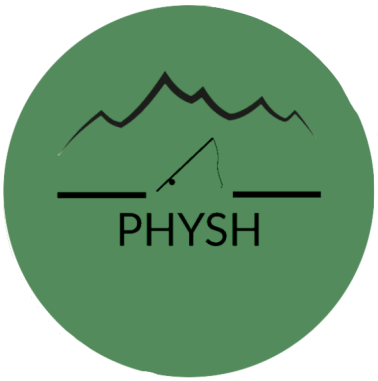
FISH - Fish Identification Search History



Requirements Specification Document

Version 1.0

Prepared by the members of Team Physh:

Ryan Mason, Scott Austin, Shelby Hagemann, Eduardo Martinez, and Jack Normand

Sponsored by David Rogowski, Ph.D.

Mentored by Vahid Nikoonejad Fard

Accepted as baseline requirements for project: FISH - Fish Identification Search History

For the client: _____ **For the team:** _____

Table of Contents

Introduction

The Arizona Game and Fish Department (AZGFD) is Arizona's government agency in charge of conserving and enhancing the wildlife and habitats around the state. The AZGFD deals with everything having to do with hunting, fishing, boating, and wildlife recreation in the state of Arizona. They are also responsible for countless projects and studies related to game and fish all across the state. One such project is data collection at the Lees Ferry fishery. Lees Ferry is located below Glen Canyon Dam, and is home to one of the premier rainbow trout populations in the state. Currently, they monitor the population here by implanting passive integrated transponder (PIT) tags into fish in the area. With these fish tagged, they are then able to sample select portions of the river to collect data. From 2011 to 2020, over 110,000 rainbow trout and 6,000 brown trout have been tagged. With all these fish tagged, scientists can examine growth rates and movement of fish, along with population size.

Our sponsor, Dr. David Rogowski is a wildlife specialist regional supervisor at AZGFD. He does various field research, and even helps oversee a different data collection method that is intended to bring in much more data on the fish than the river sampling method. This other method is a citizen science program, and gives interested anglers a PIT tag scanner, allowing them to scan the tags of the fish that they catch. This scanner returns a code, which then must be communicated to a scientist at the AZGFD. The scientist can then report back data on the fish the anglers caught, including history of the fish. This information returned to the angler is usually of interest, and can tell a lot about the fish they just caught. On the flip side, the information reported by the angler is extremely valuable to the AZGFD, as it gives key insights into the health of the fish.

Problem Statement

In order for Dr. David Rogowski to analyze and draw conclusions about the Lees Ferry fishery, he needs to have adequate amounts of information. The data collection method of sampling spots on the river only brings in so much data, and is only done 3-4 times a year. Not to mention it can only be performed on a certain portion of the river. With the inclusion of the citizen science program, the amount of data brought in should have increased significantly. However, engagement in the program is low because of several flaws in the system. The current system is visualized in the flowchart below.

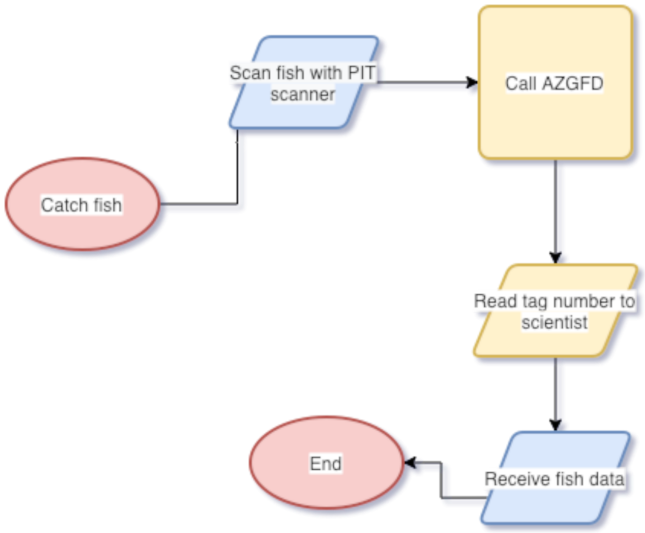


Figure 2.1

The system is relatively straightforward, but the problems arise when the angler is required to call AZGFD. The issues with this are as follows:

- The system requires another person (the scientist) to be involved
- An internet connection is required to make the phone call, and there is little to no service on the river
- A long tag number is required to be read off to a scientist, which is cumbersome and has lots of room for human error

- The scientist reads off data to the angler, which adds a lot of time to the interaction

Because of all these various deficiencies, the whole system ends up being slow and clumsy. A slow and clumsy system is one that no angler would want to take part in, and as a result data collection for the fishery is not adequate.

