A2 – Group Organisation Plan

Introduction

This brief document shall outline which responsibilities are delegated to which members of our group for our Software Development project: the Patient Interface written in Java.

It shall be structured under multiple headings – one representing each respective week of the timeline, beginning from our initial brainstorming of the manner in which to plan the project. Under each section shall be, in the form of bullet points, outlined our thought process behind plans for the future, as well as any cornerstone features we implemented or tasks we performed throughout that very week. We will also include who will perform what task and lead which feature / DB design / test suite / other in the future.

Key sources of data leveraged for the construction of this document include observing git commits and PRs to main branch, reading discord chatlogs, and analysing library room booking confirmation emails.

Every Week

- Discussions on discord regarding a plethora of crucial elements to consider both before, during, and after the development of various features. All members of the group have partaken in these discussions at various points in the development process thus far. These discussions entailed some of the following points:
    - Consideration around entity field types for TOTP / SMS / Email notifications.
    - Enforcing conventional git commits, protected branches with enforced PR to main, consistent formatting project-wide, etc.
    - Ensuring complete test cases for as many aspects of the program as possible.
    - Planning authentication with decorated-based validation before making any call, as well as JavaFX-Screen-level calls to a singleton session manager to ensure security throughout the application.
    - Discussions regarding the optimal structure of the project for both scalability and maintainability, whilst minimizing the learning curve for all group members.
    - Structure of GUI, i.e., breaking down large chunks of the skeleton of each page into individual bones (components), for ease of maintainability and to expedite development.
    - Discussions regarding who is responsible for which feature, scheduling deadlines, and ensuring the whole group is involved in both the decision-making and the development process and are content with their roles.
    - Ensuring everybody comprehends the tools they are expected to use to fulfil their roles, all the while providing explanations on certain incongruencies such as how to successfully install all dependencies and run the code despite some group members wielding different IDEs (integrated development environments), OS versions, etc.

Week 1 (starting Monday, 26th February 2024)

- **[2024.02.28] [Full Group]** Held a partial group meetup (three members) in the library, to brainstorm project ideas and begin developing the project structure and considering the

entity relationship diagram. Additionally, implemented some of the DB's to-be entities (Address, Doctor, Patient) in code. Initial commits were made to the project's GitLab repository.

- **[2024.02.28] [Josh]** Migrated from GitHub to GitLab (as per brief specification), constructed pipelines for building & testing the code on push to GitLab, and begun configuring logging & authentication. Created multiple managers exported as singletons for ease of project-wide integration.
- **[2024.02.28] [Edwin]** Delegated the task (to this group member) of producing a lo-fi interface design (for a desktop app) by the end of the week, so that we may begin building the GUI interface programmatically using JavaFX. Tool advocated for and utilized was Figma.
- **[2024.02.29] [Josh]** Improved structure of the project and implemented ORM (Hibernate) for trivial development process and mitigation of code duplication (e.g., rewriting SQL queries & writing a ton of getter and setter methods for each entity). Introduced distributions in the form of artifacts after every commit (later to be post PR), for the sake of fullness. Enforced conventional commits, integrated CSS files, refactored some code, and begun test class implementation.
- **[2024.02.29] [Scott]** Finished first basic draft of the ERD (entity relationship diagram) using an online diagram creator tool "Lucid Chart".

Week 2 (starting Monday, 4th March 2024)

- **[2024.03.01 – 2024.03.05] [Josh]** Added more tests to the test suite, improve CI, improve configs, refactor, enforce testing code before commit succeeds, change artifacts and Gradle target of dist. Remove code duplication through Lombok @Data, and login logic, and improve robustness of test cases.
- **[2024.03.06] [Full Group]** Meetup after Software Development seminar in library with the whole 5-membered group, to get everyone on the same page and fix any remaining errors at runtime regarding inadequate dependencies / insufficient versions. Brainstorm and improve ERD (adding the remaining entities after considering the brief), with proper types, committing the changes respectively. Teach group members git branches and pull requests, to ensure expedited development in the future. Also commit lo-fi model, ERM, and more to the repository, for the sake of fullness, centralization, and redundancy. Delegate tasks to various group members, for timely Sprint 1 project submission the following week.
- **[2024.03.07] [Vit]** Revamp the login screen with good styling and formatting (truer to the lo-fi diagram produced by Edwin), breaking down GUI elements into individual bite-sized components, one per file respectively. Add a validator (following Josh's initial example) for 2 fields in the Insurance entity, and 1 field in the Booking entity, with appropriate tests in place.
- **[2024.03.07] [Josh]** Integrate 2FA backend, with proper validators (processors and annotations) in place for certain entity fields, with tests in place.
- **[2024.03.08] [Vit]** Further improve the styling of the login page, creating the foundation for all other pages. Apply small fixes, including accommodating both java and pdf files for commits instead of disallowing them.
- **[2024.03.08 – 2024.03.10] [Josh]** Implement event listeners & respective event handling, as well as various performance fixes. Improve upon some of the login page styling, and create even more events, with appropriate testing for both.

