinancialInfo**

* **FinancialID**: Unique identifier for financial records.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **StudentID**: Identifier for the student.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **AmountDue**: Amount due for financial transactions.
  + Data Type: Decimal
  + Length: 10
  + Format: Numeric (2 decimals)
  + Allowed Values: Non-negative numbers
* **DueDate**: Due date for payment.
  + Data Type: Date
  + Length: N/A
  + Format: YYYY-MM-DD
  + Allowed Values: Valid dates
* **PaymentStatus**: Payment status (e.g., paid, pending).
  + Data Type: String
  + Length: 50
  + Format: Text
  + Allowed Values: Predefined statuses

**Notification**

* **NotificationID**: Unique identifier for notifications.
  + Data Type: Integer
  + Length: 10
  + Format: Numeric
  + Allowed Values: Positive integers
* **Title**: Title of the notification.
  + Data Type: String
  + Length: 255
  + Format: Text
  + Allowed Values: Any valid text
* **Content**: Content of the notification.
  + Data Type: String
  + Length: 1000
  + Format: Text
  + Allowed Values: Any valid text
* **DatePosted**: Date the notification was posted.
  + Data Type: Date
  + Length: N/A
  + Format: YYYY-MM-DD
  + Allowed Values: Valid dates
* **IsRead**: Read status of the notification.
  + Data Type: Boolean
  + Length: N/A
  + Format: True/False
  + Allowed Values: True, False

## Reports

The FAP web application will generate several reports to aid students and administration:

* **Academic Progress Report**
  + **Content**: List of courses, grades, GPA
  + **Sort Sequence**: By course, then by grade
  + **Totaling Levels**: Per semester, cumulative
* **Attendance Report**
  + **Content**: Attendance records per course, percentage attendance
  + **Sort Sequence**: By date, then by course
  + **Format**: CSV, PDF
* **Financial Statement**
  + **Content**: Amounts due, payments made, balance
  + **Sort Sequence**: By date, then by type of transaction
  + **Format**: PDF, Excel

## Data Acquisition, Integrity, Retention, and Disposal

The FAP web application will adhere to the following guidelines for data management:

* **Data Acquisition**: Data will be acquired through student inputs, instructor inputs, and automated data collection from integrated systems (e.g., class schedules, financial systems).
* **Data Integrity**: The system will implement data validation checks, unique constraints on primary keys, and regular audits. Backup systems will ensure data accuracy and recovery.
* **Data Retention**: Data will be retained for the duration of the student's enrollment plus five years. Archived data will be securely stored and accessible for compliance and reporting purposes.
* **Data Disposal**: Data will be securely deleted after the retention period. Techniques such as encryption, periodic backups, and mirroring will be employed to ensure data integrity and security.

# External Interface Requirements

This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.

## User Interfaces

Describe the logical characteristics of each interface between the software product and the users.

* **Login Screen**
  + Fields: Email, Password
  + Buttons: Login, Forgot Password
  + Constraints: Email format validation, password minimum length of 8 characters
  + Error Messages: Displayed under fields, e.g., "Invalid email address" or "Password is too short"
* **Dashboard**
  + Components: Navigation bar, student information, quick access tiles (e.g., Schedule, Grades, Financial Info)
  + Standard Buttons: Home, Profile, Logout, Help
  + Keyboard Shortcuts: Ctrl+H (Home), Ctrl+P (Profile)
  + Error Messages: Standard format in a pop-up at the top of the screen
* **Course Enrollment**
  + Fields: Course selection dropdown, Enrollment date
  + Buttons: Enroll, Cancel
  + Constraints: Validation of course availability, check for schedule conflicts
  + Error Messages: Displayed under fields or as pop-ups, e.g., "Course is full" or "Schedule conflict detected"
* **Attendance Tracking**
  + Display: Calendar view, list of courses with attendance status
  + Buttons: Mark Present, Mark Absent
  + Error Messages: Standard format below the buttons, e.g., "Attendance already recorded"
* **Financial Information**
  + Display: Table of charges, payments, and balances
  + Buttons: Pay Now, View History
  + Error Messages: Standard format in a pop-up at the top of the screen
* **Notifications**
  + Display: List of notifications with read/unread status
  + Buttons: Mark as Read, Delete
  + Error Messages: Standard format below the list, e.g., "Failed to delete notification"

