# **Project Midterm Update Report**

## **1. Abstract**

ScubaWarehouse, a leading diving gear retailer in Singapore, faces ongoing challenges in hiring experienced sales personnel due to the specialized nature of the industry. At the current midterm stage, significant progress has been achieved on the Scuba Warehouse Sales Assistant App, designed as an internal support tool for sales representatives. The foundational setup for frontend, backend, and database integration was successfully established and tested on the web, ensuring compatibility and initial functionality. The project subsequently transitioned into an MVP for iPad, validating core functionalities such as product filtering, competitor price scraping, and user interaction workflows with test data. Currently, our team is expanding product categories to include BCD and fins, alongside implementing a robust product comparison feature. This app continues to aim at enhancing customer satisfaction, improving sales efficiency, and supporting the company's strategic focus on an exceptional in-store retail experience.

## **2. Project Overview**

The Scuba Warehouse Sales Assistant App aims to empower in-store sales representatives by providing tools for effective and informed customer interactions. Central objectives of the app include streamlined product selection through customer-driven filters such as budget and product specifications, comprehensive product comparison features, real-time competitor price tracking, and a sophisticated yet intuitive recommendation system tailored to diving experience levels. The architecture leverages React Native for frontend development, Node.js for backend operations, and Firebase as a real-time database solution. Additionally, the project employs strategic API integrations and custom web scraping modules for reliable real-time price comparison, clearly aligning with ScubaWarehouse's emphasis on delivering personalized, high-quality retail service.

## **3. Architecture**

### **Scuba Diving Gear App Diagram**

A diagram of a software development

AI-generated content may be incorrect.

### **Briefing**

The **Frontend** is built using **React Native**, serving as the primary interaction point for sales representatives. It includes the **Product Selection UI**, where users filter scuba gear by customer-specific criteria. Once a product is selected, the **Product Details View** redirects seamlessly to the **Existing E-commerce** platform to display **Detailed Specs**, avoiding redundancy. The **Comparison UI** clearly illustrates differences between multiple products, supporting informed decision-making.

An optional enhancement, **Intelligent Search**, provides advanced capabilities to interpret summarized customer preferences or natural language inputs, potentially even from voice recordings or direct salesperson input. Complementing this, another optional module, **Product Recommendations**, delivers personalized product suggestions based on predefined user profiles such as beginner, intermediate, or advanced diver experience levels.

The core application logic is orchestrated by the **API Layer (Backend)**, powered by Node.js. Centralized within this layer is the **Business Logic Processing** module, which acts as the main workflow controller, directing the overall data flow and processing tasks, including requests to databases and external services. The **API Management Layer** secures and manages these requests, handling authentication, validation, logging, and monitoring.

Data handling and storage occur primarily through the **Firebase DB**, optimized for real-time operations. It hosts the **ScubaProduct Gear Filter**, enabling quick retrieval and filtration of diving products. Additionally, Firebase manages **Realtime & Historical Prices**, capturing both current and historical price data for efficient and reliable comparisons.

The **External Price Comparison** module ensures competitive pricing by integrating external data. Its **Web Scraper**collects competitor prices regularly or upon request, maintaining accurate market insights. These results feed into a **Real-Time Sync** process, delivering updated competitor pricing directly back to the frontend to enhance sales effectiveness and customer confidence.

## **4. Current Progress**

### **Sprint 1: Foundation Setup to test run on web (10/02/2025 - 24/02/2025) (done)**

**Objective:** Establish the initial architecture for the application, ensuring web compatibility and foundational UI functionality.

• **Frontend (React):** React Native environment initialized, Firebase set up, and web compatibility ensured  
• **Backend (Node.js):** API layer established with basic routes for product retrieval and filtering  
• **Database (Firebase):** Minimal database setup to build connection between the product and Google Firebase  
• **Product UI Prototype:** Initial design of product listing and filtering page for web application  
• **Feature Filter:** Implemented minimum UI for product gallery and filtering  
• **Product Research:** Identified and summarized diving gear categories for structured filtering  
• **Data Gathering:** Researched product specifications to support sales team understanding  
• **Basic Customer Interaction Mapping:** Established the workflow for in-store salesperson interactions  
• **Performance Optimization:** Addressed slow queries in Firebase and optimized filtering logic  
• **Initial Testing:** Verified basic UI functionality on web for product selection and filtering  
• **Sales Workflow Understanding:** Sales team gathered product specifications for better recommendations  
• **Environment Setup:** Resolved dependencies and ensured a stable development environment

### **Sprint 2: MVP Development for iPad (24/02/2025 - 09/03/2025) (done)**

**Objective:** Transition the MVP to iPad, validate the filtering system using test data, and begin integrating pricing comparison features.

