Marine Aquarium’s Drawing Application’s Project Plan

Team M

Table of Contents

1.0 Executive Summary

1.1 Overview

A complete software solution — available both on old mobile devices and the web — which allows users to draw their visual interpretation of the National Marine Park while visiting one of their events or their website and to provide the staff of the National Marine Park the capability to easily create, manage and store the drawings and data that is produced.

1.2 Key Objectives

1. Develop a user-friendly and intuitive Android application which allows event managers/scorers to conduct drawing activities and collect the drawings and data while also allowing participants of the drawing activity to sketch their perception and ideas of the National Marine Park.
2. Develop a database which aligns with the requirements of the project and provides the capability to store gathered data.
3. Develop an admin portal where the event managers/scorers can view, manage and score the drawings and data that have been gathered during events.
4. Develop a web-based drawing application featuring a simple and intuitive user interface which is embeddable onto the National Marine Park's website and allows visitors to participate in a similar drawing activity.

1.3 High-level Timeline

1. 20th Nov 2022 – Initial Project Plan
2. 15th Jan 2023 – Drawing Application Prototype
3. 5th Mar 2023 – Finished Android Drawing Application
4. 19th Mar 2023 – Admin Portal
5. 20th Apr 2023 – Project Completion

1.4 Delivery Date

10th of May 2023

2.0 Project Description

2.1 Background

The National Marine Park holds events at various locations, during these events the staff gets people to participate in a drawing exercise. The purpose of the exercise is to collect insights about people's perceptions of the National Marine Park. These drawings can help in figuring out what visitors find interesting, what they enjoy the most, and what they would like to see more of.

By analysing the drawings, the park management can identify the areas that require improvement or development, and accordingly introduce new exhibits, activities, or other offerings to enhance the visitor experience. For instance, if there are fewer drawings related to a particular topic, it can be an indication that visitors are less interested in that topic, prompting the management to introduce more exciting exhibits and activities in that area.

2.2 Vision

The project vision is to create a complete software solution — available both on old mobile devices and the web — which allows users to draw their visual interpretation of the National Marine Park while visiting one of their events or their website and to provide the staff of the National Marine Park the capability to easily create, manage and store the drawings and data that is produced.

The software must be straight to the point and feature a simple user interface which is easy to use for people of all ages while unsupervised.

2.3 Target Users

The project should be designed to accommodate three types of target users:

1. The event manager/scorer — individuals which will be managing the events and getting the users to participate in an activity where they will be drawing their perception of the National Marine Park using tablets. Additionally, these individuals will be scoring the drawings collected during events.

   These users require the ability to hold events, get users to participate in a drawing session and gather and store the drawings and other data in a database where it can be reviewed and scored later on.

2. An event participant — individuals which will be visiting the National Marine Park and participating in the activity of drawing their perception and thoughts of the National Marine Park. These users' age and technical abilities will vary, therefore, the user interface of the software must be simple and easy to use.

3. A participant using the National Marine Park's website — individuals who will be participating in the activity of drawing their perception and ideas of the National Marine Park while visiting their website. Similarly, these users' age and technical abilities will vary. The application must be intuitive and simple as the users will not be supervised in any way during the activity.

