National Marine Aquarium

Group M Project Proposal

Contents

Project Vision

An intuitive drawing application which allows users to sketch their perception of what the National Marine Park is. The application will likely be used unsupervised and by people of various ages, therefore, it must have a simple interface and be easy to use.

Scope

A simple drawing application which works on various devices and input methods (touch, mouse). The application should contain a small selection of brushes (pen, pencil, crayon), colours and brush sizes.

Out of Scope

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

Development Lifecycle

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.

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.

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.

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.

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.

Requirements

- The application needs to be accessible from various devices and platforms.
- An interface for viewing and evaluating images stored in a database.
- A database to store the gathered information and a way to access the data from various endpoints.
- The interfaces must be accessible and intuitive.
- The ability to export data to Excel.
- The application must be functional while offline. Data generated offline needs to be stored locally until an internet connection is established.
- A lightweight application that can run on low-end devices.

User Stories

Admin

- As an admin I want to easily view images so that I can evaluate and score them on their characteristics.
- As an admin I want to be able to easily export all data to CSV format so that I can view it using Excel.
- As an admin I would like to set up a session with the location of the session
- As an admin I would like to upload pictures at the end of the session

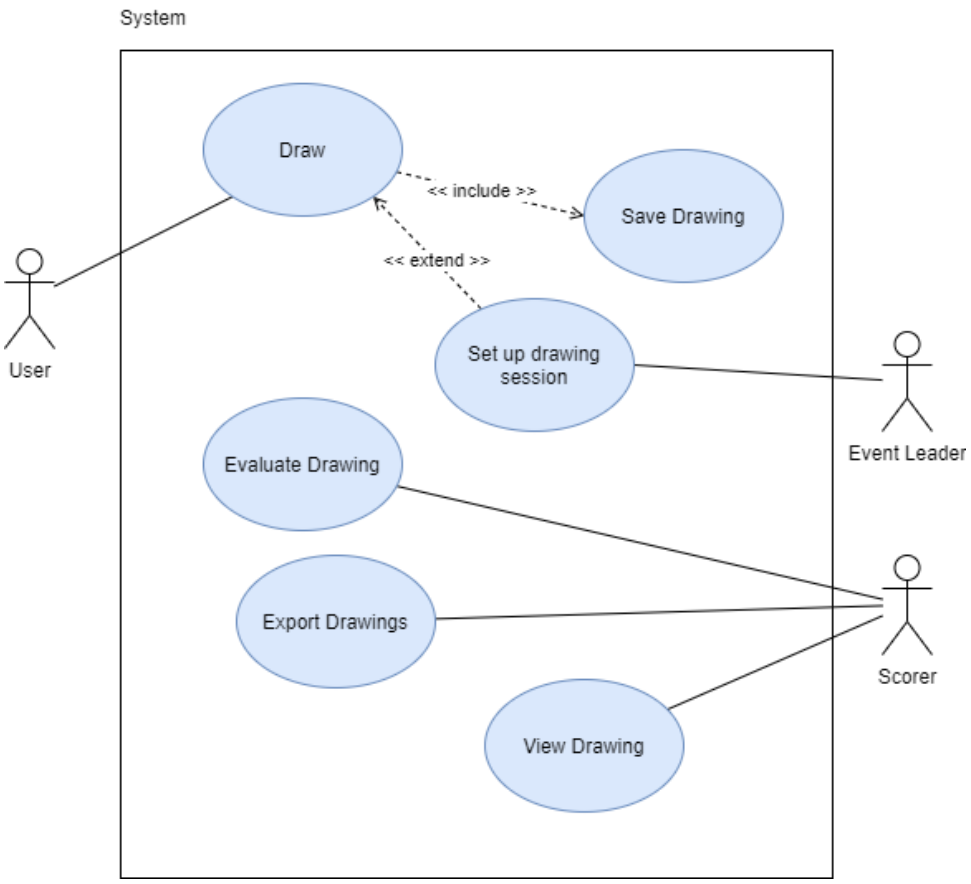
Visitors

- As a visitor I would like to be able to draw a picture
- As a visitor I would like to be able to submit my drawing
- As a visitor I would like to tag my drawing with my age

Diagrams

Use Case

A simple diagram depicting how different users will interact with the system.



Sprints

Sprint 1 (7th Nov - 20th Nov)

Tasks

1. Discuss, create, and approve upon 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.

Output

1. Completed initial project plan

Sprint 2 (21st Nov - 4th Dec)

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

   a) Create initial wireframes / mock ups.
   b) Carry out user testing.

4. Using the project plan, discuss and decide upon implantation techniques.
5. Model the architecture of the project using UML to provide a functional overview.

Output

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

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

Tasks

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

   Checklist

   a) Start developing the UI and core functionality of the Android and Web client-side applications.
   b) Receive client feedback and add requested changes into the sprint backlog.

3. Prepare for the interim submission.
4. Perform basic testing on the current application.

Output

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

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

Tasks

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

Output

1. Android drawing application design.
2. Basic implementation of the Android drawing application.

3. Comprehensive plan on how authentication and authentication will be implemented.

Sprint 5 (6th Feb - 19th Feb)

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 drawing (drawer's age must be included with the image). When a drawing is saved the canvas should be cleared, 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.

Output

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.

Sprint 6 (20th Feb - 5th Mar)

Tasks

1. Write unit tests for the Android application and ensure at least 80% coverage if possible.
2. Implement the 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.

Output

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.

Sprint 7 (6th Mar - 19th Mar)

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.

Output

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

Sprint 8 (20th Mar - 2nd Apr)

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.

Output

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.

Sprint 9 (3rd Apr - 16th Apr)

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 its fully functional.
8. Create handover documentation.

Output

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

Sprint 10 (17th Apr - 3rd May)

Spare time for polishing and contingencies.

Implementation Details

Front-End

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

Web Side

For the web side of 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.

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

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. Further research is needed to implement this.

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

Database
To be decided after discussion with the client.

Image Storing
Images themselves will be stored in a file system and the information identifying the image will be stored in the database. Said file system will most likely be on the cloud.

Rate Limiting
Rate limiting will be implemented to protect the back-end from over posting and cyber attacks.

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.

Technical Testing
During the development of the application, various parts of the system will be tested using several different testing methodologies.

Integration Testing
Testing different modules/services together.

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

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

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

User Testing

Being the nature of this project involves a wide age range and abilities, it is important that regular user testing is carried out across all phases of the project, particularly for the drawing app itself.

Initial testing will be carried out with paper-based prototypes to see how visitors will interact with a user interface compared to that of drawing materials. Testing will also be carried out with existing drawing applications on various touchscreen devices to see what differences there may be to traditional drawing.

Although the majority of user testing will be focused on the drawing application itself, especially in the initial phase of the project, some testing of a user interface for the scoring of the drawings will also be carried out, along with a little testing for the process of setting up and ending a session with the session leaders.

During development, it will also be important to keep the user testing the drawing interface once it is in a usable state. Since it will be used by such a wide range of visitors, the more testing we can do, the more feedback and information we can gather to make an application that is accessible to the widest range of people.

Less frequent testing will be carried out with the scoring interface and the setting up and closing of sessions since the relative interaction complexity with these is much less. After the initial paper prototypes are tested, it is projected that only one or two user testing sessions will be required for these interfaces.

Risk Management Plan

Reference	Event	Likelihood	Impact	Risk Exposure
R1	Changes to requirements specification during development	3	4	12
R2	Illness or absence that affects critical activities	3	7	21
R3	Illness or absence that affects non-critical activities	3	3	9
R4	Coding of a module takes longer than expected	4	7	28
R5	Testing reveals errors or issues with design	5	8	40
R6	Misunderstanding of requirements	2	4	8
R7	Change in technologies during development	1	8	8

Risk Mitigation

The main risk of the project is usability since the product will be used by a wide range of ages and abilities. For the most efficient mitigation of this, extensive user testing will be carried out during all phases of the project. Initially in the planning phase, existing applications will be used along with paper-based prototypes to research how users interact with different user interfaces and see how they generally interact with drawing on a screen against how they draw on paper to determine the following:

- Is and detail lost with drawing on a screen.
- Is there a relative speed difference with drawing on a screen against that of drawing on paper.
- Are there any intuitive things that users may do when drawing on a screen they may need to be implemented.
- Is there anything that users need more instruction on when drawing on a screen.
- Is there anything with drawing on a screen that causes frustration when drawing on a screen.
- Any other factors that may come to light with user testing.

The next biggest risks are absence of team members and any development that may take longer than expected. The best for mitigating these risks would be to add in extra sprints to allow for catch up along with some of the simpler tasks and to revise the project and general housekeeping to make sure everything is up to date.

Communication Plan

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.

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.

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.

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.

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.

Repository

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