• **Frontend (React):** Basic UI running on iPad, including product gallery and filtering  
• **Testing:** Conducted on 3 regulator products using dummy data  
• **Backend (Node.js):** Integrated product filtering API with refined logic for efficiency  
• **Web Scraping:** Built prototype scraper for competitor pricing data  
• **Database (Firebase):** Refined schema for improved real-time syncing and query performance  
• **Price Matching:** Prototype integration of external product listings for dynamic price comparison  
• **Product Selection UI:** Implemented basic selection interface for regulators  
• **Data Validation:** Verified dummy data accuracy for testing product flow  
• **Bug Fixes & Debugging:** Addressed UI inconsistencies and initial performance bottlenecks  
• **Customer Interaction Flow Testing:** Verified app usability for in-store sales interactions  
• **User Navigation Enhancements:** Improved product browsing and filtering UI based on feedback  
• **Comparison Feature Planning:** Explored methods for adding product comparison features in future sprints  
• **Design Adjustments for iPad:** Optimized the layout and performance for tablet use

### **Sprint 3: Product Expansion and Comparison (10/03/2025 - 23/03/2025) (in progress)**

**Objective:** Expand the product category to include BCD and Fin categories and implement product comparison functionality.

• **UI Design:** Developed interfaces for BCD and Fin product categories  
• **Filtering & Selection:** Extended filtering and selection logic for BCD and Fin  
• **Comparison Feature:** Implemented drag-and-drop product comparison functionality  
• **Data Fetching:** Integrated additional product details for BCD and Fin  
• **Comparison Logic:** Developed specification-based comparison for different products  
• **Performance Testing:** Ensured stability of filtering and comparison modules on iPad

### **Sprint Retrospective**

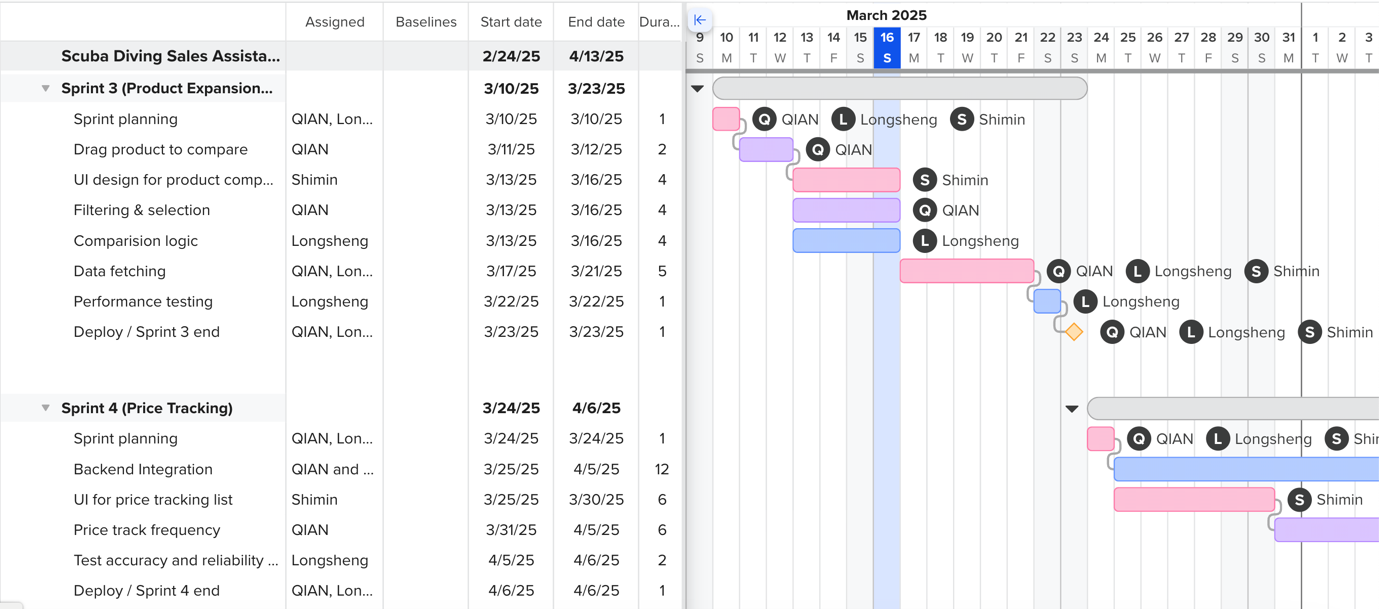
|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Start** | **Stop** | **Continue** |
| **Sprint 1** | Focus on ONE product categories: **Regulators** | Stop doing web application interface (focus on iPad) | Exploring AI to generate filters if we can scrape all product details |
| **Sprint 2** | Testing 3 regulators based on dummy data, ensuring basic UI on iPad as a MVP | Stop design detailed UI by redirecting to existing product page in scuba warehouse webpage | Exploring AI to generate filters if we can scrape all product details |
| **Sprint 3** | Expanding to BCD and Fin product categories and comparison UI | To review after Sprint 3 | To review after Sprint 3 |