2.4 Use Cases

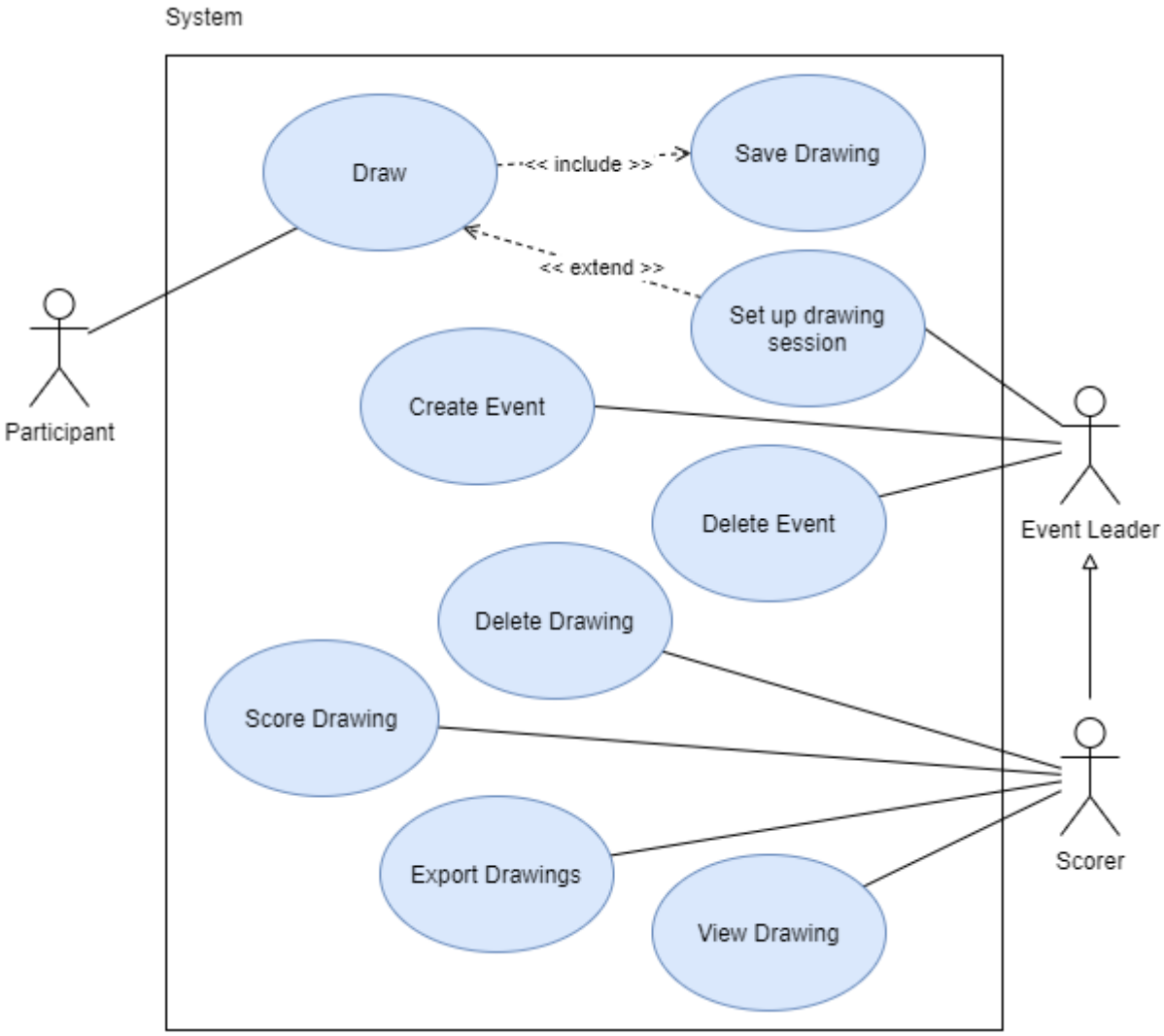

Figure 1, A Use Case Diagram

3.0 Scope

3.1 Objectives

3.1.1 Android Drawing Application

Develop a user-friendly and intuitive Android application which allows event managers/scorers to conduct drawing activities and collect the drawings and data while also allowing participants of the drawing activity to sketch their perception and ideas of the National Marine Park.

3.1.2 Database

Develop a database which aligns with the requirements of the project and provides the capability to store gathered data.

3.1.3 Admin Portal

Develop an admin portal where the event managers/scorers can view, manage and score the drawings and data that have been gathered during events.

3.1.4 Web Drawing Application

Develop a web-based drawing application featuring a simple and intuitive user interface which is embeddable onto the National Marine Park's website and allows visitors to participate in a similar drawing activity.

3.2 Functional Requirements

Functional requirements that have been identified so far:

1. The drawing application needs to be accessible from Android devices as well as the web.
2. Have an interface for viewing and evaluating images stored in a database.
3. Have a database to store the gathered information and a way to access the data from various endpoints.
4. The ability to export data to Excel.
5. The Android application must be functional while offline. Data generated offline needs to be stored locally until an internet connection is established.

3.3 Non-functional Requirements

Non-functional requirements that have been identified so far:

1. The user interfaces must be accessible and intuitive.
2. The Android application must be lightweight and functional on low-end devices.
3. Access to the database must be protected by authentication and authorisation.