*Note: Detailed user interface design will be documented in a separate user interface specification.*

## Software Interfaces

Describe the connections between this product and other software components.

* **Database**
  + System: MySQL 8.0
  + Purpose: Store all application data including user information, course details, attendance records, financial data, and notifications
  + Data Exchange: SQL queries for CRUD operations
  + Security: Data encryption, access control based on user roles
* **Authentication Service**
  + System: OAuth 2.0
  + Purpose: User authentication and authorization
  + Data Exchange: JSON web tokens (JWT)
  + Security: Encrypted tokens, secure storage
* **Email Service**
  + System: SMTP (Simple Mail Transfer Protocol)
  + Purpose: Sending notification emails to users
  + Data Exchange: MIME format for email messages
  + Security: SSL/TLS encryption for email transmission
* **Payment Gateway**
  + System: Stripe API v3
  + Purpose: Handling student payments
  + Data Exchange: JSON formatted requests and responses
  + Security: HTTPS for secure communication, PCI-DSS compliance

## Hardware Interfaces

Describe the characteristics of each interface between the software and hardware components of the system.

* **Web Server**
  + Type: Physical or virtual servers running Apache or Nginx
  + Interaction: Hosts the web application, handles HTTP/HTTPS requests
  + Protocols: HTTP/HTTPS
  + Data Exchange: HTML, CSS, JavaScript files for front-end, RESTful API calls for back-end
  + Security: SSL/TLS encryption for secure communication
* **User Devices**
  + Types: Desktops, laptops, tablets, smartphones
  + Interaction: Access the web application via web browsers
  + Supported Browsers: Chrome, Firefox, Safari, Edge
  + Data Exchange: User inputs through forms, navigation actions
  + Formats: Responsive design to adapt to different screen sizes

## Communications Interfaces

State the requirements for any communication functions the product will use.

* **Email Communication**
  + Purpose: Send notifications, password reset links
  + Format: MIME format for email messages
  + Constraints: Emails must be sent via the institution's SMTP server, attachments not supported
  + Security: SSL/TLS encryption
* **Web Browser Communication**
  + Purpose: User interaction with the web application
  + Protocols: HTTP/HTTPS
  + Constraints: Secure communication via HTTPS, no mixed content warnings
  + Security: Use of modern encryption standards, protection against XSS and CSRF attacks
* **API Communication**
  + Purpose: Integration with external services (e.g., payment gateway, authentication service)
  + Protocols: HTTPS, RESTful APIs
  + Data Formats: JSON
  + Security: Use of API keys, secure token exchange, HTTPS for all communications

# Quality Attributes

## Usability

* **Ease of Use**: The application will be intuitive and easy to navigate, with a clean and uncluttered interface. Common actions (e.g., checking schedules, viewing grades) should be accessible within two clicks from the homepage.
* **Ease of Learning**: New users should be able to perform basic functions within 5 minutes of first use. The application will provide a guided tour for first-time users to familiarize them with key features.
* **Memorability**: Frequent users should retain proficiency without needing to relearn the interface. Consistent design patterns will be used throughout the application.
* **Error Avoidance and Handling**: The application will prevent user errors through form validation and contextual help. Clear and concise error messages will guide users to correct mistakes.
* **Efficiency of Interactions**: Tasks such as enrolling in a course or checking financial information should be completed within 30 seconds under normal conditions.
* **Accessibility**: The application will comply with WCAG 2.1 Level AA standards to ensure it is accessible to users with disabilities. This includes support for screen readers, keyboard navigation, and sufficient color contrast.
* **Ergonomics**: The application will be designed to minimize user fatigue, with a responsive layout that works on both desktop and mobile devices, reducing the need for excessive scrolling or clicking.

