Project Description:

AquaAligned is a real-time water monitoring system that uses ultrasonic and temperature sensors to track tank temperature and water levels. It sends data to a web dashboard showing usage statistics and trends, helping users reduce waste and manage water more efficiently.

Requirements Summary:

Key functional requirements of AquaAligned include:

- **Real-time data logging:** The system must capture and store each water usage or refill event in a central database.
- **User authentication:** Only registered users can log in to view or modify logs.
- **Data visualization:** Weekly water usage must be presented in a chart format.
- **Log filtering:** Users should be able to filter logs by date range and activity type.
- **Sync feedback:** The system should show whether it is currently connected to the Firebase server and when the last data was received.

Non-functional requirements include:

- **Responsive UI:** The interface should work across desktop and mobile devices.
- **Scalability:** The backend should be able to handle increased log frequency without performance drops.
- **Security:** Authentication must be implemented using Firebase Authentication.
- **Data consistency:** Usage data must reflect accurately in both logs and visualizations.

Design Space:

What requirements may be difficult to realize?

- **Real-time Alerts Based on Thresholds**
  Implementing a system that accurately alerts users when the water level drops below a user-configured threshold (in percentage or liters) is technically complex. It requires continuous monitoring of live data, syncing settings between the interface and the Realtime Database, and triggering alerts reliably.
- **Reliable Firebase Sync Feedback**
  Accurately determining whether the app is actively connected to database or whether the

system has stalled is non-trivial.

- **Manual Control vs Automation**
  Allowing users to manually override sensor-reported levels (e.g., update tank level or settings) can conflict with automated sensor data streams. Ensuring that both systems stay in sync without overwriting each other is a challenge.

What are some tradeoffs that you did explore?

- **Real-Time Sync vs. Simplicity**
  Using a real-time database gives the user instant updates, but increases complexity and potential resource usage. A simpler polling model (refreshing every few minutes) is easier to implement but less responsive. The current system uses real-time listeners for tank level and periodic querying for log data.

- **Interface Complexity vs. Usability**
  We avoided adding too many controls (like advanced filters or chart customizations) in favor of a clean, fast interface with essential settings and insights only.

Which tasks will be easiest to support? Which are the hardest?

Easiest:

- Viewing recent logs and weekly usage
- Adding new usage/refill entries

Hardest:

- Implementing real time updates and data sync to the water tank
- Accurate sync status detection and feedback for the user

Design Summary:

Rejected Alternatives:

- **Design 1: Single Tabbed Interface**
  - Singular page for all data
  - *Rejected* because users wanted more organised data views, which would need more tabular structure to have different data on different tabs.

- **Design 2: Mobile-First Single View**
    - A collapsible layout optimized for mobile
    - *Rejected* because desktop users (a primary audience for dashboard monitoring) found it cluttered and inefficient.

Chosen Design: Multi-tabbed layout

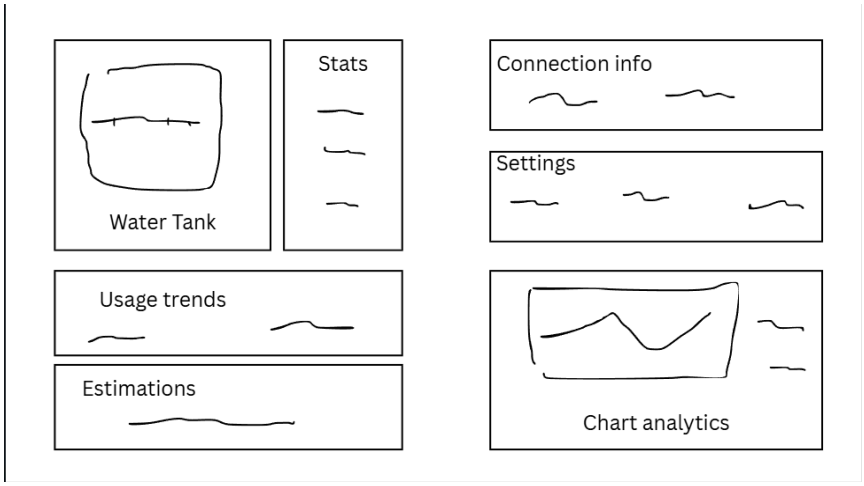
- **Sidebar Navigation**: Allows quick switching between Dashboard and Logs
- **Multiple tabs**: Organized data between the real time tank and the historical data of the chart and logs for easier understanding.

