

PROJECT DOCUMENT

GROUP 1: GEOSAFE



Welcome to GeoSafe

Introduction

Aims and objectives

The aim of the GeoSafe app is to provide useful safety information for the Cardiff area. Women are the main audience, due to [reports that women in the UK feel less safe when walking alone at night compared to men \(links to ONS census analysis\)](#), however the content may be useful to anyone who finds themselves in a potentially dangerous situation. The homepage is a simple, welcoming screen with a clearly-defined navigation bar and urgent support button. When the user scrolls down, they'll find some facts and statistics about why GeoSafe is important. The rest of the app contains a map with the nearest police stations as well as their addresses, a list of useful telephone numbers, and a list of items a person is legally allowed to carry for self-defence. Additionally the app also provides current information about local crime statistics. The objective is that this app would empower women and other vulnerable groups to feel safer and more confident when walking alone at night.

Background

GeoSafe is designed to be a simple and straightforward application for easy and fast navigation in events of an emergency. The top of the page provides links to each page

which are clearly marked and made with user experience in mind. It includes charts and a map for current crime statistics in the area, including a postcode search functionality on the map, a list of important numbers and addresses for local authorities, shelters, and helplines, and also a useful tips page which shows which items you can or cannot legally use for immediate self-defence in case of danger.

Specifications & Design

Technical and non-technical requirements

- Access to the GitHub repo
- An integrated development environment (IDE), such as VSCode, with Node.JS installed
- A web browser

To run the app, type *npm install* into the terminal in order to install all the dependencies, followed by *npm start* to launch the app.

Design and architecture

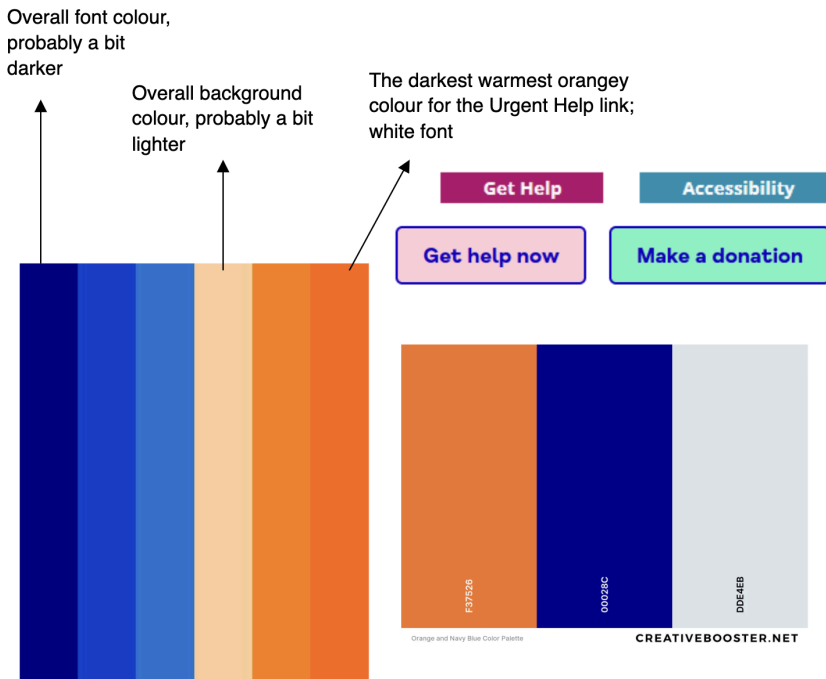
From the start our aim was to create an aesthetically pleasant, user friendly application. The first step was a logo, which we wanted to be descriptive of what our application is about and also memorable. We used the icons and settings available in Moqups to create a simple shield logo with a map pin in the centre, with an orange glow around the pin, suggestive of a beacon, screenshotted it, and then used a free Adobe feature to make the background transparent so that the logo could be used in a variety of ways/locations. We also agreed on an orange/dark blue colour scheme after a discussion on colour theory. Orange is a warm, welcoming colour, as well as being bright and associated with urgency. We discussed wanting a script-like, almost playful font (inspired by the [Mind charity's website](#)), again to increase the user friendliness of the app's appearance. To contrast the warm orange hues and playful font and to instil confidence in the user, we used dark blue/navy, which is considered to be more serious and conservative. It was important to us that the colours welcomed the user to the page and similarly so the logo would stick in memory once seen. This was a decision not only because we want GeoSafe to stand out, but also because if our users need to access the application in a stressful situation, subtle choices like this may jog their memory faster.

