Team Project Report v1

Stack Overflowed

February 19, 2026

Introduction

Summary
Footprint is a web application designed to reduce carbon dioxide emissions by fostering awareness through a challenge-based, competitive platform. Users can complete daily, weekly, and monthly challenges across a range of environmental categories, each aimed at lowering their personal carbon footprint. Progress is quantified by tracking the amount of carbon saved, which is then converted into points; enabling users to compete against one another and harness the power of healthy competition to drive meaningful environmental impact.

Addressing the Problem
Carbon dioxide is the leading greenhouse gas that impacts our society today; it is the driving factor for climate change and global warming. Data from 2022 shows that carbon dioxide is responsible for 80% of total greenhouse gas emissions in that year [5], which shows how important it is for us to reduce our carbon footprint. Together, we can find a way to reduce our carbon footprint! This application is our attempt to do exactly that, by giving users different and engaging ways to reduce their footprints through a gamified challenge-based system. The audience for this application is primarily students and staff at the University of Exeter, with potential scope for future development in other scenarios too. For example, other universities, secondary schools and potentially workplaces. The application will also be able to support moderators, who will assign challenges and also review the submissions of users' completed challenges. This is further explained in the following sections.

Our focused goals:

- Log the micro changes of carbon data and simulate data for analytical purposes.
- Reduce carbon emissions and promote eco-friendly habits across users.
- Create a fun way to introduce sustainability with a gamified experience.

Our targeted stakeholders:

- University students across all years as well as faculty members, focusing on producing a fun and engaging product which will captivate their interest and prolonged engagement.
- The universities implementing the product want to find a way to engage their students in eco-friendly practices as well as reach targets on net-zero carbon emissions.

Success measures
Reflecting on our goals, we have chosen to measure the success of our product in accuracy of our data analysis from the recorded user activity, as well as the user engagement achieved from gamifying the experience. However, as this initial sprint focuses on producing a minimal working product this sprint will be solely evaluated by the runtime and availability of our prototype as well as minimising errors and bugs interacting with the system.

How will this be done and maintained?
This will be accomplished by users uploading pictures as a form of evidence for certain tasks they have completed. The moderator will verify the evidence and can approve or deny the submission's request. If the submission is approved, the user will get the points as per the pre-assigned points' value from a table that contains the points for that specific task. If the user's submission is denied, then the user will not get any points for that activity.

Prioritised Requirements

This section will be outlining the requirements of the prototype system, discussing both functional and non-functional requirements. Using the MoSCoW prioritisation method, we categorised features into must have, should have, could have and won't have. We were able to identify the Minimum Viable Product (MVP) for the first sprint. Requirements were categorized on the basis of their necessity for the system functionality and their contribution to meeting the objectives of the first sprint.

User Stories and Acceptance Criteria

Number	User Story	Acceptance Criteria
1	As a user, I want to sign up/sign in easily.	User can sign up with their username and password of choice, and users can sign in with their username and password.
2	As a user, I want to see the challenges I can do and the points I get for completing said challenge.	Table containing challenges users can complete and the points they can get for doing it.
3	As a user, I want to easily see the leaderboard and my ranking.	A table with a list of users and how many points they have in an order of highest number of points to lowest.
4	As a moderator, I want a specific and clear space where I can easily approve or deny user evidence.	A moderator only section where they can see user submission with the option of approving their submission and user gets points for the challenge or deny user submission and user gets no points.
5	As a user, I want the submission process to be clear and know how many points I will get from the challenge done.	A clear calculation showing user how the points are derived for the specific task they are doing.
6	As a moderator, I want to have a specific tab where I can easily create/end challenges and view the analytics of each challenge.	A tab specific for moderators only where they can make challenges for users and remove previous challenges.

Carbon Calculation model

The model design was shaped by balancing computational simplicity and user engagement with a gamified web application. The primary goal was to provide an approximate for the user rather than a fully personalised carbon footprint assessment, as this would require much more input which would make users perhaps less inclined to use the service, and also makes it easier to produce as there is less input to deal with.

