Software Requirements Specification

for

FAP web application

Version 1.0 approved

Prepared by Thai Binh Duong

Group 4

19/06/2024

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

<Describe any standards or typographical conventions used, including the meaning of specific text styles, highlighting, or notations. If you are manually labeling unique requirement identifiers, you might specify the format here for anyone who needs to add one later.>

## 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

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, stimulus, response, or combinations of these, whatever makes the most logical sense for your product.>

## 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

<This section describes various aspects of the data that the system will consume as inputs, process in some fashion, or create as outputs.>

## Logical Data Model

<A data model is a visual representation of the data objects and collections the system will process and the relationships between them. Include a data model for the business operations being addressed by the system, or a logical representation for the data that the system itself will manipulate. Data models are most commonly created as an entity-relationship diagram.>

## Data Dictionary

<The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, you're better off storing the data dictionary as a separate artifact, rather than embedding it in the middle of an SRS. That also increases its reusability potential in other projects.>

## Reports

<If your application will generate any reports, identify them here and describe their characteristics. If a report must conform to a specific predefined layout you can specify that here as a constraint, perhaps with an example. Otherwise, focus on the logical descriptions of the report content, sort sequence, totaling levels, and so forth, deferring the detailed report layout to the design stage.>

## Data Acquisition, Integrity, Retention, and Disposal

<If relevant, describe how data is acquired and maintained. State any requirements regarding the need to protect the integrity of the system's data. Identify any specific techniques that are necessary, such as backups, checkpointing, mirroring, or data accuracy verification. State policies the system must enforce for either retaining or disposing of data, including temporary data, metadata, residual data (such as deleted records), cached data, local copies, archives, and interim backups.>

# 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. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Software Interfaces

<Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components. State the purpose, formats, and contents of the messages, data, and control values exchanged between the software components. Specify the mappings of input and output data between the systems and any translations that need to be made for the data to get from one system to the other. Describe the services needed by or from external software components and the nature of the intercomponent communications. Identify data that will be exchanged between or shared across software components. Specify nonfunctional requirements affecting the interface, such as service levels for responses times and frequencies, or security controls and restrictions.>

## Hardware Interfaces

<Describe the characteristics of each interface between the software and hardware (if any) components of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used. List the inputs and outputs, their formats, their valid values or ranges, and any timing issues developers need to be aware of. If this information is extensive, consider creating a separate interface specification document.>

## Communications Interfaces

<State the requirements for any communication functions the product will use, including e-mail, Web browser, network protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, handshaking, and synchronization mechanisms. State any constraints around these interfaces, such as whether e-mail attachments are acceptable or not.>

# Quality Attributes

## Usability

<Specify any requirements regarding characteristics that will make the software appear to be “user-friendly.” Usability encompasses ease of use, ease of learning; memorability; error avoidance, handling, and recovery; efficiency of interactions; accessibility; and ergonomics. Sometimes these can conflict with each other, as with ease of use over ease of learning. Indicate any user interface design standards or guidelines to which the application must conform.>

## Performance

<State specific performance requirements for various system operations. If different functional requirements or features have different performance requirements, it's appropriate to specify those performance goals right with the corresponding functional requirements, rather than collecting them in this section.>

## Security

<Specify any requirements regarding security or privacy issues that restrict access to or use of the product. These could refer to physical, data, or software security. Security requirements often originate in business rules, so identify any security or privacy policies or regulations to which the product must conform. If these are documented in a business rules repository, just refer to them.>

## Safety

<Specify requirements that are concerned with possible loss, damage, or harm that could result from use of the product. Define any safeguards or actions that must be taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the product must conform.>

## [Others as relevant]

<Create a separate section in the SRS for each additional product quality attribute to describe characteristics that will be important to either customers or developers. Possibilities include availability, efficiency, installability, integrity, interoperability, modifiability, portability, reliability, reusability, robustness, scalability, and verifiability. Write these to be specific, quantitative, and verifiable. Clarify the relative priorities for various attributes, such as security over performance.>

# Internationalization and Localization Requirements

<Internationalization and localization requirements ensure that the product will be suitable for use in nations, cultures, and geographic locations other than those in which it was created. Such requirements might address differences in: currency; formatting of dates, numbers, addresses, and telephone numbers; language, including national spelling conventions within the same language (such as American versus British English), symbols used, and character sets; given name and family name order; time zones; international regulations and laws; cultural and political issues; paper sizes used; weights and measures; electrical voltages and plug shapes; and many others.>

# Other Requirements

<Examples are: legal, regulatory or financial compliance, and standards requirements; requirements for product installation, configuration, startup, and shutdown; and logging, monitoring and audit trail requirements. Instead of just combining these all under "Other," add any new sections to the template that are pertinent to your project. Omit this section if all your requirements are accommodated in other sections. >

Appendix A: Glossary

<Define any specialized terms that a reader needs to know to understand the SRS, including acronyms and abbreviations. Spell out each acronym and provide its definition. Consider building a reusable enterprise-level glossary that spans multiple projects and incorporating by reference any terms that pertain to this project.>

Appendix B: Analysis Models

<This optional section includes or points to pertinent analysis models such as data flow diagrams, feature trees, state-transition diagrams, or entity-relationship diagrams. You might prefer to insert certain models into the relevant sections of the specification instead of collecting them at the end.>