# 4 Design

## 4.1 Design Context

### 4.1.1 Broader Context

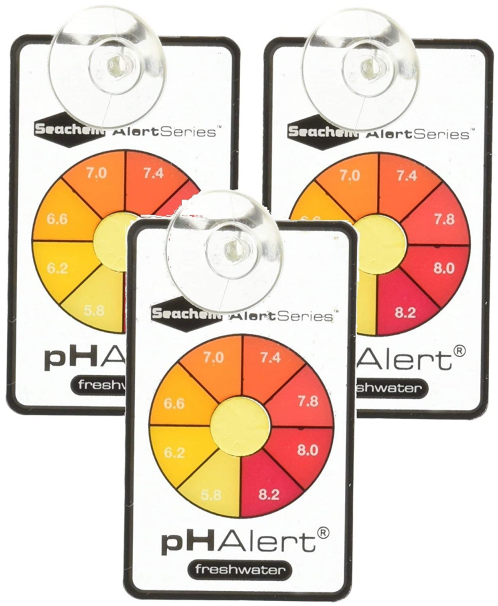
| Area | Description | Examples |
|---|---|---|
| Public health, safety, and welfare | Peace of mind -Our clients care a lot about these fish, our product will allow our clients to monitor the fish from anywhere to ensure they are healthy. Providing them with this peace of mind can help with ensuring the well being of our clients and their pets. | Increasing/reducing exposure to pollutants and other harmful substances, increasing/reducing safety risks, increasing/reducing job opportunities |
| Global, cultural, and social | The ISU office of Accounting keeps pet fish in their office in order to make it a welcoming and comfortable place to work. Our product reflects that in its features by creating an easy to use and visually appealing product. | Development or operation of the solution would violate a profession's code of ethics, implementation of the solution would require an undesired change in community practices |
| Environmental | The main goal of the product is to be able to monitor the well being of fish and the environment they live in. Our product ensures that the fish, which are living animals, are safe and taken care of. | Increasing/decreasing energy usage from nonrenewable sources, increasing/decreasing usage/production of non-recyclable materials |
| Economic | We are working to find the best components that are not too expensive so that our product may be an accessible option for many consumers.<br>Our product prevents overfeeding and health issues. This means that our clients will have to spend less on fish food as well as not replacing the fish due to casualties. | Product needs to remain affordable for target users, product creates or diminishes opportunities for economic advancement, high development cost creates risk for organization |

### 4.1.2 Prior Work/Solutions

**FISHNOSH Automatic Fish Feeder for Aquarium**

| | |
|---|---|
|  <br>[Amazon Link] | **Pros:**<br>• Works with multiple types of food<br>• Aesthetically pleasing<br>• Easy to use<br>• Fits most fish tanks<br>**Cons:**<br>• No wireless connectivity to phone<br>• Battery Powered<br>• Food amount set manually by opening/closing the window<br>• Feeding schedule not fully customizable<br>• Does not include PH/ Temperature Sensor |

**Seachem pH Alert Devices**

| | |
|---|---|
|  | **Pros:**<br>• Very simple to use<br>• Cheap<br>**Cons:**<br>• Must be in the vicinity of the tank to know if PH is bad<br>• Only lasts 3-6 months<br>• Not customizable to different fish species<br>• Takes a lot of time to change (according to customer reviews)<br>• Can be inaccurate<br>• One time use, cannot be moved from tank to tank |

### 4.1.3 Technical Complexity

1. The design consists of multiple components/subsystems that each utilize distinct scientific, mathematical, or engineering principles –AND–

   The systems that we will have are as follows

   - React Frontend
     - Able to send REST calls to send and receive data
     - Written in React Native
   - Cloud-Based Backend
     - Sends calls to both the frontend and firmware
     - Stores data in a database
     - Deployed on Google Firebase
   - Firmware
     - Controls hardware sensors to properly obtain measurements and feed fish
     - Simple operating system to control hardware interface with backend
     - Scheduler to control device feeding and schedule backend updates
     - Non-volatile memory storage for many different device parameters and data
     - Connect to backend to send and receive requests through Wifi
     - Bluetooth connectivity for initial device setup
   - Hardware
     - pH Sensor
       - 0.1 pH Accuracy
     - Thermometer
       - 0.5 °F Accuracy
     - Motor
       - Gearing in order to achieve desired speed of the dispenser
     - Food enclosure
       - Able to store food and allow for easy dispensing
     - Tank Mounting Apparatus
       - Device is able to be secured to fish tank without requiring alterations to the tank itself
     - Custom Designed PCB
       - PCB is a custom design specifically made to fit in our device with the required parts

2. The problem scope contains multiple challenging requirements that match or exceed current solutions or industry standards.

