Scenario/Problem Statement

My client, Mr Subrat Rout, owns two clinical pharmacies, partners with an orthopaedic surgeon, and employs about ten people. From my interview with him on 26/10/23, I gathered that he has a very hectic schedule. He has to manage everything manually, which leads to miscalculations and losses. It is also mentally tiring to manage a business with such a fast stock turnover.

The doctor's workload can also be quite demanding. The doctor checks up on almost 40-50 patients daily (Refer to Appendix A), and managing patient flow and providing personalised care can be strenuous. This workload leads to fatigue, making it challenging to maintain the highest level of attention for each patient.

The product is primarily being developed to automate some of the processes and reduce the workload of my client and the doctor, as those are his primary concerns.

Rationale for the proposed solution

Considering the problems my client mentioned in Appendix A, the product will be a web application, as it provides a standardised user experience regardless of the PC configuration.

The front end will be programmed using Vue, a framework of Javascript, along with PrimeVue, a component suite for Vue. These frameworks will provide the necessary props and functionality to meet the diverse requirements of the project.

The backend will be implemented using Django, a framework of Python offering security and scalability features which will prove to be integral for the development of this project.

The product will be deployed on two base URLs, one hosting the front end and one hosting the back end. Communication between the back end and front end will occur through HTTP requests via the Axios library and REST framework. Their compatibility with different data types will be crucial due to the project's varied requirements. WebSockets also were a viable option but could not be used to resolve version conflicts and incompatibility with other libraries.

Success Criteria

- Profiles for the Front Desk, Pharmacists, Doctors, and the Admin
- Only the admin profile can verify and remove employees and edit doctor specializations.
- Admin and pharmacy profiles can view and edit medicines and inventory, and lab tests.
- The front desk profile can book appointments for new and existing patients.
- The doctor profile can write a prescription for booked appointments.
- The doctor's profile allows access to the prescription at all times, and the pharmacy profile allows access for some time, abiding by medical rules and regulations.
- The pharmacy profile can dispense medicines and enter the required data for lab tests.
- Status trackers for prescribed medicines, lab tests and prescriptions.
- All required fields must be filled out before submitting; to inform the user otherwise appropriately.

- Sessions will automatically expire after a set period, requiring re-login.
- Structured way of generating tokens at the front desk, for an ordered way of writing prescriptions by the doctor.
- The system will calculate the total costs for an appointment based on consultations, lab tests, and medicines taken.
- The system will display clear error messages and success notifications for user actions to keep them informed.