Pupil Tracker

Software Requirements Specification v1.0

02.11.2024

Stanislav Stoyanov

Lead Software Engineer

# Revision History

**Date Version Author Comments**

|  |  |  |  |
| --- | --- | --- | --- |
| 02.11.2024 | v1.0 | Stanislav Stoyanov | First revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

**Signature Printed Name Title Date**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Stanislav Stoyanov | Lead Software Eng. | 02.11.2024 |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[REVISION HISTORY 2](#_TOC_250047)

[DOCUMENT APPROVAL 2](#_TOC_250046)

1. [INTRODUCTION 4](#_TOC_250045)
   1. [Purpose 4](#_TOC_250044)
   2. [Scope 4](#_TOC_250043)
   3. [Definitions, Acronyms, and Abbreviations 5](#_TOC_250042)
   4. [References 5](#_TOC_250041)
   5. [Overview 5](#_TOC_250040)
2. [GENERAL DESCRIPTION 6](#_TOC_250039)
   1. [Product Perspective 6](#_TOC_250038)
   2. [Product Functions 6](#_TOC_250037)
   3. [User Characteristics 7](#_TOC_250036)
   4. [General Constraints 7](#_TOC_250035)
   5. [Assumptions and Dependencies 7](#_TOC_250034)
3. [SPECIFIC REQUIREMENTS 8](#_TOC_250033)
   1. [External Interface Requirements 8](#_TOC_250032)
      1. [User Interfaces 8](#_TOC_250031)
      2. [Hardware Interfaces 8](#_TOC_250030)
      3. [Software Interfaces 8](#_TOC_250029)
      4. [Communications Interfaces 8](#_TOC_250028)
   2. [Functional Requirements 8](#_TOC_250027)
      1. [Grade management 9](#_TOC_250026)
      2. [Key generation 9](#_TOC_250025)
      3. [Performance evaluation 9](#_TOC_250024)
      4. [University management 10](#_TOC_250023)
   3. [Use Cases 11](#_TOC_250022)
      1. [Teacher adding grades 11](#_TOC_250021)
      2. [Student generating key 11](#_TOC_250020)
      3. [Third party getting performance overview 11](#_TOC_250019)
   4. [Classes / Objects 12](#_TOC_250018)
      1. [Student 12](#_TOC_250017)
      2. [Teacher 12](#_TOC_250016)
      3. [Admin 12](#_TOC_250015)
   5. [Non-Functional Requirements 13](#_TOC_250014)
      1. [Performance 13](#_TOC_250013)
      2. [Reliability 13](#_TOC_250012)
      3. [Availability 13](#_TOC_250011)
      4. [Security 13](#_TOC_250010)
      5. [Maintainability 13](#_TOC_250009)
      6. [Portability 13](#_TOC_250008)
   6. [Inverse Requirements 13](#_TOC_250007)
   7. [Design Constraints 14](#_TOC_250006)
   8. [LOGICAL DATABASE REQUIREMENTS 14](#_TOC_250005)
   9. [OTHER REQUIREMENTS 14](#_TOC_250004)
4. [ANALYSIS MODELS 15](#_TOC_250003)
   1. [Sequence Diagrams 15](#_TOC_250002)
   2. [Data Flow Diagrams (DFD) 16](#_TOC_250001)
5. [CHANGE MANAGEMENT PROCESS 17](#_TOC_250000)

# Introduction

This Software Requirements Specification (SRS) document outlines the requirements for a software system designed to facilitate the verification of student performance data in a university setting. The document serves as a comprehensive guide for software engineers and developers tasked with designing and implementing the system, ensuring that all necessary specifications and functionalities are clearly detailed.

## Purpose

The purpose of this SRS is to define the functionality and constraints of the university performance verification system, intended for developers, project managers, and stakeholders involved in the software’s development. This document aims to ensure all parties have a shared understanding of the system's requirements and objectives.

## Scope

The primary software products to be produced are the internal zero-knowledge and the external performance verification services.

The internal service will store student grades, allow teachers to input grades (including bulk CSV imports), and enable students to generate unique keys. The external service will provide an admin panel for managing universities and subjects and allow third parties to retrieve performance estimates without accessing personal data.

The benefit of having the proposed software system is to enhance the efficiency of grade verification while maintaining data privacy. Objectives: Provide secure, efficient access to student performance metrics while ensuring compliance with GDPR and maintaining high user experience standards. Goals: Allow third parties to easily access performance data while ensuring data integrity through digital signatures.