Solution Vision

We as a group are developing a new and improved system that will completely remove the middleman that exists in the current system, the scientist. We are developing an easy to use mobile app that allows for any angler that has access to the PIT Scanner to scan the fish that they caught and using bluetooth, transfer this ID number to our app on their phone. The app will then use this ID number to locate the respective information, this being weight, length, last location caught, species, and date/time last caught. This information will then be displayed on the user's phone to view as well as to be updated with the fish's current information. The angler would manually retrieve that information and then update that information directly on the app. The updated information will then be sent up to merge all of the existing data on the database. To accomplish this solution there are many different systems that need to be put in place and problems that we are going to need to tackle.

- React Native: Cross platform mobile development framework that implements the JS programming language, as well as HTML and CSS making frontend development more approachable.
- SQLite: A server-less relational database that will be stored locally on the users phone storing current fish data and updates allowing for offline access.
- MySQL: A server based relational database that will store all current fish data as well as approved updates.

- React Bluetooth Classic: An framework designed to allow Bluetooth connectivity within React Native applications. This framework will be used to create Bluetooth connections between the FISH app and PIT scanners.
- AWS Amplify: provides a programming model for utilizing shared and distributed data without the need to write additional code for offline and online capabilities
- Amazon RDS: the hosting platform that will hold our relational MySQL database.

Below is a diagram of our envisioned solution:

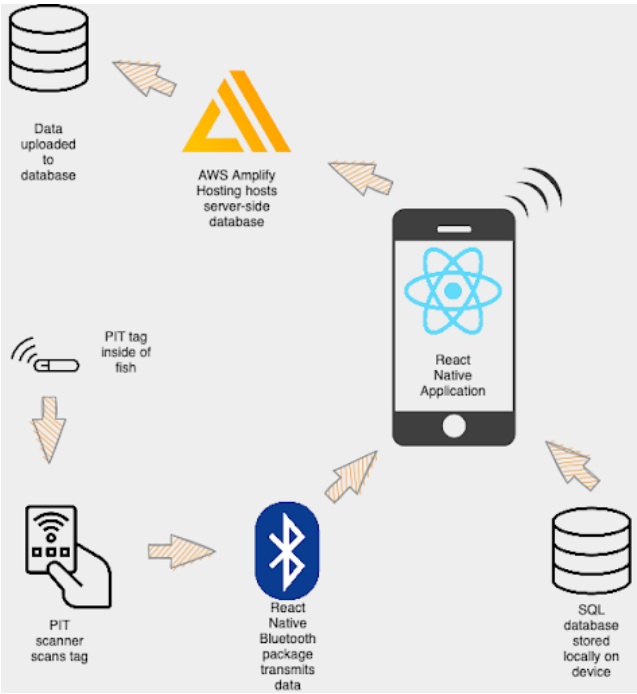


Figure 3.1

Project Requirements

Functional Requirements

- Cross-Compatibility

    - This app will be created with React Native, a framework used for cross-platform development. React Native will allow us to develop the app for both IOS and Android synchronously, saving us time and ensuring that the app is the same for both types of devices.
- Backend/Server side
    - AWS Amplify
    - Amazon RDS
- Connection to PIT Scanner via Framework
    - To accomplish bluetooth connection between the user's smartphone and the Homeagain PIT Scanner a bluetooth framework designed to be implemented into a React Native application will be used. The name of this framework is React Bluetooth classic.
- Ability to input fish ID values manually
    - In the event that an angler cannot connect their PIT scanner to their mobile device, we will have a feature where they will be able to input the fish's ID manually. When they enter the screen where they will be prompted to connect their PIT scanner and input other information on their fish, they will have the option to press a button and type in the fish's ID as shown on their PIT scanner.
- Table that uploads to main database controlled by David
    - When an angler catches a fish, they will be able to record the fish's data through our app. Once they submit this data, it will be sent to a temporary table that our client, David Rogowski, will perform quality control on, before merging it with the official Arizona Game and Fish database.
- Fish data entry screen
    - When an angler catches a fish and wishes to record its data, they will be able to enter all of the fish's information on an easy-to-follow screen, prompting them for the exact type of data that the Arizona Game and Fish Department wishes to track.
    - Drop-down menu with different fish species

