Project Definition Document

Project	Empower Events
Authors	Miles Legood, Thomas Draper, Samuel Ivuerah, Nishita Chandani, Omar Houchou, Youssef Fayad, Mathew Dawson
Group no.	19

Project Introduction

The problem we are trying to solve for Halow is to:
Develop an accessible and easy to use web application that allows adults with disabilities to have a better “voice” within the services they access. (SDG10)
A solution for charities and other services to use where the supported adults can vote on favourite/least favourite activities and give feedback, such as “like” or “dislike” and/or recording voice notes that describe how they feel.

We are solving this issue by creating a website that allows people to find and attend events. These events will be accessible to many different disabilities, so the events page needs to accommodate these users so they can easily find the events. So that the users have a voice in the services they access, we will use feedback on events they have attended and the staff that ran the event. Our website's feedback will show the charities whose events are received well by people with different disabilities. It will also allow the charities to see how the staff running the events are being received. We believe Empower Events will allow many people who were previously unable to discover events catered towards them on a website they can easily navigate.

1. Project Charter

1.1. Objectives:

- Develop an accessible web application to allow adults with disabilities to have their voices heard.
- Have feedback to be given through like or dislike rankings for many questions.
- Allow recording of voice notes for detailed feedback.

1.2. Stakeholders:

- Our target audience will be people with disabilities. Our application will enable them to vote on events held by charities.
- Charities and non-profit organisations that host events for people with disabilities and want feedback to improve their events.

1.3. Constraints:

Our constraints given to us:

- Deployed online or run in a docker container.
- Make Use of at least two external APIs to add features; these APIs have to be free to use to be fair to all groups.
- Real-time client-server streaming component.
- Using Server-side processing

Constraints we have given ourselves:

- User Customization Features: Allow users to customise their experience, such as changing text sizes, colours, or the interface layout, to accommodate a wide range of disabilities and preferences.
- Responsive Design: The web application should be fully responsive, ensuring it is accessible on various devices and screen sizes, from smartphones to desktop computers.

2. Project Description

2.1. Project Overview

Our project will be an accessible web application that allows adults with disabilities to have a better “voice” within the services they access. It will be used by multiple charities and services where supported adults can vote on activities and provide feedback such as voice recordings, text, and multiple choice answers about their feelings about previous events and the staff members running the events. Our main focus is accessibility, which includes many features to allow users with all kinds of disabilities to use the website, such as a simple and easy-to-use UI, high contrast and support for touchscreen devices.

2.2. Technical Solution

We will use a Django project with an SQLite3 database for our backend, using the Django rest framework API and a React project for the front end. This gives us a powerful and robust web application.

2.3. Feasibility

- We will be using the Google Maps API as it is free, and we can display the location of activities to our users. We will also use a weather API to help users dress correctly for the weather and use real-time processing to alert users of any changes in weather.
- We will use WebRTC to record users and store the recordings on a server.
- We will create prototypes throughout development to get feedback from our client

2.4 Accessibility features for each disability

- Visually impaired
  - Navigational by keyboard
  - Text to speech information
  - Large coloured buttons

    - No confusing colours
- Deaf
    - Give feedback via text
    - Clear text

- Autism
    - Touch screen functionality
    - No use of confusing colours
    -
- Dyslexia
    - No use of confusing colours
    - Text to speech
- Epilepsy
    - Avoid flashing/blinking animations or videos that could trigger seizures
    - Allow users to stop or reduce motion of animations
- Cognitive Disabilities
    - Use a simple, uncluttered layout
    - Provide clear instructions in plain language
    - Keep the user interface consistent across pages
- Mobility Impairments
    - Informs User if even is accessible for them (Such as if there are lifts or nearby car parks)

3. Technical Approach

3.1. Technologies

The core technologies we will use are React for the front end and Django for the backend. We selected these technologies because of the comfortability and familiarity of our designated front and backend developers and due to the power these languages provide. The code written will conform to ESLint and PEP8 code style techniques. Django as a backend technology can sometimes be slow; however, using React's virtual DOM will mitigate this issue.

3.2. Feedback system:

The system allows users to leave feedback on the events they have attended, using ratings, text, and audio to give feedback on the event and the staff running it. To create a dashboard and reports to help the charities compile the feedback to help improve the events. Also, to help assign an employee of the month.

3.3. Login system

There will be two different login systems, one for users and a separate system for charities. A user login will have access to the customer-facing side of the website. The charity login will allow a staff member to view any feedback from users and add or edit events, and this login will also provide a dashboard that can show top feedback, such as if many people have said similar things.

