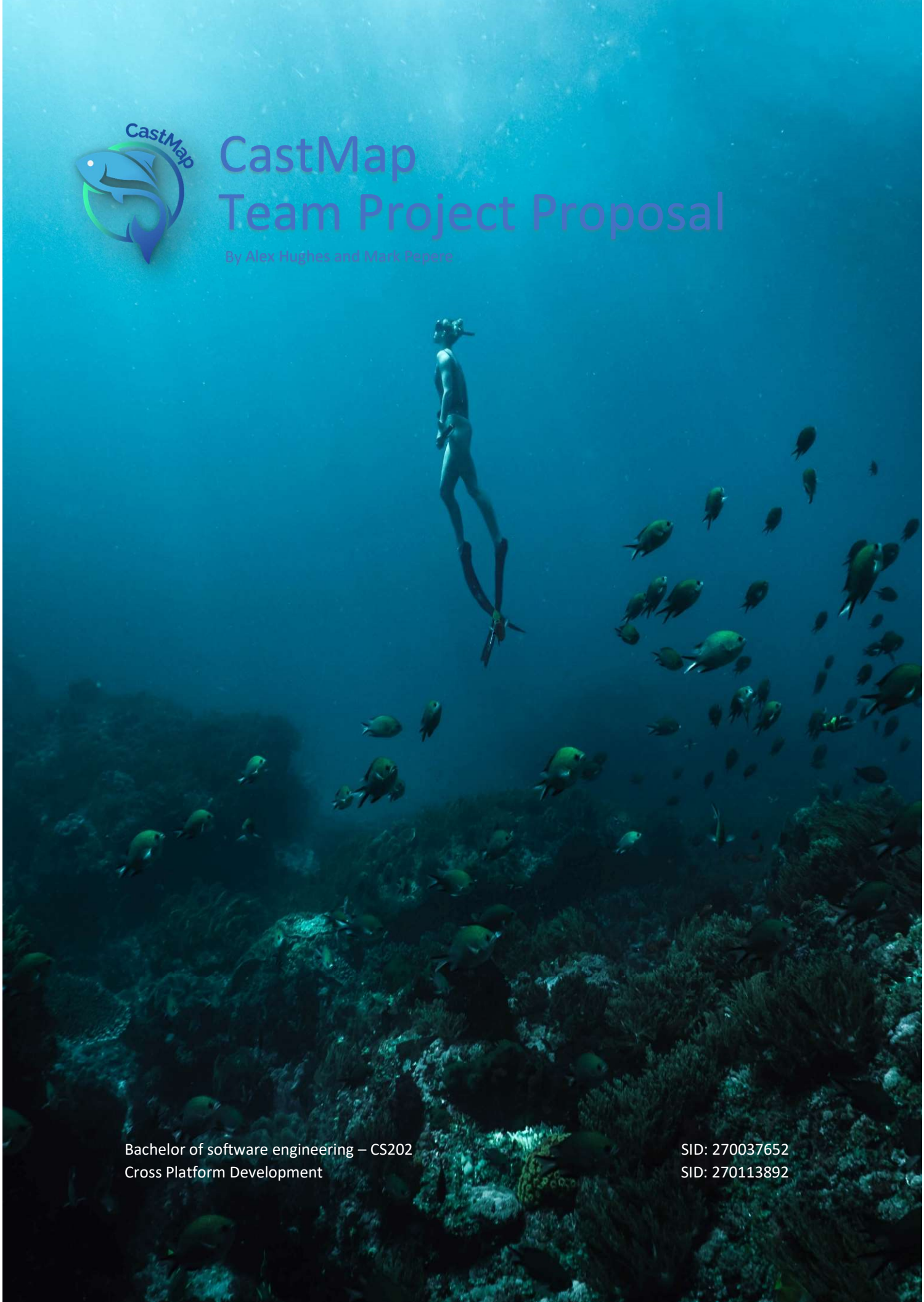
CastMap
Team Project Proposal

By Alex Hughes and Mark Pepere



Bachelor of software engineering – CS202

Cross Platform Development

SID: 270037652

SID: 270113892

Contents

Project Concept 4
- What is if for? 4
- Why is it worth working on? 4
- What will it do (Functionally)? 4
- A Description of how the application will be utilized 5

Research Phase of our Frontend Development 6
- Svelte 8
- Svelte Kit 9

Research Phase of our Backend Development 10
- Django 13

PostgreSQL 14

Folder Structure 15

Programs Used 15

Field Review 16
- What is the existing system (if any) in place to meet the business objectives? 16
- Maplaria 17
  - Summarise of Maplaria literature/web app 17
  - Comparing Maplaria to proposed CastMap 17
  - Maplaria Keynotes: 18
  - Maplaria output Images 18
- Water Pipes 20
  - Summarise of Water Pipes literature/web app 20
  - Comparing 20
  - Water Pipes output Images and model 21

Production Plan 23
- Project duration and chosen project methodology (Agile, Waterfall, etc.) 23
  - Agile 23
  - Agile Methodology Overview 23
- Project phases and work breakdown structure (WBS) 24
- Gantt chart (including milestones/project deliverables, dependencies, etc) 24
- Communication plan 25
- Scrum Meetings 25

Research Output 26
- 1. Geographic information systems, remote sensing, and mapping for the development and management of marine aquaculture 26

2. Internet of Things in Marine Environment Monitoring 26
3. A National Data Collection Framework for Recreational Fishing..... 27
4. Conflict within the fishing industry 27
5. Individual fishing quotas 28
Three user personas for the proposed system 29
UML sequence diagram 30
Use case diagram 30
Mock-ups and digital Lo-Fi wireframes of the proposed UI. 31
Mid/Hi-Fi wireframing. 32
Usability Testing of the Lo-Fi wireframes with at least five users 33
Observations 33
Interviews..... 35
References 38

Project Concept

What is if for?

The project CastMap will be about recording live data regarding areas that are being fished around New Zealand. The information will be along the lines of current people in the area fishing, people's daily and weekly quotas, the current state of the ecology, and conditions (swells, general weather, high/ low tide). This information will be displayed graphically as stats and heat maps. The application will allow for the display of research data from the Ministry of Primary Industries and live data from the fishermen and divers on the ground actively fishing. This in turn engages the people fishing and supplies MPI with live data. The longer-term outlook for this project is that the application will be hosted on the cloud with a built-in SQL database to hold and analyse gathered and exiting data.

Currently, we have discovered the Ministry of Primary Industries (MPI) conducts research annually detailing the current state of fish stock and adjusts quotas to suit. Some documents detail the stats that are publicly published. Realistically we can safely assume that the average Joe will not read or take much notice of these documents provided by MPI.

This is a worthwhile project as we can help drive sustainability for the ecosystem around the shores of New Zealand. The hope is that this application will be turned into a cloud-based native application to help share and collect information that provides education for those who use it.

While the potential for this web application goes much further the scope will be reduced to ensure that the project workload is acceptable for the project timelines.

Why is it worth working on?

This Website / Application is to ensure the protection of all species relating to fishing and diving along the New Zealand coastline. Allowing for replenishment in areas of the New Zealand coastline that have been hit hard by an excess amount of fishing or diving by the general public while also providing a dynamic solution.

What will it do (Functionally)?

Our web and phone application will be a way to serve the public information on where suitable and sustainable areas are for recreational fishing and diving to the general public.

- User Login

  User login will be standard for the public use of the application. The user will be able to:

  - Register interest in the area where they would like to fish or dive.
  - Review heatmaps to make the best decision on where the best spots will be within the local area.

- Admin Login

  Administration login will be reserved for maintenance of the front and backend.

- Display live data.
  - Live data will be displayed of the current level of activity in the area.

- Generate/update heat map of New Zealand coastlines
  - This will feed into a live and interactive heatmap display of current interest in the area.

- Manually changing daily quota allowances based on heat maps and overfishing

  3 colour system: Red, Orange, and Green.

  - (Red) The area has been hit very hard and there is a lot of fishing and diving activity.
  - (Orange) The area is in a moderate state with fishing and diving activities kept to an average.
  - (Green) The area has minimal fishing and diving activity.

A Description of how the application will be utilized

- User registers interest that they would like to fish or dive in that area.
- The Application will display heat mapped imagery of the coastal line as, a representation of previous data collected on the fishing or diving activity in that area.
- The user can then confirm registered interest in that area through a fill-out form.
- The application will then advise the user of the following:
  - 1) Yes, a full daily quota is acceptable for that day.
  - 2) Quota has been limited due to recent high activity in that area.
  - 3) Fishing and diving are currently off-limits due to extreme activity in that area.
- 1) If the user has selected a green area, they will simply hit submit which will trigger:
  - Registered interest in the area (Georeferenced coordinates can be coastguard in case of an emergency)
  - Modification to heatmap where appropriate.
  - Storing of analytics for future research and development.
- 2) If there is a limitation on quota size because the user has selected an orange area, they have 2 options.
  - Proceed to dive or fish under a limited quota.
  - Relocate to an area that is green which will allow for a maximum number for that day.
- 3) If the user selects a red area, they will be informed that the area is under rehabilitation due to high activity in the area in the past weeks.
- Once the user has completed their fish or dive, they will register with the application through a generated form of the species and quantity that was captured for that day.

Research Phase of our Frontend Development

The front end of the website is everything the users either see or interacts with when visiting the application / Website. The main concern of the frontend stack is to create a smooth user interface and straightforward internal structures. The front-end package is responsible for the design, formatting, and navigation of websites or applications.

There are 3 main elements of a frontend stack:

- HTML - it's a markup language for the creation and displaying of documents on the internet and they are the key point to the organization and placement of content on any web page.
- CSS – CSS is responsible for the format and layout of the web pages. It includes the font styles, sizes, layout, colour, and other aspects of any web page. Frontend developers typically use frameworks like SASS and LESS to make CSS more manageable and more dynamic to work with.
- JavaScript – Used to make web pages more interactive. JavaScript is a programming language that allows you to implement dynamic features on web pages; with common libraries and frameworks like jQuery, React, Angular, and Vue. Many modern applications now also use TypeScript instead of basic JavaScript.

In this area we covered reviews on many different frontend packages consisting of jQuery, Svelte, React, Angular, Vue, and many more.

Availability of Learning Resources

This area is often overlooked, you might see out there some well-constructed websites with amazing animations and eye-catching UX/UI design you still need additional courses, books, tutorials, and articles besides boring and sometimes very dry documentation to help you get started.

Creating an awesome framework is one thing but communicating the core ideas behind what you trying to deliver is something completely different.

Popularity

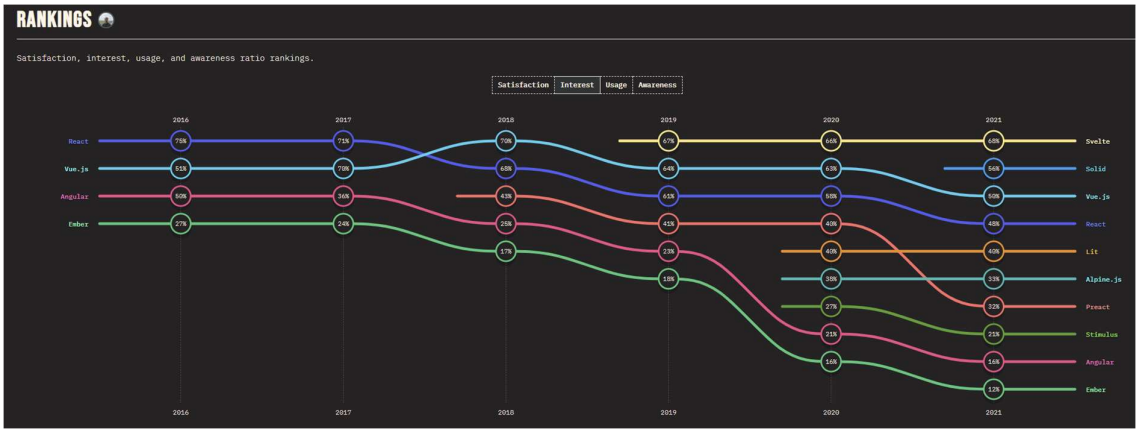


Figure 1https://2021.stateofjs.com/en-US/libraries/front-end-frameworks/

While it's easy to paste a graph and say here are the statistics and this is what we should use, we need to look a lot deeper and understand what is going on here. While we could take pride in

something exotic, we might find ourselves in a job where the company or client we are working for would require a battle-tested toolset to be used.

