
Software Requirements Specification

for

Eduplan

Version 1.0

Prepared by

**Andrade Julio
Astudillo Gerald
Bonilla David
Caizapanta Tammy**

SoftCrafters

May 21, 2025

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction	3
1.1 Purpose	3
1.2 Document Conventions	3
1.3 Intended Audience and Reading Suggestions	3
1.4 Product Scope	3
1.5 References	3
2. Overall Description	4
2.1 Product Perspective	4
2.2 Product Functions	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	5
2.6 User Documentation	5
2.7 Assumptions and Dependencies	5
3. External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces	6
3.3 Software Interfaces	6
3.4 Communications Interfaces	6
4. System Features	7
5. Other Nonfunctional Requirements	7
5.1 Performance Requirements	7
5.2 Safety Requirements	7
5.3 Security Requirements	8
5.4 Software Quality Attributes	8
5.5 Business Rules	8

Revision History

Name	Date	Reason For Changes	Version
Tammy Caizapanta	02-05-2025	Initial version	0.1
David Bonilla	18-05-2025	Final version	1.0

1. Introduction

This document is a Software Requirements Specification for a weekly academic scheduling system. This specification is based on the established IEEE 830, 1998 standards.

1.1 Purpose

This document specifies the software requirements for the Academic Schedule Management System, version 1.0. The goal is to define the system's functions, which aim to support educational institutions in organizing, controlling, and monitoring their academic activities. This SRS covers the entire main system.

1.2 Document Conventions

The requirements are written in clear and simple language. Keywords like *must*, *may*, or *should* indicate the level of priority. No special typographic styles are used. General requirements inherit the priority level from specific ones.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, teachers, and administrative staff. It begins with an overview of the system and continues with functional and non-functional requirements. Readers should start with the introduction and then move to the sections most relevant to their roles.

1.4 Product Scope

The software will allow academic institutions to register, update, and consult their schedules of activities. It will improve planning, avoid time conflicts, and enhance internal organization. It supports institutional goals related to academic control and efficiency.

1.5 References

IEEE 830-1998 – Software Requirements Specification

User Interface Style Guide (<https://developer.mozilla.org>)

Institutional Vision and Scope Document, Academic Directorate, version 2.1, March 2025.

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained application developed to support academic institutions in managing their activity schedules. It does not replace any existing system but provides a modern and practical solution to improve planning and organization. The system is independent but can be adapted to integrate with institutional platforms if needed.

2.2 Product Functions

The main functions of the system include:

- **Authentication:** Log in to the system with credentials.
- **Teacher Management (for Principals):** Search for and manage teacher information.
- **Plan Components (for Principals):** View full academic planning.
- **Weekly Planning (for Teachers):** Register, update, and edit specific days of their weekly plans.
- **Educational Content Management (for Teachers):** Add or update skills, strategies, materials, evaluation criteria, scope, integrating elements, and general academic information.

These functions are executed by different user types using controllers such as `PrincipalController` and `TeacherController`.

2.3 User Classes and Characteristics

- **Principal:** Has the highest privileges. They can view all academic plans, search for teacher information, and monitor the overall academic structure. Expected to have intermediate to advanced technical knowledge.
- **Teacher:** Responsible for entering and managing their own educational content, weekly plans, and curriculum elements. Typically has basic to intermediate technical knowledge and uses the system frequently.
- **User (Superclass):** A generic role used internally in the system's model to define shared features between teachers and principals.

These users interact through a menu-driven GUI managed by the `EduPlanSystem` class.

2.4 Operating Environment

The software is designed to run on desktop environments using Java. It can be executed on Windows, macOS, or Linux systems with Java Runtime Environment (JRE) installed. It uses plain text files for data storage and validation is handled through custom utilities like `FileManager`, `DateValidator`, `IdValidator`, and `UserValidator`.

2.5 Design and Implementation Constraints

- The system must be implemented in Java using the MVC (Model-View-Controller) design pattern.
- Data persistence is handled using flat text files.
- The program must use specific utility classes for data validation and file operations.
- Security must be ensured through input validation and role-based access control.
- It must support operations without a need for constant internet connectivity.

