Software Requirements Specification

For

Attendance management system

Prepared by Bahir Hakimi

KU Software House

October 14, 2022

Contents

[1. Introduction 1](#_Toc117240893)

[1.1 Purpose 1](#_Toc117240894)

[1.2 Intended Audience 1](#_Toc117240895)

[1.3 Project Scope 1](#_Toc117240896)

[1.4 Task Force 1](#_Toc117240897)

[2. Overall Description 2](#_Toc117240898)

[3. System Features 5](#_Toc117240899)

[3.1 Adding a New Student: 5](#_Toc117240900)

[3.2 Adding a New Teacher 5](#_Toc117240901)

[3.3 Adding a New Subject 5](#_Toc117240902)

[3.4 Adding a New Class 6](#_Toc117240903)

[3.5 Take Attendance 6](#_Toc117240904)

[3.6 Student can check their attendance report 6](#_Toc117240905)

[4. External Interface Requirements 7](#_Toc117240906)

[5. Other Nonfunctional Requirements 7](#_Toc117240907)

# Introduction

## Purpose

The main purpose of this specification is to help people who will work on this system to maintain the objectives and get started working in this project. This specification will direct people who will work on this project step by step through the process until they finish it successfully. This statement will describe specific details into every step of this project that workers will immediately locate the needs of this system to understand the purpose of doing any of the following steps into the system.

## Intended Audience

The audience of this system will be:

1. Students

2. Teachers

3. Faculty members.

## Project Scope

The scope of this system is to provide a better solution for trivial attendance system.

That will help the community use the technology in effective ways:

1. Make the attendee process easier and effective.

2. Help faculty in the attendance process every time.

3. Mange and organize the attendance through this system.

## Task Force

1. **Zubair Adelyar, Mohammad Mahdi Wahid and Safi Rahimi** (**Frontend Designers and Developers**)
2. **Shahed Ahmad, Haroon Khairy and Bahir Hakimi** (**Backend Developers and Database Designers**)
3. **Bahir Hakimi** (**Project Manager**)

# Overall Description

**2.1 Product Perspective**

At Kabul University, instructors manually take attendance in every class each day. They spend time to do that during class time. This Digital Attendance System will help them do this process in an easy way. The main scope of this project is to make attendance process more organized in every class. This project will help instructors take the attendance more easily with the help of computer. There are also many benefits for students: they can manage their attendance, absences, and late walk-ins by checking the Attendance web site. It makes it easier to have a clear picture of every student’s attendance throughout the academic year.

Currently this system will be the infrastructure for the whole system. The more complete version of system will be built on top of this system.

**2.2 Project Plan**

This project will be a beta version of the main system, thus it will have limited functionality.

The project should be developed and tested within 2 week and will be released in 30/10/2022.

**2.3 Product Features**

1. Authentication system for users based on their roles (Admin, Teacher, Student).
2. Admins can register teachers and students.
3. Admin can create or remove subjects.
4. Teachers can access the attendance form and take attendance.
5. Students can see their attendance history.

**2.4 User Classes and Characteristics**

There are three types of user in this system:

1) Admin

2) Teacher

3) Student

**2.5 Overall Description of System Workflow**

1. First, Admin must register all **Students, Teachers** and their associated **Subjects**.
2. Teachers sign in to the system and access that **particular day and particular class** attendance form (based on their teaching schedule).
3. Then the teacher takes the attendance visually and mark each student as **present or absent**.
4. After that, the teacher submits the attendance form to the server.
5. The server validates the form (**day is valid, teacher is valid, subject is valid**).
6. Finally, the server saves the data into database.
7. The students takes their **username** and **password** from the system admin.
8. The can use these credential to see their **attendance report** and **history**.

**2.6 Database Design**

Below are the main component of the system.

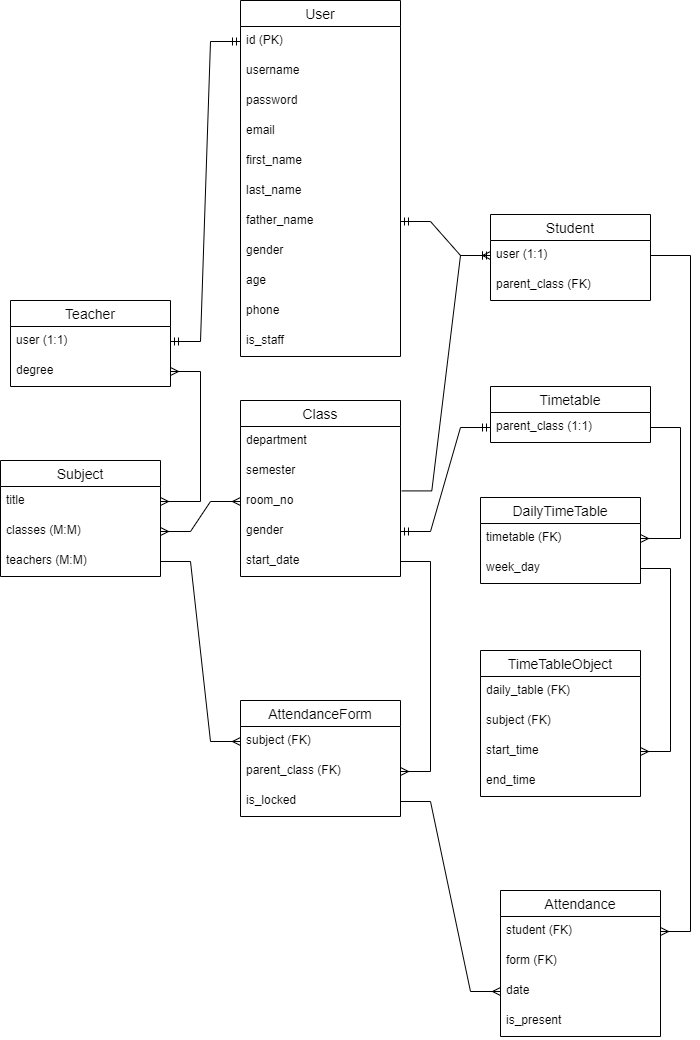
* User
* Administrator
* Teacher
* Student
* Subject
* Class
* Attendance
* Timetable