If a framework is not popular, it would mean that there are fewer developers specializing in it. Sometimes, the choice of framework for your company or client is not up to you; maybe it was made by a previous employee or some other person on the team. Chances are it will be Angular or React. Now there are other options out there like Ember and Vue. But you've got to deliberately seek out companies using them.

Learning only Angular or React will only get us so far in our careers. Of course, we'll have plenty of opportunities, but there are reasons other frameworks exist out there.

Core Features

We'll start with React and Vue. They are not frameworks; they only represent the view layer of your application. It means all the other parts—HTTP communication, form validation, etc.—are up to us to code.

Ember has almost everything. Surprisingly enough, Ember's core doesn't provide advanced form processing. It just has some input helpers and that's it. It is very opinionated and even has its data layer. Everything has to be done "the Ember way."

Angular 2 is a feature-rich framework. It ships with a lot of modules like Ember, so you have a ton of tools at your disposal out of the box. However as Angular is intended for large applications, it promotes TypeScript, which might trigger some resistance before you give it a go.

Usability

Ember is considered to be a very productive framework, at least for its core users. It comes with a CLI, which helps. You can generate routes, controllers, components, and models with their test suites. Doing all of that manually is a tedious task. Generating a new project is also possible. It will create the basic folder structure, install necessary packages, build tools, test the environment, etc. If you've never used a CLI to such an extent, you'll be very pleased. But, as I mentioned earlier, Ember is very opinionated; despite all of that goodness, you might find yourself frustrated while trying to accomplish common tasks.

Now for React and Vue, they have some kind of CLI, create-react-app, and vue-cli. But besides generating an initial project with some options, they don't offer that much compared to Ember or Angular. It is understandable as they both represent a view layer. For some developers, flexibility is the key that inherently comes with using React or Vue.

Angular 4 comes with a CLI as Ember does. You can generate components, directives, services, etc. It also generates the initial structure for an application, so you only need to worry about the product. The testing environment is nice, with every generated piece of the app, the test suite lives near to it (literally). Besides that, TypeScript might provide a real productivity boost.

Ease of Integration (with Other Libraries)

No matter how feature-rich your selected framework is, chances are we will face problems where additional tools are necessary. There are great libraries out there focusing on one problem, be it DOM manipulation, data processing, time formatting, rich text editing, etc. If we try to integrate one of those and spend hours on it every time, maybe that's not the optimal choice. This then leads us to the decision of our next topic Svelte.

Svelte

What is Svelte

https://www.techvariable.com/blog/what-is-svelte/

Svelte is an open-source, front-end JavaScript framework for creating interactive web pages. It's almost like the pre-existing frameworks such as React and Vue which enable developers for web application development. Svelte is primarily being a compiler, it is far more powerful than its predecessors, more performant, nimbler, better packaged, and packed with far more diverse features and extravagant developer experience.

Why have we chosen to use Svelte?

There are a few reasons we have decided to use Svelte over other more known frameworks and that is because of the following

- No Virtual Dom
- Performance
- Less boilerplate code required
- Easy learning curve

Here is a side-by-side comparison of all JS Frontend frameworks

You can filter on what you need to display to make you

https://rawgit.com/krausest/js-framework-benchmark/master/webdriver-ts-results/table.html

What makes it different from other packages?

There are a few main reasons that makes Svelte different from other frontend packages as listed above. There is no virtual DOM which means that Svelte itself is a compiler sending declarative code to the compiler and formatting it to imperative code when shipping it off the browser. This in turn reduces the size of the packaged code being shipped which increases overall performance dramatically.

In other words, Svelte already compiles your code during the *build* process and the final bundle only contains JavaScript that your application needs. This results in fast web apps with small bundle sizes.

Svelte Kit

What is Svelte Kit?

SvelteKit is an officially supported framework, built around Svelte. It adds key features to a Svelte app — such as routing, layouts, and server-side rendering — and makes front-end development outrageously simple.

Why have we chosen to use Svelte Kit?

While Svelte alone gives you a great development experience, we still have to decide on how we want to ship your application to the user. The classical approach would be to take a module bundler like webpack or Rollup and bundle your code into one big, fat JavaScript file. Then, you'd call it from a very basic HTML document, like so:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
    <meta http-equiv+="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=devce-width, initial-scale=1.0">
    <title>Mark Pepere</title>
</head>
<body>
    <div id="root"></div>
</body>
</html>
```

Instead of serving an almost empty HTML file to the client, SvelteKit already comes with all the HTML elements you need for the first-page view. The benefits are faster page loads and SEO boosts. There are two ways SvelteKit does this: prerendering and server-side rendering. I'll explain both in more detail below. What stays the same is that once the JavaScript has been loaded, it takes over and enables typical features of a single page application, like client-side routing.

The second difference between SvelteKit and a classical single JavaScript bundle is **code-splitting**. Instead of serving the entire app in one single Javascript file, SvelteKit splits the code into separate smaller chunks. Each chunk represents a route of your application. For example, everything that needs to be fetched for the [/home] and the [/about] routes will be loaded once the user needs it.

Another outstanding benefit of SvelteKit is that you can decide in which deployment environment your app is going to run. Nowadays, front-end developers have a variety of different platforms where applications can run. There are hosting providers for simple static files, more advanced serverless options such as Netlify, server environments where *Node servers* can be executed, and so on. With tiny plugins called adapters, you tell SvelteKit to optimize your output for a specific platform. This greatly facilitates app deployment.

However, the biggest advantage SvelteKit has to offer is its ease of use. Of course, you can manually set up your *build* process from scratch with all these features, but SvelteKit makes it as easy as possible for you, and the best way to experience this is by actually using it.

Research Phase of our Backend Development

Before we pick a backend service, we to understand what type of information we are storing within our application and how it will be accessed and utilized.

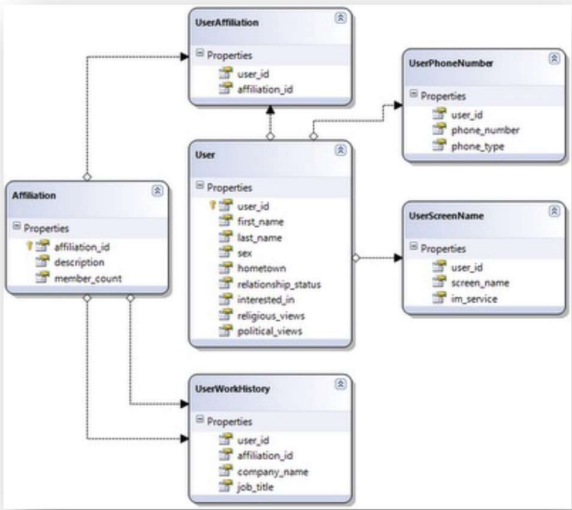
Relational vs non-Relational

Relational	Non-Relational
Organized by tables, rows, and columns	Varied data storage models
Rigid schema (SQL)	Flexible schema (NoSQL) – data stored in key-value pairs, columns, documents, or graphs
Rules enforced within the database	Rules can be defined in application code (outside the database)
Typically scaled vertically	Scales horizontally
Supports complex queries and joins	Unstructured, simple language that supports any kind of schema
Amazon RDS, Oracle, MySQL, IBM DB2, PostgreSQL	Amazon DynamoDB, MongoDB, Redis, Neo4j

Relational Database

EmployeeID	FirstName	LastName	JobRole	Location
00001	Paul	Peterson	Senior Developer	Sydney
00002	Kaleigh	Annette	Assistant Manager	Brisbane
00003	Carl	Wood	Sales Support	Sydney
00004	Vinni	Jones	Customer Services	Melbourne
00005	Stefanie	Howard	IT Architect	Brisbane

Relational Database – Multiple Tables

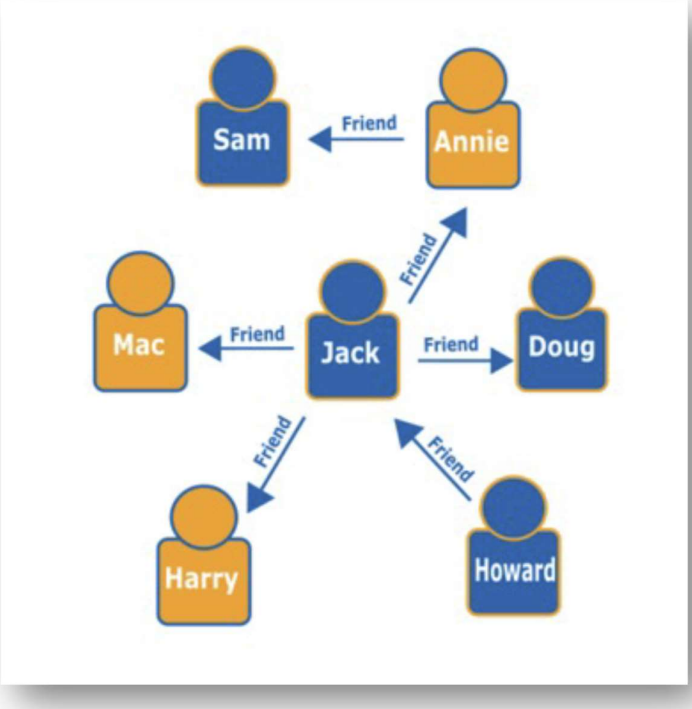


Non - Relational Database (NoSQL)

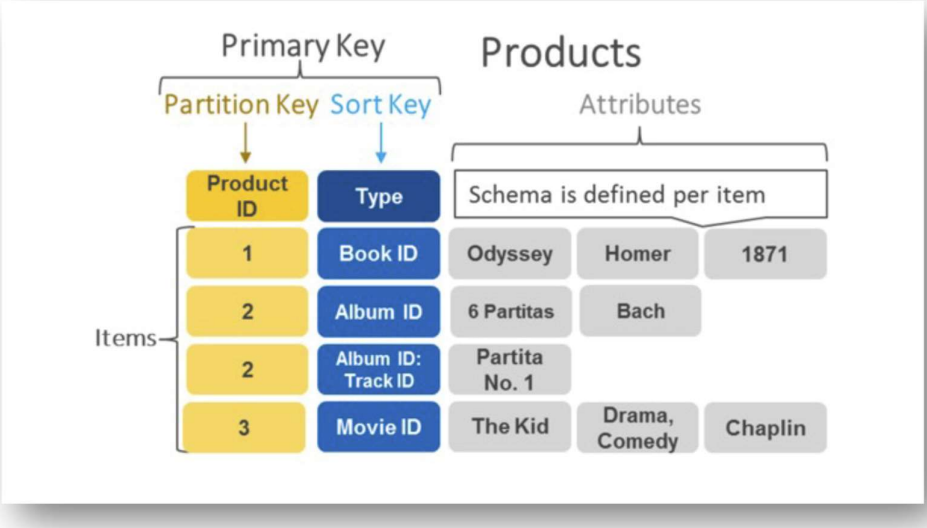
Document – e.g. MongoDB