## Definitions, Acronyms, and Abbreviations

* Internal service - zero knowledge verification service, which is hosted on-premise by university and that is allowed to store grade data.
* GDPR - General Data Protection Regulation.
* TLS - Transport Layer Security
* EdDSA - Edwards-Curve Digital Signature Algorithm
* CSV - Comma-Separated Values

## References

GDPR documentation is available at [European Commission website](https://ec.europa.eu/info/law/law-topic/data-protection/eu-data-protection-rules_en).

## Overview

The remainder of this SRS is organized into several key sections:

* General Description: describes the customer requirements and general factors that affect the product and its requirements.
* Specific Requirements: outlines specific requirements, that will be used to guide the projects software design.
* Analysis Models: lists all analysis models used in developing specific requirements previously given in this SRS.
* Change Management Process: identifies and describes the process that will be used to update the SRS.

# General Description

This section provides an overview of the factors influencing the development of the university performance verification system. It helps contextualize the software within the broader landscape of educational technology and sets the stage for understanding specific requirements.

## Product Perspective

The university performance verification system consists of two interrelated components: the internal zero-knowledge and the external performance verification services.

The internal service acts as a platform for storing student grades securely and allows authorized personnel (teachers) to manage grade data.

The external service enables third-party access to performance metrics while maintaining strict data privacy through unique student keys.

This system is designed to integrate with existing educational databases and systems within universities to facilitate seamless data transfer, particularly for bulk grade imports via CSV.

## Product Functions

Internal service

* User authentication for teachers and students.
* Entry and management of student grades, including bulk imports.
* Generation of unique cryptographic keys for students.

External service

* Provide an admin panel for managing universities and subjects.
* Offer an overview page for third parties to retrieve student performance estimates using unique keys.
* Validate and verify the integrity of performance data using digital signatures.

## User Characteristics

The users of the system will include:

* Teachers: Likely to be familiar with educational technologies and require intuitive interfaces for grade entry and management.
* Students: Varied levels of technical proficiency; need straightforward processes to generate keys and access performance data.
* Admins: Responsible for managing system-wide settings; should have experience in database management and user permissions.
* Third Parties: Users (like employers) accessing performance estimates; will require clear and concise information without exposure to personal data.

## General Constraints

* Regulatory Compliance: The system must comply with GDPR, affecting data handling and privacy practices.
* Security Requirements: Must employ robust security measures, including Argon2 for password hashing and EdDSA for digital signatures.
* Infrastructure: The internal service will be hosted on university premises, which may limit technical configurations compared to cloud-hosted solutions.
* Data Volume: The design must accommodate the anticipated volume of student and grade data while ensuring performance and response times meet user expectations.

## Assumptions and Dependencies

* It is assumed that the university will provide the necessary infrastructure for hosting the internal service.
* The software will depend on the availability of secure internet access for the external service.
* The system will rely on compatible web browsers for accessing the web frontend, without specific design requirements on the user interface.
* The performance of the system assumes the stable operation of supporting libraries and frameworks for security, data handling, and user authentication.

# Specific Requirements

This section outlines the detailed, design-oriented requirements for the university performance verification system. These requirements are structured to be unambiguous, testable, and prioritized to facilitate effective design, implementation, and testing.

## External Interface Requirements

## User Interfaces

* + - 1. Internal service:
         * Teachers: access to grade entry screens, including bulk import functionality via CSV.
         * Students: access to key generation interface.
      2. External service:
         * Admin: access to an admin panel for managing universities, subjects and categories of subjects.
         * Public: access to an overview page for performance estimates.

## Hardware Interfaces

Internal service hosted on university’s premises; external service hosted on cloud infrastructure.

## Software Interfaces

* + - 1. Internal service:
         * Integration with university databases for grade storage.
         * CSV import/export compatibility for bulk grade handling.
      2. External service:
         * Compatibility with common web browsers for user access.
         * REST API for secure data retrieval and performance estimation.

## Communications Interfaces

All communications must use TLS to ensure secure data transfer.

## Functional Requirements

## Grade management

* + - 1. Introduction

Manages the secure storage and retrieval of student grades.

* + - 1. Inputs

Grade data (individual or CSV bulk), subject name, teacher credentials.

* + - 1. Processing

Validate and store grades, ensure digital signing for data integrity.

* + - 1. Outputs

Confirmation of grade entry; error messages on invalid data.