We also used the Moqups wireframe to consider how the app would look on different devices/screen sizes, prioritising mobile-first design as is the industry best practice, however we decided that as we'd all be building, testing and running (**including markers**) on laptop or desktop screens, that we'd only build for desktops.

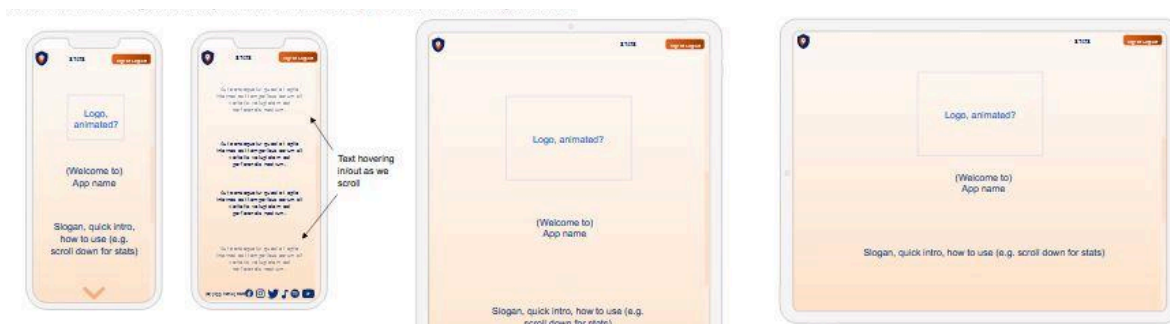
GeoSafe logo:



Screenshot of colour theme suggestions from the Moqups wireframe:



Screenshot of different screen size considerations and design choices from the Moqups wireframe:



Another important part of our design was that the application was easy to navigate. For this we made a menu on the top (as well as the logo linking to the homepage too) accessible from all pages making it easy to access any part of the application in one single click.

Implementation & Execution

Development approach and team member roles

Our development approach consisted of several meetings for planning: wireframe, ideas brainstorming, voting on an idea, planning features for the idea, and then creating and assigning tasks in Jira. It was important for us to make sure we each contribute to the project fairly. For this reason we made sure to assign ourselves at least one research and one coding task each. This allowed us all to test and develop our skills and knowledge as we developed the project.

We then mapped our app in two parts. First was the very minimum we wanted to achieve, according to our aims and objectives as well as the project assessment criteria. Second was optional ideas which we would like to implement, but which were too complicated to guarantee completion or we were simply not fully confident we can make them work. This allowed us to focus on the most important aspects first. The Jira board is also where we added these additional “nice to haves” to improve the appearance or features of the app if we had the time, but which weren’t imperative to be included.

Tools and libraries

Tools

- Moqups
- Miro
- Google Drive
- Jira
- Git
- GitHub
- Slack
- VSCode
- Google
- YouTube
- ChatGPT (for help with troubleshooting bugs and tests)

Libraries

- @react-google-maps/api
- @testing-library/jest-dom
- @testing-library/react
- @testing-library/user-event
- @tremor/react
- Axios
- Bootstrap
- Chart.js
- Chartjs-plugin-datalabels
- Marker-clusterer
- Marker-clusterer-plus
- React
- React-chartjs-2
- React-dom
- React-router-dom
- React-scripts
- Web-vitals

Implementation process (achievements, challenges, decision to change something)

We had a few ambitious ideas in our initial few meetings for UX/UI features in the app, such as a styled scroll bar, a map with pins you could click to open and see more information about the location, an animated logo, and more accessibility features such as tooltips or feedback to help with UX, however given the limited timescales, we've had to leave those on a "nice to have" list.

A major ongoing challenge in general has been understanding Git and the conventions of collaborative work/coding. Regarding the code, one challenge we faced, but eventually achieved, was getting the different colours on hovering on the Tips page. The idea was that the legal suggestions would become green on hover, and the illegal things would become red. This proved difficult, but with more time spent researching classes, it was achieved.

The map feature was a significant component on its own, but we exceeded our own expectations with the final version - an embedded Google map with a postcode search functionality that shows the figures for crime stats in pins across the map. The figures and pins cluster together as you zoom out, and when hovering over them, the side panel gives more information on the types of crimes.