The Designs:

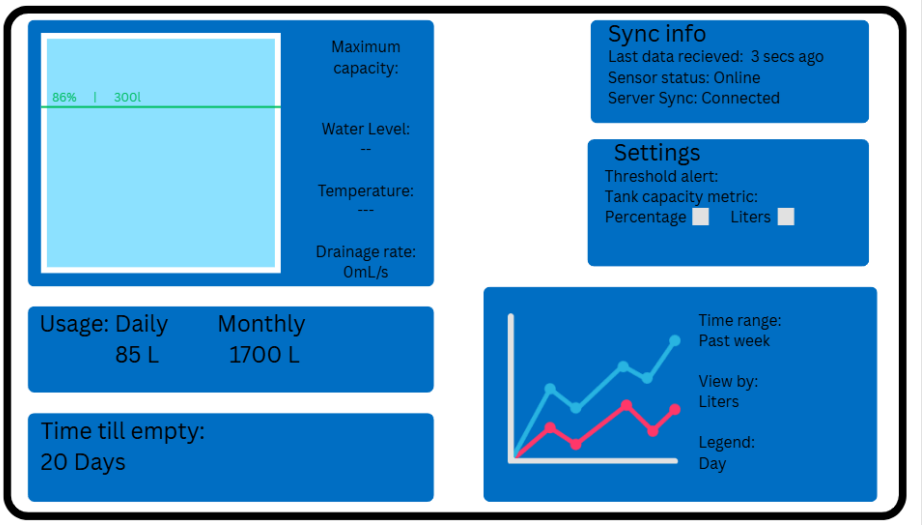
Design 1:

Story board:

Sketch:

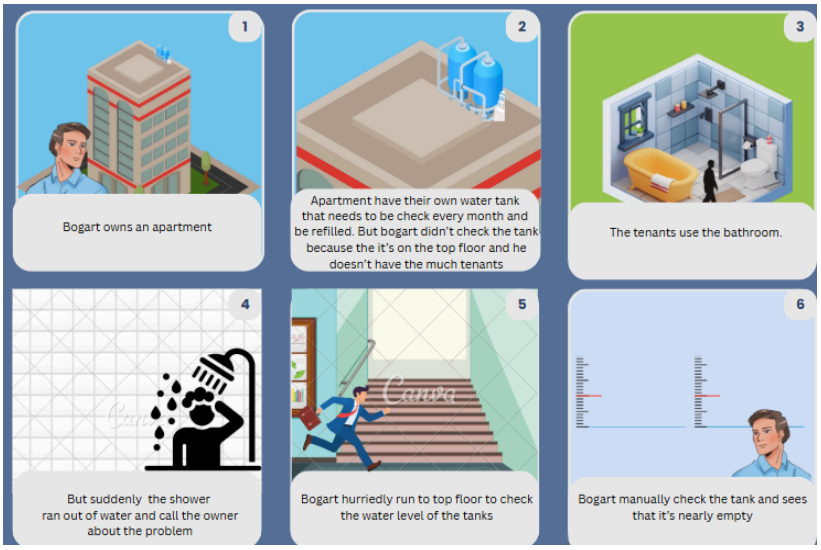


Mockup:

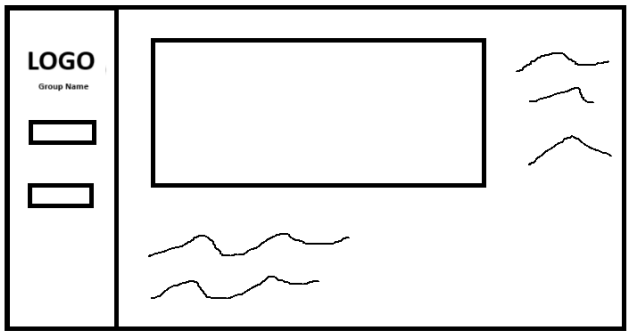


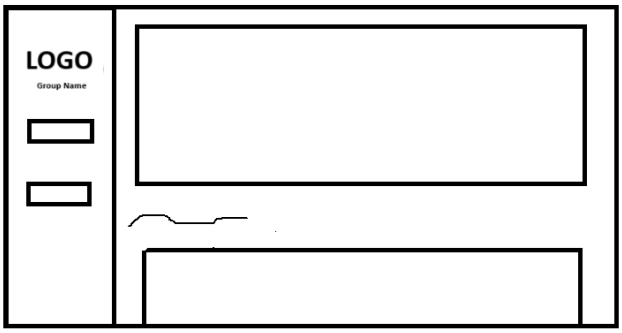
Design 2:

Storyboard:

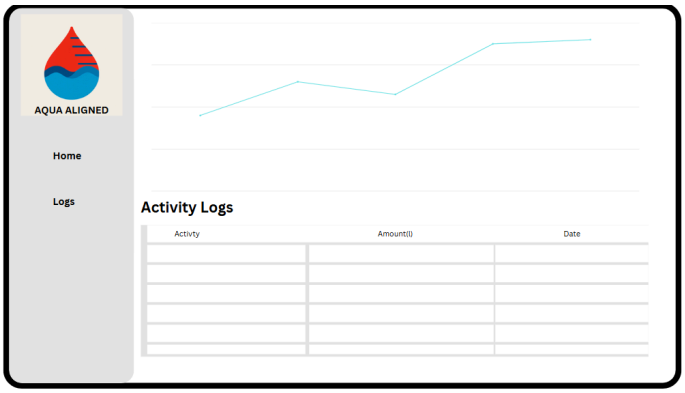


Sketch:





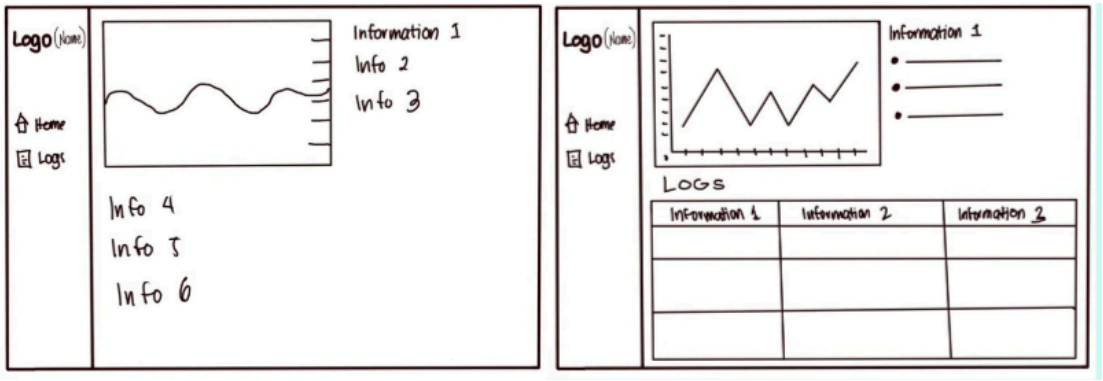
Mockup:



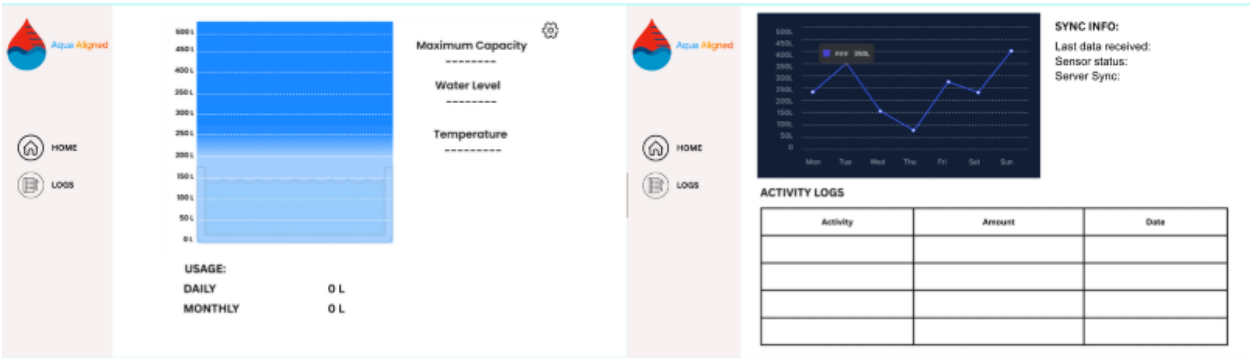
Design 3:

Storyboard:

Sketch:



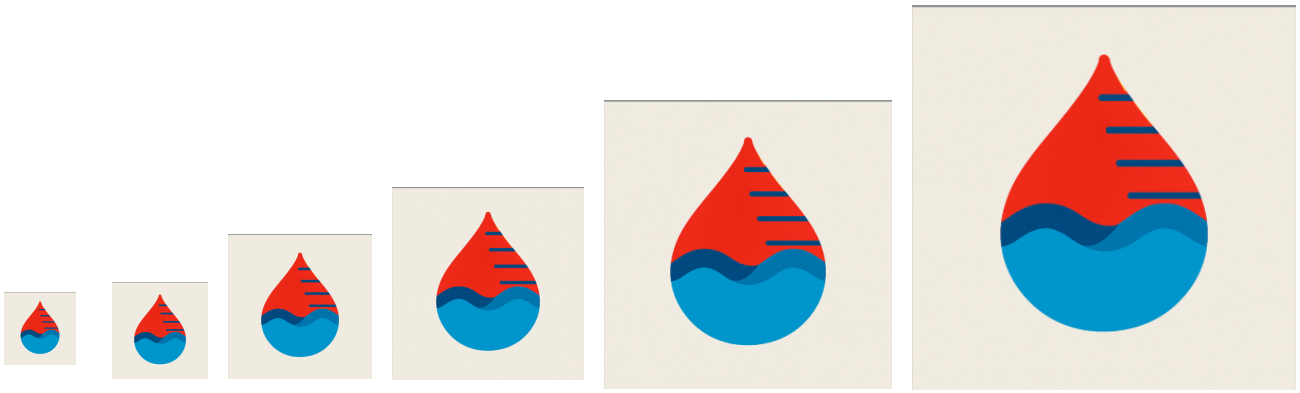
Mockup:



The focus of the first design is to have all the relevant data easily accessible. A water tank and stats to show real time data, usage trends to see the past data for analysis, and a chart to visually display the history of usage for the week for more analysis. Design 2 and 3 goes beyond it and goes for a tabular approach which separates the water tank and logs for more organized viewing.

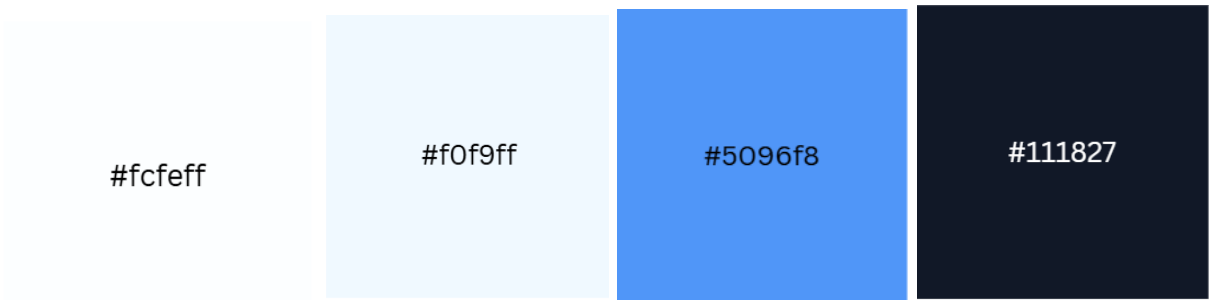
Current Design

System Logo:



36x36 48x48 72x72 96X96 144X144 192x192

Color Palette



The current chosen color palette was chosen for the connection to water and to have a soft and smooth color that embodies professionalism and organization. The darker color was used for the chart to have a visual contrast with the rest of the system.

