

Sprint 1+2 Documentation

Introduction	1
Initial Design and Project Architecture	3
Testing	5
Cards:	7
Profile Page	9
Change Deck Page	11
Authentication	12
Packs	13
Wrapper Feature	15
Bin Battles	15
Resetting the Bins/Gyms each day	19
Home page	20
Scanner Page	22
Creating Bins	23
Leaderboard Page(s)	23
Sprint Two Features:	24
Conclusion	26

Introduction

Our sustainability based game project, Ecomon, transforms the challenge of disposing, recycling, and reusing waste into an engaging and interactive card game. By combining the mechanics of a collectible card game with real world rubbish disposal and recycling points reimagined as recycling (battle) bins, the project aims to make environmental sustainability a fun adventure and challenge rather than just a responsibility.

Ecomon directly supports the University of Exeter's sustainability policies, particularly the Environment and Climate Emergency Policy Statement. The game integrates carbon literacy by linking the durability and effectiveness of cards to real world sustainability, plastics are weaker and degrade much slower, while recycled materials and plant cards are stronger and more reusable. This concept aligns with target 31 of the policy which aims for "95% reduction in single use plastic packaging". Additionally, the inclusion of plant cards based on native British trees reflects on target 41, which seeks to "Establish net positive plan for Woodland, Wetland and Campus wide biodiversity" aims to inform the players of the positive impact of their surrounding environment.

Beyond the digital mechanics, Ecomon requires engagement with the real world, players accumulate virtual rubbish (wrappers) when opening card packs and must visit

the physical recycling points on campus to dispose of these as well as take part in battles. This creates a tangible connection between the digital engagement and real world sustainability actions, hopefully this action will be mirrored in real life using the trip to deliver their own physical rubbish or habitually reinforcing the often small but integral journey to the bin. Alongside this, the competitive element of Ecomon further strengthens its environmental impact. Each day teams Reduce, Reuse, and Recycle compete against Team Fossil Fuels to reclaim recycling bins on campus. This daily battle parallels the university's mission for "transformational cultural change" turning abstract sustainability goals into repetitive action based engagement and acts as a reminder of our constant battle against our past and current carbon emissions.

To make sure our group collaboration ensured high quality code, we followed agile software development practices, incorporating the Kanban methodology through Trello and implementing structured code reviews using GitHub.

As Trello doesn't have a built in amount limiter for grouped boards we set up an automated board such as specify tracking, to go alongside specifying and specified, to become the group tracker ensuring we correctly followed the kanban methodology. We also included tiers from 1 (least important) to 4 (most important) that were simultaneously colour coordinated from red to green, representing the allocated importance of each card. Allowing our group to visualise the workflow and make certain the development pace was manageable and steady.

For version control and code collaboration, we used GitHub, owning a shared repository and enforcing a pull request approval system where each merge to the main branch required at least three approvals. This process ensures that multiple team members reviewed and tested the code before integration, reducing errors and maintaining our code quality.

We have operated under the kanban agile methodology of continuous development. Working quickly and in smaller, more manageable tasks. Everyone picked up their own tasks from the backlog and worked on it between meetings. Combining this with github branches features could be made in parallel with one another. Our channel of communication outside of meetings was Whatsapp where people could ask for help or discuss other ideas, not limited to meeting time, enabling true continuous development.

Reference: Environment and Climate Policy Statement - University of Exeter
https://www.exeter.ac.uk/v8media/specificsites/sustainability/docs/Environment_and_Climate_Emergency_Policy_Statement.pdf

Initial Design and Project Architecture

The main components of the web application are in a parent directory called **ecomon**. In the ecomon directory the front-end being utilised are Django templates styled with simple CSS. These templates interact with the Django backend python files (views.py and urls.py) to send and receive data to dynamically style the pages with context content. The data being processed in the Django backend interacts with our database, which are .sqlite3 files, to retrieve stored content about profiles and the overall application. The information stored in the database helps us keep context and content to what is happening in the overall application.