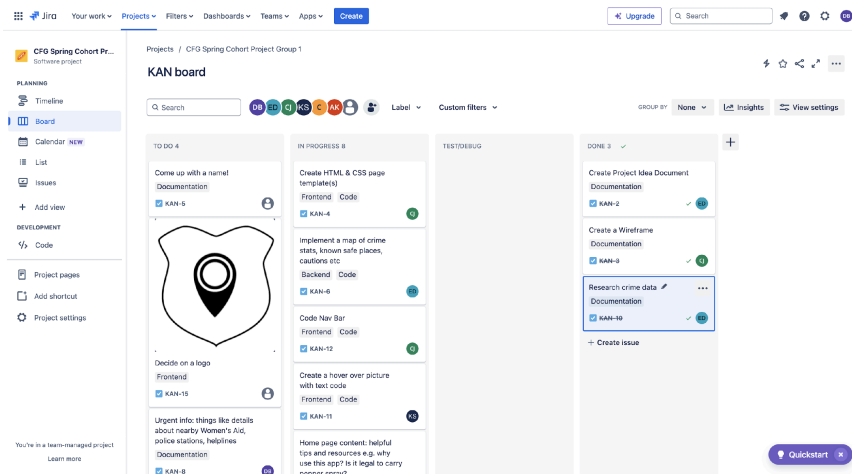
Another challenge we faced was compatibility across different devices. While building the welcome part of the homepage, which was designed to fill the screen, we had a down arrow to indicate to the user to scroll down for more information. This proved difficult to style in a way that appeared consistently across all team members' screen sizes, so we opted to remove the arrow, adding it to the "nice to have" list for if we had more time.

Agile development (agile elements like iterative approach, refactoring, code reviews, etc)

To plan and complete our project we used multiple tools to help us organise our work. We completed our first steps on Miro, where we created a board to share our strengths and weaknesses to help get a better understanding of the team's collective strengths, a mind map with ideas, and a dot voting system to elect the final idea.

After that, we moved our work to Jira where we utilised tools such as dashboards and a kanban board to track our progress. We created tasks to be completed and assigned them to the team member responsible. The kanban board in our Jira was structured in four columns (To Do - In Progress - Test/Debug - Done) which we used to track how far along each task was and which pieces of code needed to be tested. We also used the kanban tasks to hold information such as suggestions for the app name, useful links found while researching, and any pertinent notes from one team member to another regarding the task.

Jira kanban board:



We made sure to checkout new branches in git for each component of the app which we updated regularly to keep the code clean and structured. We submitted pull requests, wrote effective commit messages, and had other team members review code before merging our branches to main.

We had a couple of iterations of the welcome part of the home page and of the maps page as we learned some things were too difficult or time consuming to achieve with the time and skills/knowledge we had, however the end result has still met our initial aim and, hopefully, if the app was live, the objective.

Testing & Evaluation

Testing strategy and system limitations

We manually tested as we built components and added styling and as part of reviewing each other's code. Additionally, we wrote automated tests for some elements and features as a secondary assurance that the components should work across different devices.

Everything passed manual testing across screen sizes ranging from 1024 px to 1920 px in width, however not everything we tried to write automated tests for passed. On the Navbar component, testing that it loaded was successful, as well as that the shield logo and the Home button linked successfully to the home page. However, testing on the Stats button linking successfully to the Stats page would fail. We suspected that the issue was related to how React deals with routing, however despite many hours of research and reaching out to instructors for help, we weren't able to find a solution.

Before submission, we re-cloned the repo and ran npm install before running the app to make sure it would work for anyone cloning the repo anew, and to see if there are any packages we needed to install globally.

Conclusion

This project was a great test of not only our coding abilities, but also our communication, planning and organisation skills along with our approach both individually and in a group. We each had to first understand ourselves and what we can contribute before we could split the work most effectively amongst ourselves. Once we did this, we used various tools to help us get to our desired outcome, all of which can be found listed in this report.

We voted on which topic to focus on when creating the project. The application we chose to make was on the topic we all feel passionate about which contributed to our motivation to finish it to the best of our abilities. As a group, we quickly came to an agreement on what we want it to look like and what we want to achieve with it, but we also kept in mind that there may be a limit to our abilities. We designed the application with all this taken into account.

There were definitely some challenges which we encountered along the way, but we chose to approach them as learning experiences. The main areas we struggled with came with learning how to properly navigate Git and GitHub and how to test our code. In the end we were able to overcome these obstacles and while there are some differences between the initial idea and the final project, we can confidently say that the structure and its purpose remained what we wished it to be. Any changes made were mostly to improve the idea rather than sacrifice a feature due to lack of skill.