* + - 1. Error Handling

Flag invalid entries, provide detailed error feedback on import issues.

## Key generation

* + - 1. Introduction

Generates unique cryptographic keys for students to share with third parties.

* + - 1. Inputs Student credentials.
      2. Processing

Generate unique, secure key for performance verification.

* + - 1. Outputs

Display key to the student with copy/download options.

* + - 1. Error Handling

Notify user if key generation fails; limit retry attempts.

## Performance evaluation

* + - 1. Introduction

Allows third parties to access anonymized performance data.

* + - 1. Inputs Unique student key.
      2. Processing

Retrieve grades, calculate percentile/estimate, verify data integrity via digital signature.

* + - 1. Outputs

Display performance estimates based on percentile rankings.

* + - 1. Error Handling

Provide feedback on invalid or expired keys; log unauthorized access attempts.

## University management

* + - 1. Introduction

Admins manage supported universities subjects and subject categories for performance estimation.

* + - 1. Inputs

University details, subject names and categories.

* + - 1. Processing

Add, edit, or delete university and subject entries in the system.

* + - 1. Outputs

Confirmation of changes; updated subject/university lists.

* + - 1. Error Handling

Validate data inputs, flag invalid entries, log changes.

## Use Cases

## Teacher adding grades

* + - 1. Actors teacher
      2. Preconditions

teacher is authenticated in the internal service.

* + - 1. Flow

teacher enters grades manually or uploads a CSV file, confirms, and receives feedback on successful or unsuccessful entries.

* + - 1. Postconditions

grades are securely stored with digital signatures.

## Student generating key

* + - 1. Actors student
      2. Preconditions

student is authenticated in the internal service.

* + - 1. Flow

student requests a new key, views, and saves the generated key.

* + - 1. Postconditions

unique key is associated with the student’s record.

## Third party getting performance overview

* + - 1. Actors anyone
      2. Preconditions

student key, name and university name are known.

* + - 1. Flow

student shares the necessary information and performance overview is requested on the external service.

* + - 1. Postconditions

third party can verify students academic performance.

## Classes / Objects

## Student

* + - 1. Attributes

ID, name, degree type, key, grades

* + - 1. Functions GenerateKey, ViewGrades

## Teacher

* + - 1. Attributes ID, name, subjects
      2. Functions

AddGrade, ImportGradesCSV, ViewGrades

## Admin

* + - 1. Attributes

ID, university affiliation

* + - 1. Functions

EditUniversities, EditSubjects, EditCategories

## Non-Functional Requirements

## Performance

System must process requests with a response time under 1 second for single queries and under 5 seconds for bulk imports.

## Reliability

Ensure 99.99% uptime for services, allowing for regular maintenance windows.

## Availability

All services in the system must be available at any point in time.

## Security

Argon2 for password hashing, EdDSA for digital signatures, TLS for all communications.

Ensure GDPR compliance in all data processing.

## Maintainability

Modular code structure to allow future enhancements, particularly for subject and university management.

## Portability

External service must be compatible with popular browsers and cloud environments.

## Inverse Requirements

No storage of personal identifiers (e.g., student name, ID, grades) in the external service. External service must not allow access to raw grade data, only aggregated performance metrics.

## Design Constraints

The system must use university-approved hardware for the internal service. External service must use cloud-approved architecture for scalability.

## Logical Database Requirements

The internal services database must support secure storage of grade data and enforce access restrictions.

Relational database management systems are to be used on both internal and external services.

## Other Requirements

Comprehensive logging for all admin actions, grade imports, and performance queries for auditing and accountability.

Error logs must be anonymized and stored for system improvement and diagnostics.

# Analysis Models

This section presents essential analysis models that illustrate interactions and data flows, supporting the system’s specific requirements.