Our application is built using the Django framework, leveraging its robust architecture to manage both the frontend and backend efficiently. Django's built-in ORM seamlessly integrates with SQLite3, our chosen database, enabling efficient data storage and retrieval without the need for complex SQL queries. This structured approach allows for better maintainability and flexibility, making it easier to extend or modify different components of the application as needed.

Using the django framework the project is divided into two applications:

- **backend**: where most of the processing takes place for handling all the different features in our project including battles, profiles, packs, maps for gyms, etc.
- **accounts**: where authentication is managed.

Some of the larger tasks have files called {their name}_service.py which contains the main functions for this area. This helps break up the project and limit the number of merge conflicts due to working in a larger team.

Github is the backbone of the project where new branches are made for new components/features. Feature examples include the whole bin feature or pack opening feature. Using github actions CI/CD has been implemented by running unit tests on pull requests with main. This ensures no previous code gets broken when it's merged into main.

The application follows a standard Django MVT (Model-View-Template) architecture with client-side enhancements:

Client-Side (Frontend):

- HTML templates render the user interface
- JavaScript handles dynamic interactions, particularly with the map, scanner and battles functionality
- Client-side form validation improves user experience before server submission

Server-Side (Backend):

- Django views process user requests and determine appropriate responses
- URL routing directs requests to the correct view functions
- Django models define and manage data schema and database interactions
- Template rendering combines HTML templates with context data from views

Data Flow:

- User actions (clicks) trigger JavaScript events
- JavaScript initiates HTTP requests to Django backend endpoints
- Django processes these requests, interacts with the database as needed
- Response data is returned to the client (JSON for API calls, rendered HTML for page requests)
- JavaScript updates the DOM with new information without full page reloads

Map Functionality:

- Map interface loads gym/bin locations from Django backend via AJAX calls
- User interactions with the map (zoom, pan) are processed client-side
- Location data is cached in the browser for improved performance

Sustainability Challenges:

- User actions of opening a pack utilizes our bin_service.py file which adds a wrapper for each pack opened. The user cannot open more packs if their bin contains 3/3 wrappers. The user has to visit a recycling bin/gym in game, and initiate a battle with it. This promotes users to recycle within the app, while learning about where recycling bins are around campus.
- Through the users trying to finish their card collection, they learn more about sustainability in general as each card contains a relevant sustainability fact (there is also one for each gym/bin).

Authentication Flow:

- Django's authentication middleware validates user sessions
 - Protected views check authentication status before processing
 - Login/logout operations update session data stored in cookies
- Furthermore there is scope to have a JavaScript library as the frontend as there is a REST API for authentication.

The models were designed in a group discovery kanban to decide how the database should work and how the remainder of the app will work around it.

The models for backend: we wrote different classes for each database table namely: Team class, Card class, Gym class, PlayerCards class, Achievement class (implemented in sprint 2) and PlayerAchievements class (implemented in sprint 2).

- **Team:** Stores team information including name, color, and icon. Teams can be set as

user-selectable or not, controlling whether players can join them.

- **Card**: Represents collectible game cards with names, images, and game attributes.

Each card has two different abilities with varying power levels, health points, and an educational fact. Includes a JSON serialization method for API responses.

- **Gym**: Location-based structures positioned using latitude and longitude coordinates.

Each gym contains three cards and is owned by a player. Gyms have a cooldown period to manage gameplay pacing.

- **PlayerCards**: A many-to-many relationship tracking which cards each player has collected. Tracks whether cards are currently placed in gyms and their usage count.

- **Achievement** and **PlayerAchievements**: Implemented in sprint 2, contain the type of achievement, the name and the medal image. Whilst the PlayerAchievements monitors which players have completed each achievement and if their pack has been rewarded.

The models for accounts: uses database structure with a weak entity called Profile on the inbuilt Django User model. It stores the user with which team they belong to, all three of the chosen cards for their current active deck, their pack count and wrapper count.

