# Introduction

## 1.1 Purpose

The main purpose of this document is to introduce our project to the reader. This project focused on a web application for practicing and learning email communication.

## 1.2 Typographic conventions

|  |  |
| --- | --- |
| **Abbreviations** | **Full form** |
| LLM | Large Language Model |
| AI | Artificial Intelligence |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheets |
| JS | JavaScript |
| PHP | Hypertext Preprocessor |
| PDO | Hypertext Preprocessor Data Objects |
| MySQL | My Structured Query Language |

## 1.3 Target audience

This document will be only available within the college premises. The project is mainly targeted to people, who would like to improve their email communication skills or are looking for a platform, where they can teach email communication to other people.

The purpose of our project is to make a platform specifically for teachers and students to learn and practice email communications.

## 1.4 Scope of the project

The scope of our project is to make a platform specifically for teachers and students to learn and practice email communications. We also want to prepare this project for a future implementation of AI chat bot based on LLM, who would teach email communications without the need for a teacher. However, the implementation of the AI chat bot is not in the current plan.

## 1.5 References

So far, the project is only available privately throughout the GitHub repository via this link: <https://github.com/DavidHadek/ZSWI>

In the near future, the application will be put on a cloud and will be available for everyone online.

# Overall description

## System context

So far, there were only abstract solutions for teachers to give their students any kind of homework, but there isn’t a specific solution related to only teaching the email communications.

We thrive to make an easy platform for everyone, who would like to improve their students’ or colleagues email communications skills, therefore we mostly take inspiration from a functioning, yet abstract, teaching platform, the Google Classroom.

## Product features

The final product will include these features:

* Registration and authentication.
* Creating classes.
* Adding users (students) to the classes.
* Assigning tasks related to email communications.
* Reading and commenting tasks.
* Chat-like feature for the email communication scenario between the student and teacher.
* Profile of users with their information

## User classes and characteristics

The unregistered user will be count as a visitor, they will have no specific privileges, apart from seeing the content of pages and creating a new account.

The registered user will be able to create classes, see their profile and join other classes.

The teacher (creator of the class) will be able to maintain their students. They will be able to add new students to their class via an invite link, kick unauthorized users, assign tasks to their students and comment onto the provided assignment uploaded by the student.

Students will be able to join classes, leave classes, see tasks, upload assignments and comment to the assignments.

There will also be admins, who will see onto every class, will have any kind of privilege and will also be able to remove the user from the application for any potentially harmful behaviour.

## Operating environment

TBD

## Design and Implementation Constraints

Due to the lack of fast hardware and the lack of experience in implementation of AI and LLM, we won’t be implementing the mentioned chat bot.

# Use cases/System features

## Registration and authentication

### 3.1.1 Description and priority

Visitors will have an option to create a new account, or sign into their existing account. This feature is very important, as other features are dependent on this one.

## Creation of a new class

### Description and priority

Registered users will be able to create a new class to assign their students to and give them related tasks. This feature is important as well as the first one, from the same reason, this feature opens possibilities for more features.

### Action sequences

Registered user will provide a name for a class to differentiate it from the other classes. They will choose whether the class should be visible to public and joinable for others, or private. Both types of classes will include an invite link, which can be shared to other users to join the class. Creator can also specify the maximum number of students to join the class.

## Joining a class

### Description and priority

Registered user will be able to join, either their teacher’s class via an invite link or any public class.

## Assigning a task & completing a task

### Description and priority

Teacher will be able to assign their students an assignment (something related to the email communication). This will create a special chat between the teacher and the student.

Student will be notified of a new assignment, will be able to see the description and is ready to respond to the assignment. Teacher can see, what they have submitted afterwards.

This is the main feature of the entire project.

## Seeing and adjusting a profile

### Description and priority

Users will be able to see and adjust their profile (for example: they will be able to edit their profile picture, add a bio(graphy), or specify any additional information).

Inside their profile, there will be a table of classes, and assignments that need to be done or have already been submitted.

Classes will be available on a different page and assignments will be visible within the class. This feature is not so important.

# External Interface Requirements

## User interfaces

TBD

## Hardware interfaces

TBD

## Software interfaces

The entire web application will consist of frontend made from HTML, CSS, JS and Twig templates. The backend will be running on PHP with PDO library.

We will use MySQL database for storing any related data and PhpMyAdmin for maintaining the database.

TBD

## 4.4 Communications Interfaces

TBD