3.4 User Stories

3.4.1 Event Manager/Scorer

1. As an event manager/scorer, I want to easily view images so that I can evaluate and score them on their characteristics.
2. As an event manager/scorer, I want to be able to easily export all data to CSV format so that I can view it using Excel or other spreadsheet software.
3. As an event manager/scorer, I would like to set up an event session so that I can get visitors to participate in a drawing activity and save the drawings as a collection.
4. As an event manager/scorer, I would like to upload pictures at the end of the session so that I can review and score them later.

3.4.2 Visitors

1. As a visitor, I would like to be able to draw a picture so that I can capture my perception of the National Marine Park.
2. As a visitor, I would like to be able to submit my drawing so that the scorers can review it later.
3. As a visitor, I would like to tag my drawing with my age so that the scorers know the age of the person who drew the picture.

3.5 Out of Scope

Features within the drawing applications such as layers, rotating the drawing, colour gradients and a wide range of brushes are out of the scope of this project.

The admin portal will not provide the capabilities to create user accounts or manage their level of access to the functionality and data of the admin portal.

4.0 Project Organization

4.1 Team Roles & Responsibilities

The team consists of four developers. Each member has unique responsibilities related to the project in addition to writing the code.

4.1.1 Client Liaison – Kieran Jeffery

Kieran is responsible for organising the meetings with the client, taking notes during the meetings, getting feedback, organising user testing and being the main point of contact for the client.

a) Email

kieran.jeffery@students.plymouth.ac.uk

4.1.2 Scrum Master – Oliver Barty

Oliver is responsible for the team meetings where progress is reviewed and the work to be done is discussed and agreed upon. Olly is the person who will be guiding the team in the effective use of Scrum and Agile and lead the planning of the sprints.

a) Email

oliver.barty@students.plymouth.ac.uk

4.1.3 Product Owner – Emma Freeman

Emma is responsible for identifying product features and attributes. She will be reviewing the work done, identifying and driving the testing requirements and will lead the prioritisation of the project backlog.

a) Email

emma.freeman@students.plymouth.ac.uk

4.1.4 Technical Lead – Justas Galminas

Justas is responsible for ensuring Version Control is in place, he will manage the project's repository ensuring appropriate branching, commits and comments. He will lead in identifying and communicating appropriate design patterns and ensuring the quality of the code base.

a) Email

justas.galminas@students.plymouth.ac.uk

4.2 Communication Plan

4.2.1 Client Communication

The client will be invited to join the Kanban board and client meetings will be held every two weeks with the client liaison. Meetings will include discussion on the progress of the project. Certain meetings will involve the client being shown prototypes and design documents for feedback.

4.2.2 Team Communication

Weekly meetings will be held where we will review the progress of the project, discuss any issues or ideas that arise and decide which user stories/tasks should be worked on next.

Minutes and participants will be recorded during each meeting so we can reflect and review what has been said and decided.

4.2.3 GitHub Workflow

How the project will be version controlled. Build actions will be set up to run for each pull request to make sure the project builds without any issues.

a) Branching

Create a new development branch for each sprint.

Each task on Trello will have a unique identifier. Each person should branch out from the development branch of the current sprint and use the identifier as the name.

b) Merging

When merging task branches into the development branch, open a new pull request and request the technical lead to review it. Make sure to squash all the commits into one commit with a meaningful name and an additional description if needed.

If the instructions mentioned above are followed, sprint branches will not need to be squashed and can simply be merged. This should make the commit history clear and insightful.

c) Repository

https://github.com/Plymouth-University/comp2003_2022-team-m

4.3 Development Life Cycle

Throughout the development of this application, the Agile development lifecycle will be followed. Agile is an iterative software development lifecycle which allows the owner and users of the software to be more involved during the development, giving plenty of room for change and ensuring that the software meets the user's needs. The development lifecycle consists of four phases.

4.3.1 Initiation

During this phase, initial ideas will be discussed, and research will be carried out. This will include careful consideration of languages, frameworks, libraries as well the overall architecture of the application.

A project plan will be put together containing initial requirements and other information about the project. The requirements will be broken down into user stories.

4.3.2 Planning

This phase is focused on designing the architecture and interfaces of the application. UML diagrams are created to help visualise the functionality of some of the more complex parts of the application, as well as how users will interact with it.

The user stories are also decomposed into smaller tasks which are added to a sprint backlog. In this case, a Kanban board will be used to store all the tasks in one place.