4. Evaluation Criteria

4.1. Accessibility and Usability

- User Interface Design**:** Evaluate the simplicity, intuitiveness, and readability of the website's design, ensuring it accommodates users with various disabilities (e.g., visual, auditory, motor, and cognitive impairments).
- Customization Options: Users can adjust text sizes, colours, and contrast levels to meet their individual needs.
- Accessibility Standards Compliance: To effectively accommodate users with various disabilities, we will ensure the application meets international web accessibility standards, such as WCAG 2.1.

4.2. API Integration and Functionality

- API Compatibility: Ensure the integrated APIs are compatible with the website's framework and do not slow down the site's performance.
- API Reliability: Evaluate the APIs' uptime and response time, ensuring they are reliable and do not hinder the website's overall functionality.

4.3. Text to voice functionality

- Using large-text-to-speech at Rapid API, we will be able to send the page information and receive its reading.
- This will allow blind/ low visibility or many other disabilities to understand and attend our events.
- If the use of this api turns out not to be the most favourable, there are many other options that can be used for Text-to-speech.

5. Team Charter - Rules of Engagement

5.1. Contribution

It is of the utmost importance that each team member contributes to the project in a way that utilises their strengths whether that be programming, frontend/backend, documentation, project management and meeting minutes however each member must contribute.

5.2. Behaviour protocol:

- Every member will attend group meetings.
- Every member will attend sponsor meetings.
- Every member will attempt to complete their assigned tasks, or speak to the project manager if not possible.
- Important group decisions will take place in these meetings as it is easier to discuss together.

5.3. Meeting protocol:

- Group meetings are held once a week at 12pm on Tuesday or later if an extra software engineering lecture is scheduled.
- Every member should discuss what they have achieved and any issues that have occurred.
- Every member should leave the meeting understanding their next steps to take and what to bring to the next meeting.

5.4. Individual Roles:

- As a team we have decided to make use of a rotating Project Manager to anyone that would like to take the role.

- Then we have split the team into their specialities, so whether they will focus more on the front or back end development as shown in the table below.

Miles Legood	Front-end
Thomas Draper	Back-end
Samuel Ivuerah	Back-end
Nishita Chandani	Front-end
Omar Houchou	Front-end
Youssef Fayad	Front-end
Mathew Dawson	Back-end

6. Project Management and Methodology

6.1. Communications

We meet once a week for a weekly “stand-up” (SCRUM) to discuss progress and future plans. The meeting is led by the project manager and the manager reviews the previous sprint progress and delegates tasks to each member to be completed in the next sprint. Each SCRUM meeting will be held in person if possible, otherwise on teams. Minutes will be taken, attendance recorded and follow up meetings can be arranged if someone missed the main meeting.

To communicate with the project sponsor we will forward any questions we have to Jon, The questions have to be clear and concise so there is no confusion.

6.2. Version control

We are utilising GitLab version control to complete this application, issues and branches will be created so people work on their code separately and to reduce merge conflicts. Every merge request must have approval from 2 team members to allow for code review to check that it is up to standards and using the correct code style.

6.3 Record Keeping

All meetings are recorded by the project manager and shared with all team members in a shared file alongside with the agenda and all the notes taken in weekly meetings and/or project sponsor meetings.

6.4 Project manager

We decided on making use of a rolling project manager, so that everyone can show their skills. Our current project manager table below (Recording stints as project manager).

Miles Legood	Start of Project - 05/03/24

Thomas Draper	05/03/24 - Current
Samuel Ivuerah	
Nishita Chandani	
Omar Houchou	
Youssef Fayad	
Mathew Dawson	

6.5 Documentation practices

We will be clear and concise when making documentation, We will be updating the final audit report as we make changes and progress throughout the project. Keeping track of timings is key to making sure we are on track with the work plan.

6.6 Quality Assurance

To Ensure Quality we Will follow a pair coding technique in which two team members may work on the same issue next to each other where they would write the code together, to help spot any errors as they go. Also, code reviews are conducted before any piece of code is merged into the main branch with 2 approvals needed, one from another member of the team and one from the Project manager and if any merge conflicts occur then the project manager will resolve.

7. Work Plan

We are working on a Sprint methodology. So the tasks assigned in a scrum meeting are assigned to a sprint, that the deadline is at the end. We are assigning each sprint to be a week, but added longer for the first sprint to get everything together. These are tracked on gitlab through milestones and deadlines.