2.6 User Documentation

The following documentation will be provided:

- IEEE Software Requirements Specification of Eduplan translate in Spanish.

Assumptions and Dependencies

- It is assumed that users have basic computer knowledge to operate the system.
- The system depends on Java being installed on the host machine.
- The performance of the system depends on proper formatting of input files and validation through the utility classes.
- It is assumed that institutions will provide teachers with necessary training to operate the system efficiently.

3. External Interface Requirements

3.1 User Interfaces

The **EduPlanSystem** is currently in its initial development phase and operates as a console-based application, developed using **NetBeans Apache**. The user interface consists of hierarchical text menus, where users (Principal and Teacher) interact with the system through numerical options.

Although it does not currently include a graphical user interface (GUI), the system includes robust error-handling mechanisms to validate user inputs and prevent uncontrolled interruptions. Error messages will be clear and consistent, helping users to use the system correctly.

Future versions of the system are expected to include a graphical interface using **Java Swing** or **JavaFX**, following user-centered design standards.

3.2 Hardware Interfaces

This software does not require specific interfaces with external hardware devices. The system is designed to run on personal computers with basic specifications. The minimum recommended requirements are:

- Processor: Intel i3 or higher
- RAM: 4 GB or more
- Operating System: Compatible with Java (Windows, Linux, or macOS)
- Storage: Sufficient disk space for configuration files and **.csv** data storage

3.3 Software Interfaces

EduPlanSystem is developed using **Java version 24.0.1**. The main dependencies and libraries used include:

- **java.util.Date** and other standard Java classes for date handling
- **Apache POI**, used for managing **.csv** data files (although commonly used for Excel, it is anticipated to help with structured tabular data)

The system primarily interacts with **.csv** files to store user data, planning components, and reminders. It currently does not integrate with databases, web services, or other external systems, although such features are planned for future development phases.

3.4 Communications Interfaces

At this stage, **EduPlanSystem** does not implement any network communication interfaces such as email, browsers, or server connections.

However, future releases are expected to include communication functionalities such as:

- Remote access through a web interface
- Credential validation over a network
- Sharing of educational planning between users

4. System Features

Attached is the link to the file that has all the requirements in a better order and easy to understand for the user.

https://docs.google.com/spreadsheets/d/1f3lZYFZLjXm_mRn8-p99l4qQgwAsKYpz/edit?usp=sharing&ouid=105568818687247290971&rtpof=true&sd=true

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system must respond to user actions within a maximum of **0.5 seconds** under normal operation.
- Since this is a console-based beta version, it is expected to support **one user at a time**.
- Loading and saving of CSV files must complete within **1 to 5 seconds**, depending on the file size and data complexity.
- These performance constraints are intended to ensure a smooth and fluid experience during academic planning and data management tasks.

5.1 Safety Requirements

- No data loss is expected if the program closes unexpectedly, as the system retains data up to the last manually saved state.
- The system will not implement automatic backup features in this initial version.
- Users are encouraged to save data frequently to ensure consistency and avoid losing recent changes.

5.2 Security Requirements

- User authentication will be performed using a combination of **username and ID validation**.
- As the system runs in a console environment, **password masking is not implemented**.
- Access control will be role-based:
 - **Professors** can only access and manage data assigned to their own planning.
 - The **Director** has full access to modify, view, and manage all system data and user-related configurations.
- Unauthorized access to restricted functionalities will be denied by validating user roles upon login.

5.3 Software Quality Attributes

The system prioritizes the following quality attributes:

1. **Usability** – Menus and options will be presented clearly and sequentially to minimize user confusion.
2. **Reliability** – Functions will be tested to ensure accurate handling of valid and invalid inputs.
3. **Maintainability** – Code will be modular and documented to support future improvements and upgrades.

Other attributes such as portability, testability, and robustness are considered important but will be refined in future versions.

5.4 Business Rules

System access and capabilities are strictly role-based:

- The **Director** can only be registered **once** and has **global administrative privileges**, including the ability to edit planning, assign professors, manage classrooms, and review all data.
- **Professors** can be registered as needed and can **only manage their own academic schedules**.