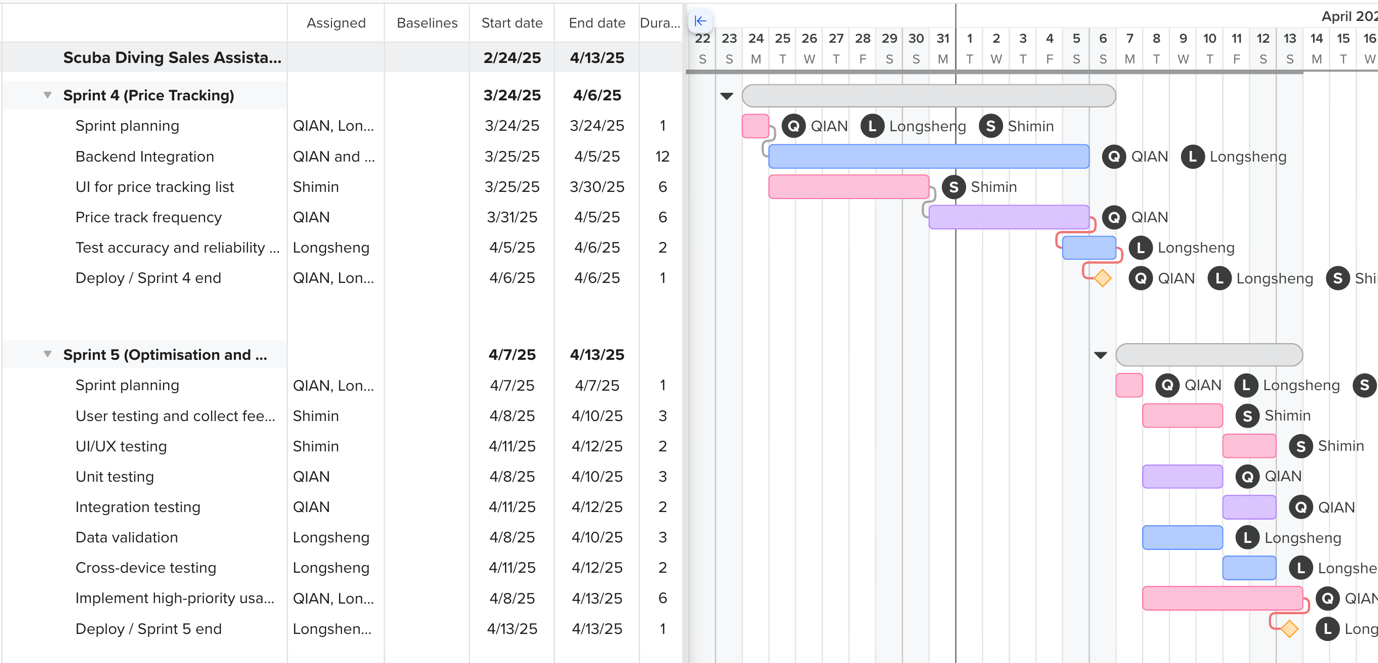
## **5. Plan for Remaining Semester**

### **Remaining Tasks:**

1. **Complete Compare Function**: Enable users to compare selected products.
2. **Complete Price Tracking**: Implement real-time price tracking using external data sources.
3. **Optimising**: Improve UI, Add features based on user feedback

### **Gantt Chart**





## **6. Individual Contributions**

|  |  |
| --- | --- |
| **Team Member** | **Contribution** |
| Zhang Longsheng | Project lead, external API integrations, competitor pricing API implementation, debug, Firebase database management |
| Chen Shimin | UI/UX design, frontend architecture and development, user interaction design, sprint coordination and project management, Database schema design |
| Chen Qian | Architecture Design, backend logic orchestration, Backend API development, ios application development |