School Master

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 28/03/2023 | 1.0 | details | Cosma Felicia-Iulia |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Non-functional Requirements 4

2.1 Availability 4

2.2 Performance 4

2.3 Security 4

2.4 Testability 4

2.5 Usability 4

3. Design Constraints 4

# Introduction

# This Supplementary Specification documents the requirements for a client-server application for school or high school administration. The application is designed to be used by teachers, students, parents, and administrators. The system provides various functionalities, such as managing timetables, assigning grades, and finalizing the students' situation.

# The system aims to improve communication between all stakeholders involved in school administration. It automates various administrative tasks, such as grade assignments and email notifications to parents, which reduces manual errors and saves time.

# Non-functional Requirements

## Availability

Quality attribute definition: The system must be available 24/7, with minimal downtime for maintenance and updates.

Source of stimulus: Users (teachers, students, parents, and administrators)

Stimulus: System unavailability or downtime

Environment: During regular school hours or outside of school hours

Artifact: System

Response: The system must be available for use by users.

Response measure: The system should have an uptime of at least 99%.

Tactics:

Implementing a redundant architecture to provide high availability.

Monitoring system uptime to identify potential issues before they become critical.

Conducting maintenance and updates during off-hours to minimize downtime during regular school hours.

## Performance

Quality attribute definition: The system must perform efficiently, ensuring a responsive user interface and minimal delays in processing requests.

Source of stimulus: Users (teachers, students, parents, and administrators)

Stimulus: Heavy user traffic, concurrent requests, large data transfers, and database queries

Environment: During regular school hours or outside of school hours

Artifact: System

Response: The system must respond to user requests in a timely manner.

Response measure: The system should provide a response time of less than 3 seconds for 95% of requests.

Tactics:

Implementing a distributed architecture to distribute the load and improve performance.

Caching frequently accessed data to reduce database queries.

Using optimized database queries to reduce processing time.

Monitoring system performance to identify potential bottlenecks and improve system efficiency.

## Security

Quality attribute definition: The system must ensure the security of sensitive data and prevent unauthorized access.

Source of stimulus: External entities (hackers, unauthorized users)

Stimulus: Attempted unauthorized access, hacking, and data breaches

Environment: Anytime

Artifact: System

Response: The system must protect sensitive data and prevent unauthorized access.

Response measure: The system should implement industry-standard security measures to protect data, such as encryption, access controls, and auditing.

Tactics:

Implementing access controls to restrict user access based on roles and permissions.

Encrypting sensitive data at rest and in transit.

Implementing auditing and monitoring to detect and respond to security threats.

Regularly conducting security audits and assessments to identify and address potential vulnerabilities.

## Testability

Quality attribute definition: The system must be easily testable to ensure software quality and reduce the risk of errors and defects.

Source of stimulus: Testing teams

Stimulus: Conducting various types of testing (unit testing, integration testing, system testing, acceptance testing)

Environment: Testing environments

Artifact: System

Response: The system must be designed to be easily testable.

Response measure: The system should have comprehensive test cases and automated testing scripts to ensure thorough testing and reduce the risk of errors and defects.

Tactics:

Implementing a modular architecture to isolate and test individual components.

Implementing automated testing scripts to ensure thorough and efficient testing.

Conducting regular code reviews and testing to identify and fix potential issues.

## Usability

Quality attribute definition: The system shall be easy to use and learn for all users.

Source of stimulus: Users attempting to access the system.

Stimulus: Users attempt to access or modify data or functionality.

Environment: The system is deployed and operational.

Artifact: The user interface and its components.

Response: The system shall have an intuitive and user-friendly interface that enables easy access to all functionality.

Response measure: The system shall have a usability score of at least 80% on standardized usability tests.

Tactics: Use user-centered design principles, perform user testing, and provide clear documentation to ensure usability.

# Design Constraints

The system shall be designed using the following technologies:

Programming language: Java

Web framework: Spring Boot

Database: PostgreSQL

Email service: SMTP protocol

UI Framework: ReactJS

These design decisions have been mandated and must be adhered to for the successful implementation of the system.