## Sequence Diagrams

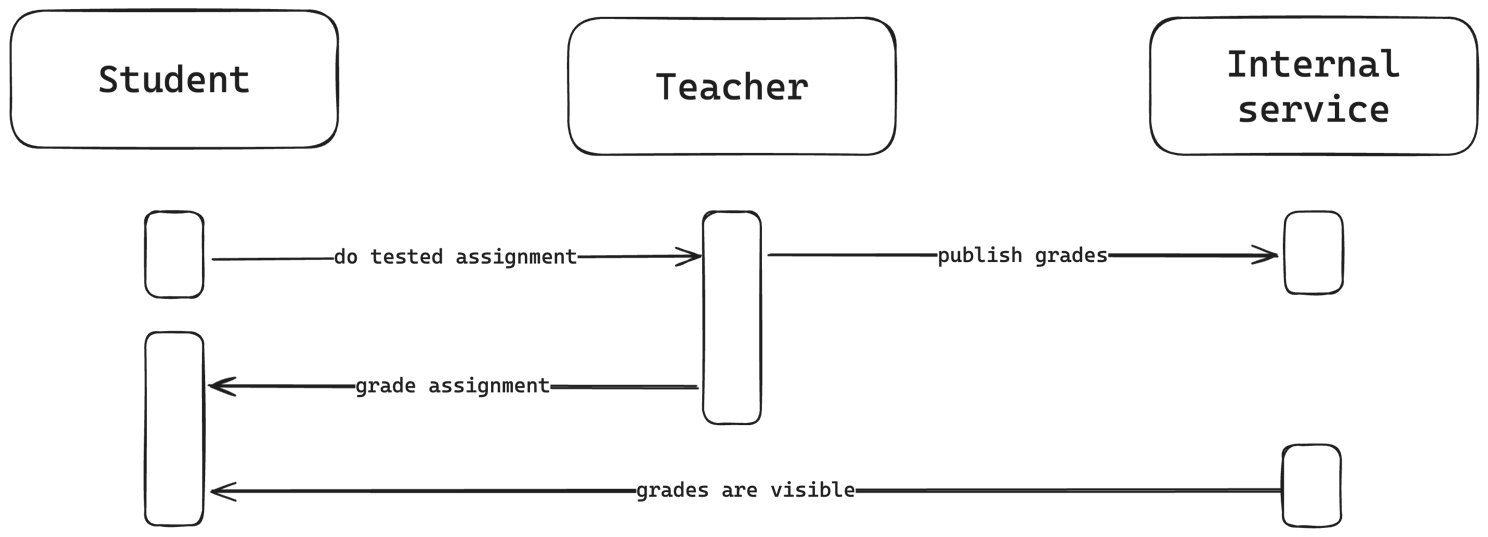
****

Figure 1 – Grading sequence diagram

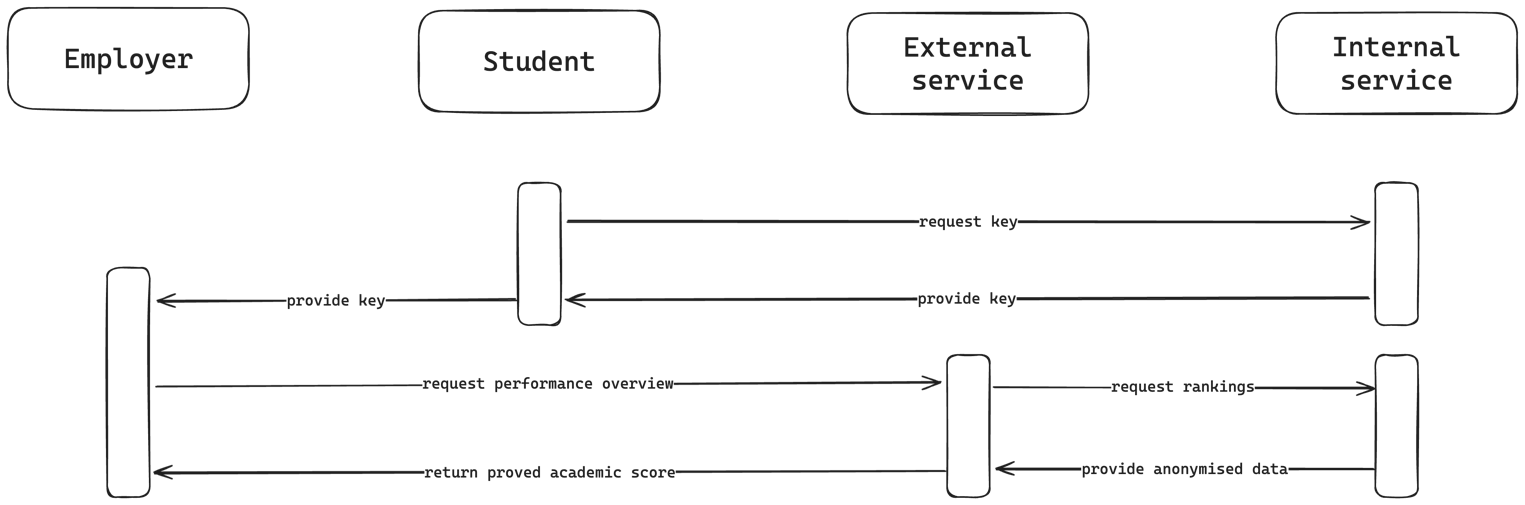


Figure 2 – Performance evaluation sequence diagram

## Data Flow Diagrams (DFD)

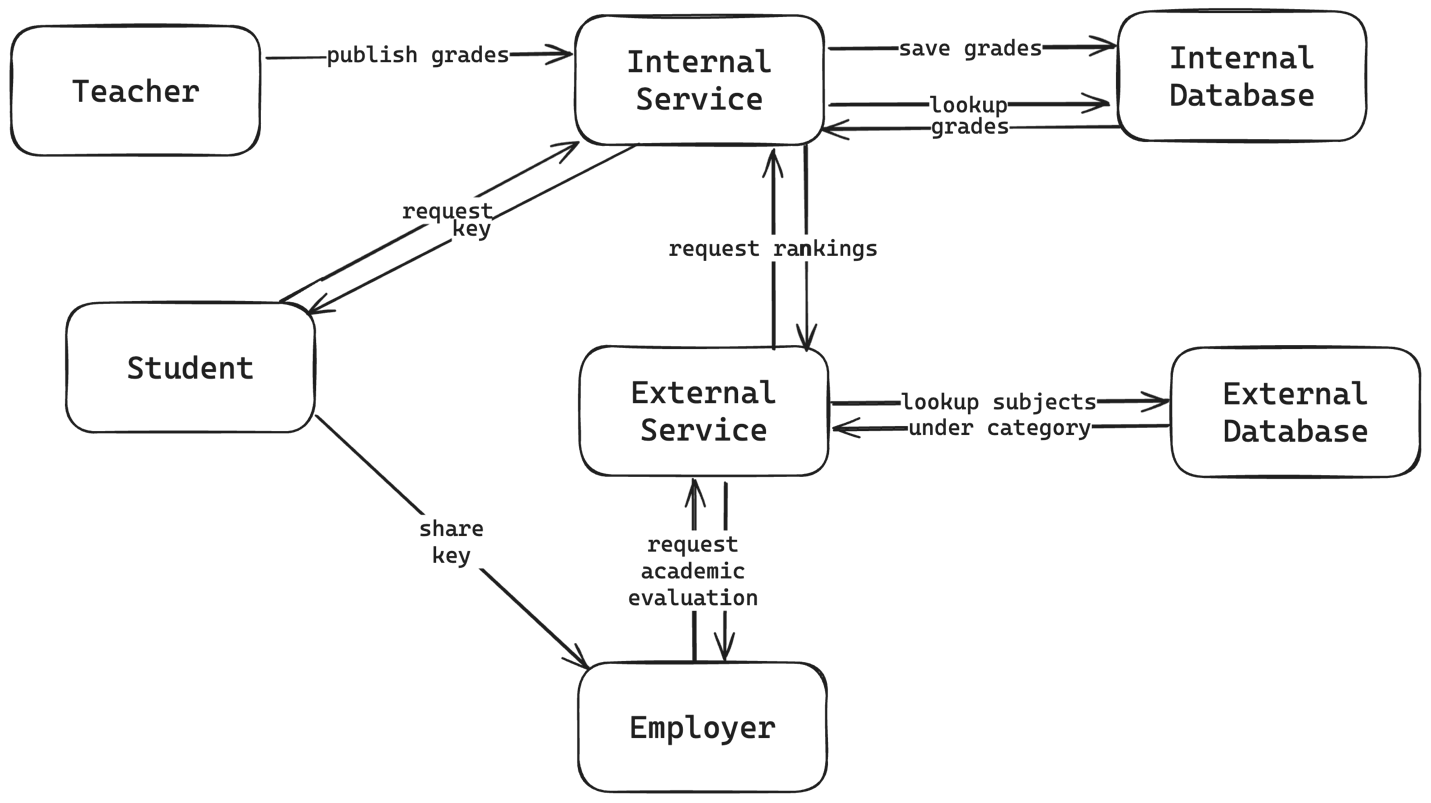
****

Figure 3 – Data flow diagram

# Change Management Process

Changes to the SRS can be submitted by any project stakeholder, including university representatives and developers. Each change request will then be evaluated by each stakeholder.