4.3.3 Implementation

After the initial planning and design phase, the project will be developed in two-week sprints where the team will work through the sprint backlog and complete the tasks that are the most important at the time. During the whole development lifecycle, the team will attend weekly meetups to discuss the progress, identify any issues, and make sure that the project is running smoothly.

4.3.4 Testing

During each sprint, functional and technical tests will be carried out to ensure that the project functions properly and meets the specified requirements. For more information on testing find the User Testing and Technical Testing sections below.

The project plan and sprint backlog will also be adjusted as needed to accommodate any changes that have been identified during testing and development.

5.0 Implementation Details

5.1 Components

1. Android Drawing Application — an application, to be used on Android tablet devices, which is developed to suit offline use.
2. Database — a database where event and drawing data are stored.
3. API — an interface which is used to communicate between applications and the database.
4. Admin Portal — a web application where data related to the project can be managed.
5. Web Drawing Application — a drawing application which can be used through web browsers and embedded on the National Marine Aquarium's website.

5.2 Front-End

After exploring the idea of a Progressive Web App (PWA), we decided not to create one and develop two separate front-end applications, due to PWA performance on older devices.

5.2.1 Web

For the web-based client applications, this includes the scorer (admin) side and the drawing application itself, TypeScript and the React UI library will be used to make development easier and quicker. Additionally, Tailwind will be used to style the user interface.

5.2.2 Android

For the mobile drawing application, Java and Android SDK will be used.

5.3 Authentication

There is no real authentication that is necessary for the drawing application itself as there is no need for user authentication. The application will simply need an API key to be able to post data to the back-end API.

5.4 Back-End

5.4.1 Language/Framework

The back end will be built using C# and the ASP.NET Core 6.0 to ensure long-term availability.

5.4.2 Database

Microsoft SQL Server will be used for the database.

5.4.3 Image Storage

Images themselves will be stored on the cloud via a storage provider and the information identifying the image will be stored in the database.

5.4.4 Authentication

The back-end API will require two types of authentication methods:

- End-points related to the drawing application, namely the image uploading end-point, will be secured by an API key.
- End-points related to the admin dashboard will be secured using credentials. Internally the API will use JWT and refresh tokens to provide the authentication functionality, including sessions.

6.0 Project Schedule

6.1 Sprints

6.1.1 Sprint 1 (7th Nov- 20th Nov)

a) Tasks

1. Discuss, create, and approve a general project plan.
2. Ensure the plan meets all the criteria gathered during the initial client meeting.
3. Highlight key elements that are required and research implementation techniques.
4. Discuss the completed plan with the client and make any necessary adjustments.
5. Prepare the GitHub repository & ensure all members have access to and are confident using Git.

b) Deliverables

1. Completed initial project plan

6.1.2 Sprint 2 (21st Nov- 4th Dec)

a) Tasks

1. Discuss the feedback received from the client meeting.
2. Decompose user stories into smaller tasks and create a sprint backlog.
3. Create UI wireframes/designs for all components of the project.

Checklist

1. Create initial wireframes/mock-ups.
2. Carry out user testing.
3. Using the project plan, discuss and decide upon implantation techniques.
4. Model the architecture of the project using UML to provide a functional overview.

b) Deliverables

1. Low-fidelity prototype.
2. Architecture/Design diagrams.
3. Sprint backlog.

6.1.3 Sprint 3 (5th Dec – 12th Dec / 9th- 15th Jan)

a) Tasks

1. Review client feedback and adjust the project vision and plan accordingly.
2. Develop the functionality of the client-side applications.

Checklist

1. Start developing the UI and core functionality of the Android and Web client-side applications.
2. Receive client feedback and add requested changes to the sprint backlog.
3. Prepare for the interim submission.
4. Perform basic testing on the current application.

b) Deliverables

1. Drawing application UI.
2. Drawing application core functionality.

6.1.4 Sprint 4 (16th – 23rd Jan / 30th to 5th Feb)

a) Tasks

1. Arrange user testing and create a testing plan and feedback forms for the client to carry out user testing of the application.
2. Design the UI for the Android drawing application.
3. Start implementing the UI of the Android application.
4. Do extensive research on authentication.

b) Deliverables

1. Android drawing application design.
2. Basic implementation of the Android drawing application.
3. Comprehensive plan on how authentication and authorisation will be implemented.

6.1.5 Sprint 5 (6th Feb- 19th Feb)

a) Tasks

1. Implement the user flow and UI of the Android application including the ability to start an event, draw, save drawings and upload drawings.