## Performance

* **Response Time**: The application should respond to user actions (e.g., loading a page, submitting a form) within 2 seconds under normal load conditions.
* **Scalability**: The application should handle up to 10,000 concurrent users without performance degradation.
* **Throughput**: The system should process up to 1,000 transactions per second during peak times.
* **Availability**: The application should have an uptime of 99.9%, with downtime not exceeding 1 hour per month.
* **Resource Utilization**: The application should efficiently utilize server resources, with CPU usage not exceeding 75% and memory usage not exceeding 70% under normal operating conditions

## Security

* **Authentication**: All users must authenticate using OAuth 2.0. Multi-factor authentication (MFA) will be required for access to sensitive information.
* **Authorization**: Role-based access control (RBAC) will be implemented to ensure users can only access data and perform actions relevant to their roles.
* **Data Encryption**: All sensitive data (e.g., passwords, financial information) will be encrypted both in transit (using TLS) and at rest.
* **Data Privacy**: The application will comply with data privacy regulations such as GDPR. User data will not be shared without explicit consent.
* **Audit Logging**: All access and modification to sensitive data will be logged. Logs will be stored securely and retained for a minimum of 1 year.
* **Vulnerability Management**: Regular security assessments and vulnerability scans will be conducted. Critical vulnerabilities will be addressed within 24 hours of discovery

## Safety

* **Data Integrity**: The system will include safeguards to prevent data corruption, such as transactional integrity and data validation.
* **Error Recovery**: In the event of a system failure, the application will have mechanisms for quick recovery, such as database backups and automatic failover.
* **User Safety**: The application will provide clear warnings and confirmation dialogs for actions that could result in data loss or other negative outcomes (e.g., deleting records).

## [Others as relevant]

* **Reliability**: The application should function correctly under expected usage conditions. The mean time between failures (MTBF) should be at least 1,000 hours.
* **Maintainability**: The codebase should follow industry best practices for readability and modularity, making it easy to update and extend. All code changes should be reviewed and tested before deployment.
* **Interoperability**: The application should integrate smoothly with existing university systems, such as the student information system (SIS) and learning management system (LMS).
* **Portability**: The application should be deployable on any modern web server running a supported version of Linux or Windows.
* **Scalability**: The application architecture should support horizontal scaling to accommodate increased load. This includes the ability to add additional servers to the cluster without significant reconfiguration.
* **Verifiability**: All requirements and functionality should be verifiable through automated tests and manual inspections. Test coverage should be at least 80%.

# Internationalization and Localization Requirements

The FAP web application must be suitable for use in multiple nations, cultures, and geographic locations. The following requirements address various aspects of internationalization and localization:

### 7.1. Language Support

* **Multilingual Interface**: The application must support multiple languages, including Vietnamese and English. All text within the application, including error messages, labels, and help documentation, should be translatable.
* **Language Selection**: Users must be able to select their preferred language from a dropdown menu or settings page.
* **Translation Management**: Use industry-standard localization frameworks to manage translations. All translatable text should be stored in resource files.

### 7.2. Date and Time Formats

* **Date Format**: Support various date formats, including DD/MM/YYYY, MM/DD/YYYY, and YYYY-MM-DD. Users should be able to select their preferred format in their settings.
* **Time Format**: Support both 12-hour and 24-hour time formats. Users should be able to select their preferred format in their settings.
* **Time Zones**: The application must be aware of and adjust for different time zones. Users should be able to set their time zone in their profile settings, and all time-related data should be displayed accordingly.