Data figures, sources and relevance.
The values for carbon emissions were picked from publicly available carbon calculators and datasets about the environment. These sources provided data on emissions in different contexts hence, had to be normalised to a per-action estimate to ensure consistency across all challenges. An example of this is that the emissions from a reusable cup were divided by the average lifespan of uses of the cup thereby we came to a conclusion of 500 uses and the emissions from recycling were converted into daily values from distributing monthly averages. To maintain a uniform measurement scale, all the carbon savings were expressed as kilograms of CO_2 ($kgCO_2$). Doing this was essential for standardisation to be able to have leaderboard comparisons and overall scoring logic.

Assumptions and model limitations
The assumptions which the model relies on to be simple.

- Three meals per day when estimating vegetarian meal savings
- Average petrol vehicle emissions per kilometre
- Full daily recycling behaviour
- One repaired item per month
- Average food weight in a Too Good To Go purchase

We had to make assumptions as we did not not have enough data on users to make a specific lifecycle. Although there is a trade-off, while this does reduce individual user accuracy, it enhances usability and reduces excessive data collection, which would negatively affect user adoption and make the system more complex.

Categorisation of frequency of activities and actions.
The possible actions were categorised into 3 groups, daily, weekly or monthly. This was done to reflect real world behaviours and lifestyles. Actions that have a high impact, like repairing an item which saves a value of 32.9kg(based on our sources and calculation), are set to monthly frequency thereby preserving leaderboard fairness and reduces the chance of distortion of cumulative totals. This action maintains system integrity while giving users more motivational incentives.

Challenge table					
Challenge	Description	Carbon Saved(KG)	Frequency	Assumptions	Links
1	Eat a vegetarian meal	0.64103	Daily	Assuming people eat 3 meals a day, rounded to 5.sf	[4]
2	Walk to campus	0.17	Daily	Assuming users would have otherwise drove a petrol car. This figure is carbon emissions saved per Km of walking.	[7]
3	Use a reusable coffee cup	0.026	Daily	Assuming users use cup until end of average lifecycle which is 500 uses	[11]
4	Use a reusable water bottle	0.47123	Daily	Assuming users use bottle until end of average lifecycle which is 500 uses	[6]
5	Recycle	1.9	Daily	Assuming recycled all materials, 61 kg divided by 31 days	[2]
6	Shop for second hand clothes	13.9	Monthly	This figure was generated on general clothing using a linked calculator	[9]
7	Line dry clothes	1.8	Weekly	The average drying cycle releases 1.8 kg's of Carbon emissions	[8]
8	Borrow book from library	0.7	Weekly	N\A	[10]
9	Donate an item to charity	13.9	Monthly	Assuming this is equivalent to buying second hand clothing	N\A
10	Repair an item of clothing	32.9	Monthly	Assuming only one item is repaired	[1]
11	Purchase 2good2go	2.7	Monthly	Assuming each bag on average saves 2.7 kg	[3]

This is a table of the challenges with their respective features like the carbon emissions saved, which taxonomy the challenge falls under and sources of where we found the carbon emission for each task.

Web Design

When planning the design for our front end webpages we used the 4 design principles put forward by the government of the UK to ensure our website is accessible to all users and create a friendly, approachable design.

Perceivable - Images include text alternatives for screen readers, content is logically structured, and the Verdana font is used throughout for dyslexia friendliness. High contrast white text on a dark background ensures readability, and the page supports zooming for low-vision users.

Operable - No flashing content is used to protect users with photosensitive conditions. Navigation tabs are persistently visible with colour-change feedback on interaction, and buttons are sufficiently large and spaced to prevent misclicks.

Understandable - The website is presented in English with a consistent colour scheme and layout throughout. Form fields have clear, visible labels, and login is kept simple, requiring only a username and password.

Robust - The website is written in semantic HTML to ensure compatibility with assistive technologies and across different browsers. All images include alternative text.

In our efforts to create such an accessible interface for our users we came up with the user friendly designs as a Lo-Fi prototype to plan our implementation of the UI. This design is shown in the transition diagram below.