Testing

Testing has proven integral to ensuring previous features have not been overwritten. Django unit testing has been implemented and can be run by `python3 manage.py test`. This is automated testing

Automated Tests include:

1. Accounts
 - a. Testing Custom User Creation Form
 - b. Testing REST API authentication
 - c. Testing initial routing, when the user is logged in and not
2. Backend
 - a. Ensuring pages that require a login are only accessible when logged in
 - b. Ensures each page renders a valid template file

With over 50 unit tests, the backend is sufficiently tested with a comprehensive suite of test cases that cover: functional testing of all features, edge case testing, user acceptance testing to ensure the app meets the project's requirements. The majority of the iteration one testing was user acceptance testing and system integration testing. This is completed by manually testing the features and all possible edge cases within the bounds of the front end. In sprint 2 we furthered tests, ensuring the entire system is fully tested and accounted for.

Manual UX tests include

1. Accounts
 - a. Creating new account
 - b. Logging out via profile
 - c. Register
2. Backend/ Wider System
 - a. Packs
 - i. Opening valid pack
 - ii. Stopped opening pack when no packs available
 - iii. Stopped opening pack when wrapper full
 - b. home
 - i. User can drag around and click gyms
 - ii. All buttons point to the correct endpoint
 - iii. Centers to current location with locate button
 - c. Profile
 - i. Logout button
 - ii. All collection successfully renders
 - d. Change deck
 - i. Selecting cards for your deck
 - e. Bins (Gyms)
 - i. Viewing a gym
 - ii. Battling against a gym
 - iii. Winning the gym
 - iv. Loosing against the gym
 - v. Wrapper recycle upon battle finished
 - f. Scanner
 - i. Scanning QR code
 - ii. Uploading QR code file

Overall the initial requirement analysis was used to validate required features in sprint 1. This document outlines the project and cross validates with the current system ensuring its user features have been implemented.

CI/CD

Github actions are used to run unit tests on every pull request to main. It downloads all requirements, migrates the database, then runs all unit tests on the current pull request. This helps the existing feature's integrity ensuring they have not been overwritten. This

has prevented bugs being merged into main numerous times, often simple file structure bugs, but saves a huge headache!

Cards:



Features:

The cards themselves are a core component of the Ecomon game, each user collects them through packs and uses them in battle to unlock further rewards. Cards are stored in their table of the database with all their stats and features outlined below:

Firstly, each card follows a template depending on their type. The three types of card are ‘plastic’, ‘recycle’ and ‘plant’ with plastic being the most common and plant being the rarest. In each class, every card is a unique item related to garbage or sustainability, shown above in the example photos.

In each card, they contain stats such as HP (Health Points) and their own unique ‘sustainAbilities’, these abilities are two moves used in battle by the user to cause damage to their opponents. Each sustainability attack move 1 will do a certain base level of damage to their opponent. For rarer cards of recycle and plant, sustainAbility two moves have two effects, a damage factor to their opponent and a heal factor to improve their own health. This increases the game's tactical aspect as you balance choosing which move to make.

Finally, each card has a sustainability fact to promote environmental awareness and provide stats about waste. These appear on the cards so when a player unlocks a new card they learn a related fact raising awareness about sustainability.

Assumptions:

No assumptions were made for the cards as they are key elements of the game that don't rely on any aspects

from the user. The group created all the cards and the team originally thought up each move name and unique feature.

Sprint Overview:

In Sprint One, 18 cards were created (9 plastic, 6 recycle, and 3 plant) as whilst being a core feature of the game, only a few are required to demonstrate the key aspects of battling and packs. For Sprint Two, double the cards were introduced, allowing for a more personable game and a greater challenge to pack all the cards. The final 36 cards vary in stats and abilities to create a different game experience for every time someone plays.

Future Sprints:

Future sprints would start to include additional types of cards such as a “made with recycled materials” items or implementing different types of support cards such as energy which could include solar, wind, coal and more.

Currently the cards have been balanced as we've constructed them however later sprints could utilise user feedback in order to create a more even playing environment.