Sprint 1: Start of Project - 05/03/24 Sprint 2: 05/03/24 - 12/03/24
Sprint 3: 19/03/24 - 26/03/24 Sprint 4: 26/03/24 - 02/04/24
Sprint 5: 26/03/24 - 09/04/24 Sprint 6: 09/04/24 - 16/04/24
Sprint 7: 16/04/24 - 23/04/24 Sprint 8: 34/04/24 - 30/04/24
Sprint 9: 30/04/24 - 07/05/24 Sprint 10 (final): 07/05/24 - 14/05/24

Colours below are detailed as:
RED: Critical
Orange: Important
Green: Low priority

1. Start Planning
    1.1. Create project concept (Sprint 1)
    1.2. Consider requirements for project (Sprint 1)
    1.3. Project proposal

1.3.1. Write and send proposal to sponsor (Sprint 1)
1.3.2. Review proposal with sponsor (Sprint 1)
1.4. Create Project Definition Document
1.4.1. Write project purpose, description, and charter (Sprint 1)
1.4.2 Define Member Roles (Sprint 1)
1.4.5. Send PDD for review and approval (Sprint 1)
1.4.6 Update the PDD when new information is proposed (Sprint 3)
1.5 Create Project Pitch
1.5.1 Create Pitch script (Sprint 3)
1.5.2 Record pitch video (Sprint 3)
1.5.3 Submit project pitch (Sprint 3)
1.6 Create development designs
1.6.1 Create database schema diagram (Sprint 1)
1.6.2 Review and assess Throughout
1.6.3 Create Wireframe diagrams (Sprint 2)
1.6.4 Create backend REST API Framework (Sprint 3)

2. Software Development
2.1. Software design
2.1.1. Decide on framework (Sprint 1)
2.1.2. Decide on packages (Sprint 1)
2.2. Building software
2.2.1. Create Front-end with React
2.2.1.0 Create React app (Sprint 1)
2.2.1.1 Create visual prototype. (Sprint 3)
2.2.1.2 Add functionality to the prototype. (Sprint 4)
2.2.1.3 Create navbar (Sprint 3)
2.2.1.4 Create the user and charity login. (Sprint 3)
2.2.1.4 Create user registration (Sprint 4)
2.2.1.5 Create the homepage. (Sprint 4)
2.2.1.6 Create events page. (Sprint 5)
2.2.1.7 Create event feedback page (Sprint 6)
2.2.1.8 Add all the features required.(Sprint 7)
2.2.1.9 Create the charity feedback page (Sprint 8)
2.2.1.10 Create notification system to alert for changes in weather. (Sprint 7)
2.2.1.11 Implement Google maps to navigate to the events (Sprint 5-6)
2.2.2. Create Django Backend
2.2.2.0 Create Django backend (Sprint 1)
2.2.2.1. Create Models (Sprint 2)
2.2.2.2. Create Sample Data (Sprint 2)
2.2.2.3. Seed Database (Sprint 2)
2.2.2.4 Create user and charity login requests (Sprint 3)

2.2.2.5 Create user register request (Sprint 4)
2.2.2.6 Create feedback get and post requests (Sprint 5)
2.2.2.7 Create get events request (Sprint 5)
2.2.2.8 Create weather change request (Sprint 6)
2.2.2.9 Create text to voice functionality (Sprint 7)
2.2.2.10 Create speech to text functionality (Sprint 7)

2.5. Testing/ Review

2.5.1 Unit testing

2.5.1.1 implement backend unit tests (Sprint 4)
2.5.1.2 Create tests that run while the website is active, to test many things such as login, and time to receive infomation. (Sprint 6)
2.5.1.3. Run the tests after every change/ commit. (Throughout)
2.5.1.4 Troubleshoot all errors encountered (Throughout)

2.5.2 Feedback

2.5.2.1 Get client and user feedback throughout(Throughout)
2.5.2.2 Use this feedback to change/ add features that will improve the experience (Throughout)

2.6. Finalising Deliverables

2.6.1. Handling and documenting deliverables (Sprint 9)
2.6.2. Evaluating and dispatching deliverables (Sprint 9)
2.6.3. Project Deployment
2.6.3.1. Deployment using Docker (Sprint 9)
2.6.4. review post launch issues
2.6.4.1 fix bugs (Sprint 8)
2.6.4.2 request feedback from users (Sprint 8)