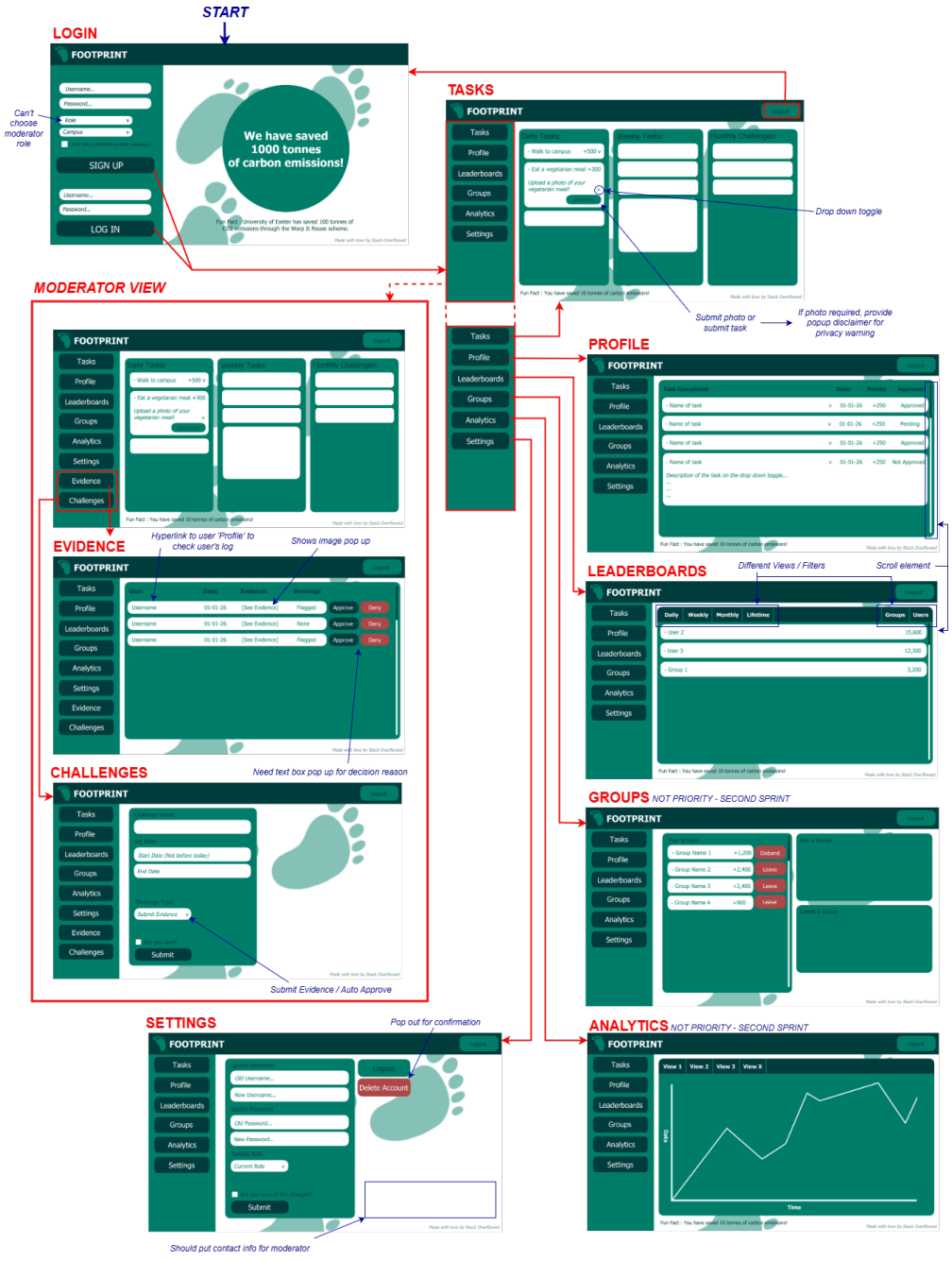

Figure 1: UI Transition Diagram

Architecture and Implementation

The project is built on a range of tools and technologies that collectively enable us to develop, deploy, and interact with the web application. From the user's perspective, the front end serves as the primary interface; allowing users to navigate the application, trigger actions via buttons, and submit data through forms and input fields. Each of these interactions generates a request that is sent to the back end via HTTP fetch and POST methods, where it is processed and the appropriate response is returned to update the interface and deliver the relevant functionality.