Profile Page



Features:

The user profile is a crucial page for a user to see all their important information and game stats. Beneath the Ecomon logo, it contains their important personal details such as their username and email. It also displays the team they chose with the team logo appearing on the side.

Similarly, the profile page contains the bin wrapper information. This is a feature of the game to promote recycling and sustainability. After a player opens a pack, the wrapper count increases by one and packs can not be opened after a full 3/3 wrapper count, at that point a player has to win at a 'recycling bin' and 'empty their bin' to be able to carry on opening packs.

The profile page also contains the players' chosen deck, this is the collection of three cards chosen by the user for battles. The profile page displays their currently chosen deck and offers a button to visit the 'change deck' page and change their active cards.

Next, the profile page contains the users' 'obtained cards'. This section checks the player_cards table of the database and for each card the player owns it allows them to view the card. For each unowned card, a template border is left in place so the player knows how many more cards they need to unlock.

Under the user's obtained cards, the achievements section appears, with personally made bronze, silver and gold medals rewarding players with a pack for winning battles, opening packs and emptying their bin. Each statistic we wanted to track was added to the profile model and each achievement can be created in the django admin with the ability to set the achievement type, amount threshold, and medal type. After each event that could trigger an achievement, a function is run to check and award if any of the milestone amounts have been reached, once this statement becomes true the PlayerAchievement model updates the date achieved and sets a boolean value for the pack awarded to ensure it isn't given again. When the user next goes to the profile page a corresponding medal will appear, as well as a pack in their total packs.

Finally, the logout button appears on the profile page allowing currently logged-in players to sign out and return to the login page to access a different account.

Assumptions

There are very few assumptions to be made on the Profile page. The only clear assumption is that anyone who accesses the page should have a valid setup profile in the database, however, this is always going to be the case as the page is inaccessible until log-in has been completed.

Sprint Overview:

In Sprint One, the user details alongside the obtained cards were introduced on the page, this was all in a simple layout that detailed the clear information the player needed to know. Also, the change deck featured where the user could see the cards they had selected for battle. A template was introduced for the achievements section that we would build upon in the next sprint.

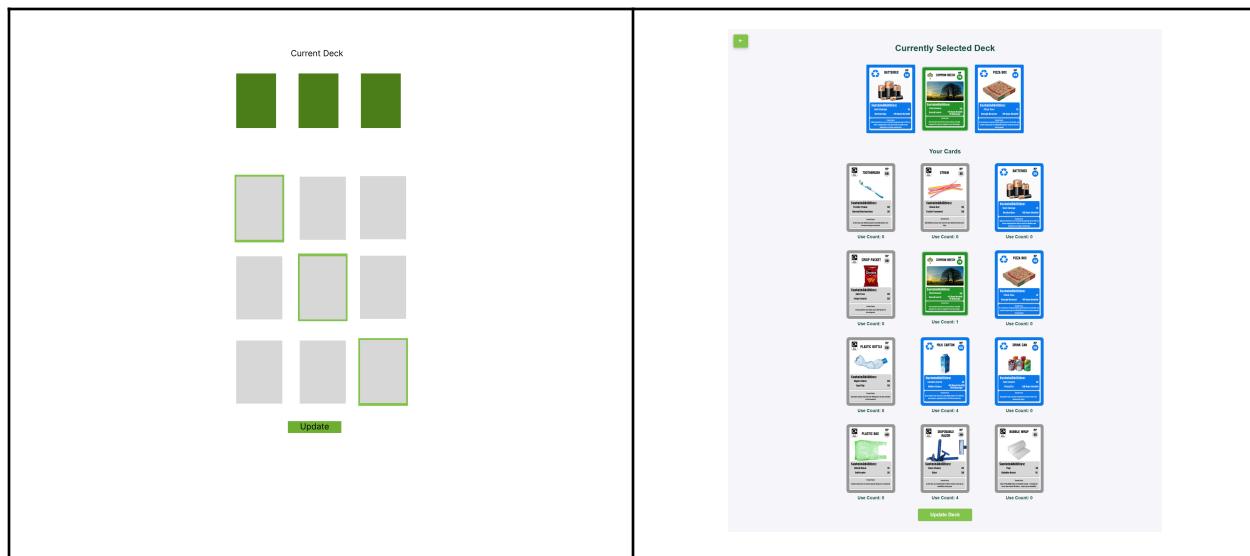
In Sprint Two, major hauls were made to the profile page. The achievements section was fully implemented and displayed with custom made medals for different milestones. The obtained cards section feature improvements, a card flip animation was implemented to increase user engagement and the degraded cards that had passed their use count would no longer display on the page. Similarly, animation effects were added to the team logo and bin lid to increase page uniqueness.

Future Sprints

The degraded cards instead of just not appearing in the profile page would still appear just greyed out with a degraded symbol over the top so the user can reflect on which cards they have acquired before.

Parts of the achievements section could be moved from the profile page to a separate section that includes a more detailed overview, a greater number of achievements as well as additional features such as achievements popping up when the user passes a milestone.

Change Deck Page



Features:

The Change Deck page is a simple follow-up to the profile page that allows the user to create their chosen deck for battle. The HTML loops through every collected player_card and displays all the ones owned by the user. Then the user can select three cards and update their deck ready for battle.

The Change Deck page also features a simple back button universal across many of the Ecomon pages that returns to the profile page where the user can see their newly updated deck

Assumptions:

An assumption made when we created the Change Deck page is that each user will have at least three cards to choose from when they prepare for battle. When the user

first creates an account they are granted two welcome packs providing three cards each. Whilst possible, it is very unlikely a user will pack 5 of the same card out of the 6 provided so we assumed in Sprint 1 that the user is near guaranteed to pack at least three different cards in the initialisation.

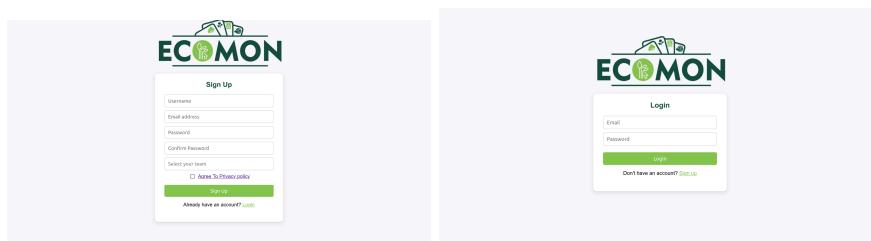
Sprint Overview:

As it's a simple page, most of the developments for the change deck page came in sprint one, the loop of displaying obtained cards and allowing a player to choose three cards and update the deck came early on in development. In sprint 2, we used feedback from playing the game to update two major features, firstly the use count of each card was displayed under each card so players know how many more times they can use it in battle, also an update with Javascript came where player's cards were added to the deck in the order they were selected. This was a small change but a major improvement, as it allowed you to ensure your best card could start first in the battle and give you a game advantage.

Future Sprints:

Currently the deck requires the user submitting all three cards meaning that if the user loses a card during battle they have to select the two cards already in their deck alongside their new card. This can be improved by being able to select the slot you want a card in and then the specific card as a small quality of life improvement change.

Authentication



Features:

The authentication page (the login page), is the first screen the user will be in contact with when initially visiting the site. They will be prompted to either login with a previous login granting them authentication, or signup and create an account with which they can do the same thing. This is done through the django template and api endpoint using a token system. This creates a mechanism for which user information is validated effectively allowing for easier access to their profile information. Once you have logged in / been authenticated you will be redirected to the home page. If when logging in you enter invalid details, the page will prompt you on the specific error. For instance, if you

have a password that is too similar to when trying to create an account it will prompt you telling you that.

The signup page also acts as a mechanism to ensure we are handling users data correctly. When signing up you must agree to the tickbox containing our privacy policy in order to create an account. This allows our storage and usage of user data to fall in line with GDPR compliance.