Font style

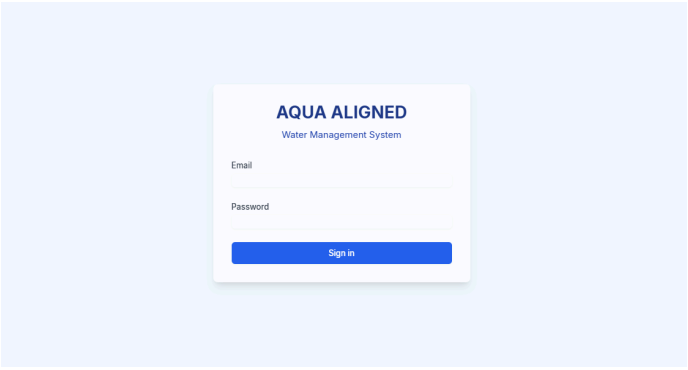
Windows: Segoe UI

Mobile: Roboto

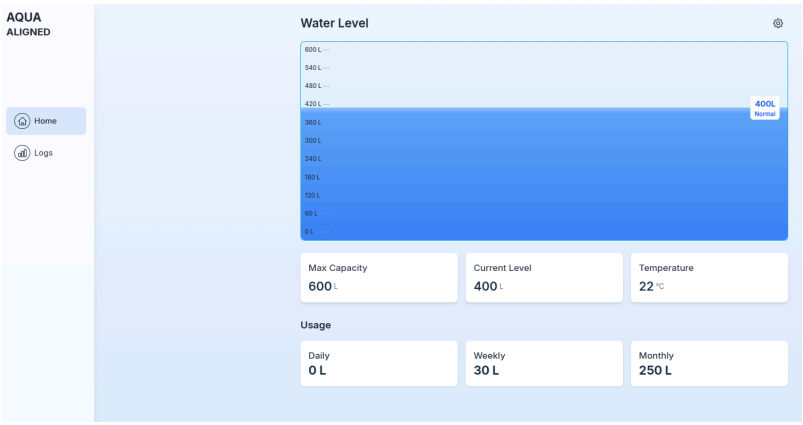
The system uses the default Talwind CSS font stack of Segoe UI for windows, Roboto for Mobile and San Francisco on macOS. This is due to this beta focusing on more functionalities. The UI font will be updated in future versions.

GUI and flow

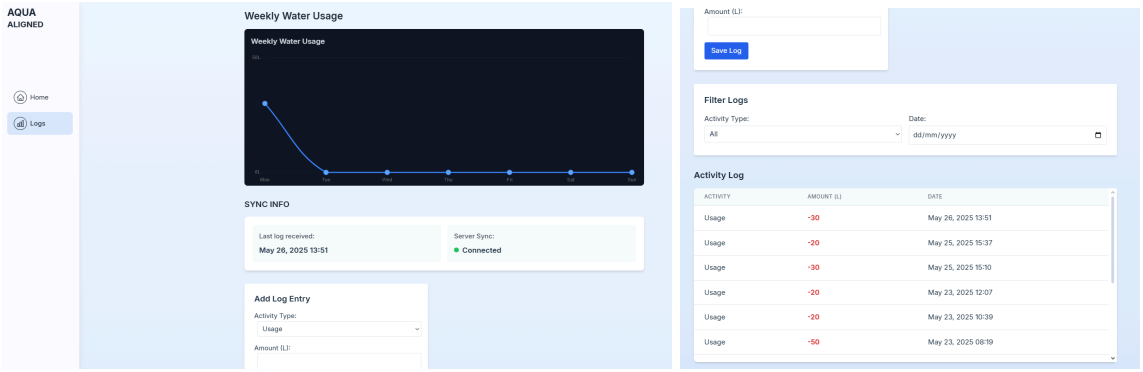
1.) Login - The Landing page of the system, and where users will start out.



2.) Dashboard - After logging in users will go immediately to the dashboard for the the real time water tank.



3.) Logs - From the dashboard the users can switch to the logs to see the logs, charts, and sync status.



Requirements changes:

During the design and development process of **AquaAligned**, several changes and additions were made to the initial requirements to better align with practical considerations, user feedback, and technical constraints.

1. Addition of Log Filtering

Original Requirement: The system should display all usage and refill logs in a list.
Revised Requirement: Users should be able to filter logs by **date range** and **activity type** (Usage or Refill).

Why it changed: During early usability testing, it became clear that viewing unfiltered logs—especially as data volume grew—was overwhelming. Users requested the ability to find specific logs or focus only on "Usage" or "Refill" data, leading to the inclusion of filtering features.

2. Clarified Real-Time Sync Feedback

Original Requirement: The system should sync with the server.
Revised Requirement: The UI must clearly indicate **whether the system is connected** to Firebase and **when the last log was received**.

Why it changed: Developers and early testers noted that it was hard to tell whether the system was working correctly or just delayed. To increase transparency and user confidence, we added a sync status section with color-coded indicators and timestamps.

Original Requirement: Display a chart of water usage.
Revised Requirement: Display a **weekly chart**, broken down by day (Mon–Sun), showing only **Usage** logs (not Refills).

Why it changed: Users found it confusing to see refill amounts included in usage data visualizations. After discussions, we scoped the chart to reflect only daily water usage, making the insights more actionable and easier to interpret.