8. Risk Analysis

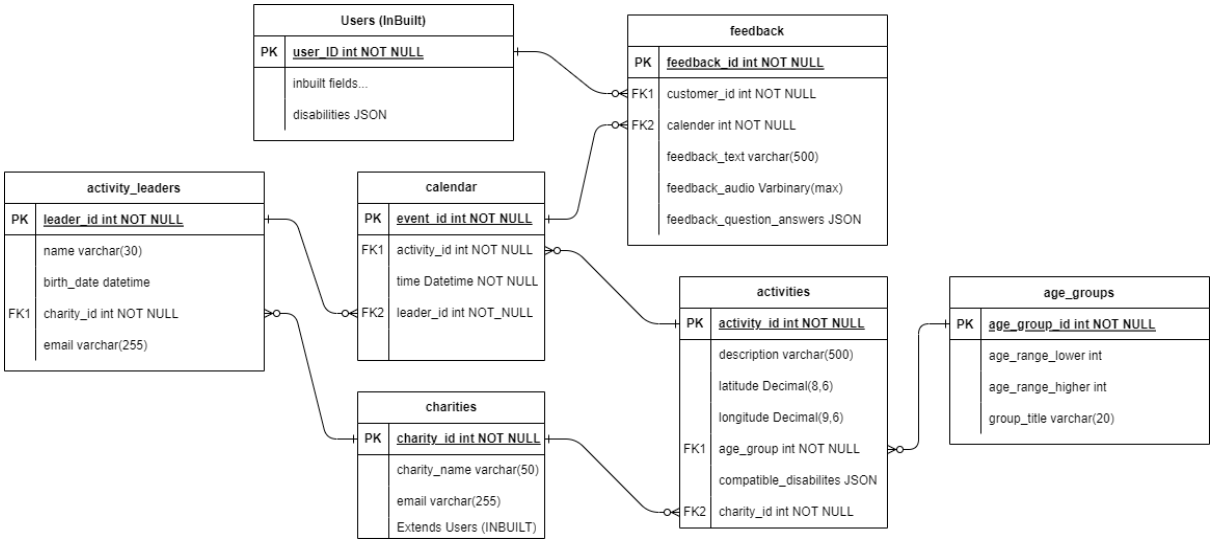
A risk analysis for a project aimed at developing a web application to empower adults with disabilities involves identifying potential risks that could impede the project's success, evaluating their likelihood and impact, outlining mitigation strategies and ranking them. The potential risks are highlighted and ranked by severity using a low, medium and high scaling with highest severities first. See Table.

Event	Probability	Impact	Severity	Rank	Type	Recovery Action
Project is delayed or off schedule	Medium	High	High	1	Performance	Lose the lease priority features to meet the deadline
Project scope is not clear	Medium	High	High	2	Scope	Ensure that all adjectives are clear to all the members from the start
Code loss or	Low	High	Medium	3	Technology	Git version control will

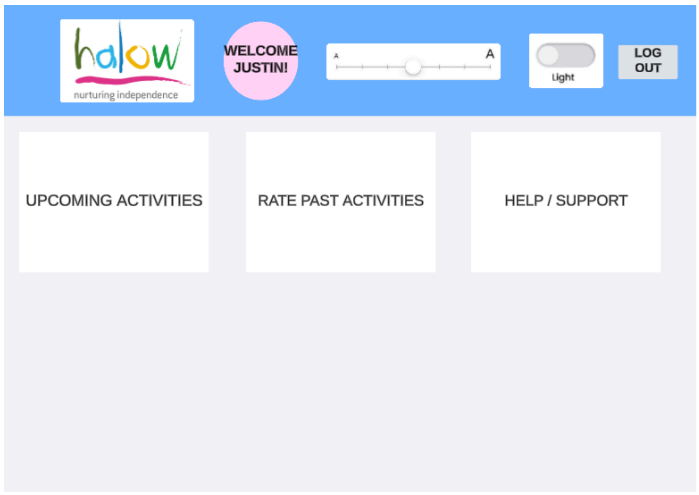
corruption						allow us to revert any changes if something goes wrong
Difficulty implementing APIs	Medium	Medium	Medium	4	Technology	Research how to implement APIs in advance. Or change the API to better suit development
Minimum criteria not met for project	Low	High	Medium	5	Performance	Propose improvements to project sponsor
External Services having a cost	Medium	Medium	Medium	6	Technology	Find a replacement API
Lack of skills or expertise	Medium	Medium	Medium	7	Resource	Tasks are delegated based on individuals skills but can collaborate if needed.
Team member leaving	Low	High	Medium	8	Resource	Reallocate tasks and contact project sponsor
Arguments, disputes and miscommunications between members	Low	Medium	Low	9	Resource	A vote or discussion will take place and the whole group will decide by popular vote
Software becomes obsolete	Low	Medium	Low	10	Technology	Find a similar replacement software

9. Development Design

- **Database entity relation diagram:**



- **Wireframe Diagrams:**
  We created wireframe diagrams to help
    - Homepage:



    - Events Page: