



Property Management System

Course Completion Group
Project: **WAA**

Property Management Portal

Content

- Team Members
- Introduction
- Environment Setup
- Best Practice
- Satisfied Requirements
- Highlight of Backend Architecture

Team Contributors

<i>Bun Thai Deng</i>	<i>ID: 617284</i>
<i>Hebron Melles</i>	<i>ID: 115609</i>
<i>Samuel Berhane</i>	<i>ID: 114960</i>
<i>Vichea Nath</i>	<i>ID: 616942</i>
<i>Yonatan Tadesse</i>	<i>ID: - 116055</i>

Introduction

- PoC for Property Management Portal
- Objectives
 - Getting hands-on experience building a full-stack project with **Spring & React** technologies
 - **Teamwork**
 - **Industry working culture**

Environment setup

- **GitHub**: Version Control & Collaboration
- **Agile + Kanban**: Addressing issues & task distribution
- Foundation layout
 - Frontend by **Vichea** & Backend by **Thai**
- Everyone was engaged with specific task - **Full-stack** implementation in mind

Best Practices

React

- Folder Structure & Component-Based Architecture
- State Management
- Immutability – **used immutable data structure and libraries**
- Error handling & Performance optimization – **used memoization**
- Pagination & Routing

Spring Boot

- Project Structure – **MVC** architecture
- Usage of Annotations - configure **beans**, handle **requests**, define **mappings**, manage **transactions**, and **validation**
- Dependency management – **Maven**
- Configuration Management – application.yml
- **Dependency Injection** – constructor injection
- **RESTful** API's – used best practices
- Exception Handling & Security (authentication & authorization)
- Performance optimization – minimized database queries
- Database Access: **Spring Data JPA**
- **Logging** – **Log4j** & **Docker** configuration

Satisfied features requirements

- **Login (as Admin/Owner/Customer)**
- **A user can be Registered**
 - **As Customer**
 - **As Owner (needs Admin Approval & must own at least a property)**
- **Reset Password (Customers & Owners both can reset their respective passwords)**

... Continued

- Profile & Dashboard (**Admins & Owners** own their respective profiles & dashboards)
- Offer history -
Customers can view their offer history (including offer statuses)
- Favorite **Customer** can keep own favorites with delete options
- Property view, search, filter (by Type, Price, NoOfRoom) & Sorting by

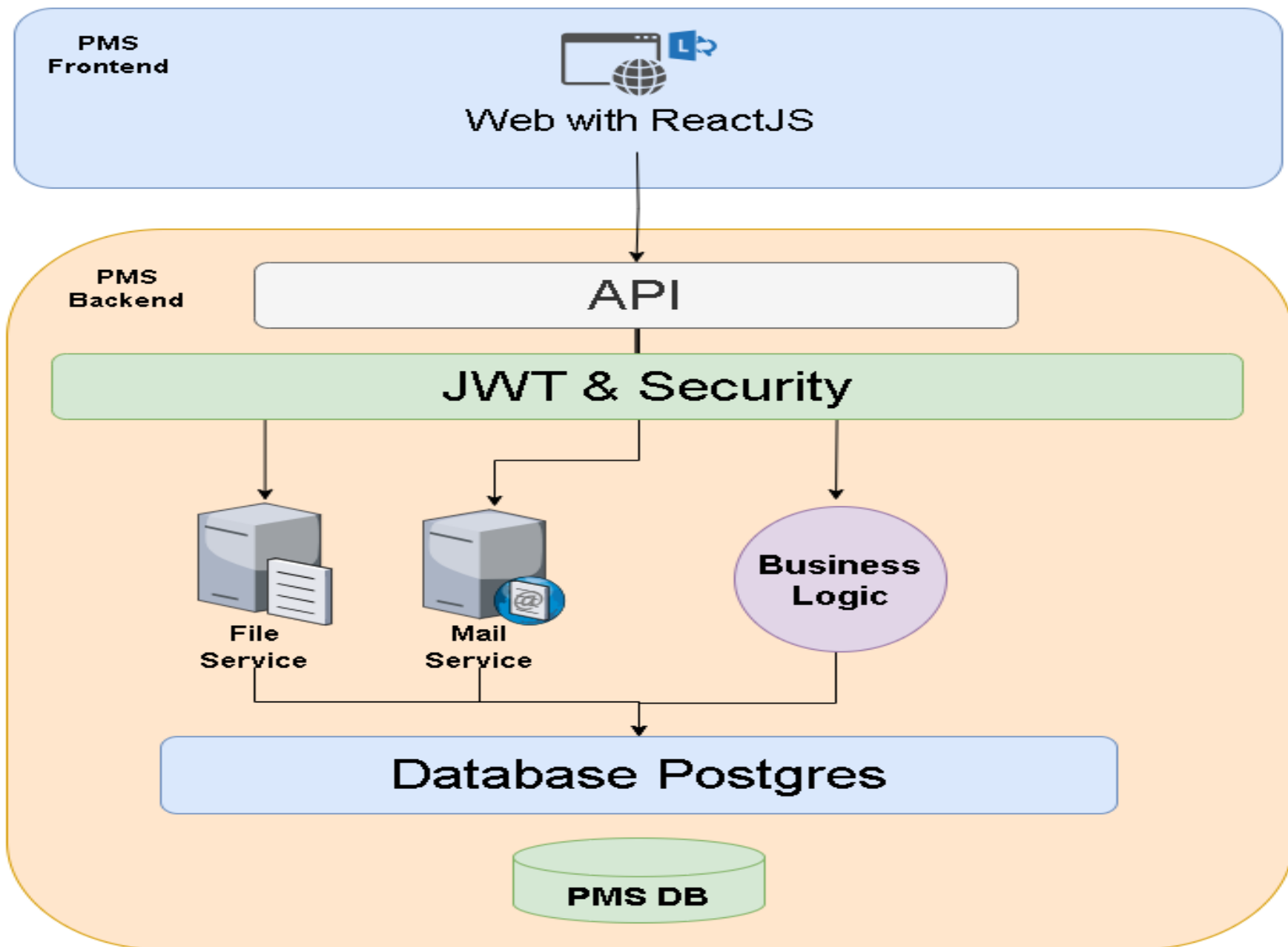
... Continued

- Save & Unsaved properties **Customer**
- Property detailed view by **Users**
- Submit, Cancel (status not Contingent) & view Offer - **Customer**
- Accept, Reject & Cancel (even a Contingent status) offer - **Owner**
- Add property (multiple images upload) & Delete - **Owner**
- Messaging – Email Messaging
- Latitude & Longitude: GIS

Backend Architecture

Backend Architecture

- JWT and Spring Security
- File service API
- Mail Service API and Internal
- Open API (Swagger)





Let's See Project Environment