**Project Report**

**Group WED-16.30-6**

VERDOUW, Midori (s3575912)

PHAM, Van (s3788106)

TJIONG, Julian (s3786866)

CHEONG, Hon Khuin Jonathan (s3642842)

SONG, Jason (s3744335)

**Vision**

a "Vision" statement, i.e., a description of the why the product is "valuable" (about 1/2 page)

**System Architecture/Design**

\*a figure showing the final basic system architecture / design --- this does not need to be a UML diagram: a box-and-lines sketch architecture is fine; focus on the 3 tiers (front, backend, datastore) and what components are in each

For the three tiers (frontend, backend, datastore) we have used React, Spring Boot and MySQL respectively. The React application uses Redux for the state management, which involves elements such as actions, reducers, store, and the view (components and containers). The front end sends API requests to the back end for getting all business schedules, for adding new person (i.e., worker and customer), for authenticating user inputs for login, etc. The back end provides such APIs as the Spring Boot REST controllers, which are supported by other layers of the back-end application, namely, services, repositories and models. Our application has models such as Person, WorkerSchedule, and Business, each of which correspond to a table in the database.

Diagram

Description automatically generated

**Refactoring**

* a report describing any refactoring, including refactoring to a Design Pattern, if appropriate, as well as the "smells" that you found (i.e., purpose for refactoring)

**Gitflow Organisation**

\*an overview of your Gitflow organisation and how often you committed

We have used the following types of branches:

* master
* deployment
* feature (e.g., feature/US#1\_customer)
* release (e.g., release/milestone2)

Each feature branch is used to implement one user story and is active only during implementation. It is based from the development and merged into development upon the completion of the implementation of the feature. A special feature branch named “feature/docker\_configuration” exists only for preserving the docker configuration needed for demo and deployment. The release branch is created based from the development when all the user stories for the milestone finished and their branches are merged.

Although we have had a short period of time (approximately 7 days) at which no commit was made, overall we have committed regularly throughout our development since Sprint 1. At the time of writing we have 247 commits in total in the development branch.

**Scrum Process**

a description of your Scrum process: how often you met, who was Scrum Master

**Deployment Pipeline**

a diagram of your Deployment pipeline setup: i.e,. CI/CD, steps in the pipeline, automation tools in the pipeline

**Testing Documentation**

documentation of acceptance test cases and evidence of test execution (when they were run; pass/fail status of each run)