Requirements

Explanation of the choice of techniques used to elicit the requirements

In order to elicit requirements for the web application, certain techniques listed in BABOK are used. These include brainstorming and interface analysis.

Brainstorming:

This is the main technique used because it is collaborative and generates a large number of ideas in a short amount of time. During a brainstorming session, the team generates as many potential requirements as possible without evaluating or criticizing them. This allows for the exploration of a wide range of possibilities and can help to identify requirements that may not be considered otherwise.

Interface analysis:

This technique focuses on identifying the connections between different components of the web application, which can then be used to elicit more specific requirements. A key part of this analysis involves identifying what interfaces are needed for each stakeholder. This technique is useful because provides an overview of the application's interoperability and can prevent difficulties when integrating multiple components during development.

Requirement specification method

In order to specify the requirements, user stories are written and compiled in the table seen below. A user story is a short, concise description of a feature or functionality that will be included in the web application, written from the perspective of the user who will benefit from it. User stories are a useful method for specifying requirements because they are easy to understand, flexible, and help to focus the development team on the needs and priorities of the end users.

Along with user stories, acceptance criteria and function/non-functional specification are shown in the table. These provide further detail to each requirement and will help the development team further on. These acceptance criteria are also flexible so they can change throughout the development cycle.

Requirement prioritisation method

In order to prioritise/rank the requirements, the MoSCoW method is used. It stands for "Must have", "Should have", "Could have", and "Won't have this time." This method allows the team to focus on the most important and feasible requirements first, while still considering the potential value of less essential requirements. It can also help to ensure that the team has a clear understanding of the priorities for the project and can make informed decisions about how to allocate resources and time.

Final prioritised set of requirements

Number	User story	Acceptance criteria	Functional /Non-functional	Priority
US01	As a fisherman, I want the app to help me locate and catch "marketable" crayfish	• Identify the locations where there is the highest likelihood of a particular crayfish • Tells the user which trapping method has the highest number of captures at that location • Ability to filter and sort crayfish data based on characteristics such as size, weight, gender, trapping method, site	Functional	M
US02	As a scientist, I want the app to give summary statistics for the data I select	• Display the mean length, weight for the crayfish selected from the dataset • Display the min and max values for each criterion • Display the sex ratio of crayfish on each site	Functional	M
US03	As a user, I want to access and use the web app smoothly 24/7	• Utilises a server to host the webapp for a large number of users (20,000+ users) without crashing or slowing down • Install caching plugins	Non-functional	C
US04	As a user, I want the app to support Chrome, Firefox, and Safari so that I can use the app with any browser	• Support recent 5 versions of major browsers • Comply with cookie policy	Non-functional	C
US05	As a scientist, I want to have download functionality so that I can use the data and charts for further analysis on my personal device	• Download datasets as .xlsx and .csv files • Download charts as .jpg, .png and .tiff	Functional	S
US06	As a scientist, I want to easily input and	• Allows the user to input their own data onto the app, after which the algorithm analyses the data	Functional	M

	process my own data into the app	• A guide showing how to enter data correctly so there will be no errors when processing the data		
US07	As a scientist, I want the app to show me regular updates on crayfish numbers on the home page	• Website displays the current number of crayfish on the home screen	Functional	C
US08	As a user, I want to use the app on my mobile device	• Automatically scale to fit screen size • Ability to switch between computer and mobile version	Non-functional	C
US09	As a user, I want my own account to work on	• Login/logout system • User specific profile to see personal changes to the data	Functional	M
US10	As a scientist, I want to have different ways of visualising my data	• Bar chart function • Time series function • Pie charts	Functional	S
US11	As a scientist, I want the app to store my data and data manipulations	• Save button	Functional	S
US12	As a user, I want to customize the visual settings of the app	• Light and dark mode	Functional	W
US13	As a user, I want to contact someone when I need assistance	• Online forum page for the collaboration of users to ask and assist one another • Contact information (email/number) for users to ask for more help	Functional	C
US14	As a user, I want to create an account using my email	• 'Create account' page • Verify email	Functional	M
US15	As a scientist, I want to see the	• History page	Functional	S

	changes I make to the data	• Shows a description and date for each change		
US16	As a user, I want a help page to describe how I can use the app	• Help page shows how to import data, show specific data, filter data, and manipulate graphs • FAQs	Functional	C
US17	As a user, I want to be notified with any updates and changes regarding my account and work to my email	• Display and alert notifications on the webpage • Sends messages to user's email addresses to further alert them	Functional	S
US18	As a user, I want to change my password and other account details	• 'Forgot my password' function • Change my password function • Change username function	Functional	M
US19	As a user, I want to see a clear overview of the app's functionality	• Home page showing the functionalities of the dashboard • Font size should be larger than 12 on a mobile screen and 16 on a desktop computer screen • Clear graphics	Functional	S
US20	As a user, I want to zoom in or focus on a particular graph/chart	• Enlarge a particular section of the dashboard • Dim the background	Functional	C
US21	As a user, I want to save graphs as images or copy them to clipboard	• 'Save as image' button for each graph • Graphs can be copied as images • Either .jpg or .png format	Functional	C
US22	As a scientist or university affiliate, I want to sign in via my institution	• SSO (Single Sign-on) feature where some users can login with affiliated accounts	Functional	W

US23	As a user, I want the app to show me specific information from parts of a graph when I select them	• If a segment is selected in a pie chart, a box pops up showing the data and number/proportion for that segment	Functional	C
US24	As a user, I want the dashboard to show relevant information for my needs	• Dashboard for fisherman is simpler: showing the general graphs, manual search function – no import/editing functionality • Dashboard for scientists is more detailed: in addition to what the fishermen see, they can access the history of all their changers and are able to import and edit their data	Functional	S
US25	As a user, I want the app to be available in different languages	• Supports 5+ languages (English, Chinese, French, etc.)	Functional	W