   Checklist

   a) Ability to enter the event ID and start the event which navigates to the drawing view.
   b) Ability to see the number of drawings that are stored locally.
   c) Ability to navigate to the view where drawings can be uploaded to the server.
   d) Ability to save drawings (drawer's age must be included with the image). When a drawing is saved the canvas should be cleared, and ready for the next drawer.

2. Implement the Android drawing functionality such that the user can draw, select colours, brushes and line width.

   Checklist

   a) Ability to select from a small range of pre-defined colours — red, yellow, orange, green, blue, pink, grey and black.
   b) Ability to change the thickness of the brush from a pre-defined range (6 options).
   c) Ability to choose from four different brush variants — pencil, pen, crayon and eraser.

3. Scaffold the API project following an appropriate structure and implement the database schema.

   Checklist

   a) Scaffold the project
   b) Implement the database schema
   c) Set up the folder and file structure following the service-repository design pattern
   d) Scaffold controllers for each resource group — Drawing, Event, User
   e) Scaffold end-points for each required operation — delete drawing, upload drawing, view drawing, update drawing, grade drawing, update drawing grade, delete drawing grade, create event, update event, delete event.

4. Create user testing feedback forms.
5. Create a user testing plan.
6. Create user testing consent forms.
7. Get and document client feedback on the project plan and the Android application using the semi-interactive Figma design.
8. Update the project plan with the most recent information and implement assignment feedback.

b) Deliverables

1. An interactive Android application which allows for navigation between the different views, drawing and saving drawings.
2. The basis of the API set-up, ready to be fleshed out in the next sprint.
3. User testing documentation.
4. Client feedback.

6.1.6 Sprint 6 (20th Feb- 5th Mar)

a) Tasks

1. Write unit tests for the Android application and ensure at least 80% coverage if possible.
2. Implement all necessary end-point functionality.

   Checklist

   a) Delete Drawing
   b) Upload Drawing
   c) View Drawing
   d) Update Drawing
   e) Grade Drawing
   f) Update Drawing Grade
   g) Delete Drawing Grade
   h) Create Event
   i) Update Event
   j) Delete Event

3. Create UI prototypes for the admin portal which match the requirements for scoring and confirm the requirements including accounts and how we plan to handle that.

Checklist

a) Login Page
b) Change Password Page
c) Forgot Password Page
d) Grading Page
e) Drawing Viewing Page
f) Event Viewing Page
g) Event Create Page

4. Implement the image storing functionality using a cloud file storage service within the API.
5. Implement the networking functionality by linking the Android app to the API and allowing for images to be uploaded.
6. Get and document client feedback on the admin UI wireframes/prototypes and project up to this point.
7. Ensure all of the project documentation is up to date including things like ReadMe's, UML diagrams, UI wireframes and communication documentation. All of it uploaded to GitHub.

b) Deliverables

1. A well-tested Android drawing application which is fully usable apart from being properly secured.
2. A fully working API which exposes end-points for storing and managing data related to the project.
3. Fully documented project to date.
4. Client feedback.

6.1.7 Sprint 7 (6th Mar- 19th Mar)

a) Tasks

1. Implement authentication and authorization within the API.
2. Implement authentication and authorization within the Android application and ensure that the API key is stored securely.
3. Write unit tests for the API and ensure at least 80% coverage if possible.
4. Implement authentication and authorization within the admin application and ensure that the API key is stored securely.
5. Implement the UI of the Admin portal based on the feedback received from the client beforehand.
6. Implement the networking functionality within the Admin portal.
7. Write unit tests for the Admin portal and ensure at least 80% coverage if possible.
8. Get and document client feedback on the current state of the project.

b) Deliverables

1. A secure API and Admin portal.
2. A working implementation of the Admin portal which is well-tested.
3. Client feedback.

6.1.8 Sprint 8 (20th Mar- 2nd Apr)

a) Tasks

1. Finish implementing the UI and functionality of the web drawing application including the ability to save drawings, providing the user's age and general location and selecting from four brush types — pencil, pen, crayon and eraser.
2. Write unit tests for the web drawing application with at least 80% coverage if possible
3. Create the poster.
4. Write up the information that is currently available on the software being produced, how to use it, etc. within the client handbook.
5. Discuss and figure out how we can host the project with an emphasis on convenience and ease of use for the client.
6. Ensure all of the project documentation is up to date including things like ReadMe's, UML diagrams, UI wireframes and communication documentation. All of it uploaded to GitHub.
7. Add the ability to export data from the database to excel format.
8. Get and document client feedback on the web drawing app UI and functionality as well as the whole project in general.

b) Deliverables

1. Web drawing application which is fully tested and functional.
2. Poster for the assignment.
3. A clear plan on how the project will be hosted.
4. A mostly-complete client handbook.
5. Exportable data.

6.1.9 Sprint 9 (3rd Apr- 16th Apr)

a) Tasks

1. Host the entire project (Web application, API and Database).
2. Finish documenting the project and finalize the client handbook, including hosting information.
3. Write the final report (might be individual or group based, to be confirmed still).
4. Create the presentation.
5. Ensure that UAT documentation is fully up to date.
6. Ensure that all project documentation and diagrams are up to date and uploaded on GitHub.
7. Run all of the unit tests and test each part of the project manually to ensure it is fully functional.
8. Create handover documentation.

b) Deliverables

1. Fully finished and hosted project.
2. User manual.
3. Handover documentation.
4. Complete project implementation documentation.

6.1.10 Sprint 10 (17th Apr- 10th May)

This sprint is slightly longer and will be used to polish up the project and ensure everything is completed, tested, documented and ready to be handed over to the client.

6.2 Milestones

6.2.1 20th Nov 2022 – Initial Project Plan

By the 20th of November 2022, the initial project plan will be completed containing information about the development of the software.

6.2.2 15th Jan 2023 – Drawing Application Prototype

By the 15th of January 2023, there will be an implementation of high-fidelity prototypes for the Android and web drawing applications.

6.2.3 5th Mar 2023 – Finished Drawing Application

By the 5th of March 2023, the Android drawing application will be fully functional and tested. The API will be working, and drawings will be unloadable to a database.

6.2.3 19th Mar 2023 – Admin Portal

By the 19th of March 2023, the admin portal will be finished where the event managers and scorers can manage the event and drawings data.

6.2.4 20th Apr 2023 – Project Completion

By the 16th of April 2023, the entire project will be completed, this includes the Android drawing application, the web drawing application, the admin portal, the API and the database. Additionally, the project will be hosted, fully tested and documented.

6.2.5 10th May 2023 – Project Handover

By the 10th of May 2023, all documents related to the project will be handed over to the client, including the source code, documentation, testing plans and user manuals.

6.3 Risk Management

The table *(see Figure 2)* below contains a list of events that may occur during the development of the project. Efforts will be made to mitigate any of these risks where possible, but in a case where the event occurs a contingency plan will be followed to resolve the issue and reduce the impact on the project's timeline, deliverables and quality.

Reference	Event	Likelihood	Impact	Risk Exposure
R1	Changes to requirements specification during development	3	4	12
R2	Illness or absence	3	7	21
R3	Coding of a module takes longer than expected	4	7	28
R4	Testing reveals errors or issues with the design	5	8	40
R5	Misunderstanding of requirements	3	7	21
R6	Change in technologies during the development	1	8	8
R7	A teammate not fulfilling their responsibilities	3	8	24

Figure 2, Risk Identification Table

6.3.1 Contingency Planning

a) Risk R1

1. Document the request for change in the requirements.
2. Discuss and evaluate the impact of the change on the project timeline and the quality of the final product with the team.
3. Update the project plan to reflect the changes.
4. Implement the changes and modify any deliverables that are affected.

b) Risk R2

1. Identify the responsibilities that are critical to the project's success and ensure they are covered by other team members.
2. Document the absence and inform the module leader.
3. Monitor the situation and adjust the contingency plan as needed.
4. Adjust the project's timeline and deliverables depending on the length of the absence.

c) Risk R3

1. Identify the cause.
2. Allocate additional personnel to assist with the task.
3. Communicate with the client.
4. If inevitable, adjust the project's timeline or the deliverables depending on the requirements and the priority of the task.

d) Risk R4

1. Identify the issue found during testing.
2. Assess the impact of the issue on the project's timeline and quality.
3. Prioritise issues depending on the impact.
4. Develop an action plan which includes the steps needed to resolve the issue.
5. Allocate personnel to the issue.
6. Implement the fixes.
7. Test the issue, ensure it fulfils the requirements and does not introduce new issues.
8. Update the project plan to reflect the changes to the project's timeline, deliverables and other relative information.
9. Communicate the issues and the step that were taken to resolve them with the client.

e) Risk R5

1. Identify the root cause of the misunderstanding and prevent it from happening again.
2. Assess the impact of the misunderstanding on the project's timeline.
3. Communicate with the client.
4. Revisit the requirements and clarify them with the team and potentially the client.
5. Identify a solution to rectify the issues caused by the misunderstanding.
6. Implement the solution.
7. Test the solution.
8. Communicate the solution to the client.

f) Risk R6

1. Stay informed and up to date with the technology trends.
2. Avoid using technologies that are new and have no or little use in the industry.
3. Find a solution to minimise the impact of the changes.
4. Implement the solution.
5. Document the impact of the changes, re-evaluate the project plan and adjust the project's timeline and deliverables accordingly.
6. Communicate with the client.

g) Risk R7

1. Identify the problem and the responsibilities not being fulfilled.
2. Communicate with the team member, discuss the issue and determine the cause of the problem.
3. Work with the teammate to resolve the issue.
4. Monitor progress and provide support if needed.
5. If the teammate does not communicate or does not improve, escalate the issue with the module leader.
6. Assess the impact on the project's timeline and deliverables.
7. Communicate with the client.

7.0 Quality Plan

7.1 Objectives

1. Find bugs, errors and defects.
2. Ensure security and usability standards.
3. Validate the requirements and ensure the project meets the client's needs.
4. Comply with standards and regulations such as data privacy laws and accessibility guidelines.

7.2 Technical Testing

During the development of the application, all parts of the system will be tested using various testing methodologies.

7.2.1 Integration Testing

Testing different modules/services together.

7.2.2 Unit Testing

Use unit testing for some of the more complex functions/components within the application.

7.2.3 Acceptance Testing

Testing the application to make sure that it meets the requirements.

7.2.4 End-to-End Testing

Testing the UI of the application to make sure that it functions as expected and is accessible to all users.

7.3 User Testing

Being the nature of this project involves users with a wide age range and technical ability, it is important that regular user testing is carried out across all phases of the project.

Initial testing will be carried out with the client using paper-based/low-fidelity prototypes to gather some initial feedback and ensure that the project requirements have been understood correctly.

With each iteration, all aspects of the project will be presented to the client where they can test it themselves or even conduct testing with the visitors during their events. Initially, we planned on conducting user testing with the visitors of the National Marine Park ourselves, but we were not able to get ethical approval for that.

7.4 Acceptance Criteria

To consider the project acceptable for release the following criteria must be met:

1. All of the unit tests must pass without any failures.
2. The software must perform all of the functionality specified within the requirements.
3. The software must meet the hardware requirements specified within the requirements.
4. The software must be free of any security vulnerabilities.
5. All known bugs must be addressed and resolved before release.
6. The software must be fully documented and have clear user instructions.

8.0 Project Deliverables

8.1 Source Code Repository

A GitHub repository containing code for all pieces of the project — Android application, web application (web drawing application and admin portal), database, and API.

8.1.1 Acceptance Criteria

1. The code is well documented and commented.
2. Each component of the project contains a ReadMe document outlining the implementation details.
    a. Android Drawing Application
    b. Web Application (admin portal and drawing application)
    c. API
    d. Database
3. All components of the project are functional and meet the functional and non-functional requirements.

8.2 Design & Implementation Documentation

8.2.1 Acceptance Criteria

1. All documentation is present in the GitHub repository.
2. All diagrams and drawings include a source file which can be easily edited.
3. Diagrams are drawn following industry standard practices.

8.3 User Manual

A user manual detailing how to use the software in its entirety.

8.3.1 Acceptance Criteria

1. The manual is written in simple language and is easy to read and understand for individuals with non-technical backgrounds.

2. The manual contains all of the necessary information regarding the use, storage and hosting of the software.
3. The manual contains step-by-step instructions on how to operate all components of the project.
4. A separate document is produced containing all of the necessary credential information for hosting and external services.
5. The user manual is included in the GitHub repository.

8.4 Testing Documentation

Testing plans and documentation.

8.4.1 Acceptance Criteria

1. Clear and extensive documentation showing how the software was tested, including the results and metrics.
2. Testing documentation is included in the GitHub repository.