

**Software Requirements**

**Specification**

**for**

**Eduplan**

**Version 2.3**

**Prepared by**

**Andrade Julio**

**Astudillo Gerald**

**Bonilla David**

**Caizapanta Tammy**

**SoftCrafters**

**May 21, 2025**

***Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

# 

# Table of Contents

[**Table of Contents 1**](#_heading=h.cdk1v69xolq1)

[**Revision History 3**](#_heading=)

[**Overall Description 3**](#_heading=h.h5707mm0p3wg)

[**System Features 3**](#_heading=h.6z25o6ie15b1)

[**System Features Correction 3**](#_heading=h.6z25o6ie15b1)

[**1. Introduction 4**](#_heading=h.7hz9vyp72oi)

[1.1 Purpose 4](#_heading=h.8jtmtydoudsi)

[1.2 Document Conventions 4](#_heading=h.61n4xcv71ju8)

[1.3 Intended Audience and Reading Suggestions 4](#_heading=h.i5hpiw84u6wi)

[1.4 Related Staff 4](#_heading=h.yz7m26iv66bg)

[1.5 Product Scope 5](#_heading=h.bqwfzjrbykpf)

[1.6 References 5](#_heading=h.r6m642yq3xct)

[**2. Overall Description 6**](#_heading=h.ccy8zpnenlib)

[2.1 Product Perspective 6](#_heading=h.pqhipf5le28j)

[2.2 Product Functions 6](#_heading=h.u4lmcln3ijuj)

[2.3 User Classes and Characteristics 6](#_heading=h.yihojqq2ifyd)

[2.4 Operating Environment 7](#_heading=h.1or8jn7j6e7f)

[2.5 Design and Implementation Constraints 7](#_heading=h.y4ts5mr5urou)

[2.6 User Documentation 7](#_heading=h.8nqgdiesdake)

[**3. External Interface Requirements 8**](#_heading=h.i53luev6wbsn)

[3.1 User Interfaces 8](#_heading=h.8p7an11rzbcj)

[3.2 Hardware Interfaces 8](#_heading=h.3hga41bp8pul)

[3.3 Software Interfaces 8](#_heading=h.jvfaxcekkzub)

[3.4 Communications Interfaces 9](#_heading=h.qx79swao5v0u)

[**4. System Features 10**](#_heading=h.y899g3o6brpr)

[1. RF001 10](#_heading=h.d3ze8qrunecw)

[2. RF002 10](#_heading=h.yf8xsodglegi)

[3. RF003 11](#_heading=h.xxy0c8dbu7xs)

[4. RF004 11](#_heading=h.vgna5xoro6q0)

[5. RF005 12](#_heading=h.bzwu2rb5aoyw)

[6. RF006 12](#_heading=h.d9qwr6624jh9)

[7. RF007 13](#_heading=h.u1vhve8loce5)

[8. RF008 13](#_heading=h.nnge4preioxh)

[9. RF009 14](#_heading=h.8r70aaw0v8n8)

[10. RF010 14](#_heading=h.4zi6x55l2djm)

[11. RF011 15](#_heading=h.g26dmd6bxx21)

[**5. Other Nonfunctional Requirements 16**](#_heading=h.pytufuqdaa7v)

[5.1 Performance Requirements 16](#_heading=h.ar2l0lozpex5)

[5.2 Security Requirements 16](#_heading=h.iofuloya9n1n)

[5.3 Software Quality Attributes 17](#_heading=h.fu4ogxlgpfnt)

[5.4 Business Rules 17](#_heading=h.6evwcaliaz3b)

# 

# Revision History

| **Author** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Caizapanta Tammy | 02-05-2025 | Initial version | 0.1 |
| Andrade Julio | 07-05-2025 | Introduction | 1.0 |
| Bonilla David | 10-05-2025 | Overall Description | 1.1 |
| Astudillo Gerald | 10-05-2025 | External Interface Requirements | 2.0 |
| Caizapanta Tammy | 14-05-2025 | System Features | 2.1 |
| Andrade Julio  Bonilla David | 16-05-2025 | System Features Correction | 2.2 |
| Andrade Julio  Astudillo Gerald  Bonilla David  Caizapanta Tammy | 20-05-2025 | Final version | 2.3 |

# Introduction

This document is a Software Requirements Specification for a system that assigned a daily activity to a weekly Micro - Curricular plan. This specification is based on the established IEEE 830, 1998 standards.

## Purpose

This document specifies the software requirements for Eduplan 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.

## 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.

## 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.

## Related Staff

| Name | Julio Andrade | Gerald Astudillo | David Bonilla | Tammy Caizapanta |
| --- | --- | --- | --- | --- |
| Email: | [jaandrade24@espe.edu.ec](mailto:jaandrade24@espe.edu.ec) | [giastudillo@espe.edu.ec](mailto:giastudillo@espe.edu.ec) | [dabonilla15@espe.edu.ec](mailto:dabonilla15@espe.edu.ec) | [tacaizapanta@espe.edu.ec](mailto:tacaizapanta@espe.edu.ec) |
| Position: | Leader / Programmer | Programmer | Programmer | Programmer |

## Product Scope

The software will allow the educational staff of the "Mamá Cigüeña" Child Development Center, create, modify, 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.

## References

*IEEE 830-1998 – Software Requirements Specification*

*User Interface Style Guide (*[*https://developer.mozilla.org*](https://developer.mozilla.org)*)*

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

# Overall Description

## 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 practical solution to improve planning and organization. The system is independent but can be adapted to integrate with institutional platforms if needed.

## 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.
* Weekly Planning (for Teachers): Create, update, and modify 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.

## 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.

## 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.

## 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.

## 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.

# External Interface Requirements

## 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.

## 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

## 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.

## 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

# System Features

## RF001

| Requirement ID | RF001 |
| --- | --- |
| Requirement Name | Log in |
| Type | Functional Requirement |
| Actor | Director / Teacher |
| Requirement Description | Develop a secure login interface requiring users to provide credentials for system access. |
| Inputs | Username, Password |
| Outputs | Authentication |
| Requirement Priority | High |

## 

## RF002

| Requirement ID | RF002 |
| --- | --- |
| Requirement Name | User Registration |
| Type | Functional Requirement |
| Actor | Director / Teacher (New User) |
| Requirement Description | Implement a robust user registration module allowing new users to create accounts. Upon registration, the system shall automatically generate a unique, random User ID for each new entry. |
| Inputs | Username, Password, Role (Teacher/Principal) |
| Outputs | User Account Created, Unique User ID Generated |
| Requirement Priority | High |

## RF003

| Requirement ID | RF003 |
| --- | --- |
| Requirement Name | Role-Based Access Control |
| Type | Functional Requirement |
| Actor | User (Director / Teacher) |
| Requirement Description | Integrate a role selection interface during the initial login or registration process, enabling users to designate their role as either "Teacher" or "Principal." |
| Inputs | Role Selection |
| Outputs | User Role Assigned |
| Requirement Priority | High |

## RF004

| Requirement ID | RF004 |
| --- | --- |
| Requirement Name | Authentication and Authorization |
| Type | Functional Requirement |
| Actor | System |
| Requirement Description | Implement a mechanism to authenticate user credentials against stored user data and authorize access based on assigned roles. |
| Inputs | Authenticated Credentials, Assigned Role |
| Outputs | Authorized Access to Specific Interface |
| Requirement Priority | High |

## RF005

| Requirement ID | RF005 |
| --- | --- |
| Requirement Name | Login Attempt Management |
| Type | Functional Requirement |
| Actor | System / User |
| Requirement Description | Limit login attempts to a maximum of three consecutive failures. Upon exceeding this limit, the associated user account shall be locked for a period of one minute. |
| Inputs | Failed Login Attempts |
| Outputs | Account Lockout (60 seconds) |
| Requirement Priority | High |

## RF006

| Requirement ID | RF006 |
| --- | --- |
| Requirement Name | Dynamic Interface Assignment |
| Type | Functional Requirement |
| Actor | System |
| Requirement Description | Automatically assign the appropriate user interface (Teacher or Principal) based on the authenticated user's role. |
| Inputs | Authenticated User Role |
| Outputs | Display of Role-Specific Interface (Teacher UI / Principal UI) |
| Requirement Priority | High |

## RF007

| Requirement ID | RF007 |
| --- | --- |
| Requirement Name |  |
| Type |  |
| Actor |  |
| Requirement Description |  |
| Inputs |  |
| Outputs |  |
| Requirement Priority | High |

## RF008

| Requirement ID | RF008 |
| --- | --- |
| Requirement Name |  |
| Type |  |
| Actor |  |
| Requirement Description |  |
| Inputs |  |
| Outputs |  |
| Requirement Priority | High |

## RF009

| Requirement ID | RF009 |
| --- | --- |
| Requirement Name |  |
| Type |  |
| Actor |  |
| Requirement Description |  |
| Inputs |  |
| Outputs |  |
| Requirement Priority | High |

## RF010

| Requirement ID | RF010 |
| --- | --- |
| Requirement Name |  |
| Type |  |
| Actor |  |
| Requirement Description |  |
| Inputs |  |
| Outputs |  |
| Requirement Priority | High |

## RF011

| Requirement ID | RF011 |
| --- | --- |
| Requirement Name |  |
| Type |  |
| Actor |  |
| Requirement Description |  |
| Inputs |  |
| Outputs |  |
| Requirement Priority | High |

# Other Nonfunctional Requirements

## 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.

## 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.

## 

## 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.

## 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**.