### 7.3. Address and Telephone Number Formats

* **Address Formats**: Support different address formats, including variations in order and components (e.g., street name before number, postal codes).
* **Telephone Number Formats**: Support international telephone number formats, including country codes and varying lengths.

### 7.4. Legal and Regulatory Compliance

* **Data Privacy**: Ensure compliance with international data privacy laws, including GDPR, CCPA, and others relevant to the regions where the application will be used.
* **Export Controls**: Ensure the application adheres to international export control regulations.
* **Accessibility**: Ensure the application meets international accessibility standards, such as WCAG 2.1.

### 7.5. Symbols and Characters

* **Character Sets**: Support international character sets, including UTF-8, to ensure proper display of text in various languages.
* **Symbols**: Ensure that symbols used in the application are culturally appropriate and understandable in different regions

# Other Requirements

#### ****Legal, Regulatory, and Financial Compliance****

* **Data Protection Regulations:** The system must comply with GDPR, FERPA, and other relevant data protection and privacy laws.
* **Accessibility Standards:** Adherence to WCAG 2.1 guidelines to ensure accessibility for all users, including those with disabilities.
* **Financial Transactions:** Compliance with PCI-DSS standards for secure handling of online payments and financial data.

#### ****Installation, Configuration, Startup, and Shutdown Requirements****

* **Installation:** The system should support easy installation with clear instructions and minimal technical requirements.
* **Configuration:** Provide user-friendly configuration options to customize the system according to university policies and requirements.
* **Startup:** Ensure that the system can start up quickly and reliably, with minimal downtime.
* **Shutdown:** Implement safe and secure shutdown procedures to prevent data loss and maintain system integrity.

#### ****Logging, Monitoring, and Audit Trail Requirements****

* **Logging:** Comprehensive logging of user activities, system events, and error messages for troubleshooting and analysis.
* **Monitoring:** Real-time system monitoring to track performance, detect issues, and ensure availability.
* **Audit Trails:** Maintain detailed audit trails for critical actions such as data modifications, user access, and administrative changes.

Appendix A: Glossary

 **API (Application Programming Interface):** A set of rules and tools for building software applications, allowing different systems to communicate with each other.

 **FERPA (Family Educational Rights and Privacy Act):** A federal law that protects the privacy of student education records.

 **GDPR (General Data Protection Regulation):** A regulation in EU law on data protection and privacy for all individuals within the European Union.

 **GPA (Grade Point Average):** A standard way of measuring academic achievement in the U.S., typically on a scale from 0 to 4.0.

 **PCI-DSS (Payment Card Industry Data Security Standard):** A set of security standards designed to ensure that all companies that accept, process, store or transmit credit card information maintain a secure environment.

 WCAG (Web Content Accessibility Guidelines): A set of guidelines for making web content more accessible, particularly for people with disabilities.

Appendix B: Analysis Models

#### ****Data Flow Diagrams****

* **High-Level Data Flow Diagram:** Illustrates the overall flow of information within the university academic portal, highlighting key processes such as user authentication, course management, and academic records access.
* **Detailed Data Flow Diagrams:** Breaks down major processes into finer details, showing data inputs, processing steps, and outputs.

#### ****Feature Trees****

* **Course Management Feature Tree:** Visual representation of all features related to course management, including registration, catalog access, and schedule management.
* **Communication Tools Feature Tree:** Outline of communication features such as messaging, announcements, and email notifications.

#### ****State-Transition Diagrams****

* **User Authentication State-Transition Diagram:** Depicts the various states and transitions involved in user authentication, from login attempts to successful login and session management.
* **Course Registration State-Transition Diagram:** Shows the different states in the course registration process, including selection, validation, and confirmation.

#### ****Entity-Relationship Diagrams****

* **User Management ER Diagram:** Illustrates the relationships between different entities involved in user management, such as users, roles, and profiles.
* **Academic Records ER Diagram:** Shows the connections between entities related to academic records, including students, courses, grades, and transcripts.