```JSON
[
    {
        "year" : 2013,
        "title" : "Turn It Down, Or Else!",
        "info" : {
            "directors" : [ "Alice Smith", "Bob Jones"],
            "release_date" : "2013-01-18T00:00:00Z",
            "rating" : 6.2,
            "genres" : ["Comedy", "Drama"],
            "image_url" : "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@._V1_SX400_.jpg",
            "plot" : "A rock band plays their music at high volumes, annoying the neighbors.",
            "actors" : ["David Matthewman", "Jonathan G. Neff"]
        }
    },
    {
        "year": 2015,
        "title": "The Big New Movie",
        "info": {
            "plot": "Nothing happens at all.",
            "rating": 0
        }
    }
]
```

Graph – e.g. Amazon Neptune



Key-value – e.g. Amazon DynamoDB



Operational vs Analytical

Operational / transactional	Analytical
Online Transaction Processing (OLTP)	Online Analytics Processing (OLAP) – the source data comes from OLTP DBs
Production DBs that process transactions. E.g., adding customer records, checking stock availability (INSERT, UPDATE, DELETE)	Data warehouse. Typically, separated from the customer-facing DBs. Data is extracted for decision making
Short transactions and simple queries	Long transactions and complex queries
Relational examples: Amazon RDS, Oracle, IBM DB2, MySQL	Relational examples: Amazon RedShift, Teradata, HP Vertica
Non-relational examples: MongoDB, Cassandra, Neo4j, Hbase	Non-relational examples: Amazon EMR, MapReduce

Django

What is Django?

Django is a free and open-source web framework written in the Python programming language. Named after a jazz guitarist Django Reinhardt, it is used by some of the largest websites in the world including YouTube, Spotify, Instagram, Mozilla, and NASA, but also lightweight enough to be a popular choice for weekend side projects and start-ups. Its "batteries-included" approach means it comes with a lot of features straight out of the box and a powerful website can be generated quickly in the hands of a skilled developer. Django is a high-speed MVC (Model-View-Template) framework for websites as displayed below:

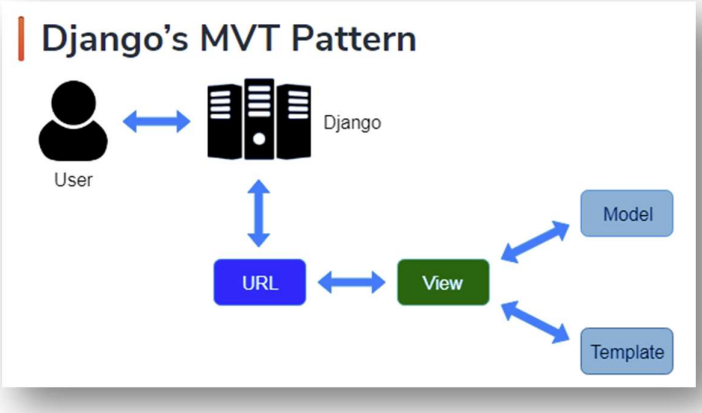


Figure 2https://www.educba.com/what-is-django/

Why have we chosen to use Django?

Mature

The first point is Django is a Mature platform meaning it is well developed and documented. Django was created in 2003 by Adrian Holovaty and Simon Willison from Lawrence Journal-World started using python to build applications.

Stable

Django as a backend platform is very stable. If you look at the likes of YouTube, Spotify, Instagram, and Mozilla there is clear evidence that the framework is very stable.

Easy To Learn

The learning curve for Django is relevantly straightforward if you have experience with backend frameworks. There is a large number of resources from courses, books, tutorials, and articles to select from.

Huge Community

Being well established and widely used means that there is a huge community helping develop and provide feedback on Django as it is an open-source package. Running into development problems when having a large support group behind the package can go a long way as typically someone would have run into issues that we might encounter along the way.

PostgreSQL

What is PostgreSQL?

PostgreSQL is a free and open-source object-relational database system that extends the SQL language combined with lots of features that safely store and scale the most complicated data workloads. The origins of PostgreSQL date back to 1986 as it was a part of the POSTGRES project at the University of California at Berkeley and it has more than 30 years of active development on the core platform. This means that the community is well established and there is no shortage of resources.

PostgreSQL has earned a strong reputation for providing solid architecture, reliability, data integrity, robust feature set, extensibility, and the dedication of the open-source community behind the software to consistently deliver performant and innovative solutions. PostgreSQL runs on all major operating systems, and has been ACID-compliant since 2001, PostgreSQL has powerful add-ons such as the popular PostGIS geospatial database extender.

Why have we chosen to use PostgreSQL?

PostgreSQL aims to help developers build apps, admin protect data integrity and build fault-tolerant environments, and help you manage your data no matter how big or small the dataset. PostgreSQL is highly extensible.

PostgreSQL confirms with the SQL standard that such conformance does not contradict traditional features or could lead to poor architectural decisions. Many of the features required by the SQL standard are supported, though sometimes with slightly differing syntax or function.

Folder Structure

When it comes to folder structure the planning of this will take place in CS203 when we start to develop the application but we will be using VS Code with a .svelte folder structure similar to the below image.

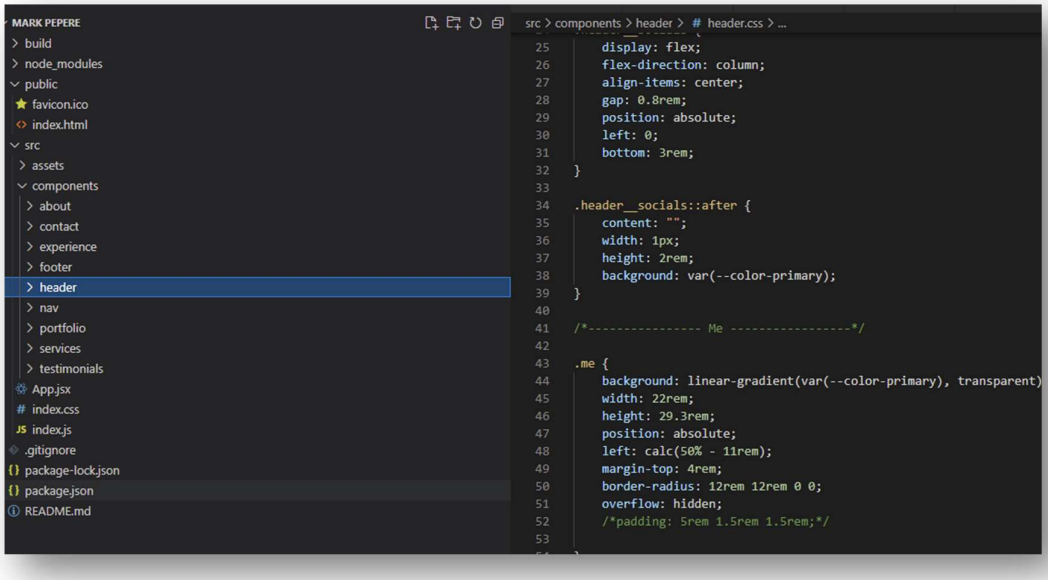


Figure 3 Mark Pepere, Visual Studio Code

Programs Used

The above tools will be used from a framework standpoint but there is more that we will need to construct our application. Below are more resources that we will be utilizing

Notion

Notion will serve as a second brain for storing all information from project planning to a useful website.

Vision Studio Code

VS Code will be our IDE of selection with great integration with Svelte and powerful debugging tools.

Microsoft Word

MS Word for report writing etc.

GitHub

GitHub will be our version control for all code generation, push and pull requests from myself and Alex.

Field Review

What is the existing system (if any) in place to meet the business objectives?

In terms of direct completion, there is nothing in New Zealand that offers anything close to our application but there are some applications that share similar ideas that we will be utilizing:

www.google.co.nz/maps

Google maps Heat map display will be a big part of the project where the coastal line will be constantly in traffic like state of green, orange, and red to display current the activity in fishing and diving.

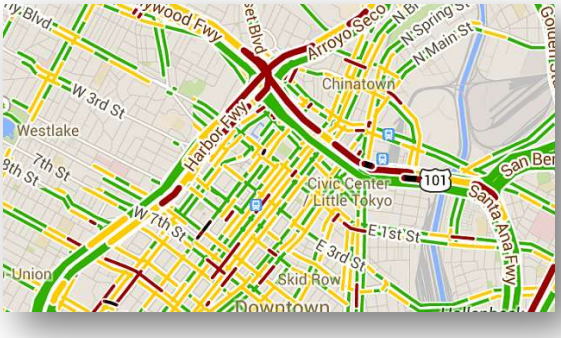


Figure 4 Google maps. (n.d.). Google Maps. https://www.google.co.nz/maps

www.windy.com

windy.com offers the high level of design we are after when it comes to live and interactive heatmap displays.

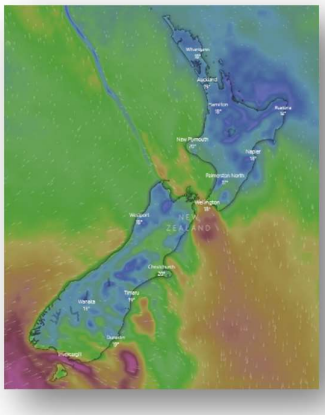


Figure 5 Windyty, S. (n.d.). Windy.com/. https://www.windy.com

https://globalfishingwatch.org/map

The global fishing watch is probably the closest website we could find that can close to our application. This website displays areas of high activity fishing. But the site itself suffers from a lot of technical difficulties. From slow load times of map information to a less than desirable UX design.

Maplaria

Summarise of Maplaria literature/web app

Maplaria is a web application developed to help analyse and break down the extent of the disease Malaria in countries such as Kenya and Tanzania. This application works based on statistical data to depict the state of disease in a geographical region. This application was developed due to a lack of statistical expertise and computational equipment in the poorer countries that currently still suffer from Malaria outbreaks.

The main objective is to provide data to policymakers to allow a better understanding of affected locations, help make decisions on treatments, the allocation of needed treatments, and collect data for real-time observations. The web application is designed with a user-friendly approach to ensure that people with little statistical knowledge will still be able to understand and operate the application. The baseline data is taken from currently collected data that allows a metaphorical base to be built on.

This geostatistical mapping model allows for a much wider view of the spread of the endemic countries in sub-Saharan Africa. Overall, we found that this application was a smart step in the right direction to ensure people in the aforementioned countries get the medical attention needed.

Comparing Maplaria to proposed CastMap

When we compare this application's purpose to the proposed CastMap web application, we can see that functionality would be similar. The functionality of Maplaria is to allow officials with limited knowledge to upload and adjust data directly which is then compiled with existing baseline data gathered from outside sources. This data is then displayed as geostatistical heat maps on the web app and provides a graphical output. When we compare CastMap, the intention is to get data from NZ fisheries and the Ministry of Primary Industries (MPI) to obtain a statistical baseline on the current ecology and sea creature numbers/quotas, this will then be supplemented with real-time data provided by people on the ground, fishing, diving, and officials from government agencies. CastMap's web app outputs will be of a similar style to Maplaria as we would also like to depict the data collected graphically.

The graphical output is to provide the average joe with information based on popular/overfished spots around the coast of New Zealand. The idea is to catch specific data such as quotas and the density of people currently fishing in an area but also ensure that we can inform people to reduce overfishing and help the ecology around the shores of NZ. The data gathered will provide NZ fisheries and MPI with information and data to better understand and ensure the life cycles of NZ shores and particular sea species.

When looking into the actual design of Maplaria we can see that it has been tailored to be very easy to use which will allow for increased engagement and directly relates to how much data is supplied. We intend to follow this thought pattern as we recognise that fishermen and divers are mostly hobbyists and won't want to spend large amounts of time or effort filling out forms.

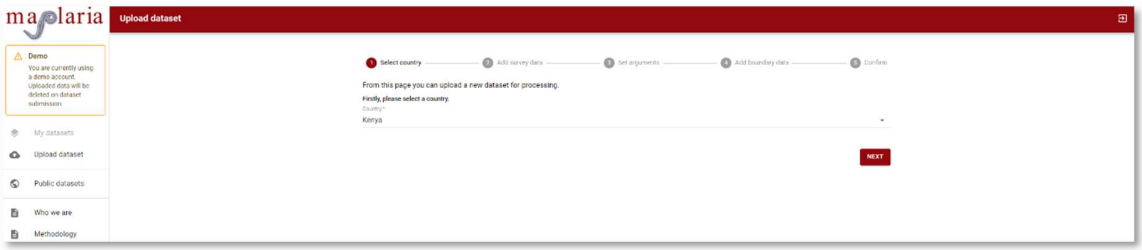
In summary, Maplaria was a great case study to review given that the purpose of the application aligned nicely with our intended output. Some key differences are that Maplaria is designed for a representative or official person with some knowledge of statistics and app functionality. CastMap on the other hand will have a much simpler data input, real-time data that we hope will be fed by the fishermen and divers will be a matter of button press. For example, to acknowledge conditions

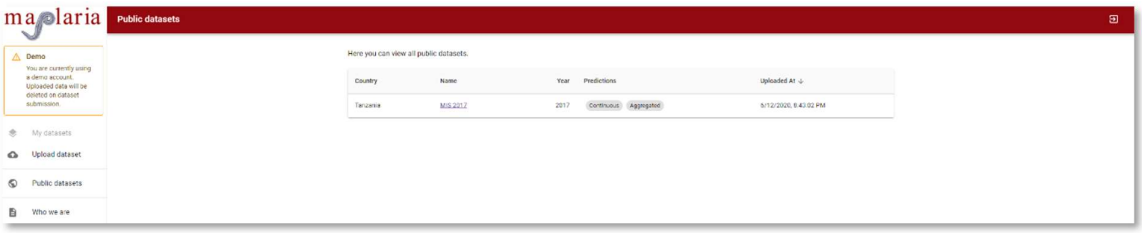
met for max quota allowance per person/per day means that the person will press the quota complete button, and this will register that one person has taken their quota (within official range) for the day (date) and at location (GPS or GIS).

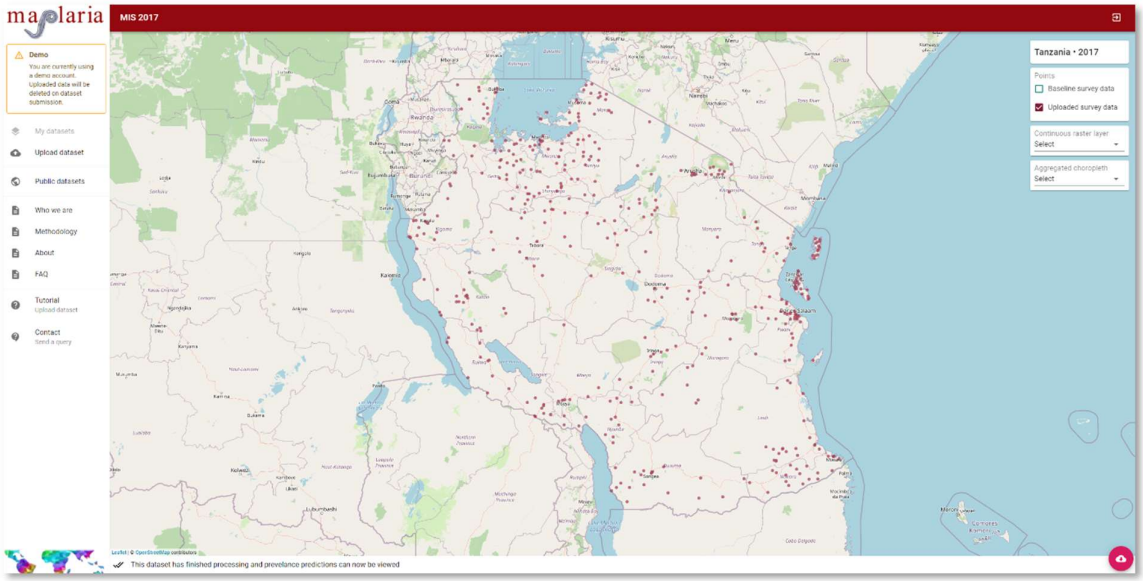
Maplaria Keynotes:

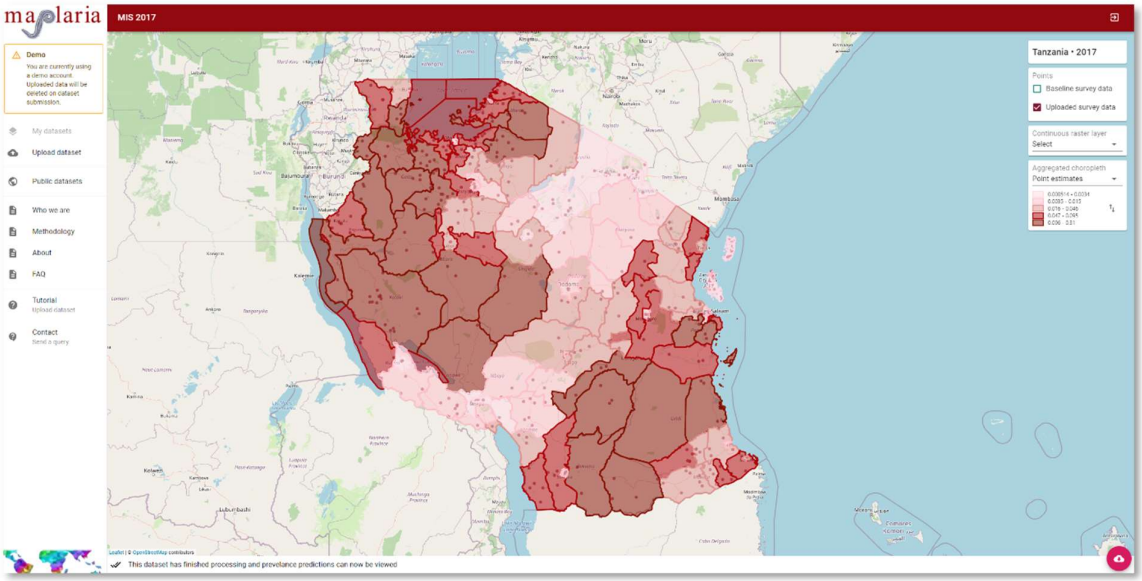
- Simple UI/UX for all to manage
- Heat Maps
- Coloured Indicators
- Percentage Indicators
- Upload and Download
- Low user journey
- Simple page & menu layouts
- Graphs
- Baseline data from outside sources
- GIS
- Statistical analysis
- Database/Data Management

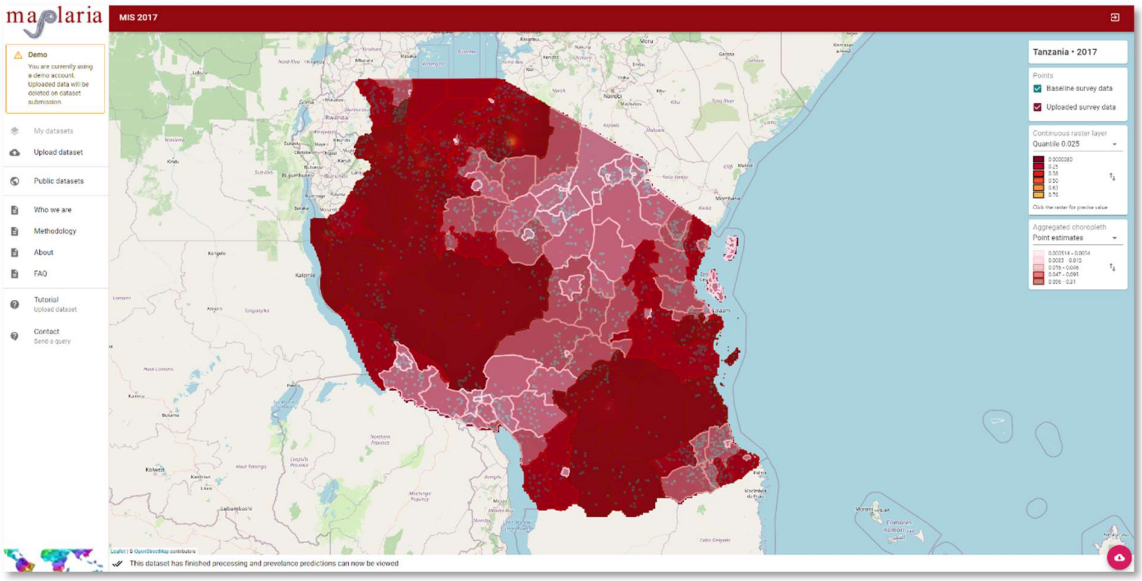
Maplaria output Images






maplaria
MIS 2017
Demo
You are currently using a demo account. Uploaded data will be deleted on dataset submission.
My datasets
Upload dataset
Public datasets
Who we are
Methodology
About
FAQ
Tutorial
Upload dataset
Contact
Send a query
Tanzania • 2017
Points
Baseline survey data
Uploaded survey data
Continuous raster layer
Select
Aggregated choropleth
Select
This dataset has finished processing and prevelance predictions can now be viewed


maplaria
MIS 2017
Demo
You are currently using a demo account. Uploaded data will be deleted on dataset submission.
My datasets
Upload dataset
Public datasets
Who we are
Methodology
About
FAQ
Tutorial
Upload dataset
Contact
Send a query
Tanzania • 2017
Points
Baseline survey data
Uploaded survey data
Continuous raster layer
Select
Aggregated choropleth
Point estimates
This dataset has finished processing and prevelance predictions can now be viewed


maplaria
MIS 2017
Demo
You are currently using a demo account. Uploaded data will be deleted on dataset submission.
My datasets
Upload dataset
Public datasets
Who we are
Methodology
About
FAQ
Tutorial
Upload dataset
Contact
Send a query
Tanzania • 2017
Points
Baseline survey data
Uploaded survey data
Continuous raster layer
Quantile 0.025
Click the raster for precise value
Aggregated choropleth
Point estimates
This dataset has finished processing and prevelance predictions can now be viewed

Water Pipes

Summarise of Water Pipes literature/web app

Water pipes are an application that allows complaints and reporting about issues to be submitted by anybody. This application allows for fast on the go reporting of damage or issues that pertain to pipe infrastructure to the correct authorities. Water Pipes run on a geographic information system (GIS), which means that location and local authority can be allocated instantly allowing for quick response times. This enables authorities to quickly assign the relevant resources/contractors to remedy problems very quickly. This broad information collection technique help reduces overall service cost to both authorities and the taxpayer.

The application is written in Java and C++ which means that the program is highly efficient. This application has been designed for mobile and therefore is native. Allowing it to be native to mobile makes complete sense as the reporting aspect of this application is enabled for everyone to use.

The collected data is fed into an SQL database that allows authorities to analyse and report. Analysing this data will allow for authorities to project work that may need to be undertaken shortly. for example, different sections of a water pipe have been reported as an issue(broken/flooding/collapsed)

Comparing

When we compare this to the ideation of the proposed CastMap there are some similarities. We can see that the fault reporting is a key takeaway from Water Pipes, this essentially is what CastMap will do by having data live from the ground. CastMap will represent data geographically on a heat map much like the GIS output of Water Pipes.

When we review the broader scope, CastMap would be best suited and utilised as a native application much like Water Pipes. This provides a more valuable data collection as it will be live. Again, the data management approach is through SQL for Water Pipes and again CastMap will be on a similar (yet to be decided) structure.

Water Pipes is built using Java and C++, CastMap will be built using Python and Svelte

In summary, Water Pipes have been designed to meet a couple of KPIs.
These include:

- Accurate location data
- Efficiency
- Fault reporting
- Data management

CastMap's KPIs will pretty much align with the above Water Pipes KPIs

Keynotes:

- GIS
- Database/Data Management SQL
- Java & C++

Water Pipes output Images and model

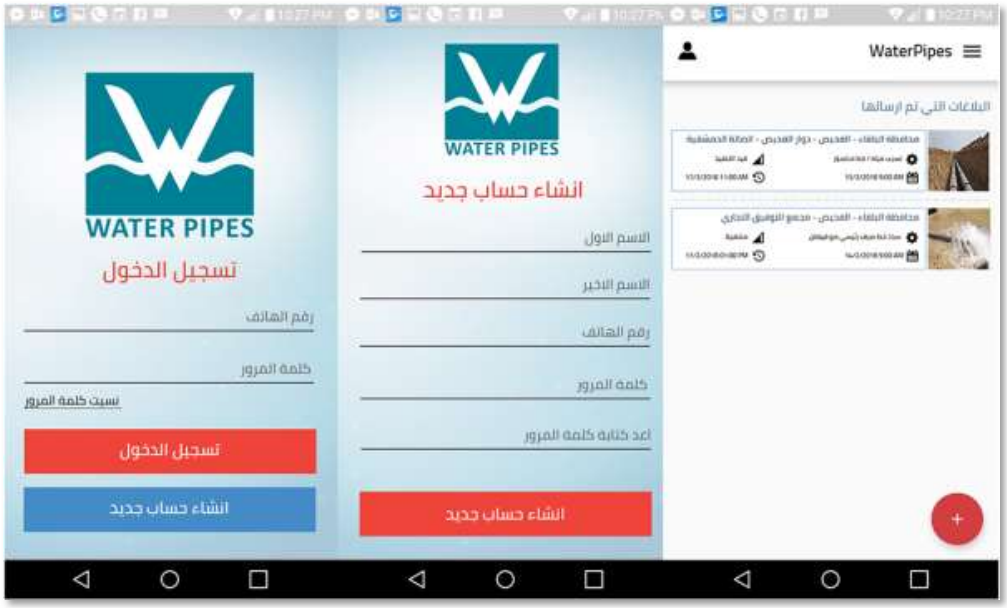


Figure 6 Profile Page

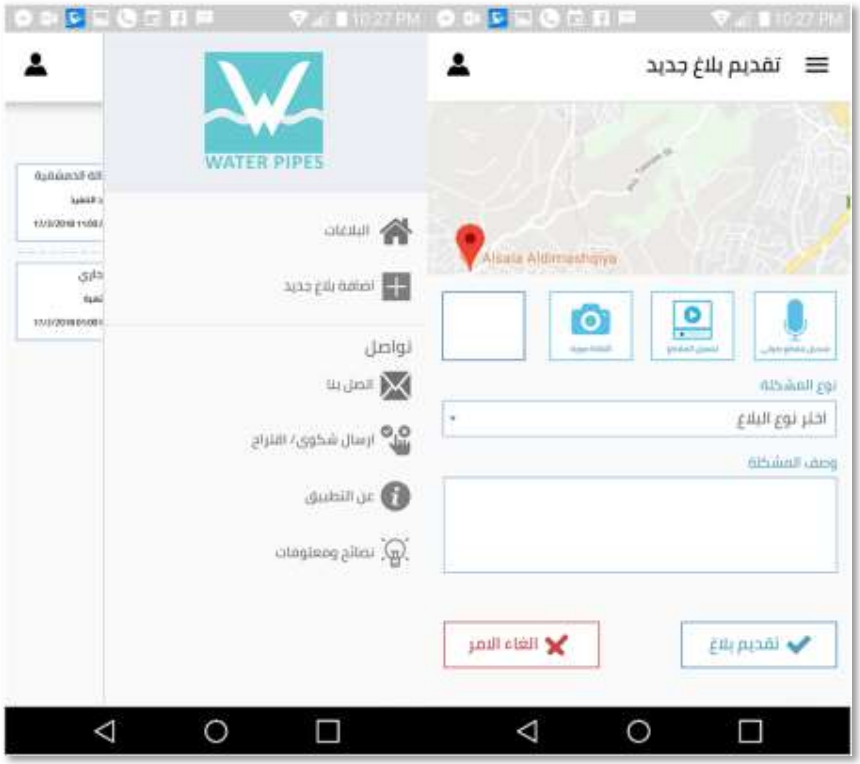


Figure 7 Profile Page and Setup

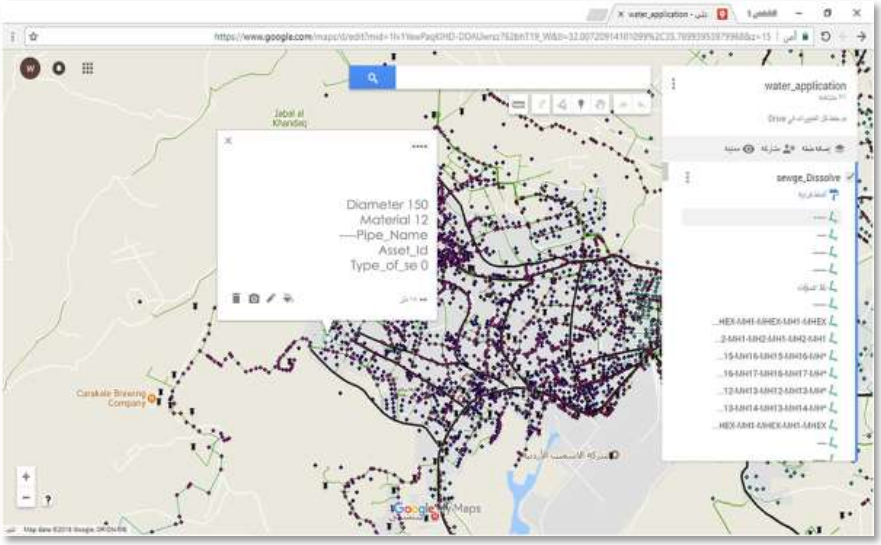


Figure 8 GIS System

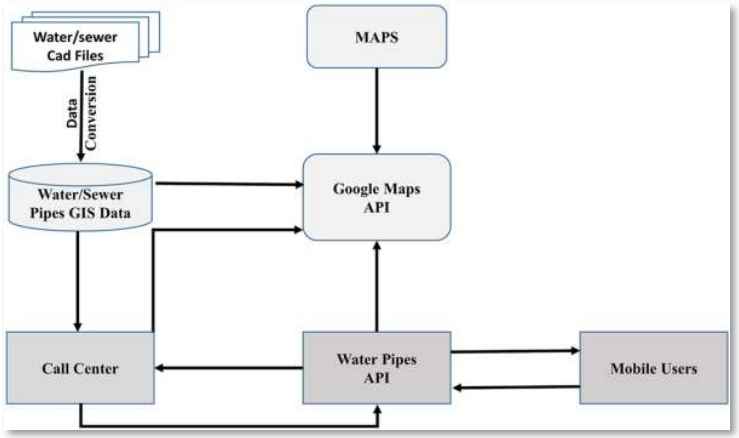


Figure 9 Use Case for Data exchange

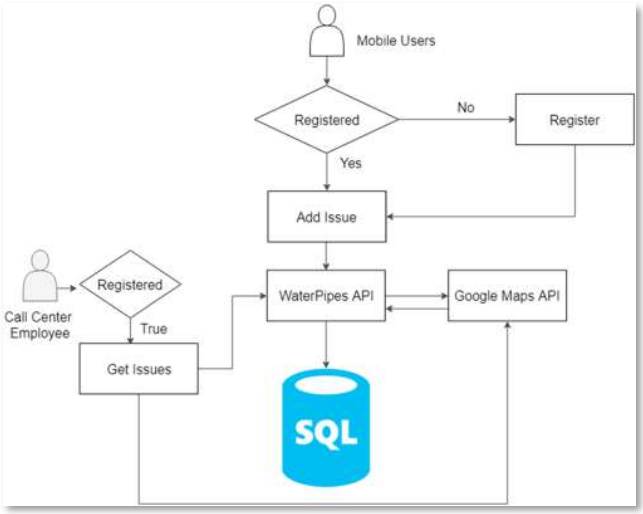


Figure 10 Use Case for Data feed to SQL

Production Plan

Project duration and chosen project methodology (Agile, Waterfall, etc.)

Agile

Both myself and Alex have come from the construction industry where the Waterfall methodology is widely used. Applying the waterfall method allows for having a clear direction on where we are going and when we are delivering. Applying the Agile model will allow us as a group to tac faster and communicated more often which I feel is a great aspect of the agile model. For this reason, we will be going with the agile model.

Agile Methodology Overview

Agile abandons the risk of spending months/years on a process that ultimately fails because of some small mistake in an early phase. It relies instead on trusting employees and teams (Myself and Alex) to work directly with customers to understand the goals and provide solutions in a **fast and incremental way**.

Faster, smaller. Traditional software development relied on phases like outlining the requirements, planning, design, building, testing, and delivery. Agile methodology, by contrast, looks to deploy the first increment in a couple of weeks with scrum meetings and the entire piece of software in a couple of months.

Communication. Agile teams within the business work together daily at every stage of the project through face-to-face meetings. This collaboration and communication ensure the process stays on track even as conditions change.

Feedback. Rather than waiting until the delivery phase to gauge success, teams leveraging Agile methodology track the success and speed of the development process regularly. Velocity is measured after the delivery of each increment.

Trust. Agile teams and employees are self-organizing. Rather than following a manifesto of rules from management intended to produce the desired result, they understand the goals and create their path to reach them.

Adjust. Participants tune and adjust the process continually, following the KIS or **Keep It Simple** principle.

Project phases and work breakdown structure (WBS)

Code	Sub Code	WBS Title
1.1		Project planning
	1.1.1	Team Roles
	1.1.2	Project plan and Develop Milestones
	1.1.3	Communication planning
	1.1.4	Resource allocation
	1.1.5	Agile model scrum meeting scope
	1.1.6	Planning Approval
1.2		Requirements analysis
	1.2.1	Identification of NFR and FR
	1.2.2	Requirements elicitations
	1.2.3	Classification and organisation
	1.2.4	Negotiation
	1.2.5	Requirements validation
	1.2.6	Milestone and signoff
1.3		**System modelling & prototyping**
	1.3.1	Activity Diagram
	1.3.2	Use Cases
	1.3.3	Sequence Diagram
	1.3.4	Low-fi / Mid – fi design
	1.3.5	Screen layout / Window layout
	1.3.6	Form elements
	1.3.7	Low-fi / Mid – fi user testing approval
	1.3.8	Hi – fi Prototyping
	1.3.9	Modelling & Prototyping Approval

Code	Sub Code	WBS Title
1.4		**Software Testing**
	1.4.1	Development Testing - Unit Testing
	1.4.2	Release Testing
	1.4.3	User Testing
	1.4.4	Alpha Testing
	1.4.5	Beta Testing
	1.4.6	Acceptance Testing
	1.4.7	Software Testing milestone completion
1.5		**System and user Documentation**
	1.5.1	A section describing an overall system architecture
	1.5.2	For each component in the system, its functionality, and interfaces User Documentation
	1.5.3	Installation document
	1.5.4	User Manual
1.6		**Closeout**
	1.6.1	Review
	1.6.2	Installation document
	1.6.3	User Manual
	1.6.4	Handover
	1.6.5	Lessons Learnt

Figure 11 https://www.notion.so/Topic-1-Work-Breakdown-Structure-WBS-0bda90ddff7f45fd8824f987d92980fd#375d60e4fea94d719f01567d1e12ee35

This section of the WBS was created in Notion and filters right through to the Gantt chart Below.

Gantt chart (including milestones/project deliverables, dependencies, etc)

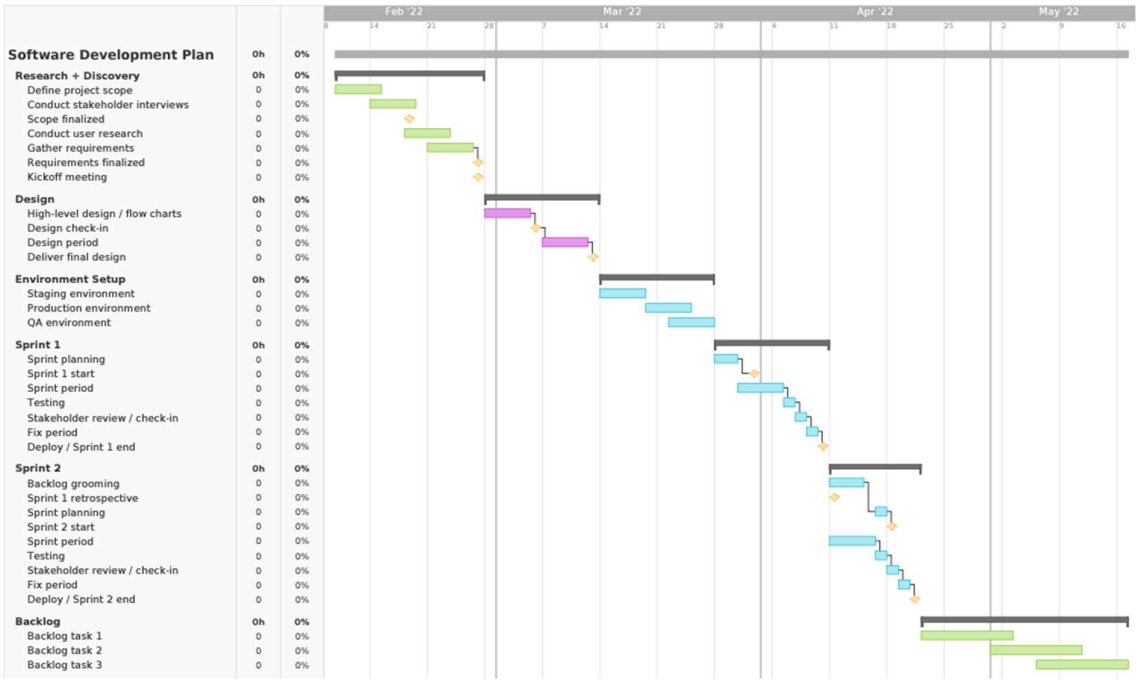


Figure 12 https://prod.teamgantt.com/gantt/schedule/?ids=3020687#ids=3020687&user=&custom=&company=&hide_completed=false&date_filter=&color_filter=

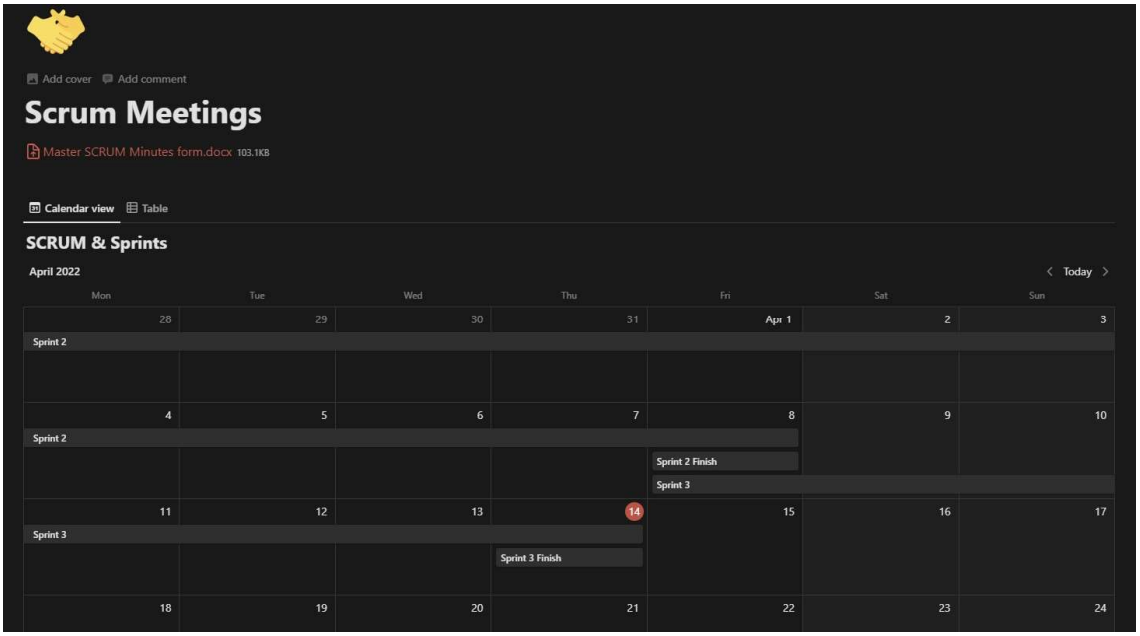
In creating the Gantt chart we started with Milestones and then build on that drilling into the milestones to ensure that each section is as fleshed out as possible.

Communication plan

Types of Communication	Purpose of Communication	Owner	Audience	Frequency (weeks)	Documentation
Alpha Testing Meeting	Review of feature testing	Product Owner	Scrum Master /Dev team	2	Sprint Meeting Minutes 1
Beta Testing Meeting	Review of Stress testing of the servers	Scrum Master	Dev team	2	Sprint Meeting Minutes 2
Modeling & Prototyping	Review of Prototyping	Scrum Master	Dev team	2	Sprint Meeting Minutes 3
Activity Diagram Meetings	Review of Activity Diagram	Scrum Master	Dev team	2	Sprint Meeting Minutes 4
Low-fi / Mid – fi design Meetings	Review of Low & Mid fi design	Scrum Master	Dev team	2	Sprint Meeting Minutes 5
Screen layout / Window layout Meetings	Review of Screen layout etc	Scrum Master	Dev team	2	Sprint Meeting Minutes 6
Installation document Meetings	Review of Install doc	Scrum Master	Dev team	2	Sprint Meeting Minutes 7
Lessons Learnt	Review of Full Process	Product Owner	Scrum Master /Dev team	2	Sprint Meeting Minutes 8

Scrum Meetings

As we are using the Agile model scrum meeting have been held every 2 weeks to ensure that we are still on task while also discussing the direction of the application and ensuring that milestones are being met. We log all conversations within Notion as displayed below.



Research Output

1. Geographic information systems, remote sensing, and mapping for the development and management of marine aquaculture.

Summary

(Kapetsky & Aguilar-Manjarrez, 2007) explored and promoted the use of Geographic information systems (GIS) and how they can be utilised for the monitoring and management of aquaculture. The GIS system's primary selling point is the capture of data in many forms, these forms comprise, satellite, undersea, ground, and airborne sensors. This data applies to near and offshore geographical locations, data can be used to discover the suitability of the environment, cultural structure, and access to the environment. This data allows for accurate monitoring and managerial decisions to ensure the health of aquaculture in each geographical location.

The author's perspective is global and represents a mixture of current and proposed uses for GIS technologies that are currently underutilized, the primary goal is to ensure the wellbeing of aquaculture and sea ecology is monitored and looked after. As such the author's purpose is to enlighten a variety of people, organizations, and governments about the advantages of geographical information systems and the tools that can be integrated. Overall, the book allows the user to make their mind up by laying out a multitude of tools, applications, and research.

Finally, comparing this book to the project ahead, there are some obvious relations between them. CastMap will be worked on a similar system to the GIS (potentially the same) with means that our graphical output on the web application will be directly related to the geographic locations around NZ shores. Other similarities include the capture, storage, and processing of data. The key difference here is that (Kapetsky & Aguilar-Manjarrez, 2007) book refers to the collection of data around the swells, microbiology, and broader aquaculture. CastMap's primary data collection will be related to the location of divers or fishermen, quotas caught, types of sea life, the overall health of species, weather, and sea movements.

2. Internet of Things in Marine Environment Monitoring

Summary

(Xu et al., 2019) Journal presents a review of the 'Internet of things (IoT) capability concerning the marine monitoring space. Touching on IoT, we know that IoT refers to a network of connected Electronics, the connection allows for data to be monitored and stored based on specified parameters. In this case, the journal reviews how IoT can be/is already integrated into marine monitoring systems. Commonly, the collection of data from sensors and other monitoring equipment means the quantity is large and thus a challenge to process, store and analyse. The Journal covers many challenges and opportunities. Some of the key challenges are big data outputs, differing systems used around the globe to monitor and process, and the accuracy of data. The journal suggests that working within the bounds of IoT could be beneficial to multiple parties at the same time, neighboring seas and oceans would have matching data from similar systems and thus accuracy would be increased. The application of IoT is incredibly large as stated in the journal (Xu et al., 2019, p. 3) IoT-based marine environment monitoring application areas includes: (1) Ocean sensing and monitoring; (2) water quality monitoring; (3) coral reef monitoring; (4) marine (either offshore or deep-sea) fish farm monitoring; (5) wave and current monitoring. Different applications use different IoT system architectures, sensing and control technologies, and communication technologies.

How do we relate this to the proposed CastMap? If we look at some of the overarching principles that the journal sheds light on, we can relate aspects to the proposed project. As previously mentioned, the scope for CastMap has been narrowed down, but in the future, the web application could be a forerunner to commercial fishing and deep-sea/offshore monitoring. This shows that the longevity of this project is strong. The confirmation that accurate data is key for monitoring is clear in the journal, we intend to follow this ethos but ensure it is easy to upload data to CastMap. Finally, as has been stated the expected data volume will be large and thus highlights the importance of ensuring that backend and databases such as PostgreSQL are key.

3. A National Data Collection Framework for Recreational Fishing

Summary

(Karlsson, 2016) This report intended to highlight the need for increased knowledge on fish populations and the impact that recreational fishing is having on ecosystems. The authors try to highlight the ability and extent of studies that can be undertaken. In this case, the geographical location is centered around Sweden. The report looks at the current knowledge and identifies that information is limited at best. knowledge and data are key to an accurate valuation of the ecosystems that are been fished. The report delves further into why such information a data is important. e.g., The ability to understand the impact fishing has on the ecosystem as some fish help regulate the system, and if they are fished what is the knock-on effect.

Some of the issues highlighted were that fishermen are not informed enough to know if they are contributing to the impact on said ecosystems around Sweden, another key pain point for this report is the lack of science and actual data surrounding recreational fishing.

Key takeaways for the CastMap:

- The importance of data collection in this area
- The growth in the scientific world has increased vastly
- Knowledge and studies are showing more evidence of ecosystem damage.

4. Conflict within the fishing industry

Summary

This article was written by David Thomson from Edinburgh dated July 1980 is the distinction between small and large fishing companies and the comparison between the two. David coins them as two distinct sectors. One is composed of main company-owned large-investment fishing units. And the other is mainly privately owned. David then goes on to state that historically the two different sectors we always been in conflict but as both industries increase in size and profit margins increase so too is conflict.

David then goes on to talk about the Dangers and remember this article was written in 1980

David states that we could if we follow this course

- Destruction of coastal fish stocks due to uncontrolled fishing efforts
- Greatly increased migration from rural areas to cities with consequent:
    - The disintegration of rural society in the coastal areas
    - Possible political instability as small-scale fishermen is driven into the hands of radical groups advocating violent action to remedy the situation.

Thus, the problem of conflict within the fishing industry is not just a matter of resource conversation, economics, or legislation. It becomes a political problem, dealing with the ultimate survival of fishing communities and the many millions of people it represents.

Key takeaways for the CastMap:

This article provides strong evidence dating back to 1980 that regulation is needed now more than ever. CastMaps' focus is on exactly this, ensuring the regulation of coastal fishing and the protection of all related wildlife.

https://digitalarchive.worldfishcenter.org/handle/20.500.12348/3672

5. Individual fishing quotas

Summary

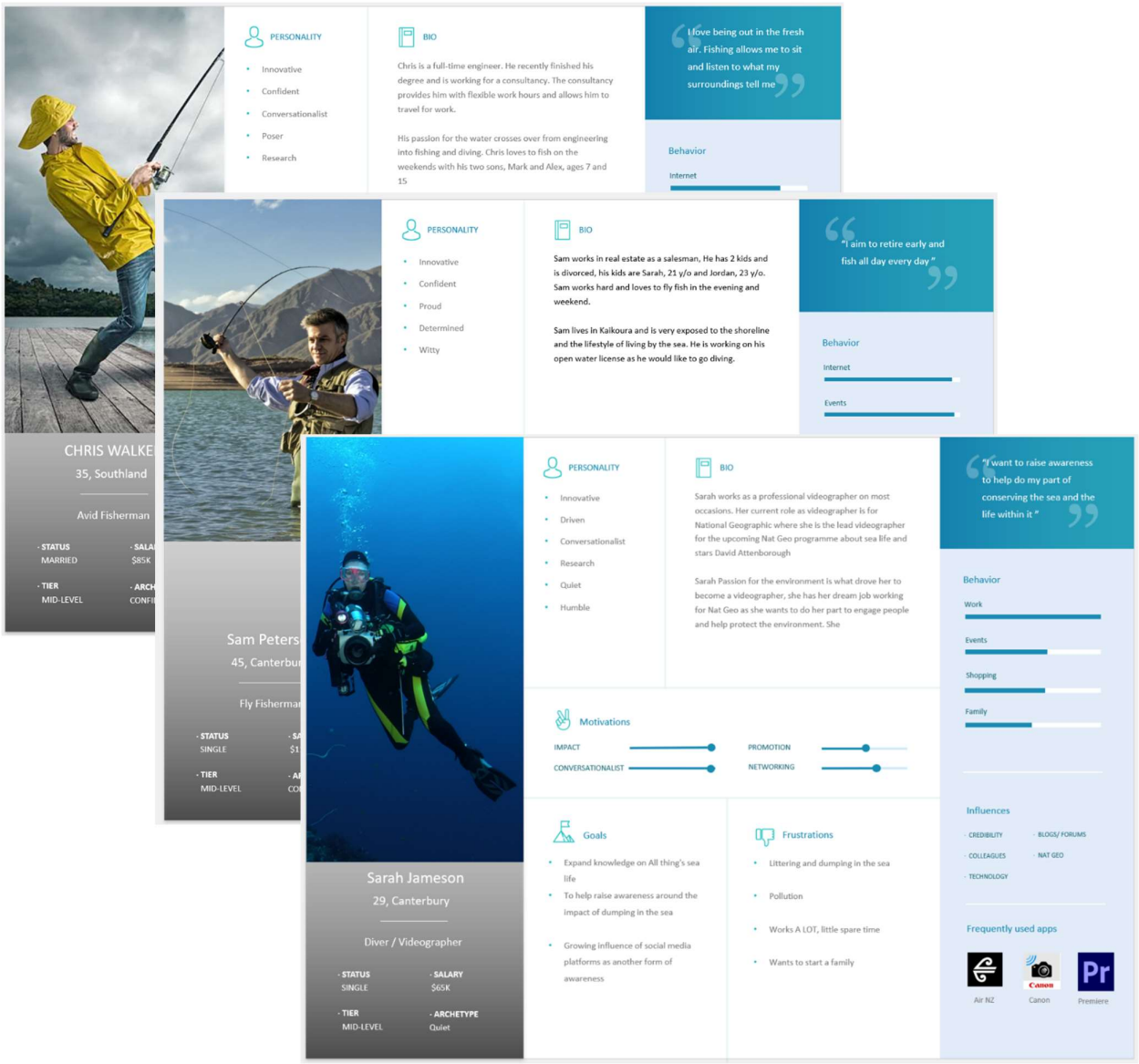
This article was written and published by Beth Baker published in 1999. The article itself talks about the different ways in which commercial fishermen could complete their quotas for the season. At the time this article was written commercial fishing companies would reach their quota but would still fish over their limits for increased capital gain. This in turn leads to overfishing.

Key takeaways for the CastMap:

This again provides evidence that something like CastMap is not only needed for humanity but the protection of wildlife as well. Overfishing from the general public or Commercial fisheries is a bad thing overall and we would all benefit from something that would provide a dynamic quota system.

https://academic.oup.com/bioscience/article/49/3/180/242429?login=false

Three user personas for the proposed system



When creating the personas, we looked at the different areas of the audience that we will be targeting. Fishing and diving are very different in their regard. So it was important to capture this within different personas.

Personas Link below

Persona - Chris.pptx

Persona - Sam.pptx

Persona - Sarah.pptx

UML sequence diagram

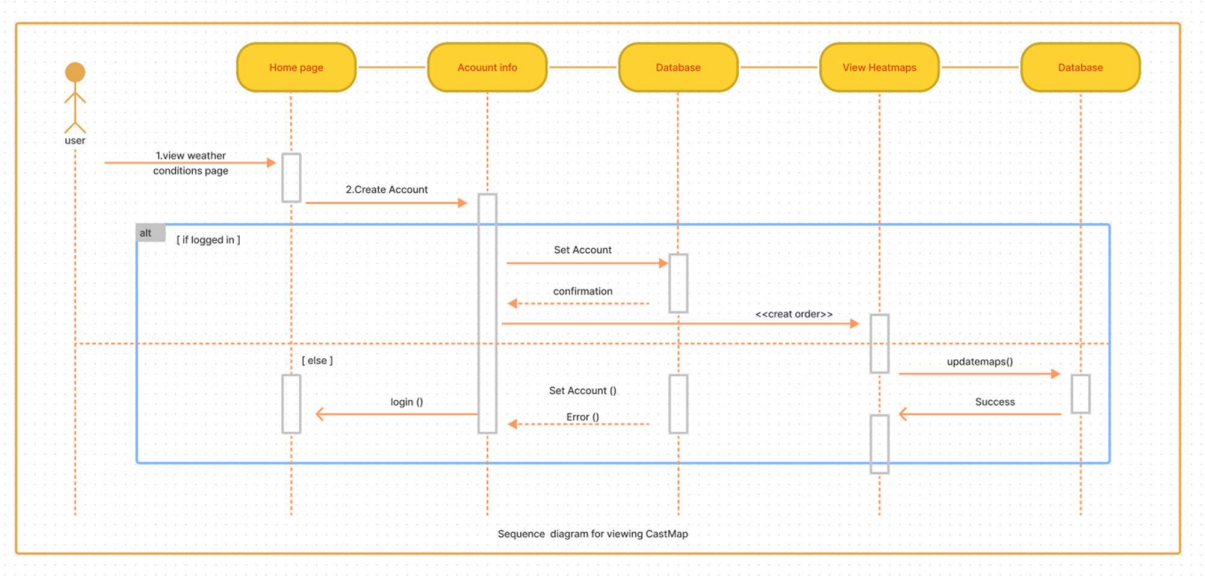


Figure 13 https://www.figma.com/file/mUi4yPtzSo3IQix4UAzaEJ/CastMap?node-id=45%3A479

I know that when it comes time to code the project in CS203 changes will be made to Functional and non-functional requirements which will have an impact on the above and below diagrams.

Use case diagram

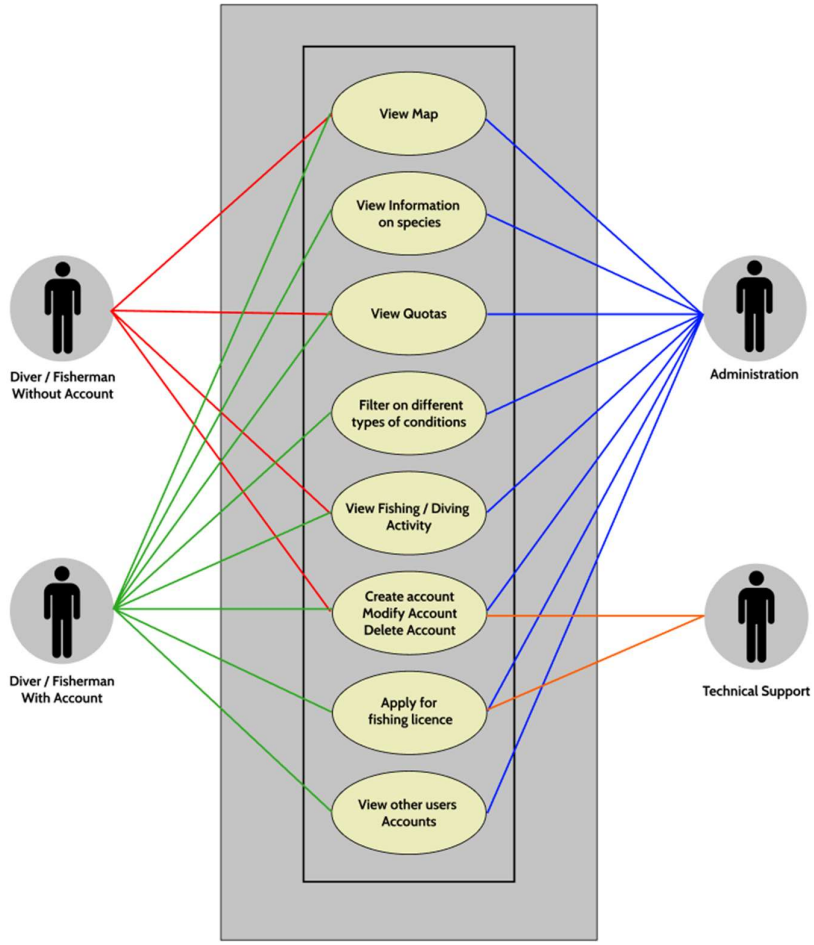


Figure 14 https://www.figma.com/file/0iMbQvMEPqH9ozD8gq3oov/CS106-Use-Case-Diagram?node-id=0%3A1

Mock-ups and digital Lo-Fi wireframes of the proposed UI.

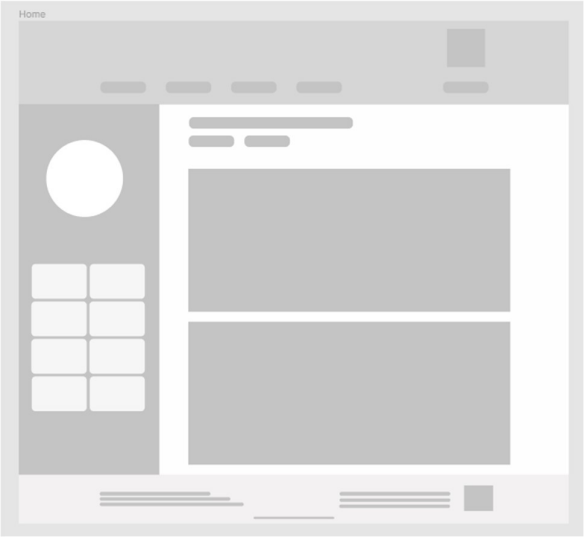


Figure 16

https://www.figma.com/file/mUi4yPtzSo3IQix4UAzaEJ/CastMap

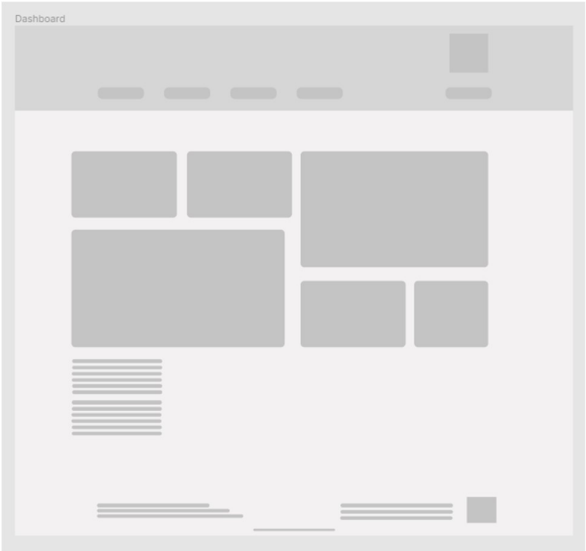


Figure 15

https://www.figma.com/file/mUi4yPtzSo3IQix4UAzaEJ/CastMap

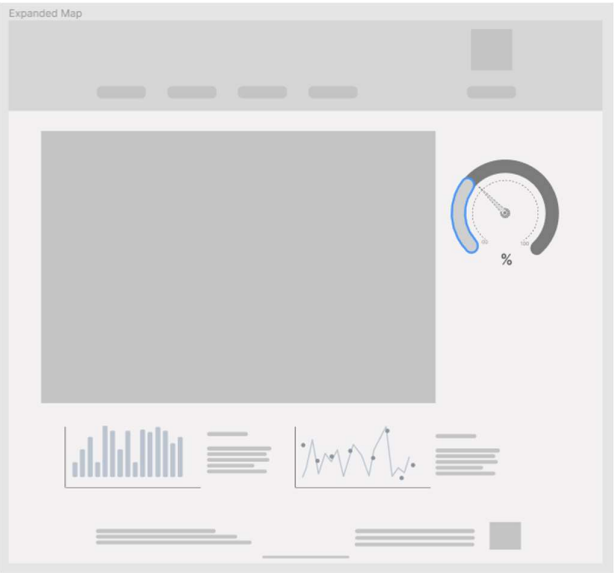


Figure 18

https://www.figma.com/file/mUi4yPtzSo3IQix4UAzaEJ/CastMap

Figure 17

https://www.figma.com/file/mUi4yPtzSo3IQix4UAzaEJ/CastMap

Mid/Hi-Fi wireframing.

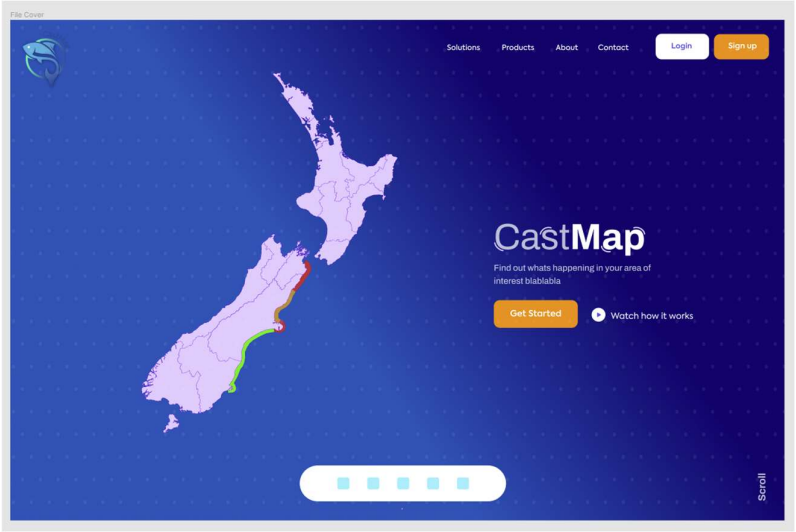


Figure 19https://www.figma.com/file/mUi4yPtzSo3IQix4UAzaEJ/CastMap

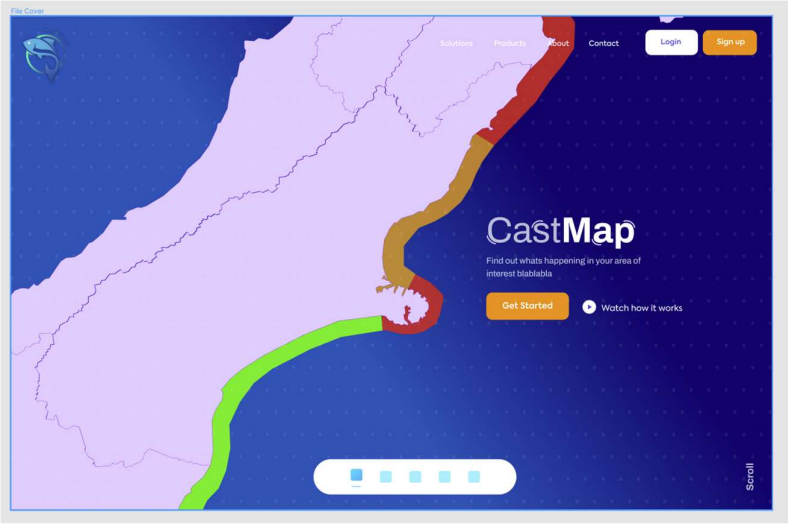


Figure 20https://www.figma.com/file/mUi4yPtzSo3IQix4UAzaEJ/CastMap

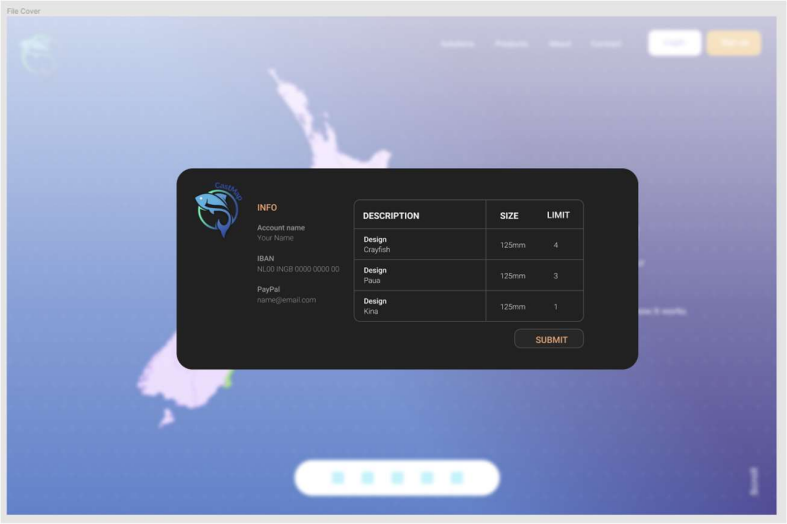


Figure 21https://www.figma.com/file/mUi4yPtzSo3IQix4UAzaEJ/CastMap

Usability Testing of the Lo-Fi wireframes with at least five users

User testing was completed by sitting down with the user and presenting the Lo-fi wireframe and questions.

1. Please locate the basic buttons and interactions were in the Lo-fi (back, menu, login)
2. Please point to the area you think the graphical output would be located (Maps/data)
3. Multiple areas contain stats, do you feel there is too much to view or just enough?
4. Based on the concept of the web app and Lo-fi can you think of any Integration that would be useful to include?
5. Would prefer to log in or use guest access to share data?
6. How do you like the titled layout on the main page?

Observations

User 1 Name: Jamie Long (23y/o)
Date: 05/04/22

Observations/Questions:

1. Basic button functions were found as expected, mentioned header was cluttered
2. Maps and graphical data are found easily
3. Two many
4. Weather, personal tracker
5. Login if there was an advantage to the user
6. Again, too cluttered

User 2 Name: Sarah Baynes (29y/o)
Date: 05/04/22

Observation/Questions:

1. Most buttons found based on general knowledge
2. A user pointed to tiles before larger map squares some confusion
3. Too crowded
4. Local risks i.e., rockfall warnings for remote areas
5. Would use guest
6. Wondered if it might be too much info

User 3 Name: Brian Long (52y/o)
Date: 06/04/22

Observation/Questions:

1. Some prompting was needed to find the location of the buttons, instruction may not have been clear
2. Maps and data found
3. Over-crowded
4. Swells and additional info for fuel and boat ramps
5. Would log in and store personal info like a fishing license
6. Too much info

User 4 Name: Mark Longman (43y/o)
Date: 07/04/22

Observation/Questions:

1. No issues finding functions
2. Found maps purely due to size, like the expanded function
3. Crowded
4. Links or locations to find fishing equipment local to the fishing area, weather
5. Yes, I like the idea of having one-stop-shop
6. Could tell, without more context

User 5: Name: Chris O'Brien (40y/o)
Date: 07/04/22

Observation/Questions:

1. Found the button area after some time
2. Maps found easily
3. Enough
4. Local shore info like high tide and weather
5. Would register and login
6. Didn't mind the tiles, depends on what is depicted

User Testing Summary

Overall, the Lo-fi testing was great, we discovered much about the complexity. The Lo-fi is a touch complex and needs to be trimmed down. CastMap's concept is based on making it easy for the users to upload and get information. Too many buttons and pages take away from the UI/UX. this will be trimmed down to ensure we meet a simpler design.

Standard basic buttons such as 'Back, menu, and login/register' are located universally and seemed easy for most to find. We expected this to be the case as we deliberately kept to standard conventions with these buttons.

A-List of suggested integrations was generated through this process:

1. Weather
2. Local info (fuel, fishing shops, and boat ramps)
3. Sea and shore info (swells, high tide, etc)
4. Local warnings (especially from remote areas e.g., rockslides, tsunamis)

Finally, overall people weren't sure about the UI/UX, and will need to be looked at in further detail, likely simplified. We were happy with this first round of user testing and will adjust and pivot to the user's needs. User need is important simply because without users no data will be shared.

Interviews

In this section, we have gone out and interviewed members of the public that will relate to CastMaps. The 3 members are as follows.

Robbie (Ngāi Tahu - https://ngaitahu.iwi.nz/)

Robbie's background

I met Robbie while on a family trip to magnet bay at the bottom of the bank's peninsula with the family where it explained that he is on the board for Ngāi Tahu. Robbie's job is to provide increased quota for families where a Tangihanga or mārena (Funeral or Wedding) might be coming up. This is a limited-time deal allowing the families to take advantage of what mother nature must provide. When speaking with Robbie on CastMap and what Alex and I are looking to offer within our application he had a few suggestions.

- Robbie finds it hard when having to provide a formal application and the processing time.
- Robbie thinks that with an application like this it will become a lot easier to determine who has a quota and how much they are allowed to take.

Questions directed to Robbie

How often do you get requests for increased quotas?

In terms of requests normally, the busier season is around summertime when fishing and diving and outside activities pick up in general. Natural Weddings follow suit.

Do you see anyone from the general public without increased quotas taking more than what is specified on the New Zealand fishery's Website?

Now and again, I will catch a group of people with catch bags where catch bags are larger than they should be. It's the ones who don't care about the size that I have a problem with.

How do you think something like CastMap could help you out?

When I think about everyday use and what I would be looking for I want an application that has it all. Typically, I will go to 3 different websites before I go for a dive because there isn't an all-in-one package yet that displays Swell, Wind, and Visibility. Having CastMap on the market would reduce the need to go all over the internet every time before going out for a dive.