        - In order to avoid any confusion or misspellings, the angler will be able to select their breed of fish from a dropdown menu that will include each different potential species that the app will be used to track.
    - Weight
        - There will be a text box or sliding scale where anglers will be able to enter the weight of their caught fish. They will have an option to insert the length of their fish in either ounces or grams.
    - Length
        - There will be a text box or sliding scale where anglers will be able to enter the length of their caught fish. They will have an option to insert the length of their fish in either inches or centimeters.
    - Date and time (automatic?)
        - The angler will be prompted to input the date and time in which they caught their fish. The app will be able to automatically record the date and time of entry, however the angler will have the option to change these values if they are retroactively inserting fish data.
    - Fish ID
    - Location (optional)
        - When an angler is inputting their fish data, they will be asked to include the closest river mile marker to their location. This will be optional for anglers who prefer to remain more private.
- Optional profile system
    - Anglers using the FISH app will have the option to either create a profile to link their fishing records to themselves or remain anonymous. Creating a profile will not be required for someone to use this app, however it will give them the opportunity to look up their fishing records and potentially their standings on leaderboards (which would be a part of a stretch goal for this project–gamification).
- Instructions interface
    - Tutorial

        - To ensure that our app is easy to use and understand, we will offer a tutorial to our users where they can learn its functions and features. Initially, this feature will involve a set of screens that users will swipe through when they open the app for the first time, which will include instructions on how to use the app and will highlight the app's features. As a stretch goal, we would like to introduce an interactive tutorial, where users will be prompted to go through the app themselves, being guided by instructions and being given information on how the app works.
    - Help Button
        - In the event that a user is ever confused on how to use the app, there will be a "help button" on each page within it where users can find instructions on how to navigate the app and use all of its features.
- Administrator login
    - In order for our mentor, Dr. Rogowski, to easily access data from the app, we will create an administrator account in which he, along with any authorized Arizona Game and Fish employees, can have authorization to certain information in the app.

Non-Functional Requirements

- Simple UI
    - The user interface will be simple and clear in nature. The anticipated audience for our application is expected to be less familiar with modern technology. Thus, a simple user interface ensures that our application is accessible to all. Furthermore, the application will provide the user with a tutorial on proper usage of the app.
- Data Accuracy
    - Gathering proper information on fish is essential to the Arizona Game and Fish Department's research. Our goal is to accurately capture the fish data from local anglers. In addition to ensuring that we accurately record that data, we must anticipate user error in data reporting. Our stretch goal is to have automatic outlier

detection and flagging, but at minimum we will have a portal where an administrator can approve new entries before entering the main database.

- Speed
    - With an application that focuses on data acquisition and displaying, we must ensure that data is recorded and displayed to the user quickly and effectively. Our goal is to serve the user with accurate data within 1 second of a request.

Environmental Requirements

- Build on Existing Database
    - The Arizona Game and Fish Department has been gathering data for research some years prior to our introduction to this project. Thus, we will ensure that we can properly integrate into their currently implemented systems. The AZGFD currently hosts PostgreSQL and SQLite databases, although our system does not directly interact with these databases we will be using SQLite and MySQL to ensure integration.
- PIT Scanner
    - The Arizona Game and Fish Department's current research uses PIT-tagged fish and PIT Scanners. The PIT scanner must be used in order to get the PIT tag of a fish. Our application will take advantage of the PIT scanner's bluetooth capabilities to connect seamlessly with our application.
- Mobile Application
    - Our application will be used by local anglers while fishing, meaning our program must be accessible on mobile devices. Therefore, we will be building a cross-platform mobile application.
- Adaptability/Scalability
    - The Arizona Game and Fish Department is not the only program that utilizes the mark and recapture program for research. With this in mind, AZGFD would like this software to be suitable for use by other research programs across the world.