These requests are processed and managed by a Java-based back end built around a Controller-Service design pattern, which divides the system into modular layers, each with a distinct responsibility. The controller layer acts as the entry point, receiving requests from the front end and routing them to the appropriate service classes. The service layer houses all core business logic, including evidence handling, moderation, and leaderboard updates, performing the necessary calculations and operations before returning results to the user via the controllers. Underpinning this is the model layer, which defines the structure of data objects as they are stored in the database, and the repository layer, which sits closest to the database and handles all data access operations, allowing the service layer to focus purely on logic rather than database management. This layered, modular architecture makes the system straightforward to test, debug, and extend without disrupting existing functionality

Central to the application is a relational database built around a user-centric design, ensuring that all stored data is associated with a specific user and that each user has an independent experience within the system. The USERS table stores core account information, including usernames, hashed passwords, roles (user or moderator), campus, and year of study. The CHALLENGES table defines each challenge, storing its title, description, point value and other classification data, while the USERS-CHALLENGES junction table links the two, tracking which challenges each user has and has not completed. The EVIDENCE table manages image files submitted for challenges that require proof of completion. The LEADERBOARD is constructed dynamically on request by querying the USERS table, ordering users by points, and returning the top ten results. Finally, a GROUPS table is planned for the second sprint, which will allow users to connect with one another under shared group titles. These tables and their relationships can be seen in the table below.

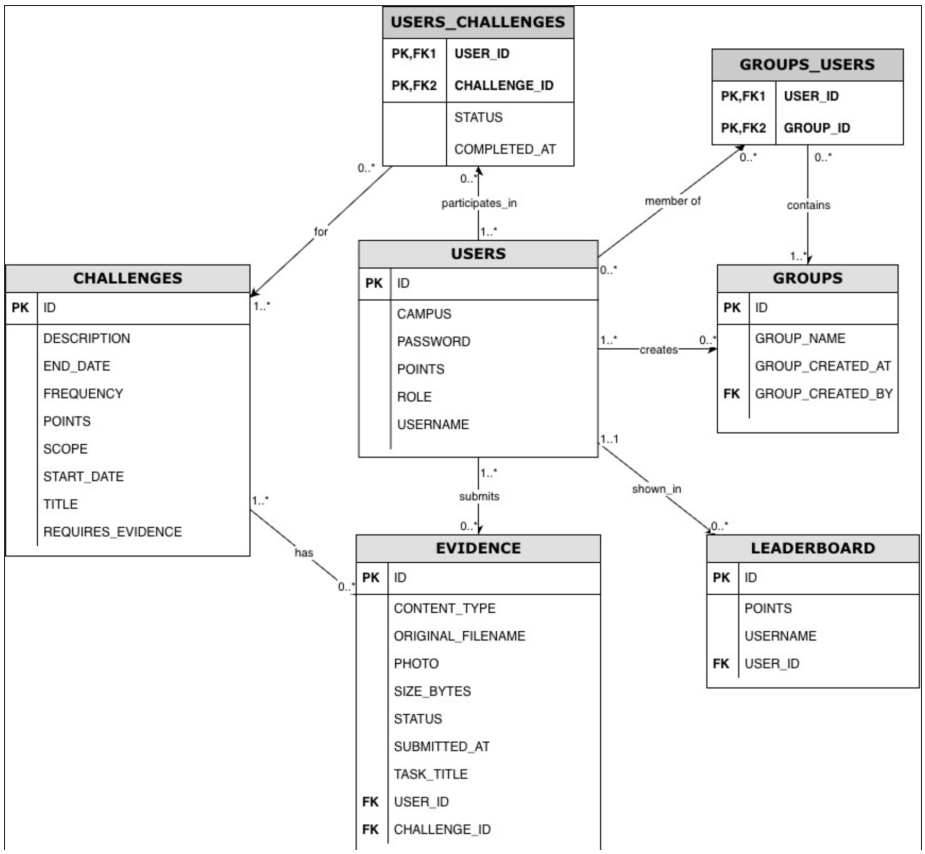

Figure 2: ER diagram

Overall, the architecture underpinning this application has been designed with clarity, scalability, and maintainability in mind. The Controller-Service pattern enforces a clean separation of concerns across the system, ensuring that request handling, business logic, and data access remain distinct and independently manageable. This makes the codebase straightforward to extend, test, and debug as the project evolves. Paired with a user-centric relational database that ties all data to individual accounts, a responsive front end, and a robust Java back end, the system delivers a seamless and engaging experience for users.