Here is the description of each component.

* User is used to store the username is and password.
* Administrator is a User but has its specific properties.
* Teacher is also a User with its own properties; Each Teacher is associated with one or more Subjects.
* Student is also a User and it is associated with only one class.
* Subject is used to store each subject and its associated with one or more Teachers and one Class
* Class is used to store the class info e.g. Room number, floor, semester etc…
* Attendance is for storing the presence or absence of a student.
* Timetable we use it to validate attendance form submission.

The relationship between these components is as follow:

**ER diagram:**



# System Features

**Functional Requirements**

## Adding a New Student:

Function: Sign up a new student to the system.

Priority: Top (Required for first release)

Requirements: To add a new student to the system the person should be a student of the faculty and should be added by the system admin.

**List of Endpoints**:

* Add\_student()
* Get\_student()
* Update\_student()
* Delete\_student()
* Get\_student\_list()

## 3.2 Adding a New Teacher

Function: Sign up a new Teacher to the system.

Priority: Top (Required for first release)

Requirements: To add a new teacher the person should a teacher of the faculty and will be added by system admin.

**List of Endpoints**:

* Add\_teacher()
* Get\_teacher()
* Update\_teacher()
* Delete\_teacher()
* Get\_teacher\_list()

## Adding a New Subject

Function: Register a new Subject to the system.

Priority: Top (Required for first release)

Requirements: To add a new subject the subject must be part of curriculum and will be added by system admin.

**List of Endpoints**:

* Add\_subject()
* Get\_subject()
* Update\_subject()
* Delete\_subject()
* Get\_subject\_list()

## Adding a New Class

Function: Add a new Class to the system.

Priority: Top (Required for first release)

Requirements: To add a new class the class must be an existing class and will be added by system admin.

**List of Endpoints**:

* Add\_class()
* Get\_class()
* Update\_class()
* Delete\_class()
* Get\_class\_list()

## 3.5 Take Attendance

Function: The teacher should be able to take attendance of the student through a form.

Priority: Top (Required for every class attended)

Requirements: When the teacher goes to class, he must sign in and take the attendance of students based on visual perception.

**List of Endpoints**:

* Get\_attendance() 🡺 takes the class id and subject id and returns attendance record of this month with a field which shows whether it is editable or not.
* Submit\_attendance() 🡺 takes the attendance form id and list of students with their attendance (present or absent) and returns the updated report.
* Get\_attendance\_report() 🡺 takes the class id and subject id and returns attendance record of that particular class and subject

## 3.6 Student can check their attendance report

Function: Every student can check his or her attendance history.

Priority: High

Requirements: For student to be able to check their attendance report they should be given a username and password by the administration to login and check their report.

**3.7 Adding Timetables**

* Get\_timetable()
* Add\_timetable()
* Update\_timetable()
* Delete\_timetable()
* Get\_timetable\_list()

**3.8 Additional endpoints**

* Get\_user\_type() 🡺 takes username and returns user’s role(Admin, Teacher, Student)

# External Interface Requirements

**4.1 User Interfaces**

**Login Page:**

This is the main login in the system, which users will use to sign in to system (Admin, Teachers and Student).

**Admin Panel:**

In this page, the list of students, classes, subjects and teachers, additionally, we will have the buttons and forms for adding students, teachers, class and subject, it should also have a section to check and manage attendance report.

**Error message view:**

We use this page to show deferent kinds of errors (404,403, server Errors and more).

**Student, Teacher, Subject and Class Registration Page:**

These pages is only accessible by the admin; no other users will have access to these forms.

These for will also handle validation errors.

**Student Report:**

This page will be accessible for the students and will be a part of Admin page, when a student logs in he/she will only see this page.

**4.3 Software Interfaces**

The system will use:

1) ReactJS or NextJS for Frontend

2) Django Rest framework or NodeJS + Express for Backend

3) Sqlite3, PostgreSQL or MongoDB for Database.

# Other Nonfunctional Requirements

Will be added later…