What is some Website you utilize for Fishing and Diving I.e., checking swell, wind, visibility?

- Windy.Com
- MagicSeaweed.com
- Metservice

What are your thoughts on creating an account for a profile view?

Takeaways

After chatting with Robbie throughout our stay it was clear that Robbie thinks we live in a world where we should take advantage of automation where we can. Robbie has a problem where CastMaps could automate a process that is typically drawn out and very long-winded.

Nick Morris (sortitout - https://sortitout.co.nz/)

Nick Morris's background

Nick runs a business called Sort IT Out. Sort IT Out is an experienced locally owned IT Support and IT Managed Services company formed in Christchurch. Nick provides software to businesses all over Christchurch where needed. The reason I decided to contact Nick for an interview is that he also is an avid fisherman as well. Nick can provide feedback with experience on the software side and the fishing side as well.

Questions directed to Nick Morris

How often do you go fishing?

I typically go fishing down south in the canals whenever I can get down there. Trout and salmon fishing is the game I enjoy the most.

Where do you go to get information about the types of species you are fishing for? Information like what they eat, breeding patterns, and habitation.

New Zealand fisheries have this type of information displayed on their website, but it is all in PDF format. The website itself is not that user-friendly with poor UX/UI design just making it a bad experience overall to visit. Accessing this site through your phone is impossible as my phone has trouble opening PDFs.

How do you think something like CastMap could help you out?

It would be nice to have information about what I'm looking to catch on an application that I can accesses from everywhere, Information that could be updated freely

What is some Website you utilize for Fishing and Diving I.e., checking swell, wind, visibility?

- Windy.Com
- MagicSeaweed.com
- Metservice

What are your thoughts on creating an account for a profile view?

Having to fill out a profile form is another step in the process when going for a dive or a fish. It would be nice to have if there were added features like more in-depth detailing around the activity that you are performing. But personally, if it doesn't largely benefit me, I probably wouldn't consider it.

What are your thoughts on filling out forms relating to fishing or diving?

Again, it is another step in the process. I would need to see what CastMaps Profile benefits would be before making that step.

Takeaways

The general takeaway from Nick is that the information currently provided is not easily accessible on all devices and the site itself is outdated with poor UX/UI design. He goes on to say that this is an area of the market that has not yet been captured. CastMap could provide information on different species displayed on cards with information ranging from quota limits for that period to what they eat, breeding patterns, and habitation.

Sam Hemopo (Hemopo Builders)

Sam's background

Sam is a local builder in Diamond harbor. Sam is a close friend and dives a lot over the summertime. Sam typically tries to avoid technology where he can but realises that sometimes it is just needed. Sam typically when going for a dive goes up to Kaikoura so he needs to check the conditions as the trip is around 3 hours there and 3 hours back. Sam takes advantage of Windy.com for information on wind and swell and he uses magic seaweed to double-check on the swell. He calls up friends in Kaikoura to check what the current visibility is like as most sites are not that accurate.

Questions directed to Sam

How often do you go diving?

Normally I will start to go diving around the start of summer and then go right through until March the next year typically 2 – 3 times a month. The condition starts to get a bit cold after that.

Where do you go to get information about the types of species you are fishing for? Information like what they eat, breeding patterns, and habitation.

Normally diving in Kaikoura, I will target Crayfish, Paua, Butterfish, and blue cod. I have been diving there now for over 10 years.

How do you think something like CastMap could help you out?

I would normally use your standard websites to check conditions but the level of detail in the information I need in planning a trip to Kaikoura needs to be at a higher level than normal because Kaikoura is not just down the road, It's a 6 – 7-hour trip there and back. If CastMaps could offer levels of activity/interest in the area I could better plan ahead. For example, if there is a high level of activity with diving, I might consider diving somewhere else closer because the chances of catching something would be reduced.

What is some Website you utilize for Fishing and Diving I.e., checking swell, wind, visibility?

- I normally just use the standard websites like: Windy.Com, MagicSeaweed.com, and Metservice

What are your thoughts on creating an account for a profile view?

Looking at other applications I have noticed that while typically you can create an account there is also a paid version or option which provides extra benefits to the customer/user. CastMaps could take advantage of this where sign-up would not be required but in doing so there would be added functionality within the application itself.

What are your thoughts on filling out forms relating to fishing or diving?

I don't like filling out forms but sometimes you just have to for better functionality or more direct information.

Takeaways

Chatting with Sam it sounds like there is a bigger than a normal deal of planning involved with his preparations for diving. Sam uses whatever tools he can get his hands on to ensure that the trip is worth it.

References

Kapetsky, J. M., & Aguilar-Manjarrez, J. (2007). *Geographic information systems, remote sensing, and mapping for the development and management of marine aquaculture*. Food & Agriculture Org.

Xu, G., Shi, Y., Sun, X., & Shen, W. (2019). Internet of things in marine environment monitoring: A review. *Sensors*, *19*(7), 1711. https://doi.org/10.3390/s19071711

Karlsson. (2016, October). *A national data collection framework for recreational fishing*. Search open access publications in the SLU publication database - Open access publications in the SLU publication database. https://pub.epsilon.slu.se/14150/

Conflict within the fishing industry. *(n.d.). WorldFish Repository Home. https://digitalarchive.worldfishcenter.org/handle/20.500.12348/3672*

Individual fishing quotas—A complex and contentious issue. (1999, March 1). OUP Academic. https://academic.oup.com/bioscience/article/49/3/180/242429?login=false

(n.d.). Figma. https://www.figma.com

The state of JS 2021. (n.d.). The State of JS 2021. https://2021.stateofjs.com/en-US/libraries/front-end-frameworks/

What is svelte? A comparison between svelte & react. (2021, September 24). TechVariable. https://www.techvariable.com/blog/what-is-svelte/

Interactive results. *(n.d.). RawGit. https://rawgit.com/krausest/js-framework-benchmark/master/webdriver-ts-results/table.html*