Evaluation

When thinking about the evaluation of the implementation, there are many techniques that need to be used to effectively test the entire program. We have opted to use a combination of unit testing, integration testing, and end-to-end (E2E) testing.

JUnit 5 is being used for both unit testing and integration testing as it is very well known and provides easy integration with the Spring framework used to develop the website. Unit testing is used to test the individual methods and their functionality regardless of the database and frontend. Mostly, this is done using the Mockito Testing Framework, which allows us to mock the repository/database classes and control any access to them.

With integration testing, the annotation *@DataJpaTest* on the test classes allows us to auto-wire the repositories, as opposed to mocking them, helping us to test the code and functionality with a functioning repository system. Regarding E2E testing, we selected a vertical slice of all functionalities implemented during the first sprint. This vertical slice reflects the most important features, and they will be manually tested by going through the production website recording successes and failures.

The overall results of the tests helped us to understand what needs to be completed for the next sprint. All unit tests have been passed, giving us good coverage on the classes that were implemented, which is reflected in the final production of the website, where all features that were implemented, work as intended and produce no errors during runtime. The end-to-end test also proves really useful to the evaluation of the implementation, as it clearly shows us what features have passed, partially passed or failed given our criteria for the MVP.

Sprint plan for CW2

In the second sprint, we will implement the analytics tab, which includes graphs and statistics, regarding users' recent activity amongst others. Furthermore, we will be implementing groups where users can create or join a group where some of them will require a specific code. This is done to ensure that societies/departments in the university can only have their own members joining the group. In addition, there will be incentives like streaks and badges to motivate users. The latter will also be able to filter the leaderboard by date. Moreover, additional security measures will be added.

Sprint 2 goals		
1	As a user, I want to easily be able to join/create groups with a code for only members of a specific society to be able to join.	A group system where users can join by having a unique code for each.
2	As a user, I want to be able to see what badges I can receive from challenges and see the badges of other users.	A list of badges users can get for matching a certain criteria and the badge will be present on user profile so they can see and so that other users can see.
3	As a user, in the case my submission is denied, I would like to know why the moderator rejected my submission	In the case moderator deny a request, moderators will return a text explaining why they made the decision to reject it.

Management

To structure the development of our project, we adopted an agile approach with the goal of producing a minimum viable product within our first sprint. Agile was well suited to our timescale, as its short development cycles allow deployable code to be produced quickly and enable workstreams to run in parallel rather than sequentially. We also held weekly meetings to review progress using scrumboards, which helped the team stay coordinated and on track. Together, these practices allowed us to avoid the rigidity of hard methodologies such as the waterfall methodology, where each phase of development must be completed before the next can begin, which would not have been compatible with the time constraints we were working within.

In order to facilitate a better structure for managing responsibilities we designated roles to each member. While we endeavoured to support one another when needed, this structure enabled us to focus on our respective tasks and provide a clear outline on who would be responsible and consulted on different areas of the project, giving us a clearer method to delegate tasks and navigate a roadmap to complete our first sprint.

The following entails the responsibilities and contributions of each member of the team:

Madi — Project Lead Responsible for organising and leading meetings, ensuring final product quality, helping facilitate the requirements and design of the project and liaising with all the members and facilitating communication between the team. Acted as scrum master, managing the scrum boards and maintaining the risk register to avoid a variety of issues from arising such as scope creep. Also provided support to UX design by creating the digital designs for the UI assets and layouts.

Ben — Technical Lead In charge of managing the architecture and quality of the code, ensuring that it is maintained for future developer use. Delegated tasks to members of the development team based on priority and individual strengths, while also taking on development tasks where needed on all elements of the code.

Jai — Data Lead Managed the structure and design of the database, and how the data will be securely stored. Supported documentation in conjunction with this.

