Student Advisor Ticketing System - SATS - Documentation

Table of Contents

[Background 3](#_Toc198209985)

[General Application Description 3](#_Toc198209986)

[How to Launch Test Build 3](#_Toc198209987)

[Linux Server General Information 4](#_Toc198209988)

[Future Implementation 4](#_Toc198209989)

# Background

During discussions with Nazneen Kanga, MSc and Heather Dunlop, we found that there were many issues with their current booking system. One main issue they faced was students booking appointments for reasons that they were unable to resolve. This caused a temporary hold on the timeslot and prevented other students from booking an appointment for that date/time. The application our team developed solves this issue, as well as many others, while sustaining the original layout of their current systems.

# General Application Description

The SATS application is an app used to streamline the process of students booking appointments with their advisors. The app first prompts the user (student) to get certain information ready like their name, student number, DC email, and their reason for visiting. They will then be prompted to enter their information individually as they proceed through each screen. Once they select their campus and program, they can select a date and time (and whether they want their meeting to be in person or online) and the program will dynamically match the program to the advisor and send a confirmation email to the advisor to verify if that time slot works or not for them. If the time slot works, they can accept the meeting and it will be added to their calendars but if declined, an email will be sent to the student informing them of the situation.

# How to Launch Test Build

Launching the Test Build is relatively simple. First you will need to use one of the current branches from the repository or create your own branch. Once you have a branch, open it in your preferred IDE (This project was developed on Visual Studio Code for a bit of background). Firstly, to download all the dependencies for the project, enter the command “npm install” in the terminal. Once that command is completed, there are 3 commands you need to put in each of the 3 terminals respectively but before that, you will need to install an emulator for an android device. There are 3 methods that work, and you can choose your preferred method. The first method is using bluestacks. Once you install bluestacks, you can then install the “Expo Go” app. Another method is installing Android Studio and using their android emulators to connect to the application. Once you have your preferred emulator, in the first terminal, you will need to type the command “npx expo start” in the root folder. Once that command is completed, navigate to the 2nd terminal and navigate to the services folder by using the command “cd services”. Once in the services folder, enter the command npx ts-node fetchProgramService.ts (Make sure you are in the services folder, or the command will not work). Finally, navigate to the 3rd terminal and enter the command “cd services” again. Once in the services folder, enter the command “npx ts-node authService.ts”. Once these 3 commands are running, you are ready to use the test build. If you choose to use bluestacks, open Expo Go and enter the exp ip address and wait for it to build. Similarly, if you choose to launch it on your personal mobile device, use the QR Code and follow the instructions to launch it. Lastly, if you’re using Android Studio, open the device manager and connect to a medium phone. Once the emulator is running, in the first terminal where you entered “npx expo start”, press “a”. This will start the application.

# Linux Server General Information

You need Docker and docker compose installed for further instructions.

In order to start the server, navigate to the GitHub repository with the SATS app, clone the repo, and after cloning the repo, start the “docker compose up -d”; you need to be in the same directory as docker-compose.yml (root directory of the project).

Currently, you need to install ZeroTier One on the server (and on any device interacting with the server and the tablet). Within ZeroTier One website, create a network (or join preexisting one). On the server you must install ZeroTier One using the instructions provided on their website and configure the network using ZeroTier One-CLI. On the tablet, you need to download the APK from their website, connect to the same network as the server, and make sure to enable it (you can verify that by observing the VPN icon showing at the top taskbar of your tablet). For any additional devices, repeat the steps above. Note that further down the line, you can connect to the server via SSH (including from home).

# Future Implementation

* New scraper that will differentiate Oshawa and Whitby Campus. The scraper will scrape the program webpage to match the program itself with the corresponding campus. Campuses can have many advisors and advisors can have many campuses.
* Dean requested an additional database/table to store relationship between programs and its campus.
* Prepared statement query that will match the name of the advisor to their calendar link
* Implement Caching for calendar page to cache the advisor’s calendar (to prevent overloading the system with requests for the calendar)