   - User permissions in app prohibit monitoring/modification of unauthorized tanks
   - Data about fish tanks is stored and evaluated
   - The firmware and app are easy to connect to without advanced software knowledge
   - App will be user-friendly
   - Hardware will need to feed fish an accurate amount of food during each feeding
   - Backend, Frontend, and Firmware will follow good coding practices as well as IEEE standards

## 4.2 Design Exploration

### 4.2.1 Design Decisions

List key design decisions (at least three) that you have made or will need to make in relation to your proposed solution. These can include, but are not limited to, materials, subsystems, physical components, sensors/chips/devices, physical layout, features, etc. Describe why these decisions are important to project success.

- Frontend Design Decision - Language to use for app development
    - As the user-facing application for our project, we needed to ensure that the frontend application of our project is easy to use, fast, and visually appealing. As an integral part of the project, we needed to decide on a language that would make it easy for us to develop the application as well as provide a good user experience for our clients.
- Backend Design Decision - Cloud service for app
    - There are a couple important considerations for cloud service providers and it is important that we look into one that will fit our use cases. For instance, we need to make sure that the service is always available as well as able to handle the amount of data that it needs to handle
- Firmware/Hardware Design Decision - Base Hardware for Device
    - Choosing the main processing power for our device left us with a couple different options of moving forward. We were able to choose between creating a solution based off an off-the-shelf Raspberry Pi design or developing our own custom chip-down design for our product. This is an important decision to make due to this being the main procession power for our device and this device will also be completing the backend to hardware connection. The chip-down design ends up being a more customizable and cheaper solution allowing us to add in more functionalities than we otherwise would have. While the raspberry pi would be much more intuitive to implement, we are confident that we have the knowledge to use the chip-down design. For these reasons, we opted for the chip-down design.
    - *Enclosure Design* - Choosing our differing material to use for our design to account for cooling and being able to be used by the water left us looking into differing plastics to 3D print our enclosure. We also had to look into the 2 different designs of prototypes for the enclosure and see which fit our clients needs closer. Lastly, we had to make a decision on how the enclosures would hold the included hardware which led us to designing slots of each

### 4.2.2 Ideation

For at least one design decision, describe how you ideated or identified potential options (e.g., lotus blossom technique). Describe at least five options that you considered.

For the Frontend framework design decision, we came up with React Native, Swift, Flutter, Android Studio, and AWS Amplify for mobile development. We used a technique similar to rapid ideation to come up with various frameworks to develop and compare with. AWS Amplify is a part of the greater whole of the array of AWS services. The base framework allows us to develop everything from the backend to CI/CD and including the frontend; although it seems that the base application visuals and available libraries are left to be desired. In addition, there comes a cost factor on different payment tiers as well as a knowledge barrier.

Likewise, for Flutter, we are unfamiliar with the semantics of the frontend and CLI language for DART as well as knowledge on integration. Swift also had similar cons, where we are also unfamiliar with the language. Since we were aiming to develop an application for iOS devices. It seemed that it could have been a viable option if not for the complexity of implementation and non-understanding of the coding language/environment.

Thus leaves React Native to be considered as the last and primary option. React Native supports creating web applications for mobile devices which could also be deployed to a website with little to no changes. We are also familiar with the framework including Node.Js. React Native's has a wide range of libraries available for development which include being able to test and view the development application on devices through an external app.

### 4.2.3 Decision-Making and Trade-Off

Hardware Weighted Decision Matrix

| | Cost | Part Availability | Ease of Implementation | Functionality Provided | Size | Total |
|---|---|---|---|---|---|---|
| Criterion Weight [0-1] | 0.5 | 0.75 | 0.4 | 0.8 | 0.6 | - |
| Chip Down Design [1-10] | 5 | 7 | 4 | 10 | 9 | 22.75 |
| Raspberry Pi [1-10] | 4 | 7 | 8 | 7 | 5 | 19.05 |

Frontend Decision Matrix

| | Ease of Implementation | External Libraries Available | Appearance of Application | Knowledge of Language | Porting to iPhone for Clients | Total |
|---|---|---|---|---|---|---|
| Criterion Weight [0-1] | 0.5 | 0.5 | 0.6 | 0.8 | 0.5 | - |
| Android Studio [0-10] | 8 | 5 | 2 | 8 | 2 | 15.1 |
| Swift [0-10] | 5 | 7 | 9 | 2 | 10 | 18 |
| React Native [0-10] | 9 | 10 | 9 | 7 | 8 | 24.5 |
| Flutter | 7 | 6 | 6 | 2 | 7 | 15.2 |
| AWS amplify | 5 | 7 | 4 | 5 | 7 | 15.9 |

Backend Decision Matrix

| | Ease of Implementation | Learning resources available | Scalability | Cost | Performance | Availability | Total |
|---|---|---|---|---|---|---|---|
| Criterion Weight [0-1] | 0.20 | 0.15 | 0.10 | 0.25 | 0.15 | 0.15 | - |
| Google Firebase [0-10] | 8 | 7 | 5 | 8 | 7 | 7 | 7.25 |
| AWS [0-10] | 5 | 6 | 7 | 5 | 8 | 8 | 6.25 |
| Microsoft Azure [0-10] | 6 | 4 | 6 | 6 | 8 | 8 | 6.3 |