Marko — Developer Completed development tasks on all elements of the code to incorporate different features and functionalities to the system as well as establishing security systems across the project.

Phoebe — Developer Provided important analytical information from research to facilitate the data used in challenges and tasks, and worked on the HTML as well as CSS elements of the code.

Zarreen — UX Design Considered all the legal and ethical aspects of our product, including our license justification. Ensured that the standards we have set are upheld across our code by coordinating with the development team. Worked on design aspects of the project, keeping accessibility and stakeholders in mind, and helped produce the demo slides for our team presentation.

Davi — Testing Facilitated all of the testing suites and implementations, coordinated with developers to make sure testing coverage was broad and that the final product was bug free and effective. Also provided support in documentation and code.

Armin — Documentation Responsible for the report of our first sprint prototype, has coordinated with other members of the group to ensure information is accurately documented and helped with database design.

For a full list of logs and contributions made by each member please see the second part of our `meeting_minutes.pdf` in the directory `2_process`.

Risk Management

When producing a large-scale product for a client, it's easy to overlook vital risks when focusing solely on the criteria required to produce the final outcome. Once challenges arise and appropriate mitigation or contingency isn't put in place it can require more resources to be drained to amend these issues. Common points of failure for a project can come from a lack of testing (R7, R16) when not considered in advance of development leading to problematic bugs the client encounters. Other areas of failure may originate from a lack of clear definition in requirements when understanding the problem (R1, R8, R10) or they may stem from within the functionality of the team itself (R3, R9, R18).
In order to minimise risks, a risk register was maintained through the first sprint to cover all these individual risks that were owned and mitigated by respective members of the team. This provided a useful ongoing log to track potential issues that would arise and pre-emptively respond to minimise potential negative outcomes by putting mitigation in place to aptly prevent it occurring.
For the risk register please find it in the directory `2_process`.

First Sprint Strategy Review

After reviewing our first sprint we believe that we collaborated effectively and efficiently together in order to create a strong minimal viable product for our first sprint. This now means that we can focus on our goal of engagement in our second sprint with this foundation now laid in place, while we also conjunctively make improvements to our current implementation along the way.
Discussing with the team on our collaborative efforts, we want to move forward in the same process we did for sprint one, focusing on our shorter weekly iterative sprints and delegating our tasks weekly. However we would like to revise our risks more thoroughly. Certain risks such as R20 and R21 only became prevalent when reviewing our sprint based on unforeseen circumstances and perhaps could have been mitigated sooner had the risks been revised and discussed more frequently in the team. Given the time scale we have to achieve this project, our team wants to prioritise managing our scheduling and deadlines effectively to create the best product, which at its core means handling our risks as adeptly as possible.

References

[1] CFSD Carbon Footprint Calculator. Carbon savings calculator result: landfill 3.5, emissions 32.9, repairs 1, car miles 137, showers 21, flights 201, 2025. Accessed: 2026-02-18.

[2] Changeit. How much co can i save by recycling?, 2025. Accessed: 2026-02-18.

[3] Ecoidea. How one app saves millions of meals and cuts co emissions, 2025. Accessed: 2026-02-18.

[4] Exeter City Council. How much carbon could you save by going veggie or vegan every so often?, 2025. Accessed: 2026-02-18.

[5] Department for Energy Security and Net Zero. 2022 uk greenhouse gas emissions, final figures: Statistical release. National statistics, GOV.UK, 2024. Accessed: 2026-02-18.

[6] Nevett With Nature. How much impact does a reusable water bottle actually make? a look at 1, 5, and 10 years, 2025. Accessed: 2026-02-18.

[7] Hannah Ritchie. Which form of transport has the smallest carbon footprint?, 2023. Accessed: 2026-02-18.

[8] SMOL. Line drying vs tumble drying, 2025. Accessed: 2026-02-18.

[9] Strocco's. Carbon footprint calculator, 2025. Accessed: 2026-02-18.

[10] Sutton Libraries. Green libraries week 2025, 2025. Accessed: 2026-02-18.

[11] Zero Waste Scotland. The environmental benefits of reusable plastic over single-use products, 2023. Accessed: 2026-02-18.