# Introduction

## Purpose

The purpose of this document is to build an REST API Service to school information system.

## Document Conventions

This document uses the following conventions.

DB – Database

ER – Entity Relationship

GET – The GET method is used to retrieve information from the given server using

a given URI. Requests using GET should only retrieve data and should have no other

effect on the data.

POST – A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.

## Designed audience and reading offers

# This project is a prototype of a school information system. It is useful for school teachers as well as students.

## Scope of the project

The purpose of the school information system is to make it easier for students and teachers to work with subjects, grades and schedules. First of all, we hope to provide comfortable user experience  and system stability.

# Overall Description

## Product Perspective

* **Teacher details:**

It includes username, password, name, last name, email and id of department, where teacher works.

* **System administrator details:**

It includes username, password, name, last name.

* **Student details:**

It includes username, password, name, last name, email and id of faculty, where student is studying.

## Product Features

* **Teacher Features:**
  + **CRUD**
  + **GET method:**
    - **Get teachers school schedule.**
    - **Get teachers lessons.**
  + **POST method:**
    - **POST evaluate students.**
    - **POST change profile information.**
* **System Administrator Features:**
  + **CRUD**
  + **GET method:**
    - **GET a list with all teachers.**
    - **GET a list with teachers by criteria: Id, First Name, Last Name, Department.**
    - **GET a list of teacher lessons by criteria: Id**
    - **GET a teacher’s school schedule by Id.**
    - **GET a list with all students.**
    - **GET a list with students by criteria: Id, First Name, Last Name, Departments.**
    - **GET a list of student lessons by criteria: Id**
    - **GET a student’s school schedule by Id.**
    - **GET all department.**
    - **GET department by Id.**
    - **GET a list of department lessons by criteria: Id, Name.**
    - **GET number of teachers on select department.**
    - **GET number of all teachers.**
    - **GET number of teacher’s students.**
    - **GET number of all students.**
    - **GET remove a teacher by Id.**
    - **GET remove a student by Id.**
    - **GET remove a lesson.**
    - **GET remove a department.**
    - **GET remove an event.**
  + **POST method:**
    - **POST add a new teacher.**
    - **POST add a new student.**
    - **POST add a new department.**
    - **POST add a new event. Lesson, Teacher, Class Room, Time.**
    - **POST add a new lesson.**
* **Student Features:**
  + **GET methods:**
    - **GET student profile: First Name, Last Name, Date at Birthday, Current Course, Number of Courses.**
    - **GET student school schedule.**
    - **GET current lessons.**
    - **GET compulsory lessons.**
    - **GET compulsory-option lessons.**
  + **POST methods:**
    - **POST change profile information.**
    - **POST change schedule for selected lesson.**
    - **POST add lessons to current courses.**
    - **POST remove lessons to current courses.**

# NON functional requiremnts

* **ELECTRONIC DIAGRAM**

An E-R diagram is a method for representing the logical structure of a database in graphical form.

* **USABILITY**

Thanks to REST API It is easy for a customer to use the School Information System.

* **SECURITY**

The system and its data protected against attacks by using Prepared Statement.

## SOFTWARE QUALITY ATTRIBUTES

* **AVAILABILITY**

A lecture hall or a class for lessons should be available on the specified date and time.

* **CORRECTNESS**:

The department must have the correct list of teachers, the faculty must have the correct list of specialties, and the speciality must have the correct list of subjects and students.

* **MAINTENANCE**

Administrators must maintain the correct school curriculum.

* **USABILITY**

The subject schedule should satisfy a maximum number of teachers and students.

# System limits

* The student will not be able to write down optional subjects, there is a specialty that has a list of subjects and a student can only have subjects from this list.
* The teacher or administrator will not be able to send messages to students.
* The system does not provide the ability to have two or three roles.. A student cannot also be a teacher; a teacher cannot be an administrator.

# Unexpected problems

# We had had a few problems with implementation like:

* Realisation inheritance with School\_user and Student and School\_user and Teacher.
* Creating right UML Diagram which will represent our ideas about the System.
* Creating timetable for Students. There were 5 entities which had to be in relation to make the right timetable. But its a very difficult relationship, where could be a lot of errors. Finally, we desided to remove 2 entities from this relation.

# Used technologies

# Spring boot

# JPA

# REST API

# GIT

# Postgres SQL

# Unit Tests