

# 1. Introduction

Our project is to help make a scheduling system for a school to allow students and parents to make appointments with a therapist. The scheduling system allows the children, parents/guardians, therapists, and school administration to access the website.

## 2. System Overview

### 2.1 Background

Start with a system overview from the perspective of a system administrator. What will he/she do with the system and why?

- The admin will use this system to help schedule meetings between a therapist and student/guardian
- The admin will use this to securely bridge the gap and allow these two groups to easily make appointments as well as manage these appointments
- The admin will verify and accept therapists when they first sign up for the scheduling website

Hardware and Software Requirements

- PC (windows or apple)
- Python with required libraries(flask, venv, required backend dependencies)
- React library (required frontend dependencies)
- Gmail account with 2 factor authentication
  - Create an app password (this will allow you to send automated emails from the account to users)

## 3. Administrative Procedures

### 3.1 Installation

1. Users must clone github in a terminal using:  
git clone <https://github.com/adams-x0/team3github.git>
2. Users should go to the frontend folder and npm install all dependencies mentioned in the npm\_requirements.txt file in the root directory of the repository. (Ensure you have npm installed)
3. Users should then go to the server directory
4. User should activate virtual environment, if user is mac, run source venvMac/bin/activate, if user is linux or windows, run source team3flaskvenv/Scripts/activate
5. Then install all dependencies in requirements.txt (Ensure you have python3 and pip installed)
6. Open two terminal windows, and navigate to frontend in one, and backend in the other
7. In the frontend terminal, run npm run dev, and in the backend run python3 [app.py](#)
8. Finally, navigate to your search engine and put <http://localhost:5173/login> in the url

### 3.2 Routine Tasks

Discuss any routine tasks that must be performed such as creating and

Maintaining user accounts.

- When there is a new admin that needs to be created, an admin will have to add the new admin through the database or use a python script
- If a user wants to change their password they will need to email an admin so that their password can be changed
  - Same with email needs to be changed through the database
- An admin will have to manually update a student account to a parent account if a student becomes of age.

### **3.3 Periodic Administration**

Discuss any tasks to be performed periodically such as system backups and the

Cleaning up user accounts.

- The admin has to go in and delete the accounts that aren't going to be used anymore
  - Currently no delete account button on the users side
- The system database should be manually backed up every week in order to prevent serious data loss or data
- Students and Therapists must remove all past appointments regardless of acceptance.

## **4. Troubleshooting**

### **4.1 Dealing with Error Messages and Failures**

Create logs to help developers understand and troubleshoot issues, monitor application performance.

Provide tips on how to deal with serious error messages and failures.  
fix bugs by deploying updated versions of the application that contain the bug fixes.

### **4.2 Known Bugs and Limitations**

Provide specific information (e.g., code location, description of the bug, why the bug could not be fixed) on any known bugs and/or limitations of the system. Present specifics in the context that they are likely to affect end- user and/or administrator tasks and activities. Discuss how to deal with these bugs and limitations. Note that points may be deducted if known bugs surface in your product demo but are not discussed in this section.

- Emails currently aren't sending reminder emails the day before
  - Send out reminder emails manually
  - Couldn't get the scheduler to work with the email system and be able to test it
- Currently can't change/modify appointment (not counting accepting and cancelling)
  - Will need to take emails from the email account and any inbound emails use those to change the necessary appointment information
- A user attempts to register using an email or phone number that must be unique but is already associated with an existing account

- Should alert user trying to register that email address already exists
- No way to tell difference between two children if they have the same name
  - Children are unlikely to have the same name
- When cancelling appointments, appointments are slow to remove themselves from the list of appointments, meaning you can spam cancel until the appointment is removed.
  - In the student and parent dashboard, likely caused by the get appointments endpoint being called late or the request taking a long time.