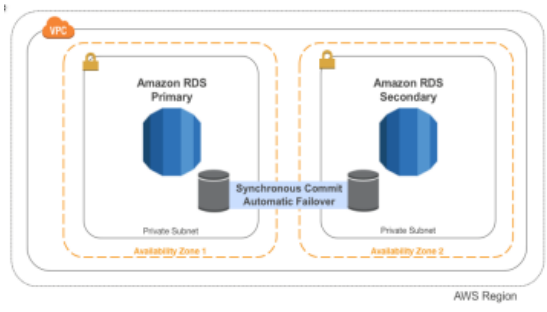
Potential Risks

Due to the project's need to grab accurate data for user display and ability to upload to the data, the mobile application and cloud server have risks involving the manipulation, integrity, and preservation of the database. Although all risks have different probabilities of happening, it is critical to identify that failures can happen both on the client and server side at any given time. The risk will be broken down into two main components:

1. Server Risks
2. Client Risks

Server Risks

The server risks will take into account the possibility of a server crashing, and the need to maintain backups for our client. Although our mobile application will be hosted in the cloud, our chosen infrastructure Amazon Web Services maintains physical locations where the server will be located. The locations are also known as availability zones, our server will be hosted in the california availability zone, and in the case the one goes down there are two other zones within the same region but a long enough distance for our server to continue running. Displayed in the figure below is a demonstration on how this issue will be taken care of :



A database instance will be launched in two out of the three available zones in the California regions. In the act of one of the regions being down, an automatic synchronization between booth regions will happen that will preserve the data and keep the server alive.

Client Risks

Now evaluation client risks involve how user input can affect either the server or the mobile application itself. These can be broken down to:

1. Inaccurate or harmful data submitted to the database

2. Inability to connect via bluetooth
3. Multiple users attempting to update and view fish data at the same time

1. Our client requires valid data for their research, so we will make precautions to set up a standard deviation of fish data, and flag input that appears too far off. However, scenarios such as an angler who decided to input the identification number manually, and incorrectly enters the wrong number but accesses a fish data within the database, consequently will be updating the wrong data without it being flagged.

To tackle this issue, we will use a feature on AWS known as AWS backups, here we will back up data frequently. For instance, if our client is verifying data and notices an anomaly(s), they will be able to access a backup vault, where they can restore valid data from the previous backup, discard the invalid data, while keeping the integrity of the database. Due to the nature of gathering data, these backups happen depending on the time of the year, on the seasons most active of gathering fish data, we will set up a backup frequency on a weekly basis and during slower time, on monthly basis.

2. Bluetooth connection can be affected by different interferences, and might not even be implemented in some devices. The risk of not getting valuable data due to not being able to connect via bluetooth is likely. Our Solution to this would be to have the user be able to manually input the identification number, and still be able to access the corresponding fish data.

3. As users are accessing the database, it's possible for an angler to be viewing/updating fish data while another angler is updating the database. This can lead to the risk of a user being presented with inaccurate information that is not in real time.

To prevent two updates happening in parallel, we will implement a critical-section while updating fish data. Within our update function to the database our function will gain a mutex lock where it holds a lock while performing the critical-section, and releases it once the update has been performed. This will prevent any user from being able to update the database while another user is performing an update.

Project Plan

In order to meet the minimum viable requirement for this project for Arizona's Fish and Game research team, we need to conduct testing, implementations, and overall a unique mobile application that improves engagement with collecting data. To deliver our promise to our client, we made a progression plan of all implementations, testing, and launching.