Week 3 (starting Monday, 11th March 2024)

- **[2024.03.11] [Josh]** Write the Testing Specification Documentation in full, for the entire test suite (near 100% coverage of the program as of this date).
- **[2024.03.11] [Vit]** Write the Group Organisation Plan document in full, somewhat retrospectively.
- **[2024.03.11] SPRINT #1 DUE TODAY:** Upload project as zip with Testing Specification Documentation and Group Organisation Plan document included. Also include ERD and lo-fi prototypes in the zip file. Ensure it can be run on the assessor's computer without a hitch, by testing prior to upload, and including concise instructions on how to install dependencies and run the code.

Week 4 (starting Monday, 18th March 2024)

- **[2024.03.18] [Vit]** Expected to have finished the session manager implementation, as well as the home screen page, for visual confirmation (not just backend) that the user is successfully logged in. Additionally, implement GUIs and backend logic for signing up and resetting one's password.
- **[2024.03.18] [Vit]** Register user functionality (both backend and GUI), with appropriate error / success feedback popups.
- **[2024.03.18] [Vit]** The system should allow a new user to register as a patient, choose a doctor from the list of all doctors. The system should then send confirmation messages to the patient and the doctor.
- **[2024.03.19] [Scott]** The system should allow a patient to change their doctor using the list of all doctors. The system should then send confirmation messages to the patient and the doctor.
- **[2024.03.20] [Rhys]** The system should allow a patient to arrange a booking with their doctor by entering the date and time. If the doctor is not available for the chosen date and time, the system should warn the patient. Otherwise, the system should then send confirmation messages to the patient and the doctor.
- **[2024.03.21] [Edwin]** The system should allow a patient to view their bookings by entering a month and year.
- **[2024.03.22] SPRINT #2 DUE TODAY:** Upload project as zip with all the appropriate files included. This includes implementing four extra pieces of functionality, ensuring code adheres to standard conventions and is commented, and JUnit tests all passed, with a 2-page test specifications document.

Week 5 (starting Monday, 25th March 2024)

- **[2024.03.28] [Josh]** The system should allow a patient to reschedule a booking with their doctor by entering the date and time. If the doctor is not available for the chosen date and time, the system should warn the patient. Otherwise, the system should then send confirmation messages to the patient and the doctor.
- **[2024.03.28] [Vit]** The system should allow a patient to view the visit details regarding a past booking, for which the doctor provided visit details and prescriptions.
- **[2024.03.29] [Scott]** The system should allow a patient to view all doctors with their summary information, e.g. name, phone number.

- **[2024.03.30] [Rhys]** The system should allow a patient to view a doctor's details, e.g. name, phone number, background, along with the doctor's availability for bookings in a given month and year.

Week 6 (starting Monday, 1st April 2024)

- **[2024.04.1 – 2024.04.07] [Full Group]:** The entire group should be involved in the final stretch of the sprint, to brainstorm any final changes that need to be made in accordance with the brief's business rules & specification. Additionally, any refactoring, GUI adjustments, or backend code cleanup can be done in this period of time, all the while ensuring that all tests pass and the program runs smoothly.
- **[2024.04.07] SPRINT #3 DUE TODAY:** Upload project as zip with all the appropriate files included. This includes implementing authorisation checks & 3 extra pieces of functionality, properly formatted & commented code, Junit test classes with all tests passed, 2-page test spec document explaining the tests, database design document, quality assurance document, and finally a video demonstration.