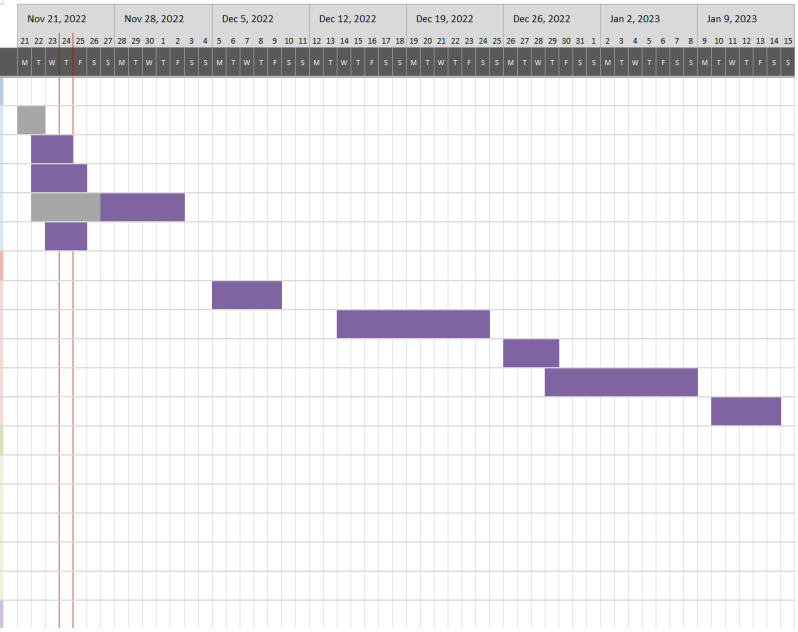
F.I.S.H Fish Identification Search History

Team PHYSH

Project Start: Thu, 11/24/2022

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Stage 1				
Task 1	Reqirement Specificaion Doc	100%	11/19/22	11/22/22
Task 2	Project info mini-video	0%	11/22/22	11/24/22
Task 3	Teahcnical Prototypes Demos	0%	11/22/22	11/25/22
Task 4	Design Review 1 presentation	50%	11/22/22	12/2/22
Task 5			11/23/22	11/25/22
Stage 2				
Task 1	Launch Cloud Server	0%	12/5/22	12/9/22
Task 2	React Native/Aws implementation	0%	12/14/22	12/24/22
Task 3	API Connection	0%	12/26/22	12/29/22
Task 4	API Functianilites Implementation	0%	12/29/22	1/8/23
Task 5	Bluetooth implementation	0%	1/10/23	1/15/23
Stage 3				
Task 1	GUI Front End	0%	1/15/23	1/29/23
Task 2	GUI Back End	0%	1/30/23	2/3/23
Task 3	Final Testing	0%	2/4/23	2/11/23
Task 4	Launch on cross-platforms	0%	2/12/23	2/16/23
Task 5	Steach Goals Feasibility	0%	2/4/23	3/6/23
Stage 4				



The milestone must be fully completed and of quality to deliver our minimum viable product for our client, throughout December to April, the team will work aggressively to deliver a quality product. By breaking the development into three stages, where stage one includes the preparation and feasibility of the project. Stage two, include preliminary testing and setting up of all our technologies ensuring everything is working cohesively, and stage three will be implementing all functionalities while developing both GUI front and back end. By the end stage three all requirements will be met, ensuring a minimum viable product, while leaving time for stretch goals consideration and possible implementation.

Conclusion

The Arizona Game and Fish Department are responsible for gathering data over a large sample of rainbow trout and brown trout. However, due to the lack of a system that allows efficient collection of data, it produces a problem of engagement with the angler and researcher. This project tackles both engagement and efficiency that can overall benefit the research of the Arizona Game and Fish Department.

The solution to our clients problem requires a mobile application that allows anglers to swiftly engage with the database and pit tag. The mobile application is tailored to display information on the corresponding fish found, to improve the engagement of all anglers. While also being able to upload valid information for our client to use in their research and main database. The cloud server will enable anglers to access real-time data while also being able to perform their function offline, as a connection to a network is difficult in regions of collecting fish data. Combining bluetooth connectivity to the scanner, AWS cloud server, and React Native cross platform development will provide the tools necessary to solve these two challenges.

As of now, the team is beginning to launch the cloud server and testing the environments that it will interact with. We are also testing the reliability of the bluetooth connectivity and setting up precautionary measures for risks that can happen during and after project development. The project is at a point to begin development and the team is eager to begin implementation.