Assumptions

One assumption we have made is that user emails will be trusted as “verifiable emails” this is because we cannot verify every email entered due to email server and domain limitations.

Future Sprints:

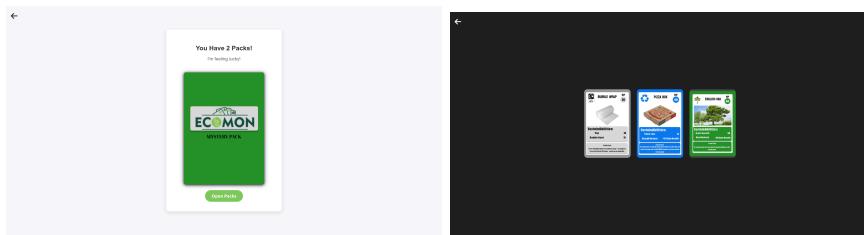
No further improvements are needed.

Sprint overview:

In sprint one, we added the main login page and the signup page to create a new account. Once logged in, you are authenticated and sent to the home page. Finally, we added the privacy policy tickbox to allow users data to be handled safely.

In sprint two, smoothed out the bugs in authentication. Added relevant prompts on users entering the incorrect details and when creating an account it will state the exact issue with the submission.

Packs



Features:

The packs page is one of the main pages of the app which allows users to open packs they have in their “inventory”. This page focuses on trying to “gamify” the process while still educating the user on important sustainability information. These packs contain a variety of cards based on sustainability concepts which each hold unique fun facts

around the specific cards. This allows the user to be educated about environmental ideas while playing.

The number of packs a player has is validated by the backend and displayed on the page clearly. We also added a regular pack addition aspect, where if it has been more than three days since the user has last received a pack, the user receives a new pack. When opening there is a custom animation to catch the players eye and add some of gamification features as discussed earlier. The cards that are unboxed are one of three “types” which each have different chances of being opened; this allows for added engagement when getting new cards. To make collecting all the cards more difficult, the player can pack duplicate cards.

Assumptions

There are no significant assumptions that have been made around the opening of packs.

Sprint Overview:

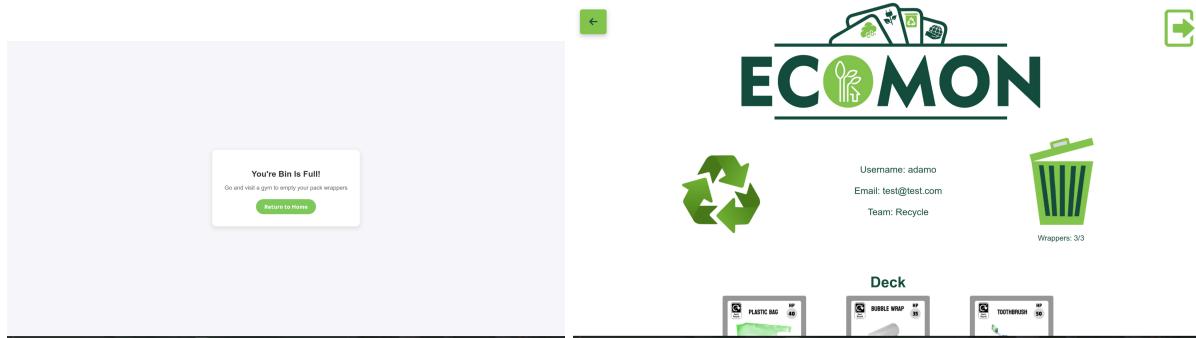
In sprint one, we created the pack opening page as well as the cards inventory page. The packs page has a couple variants, one being the no packs variation page, where the user has no packs left, as well as the regular pack opening page.

In sprint two, we added further animations to enhance user experience and make the app more eye-catching. We also added an interval where the user can more regularly receive packs to keep them playing the game.

Future Sprints:

In future sprints we will likely polish the “unboxing” animation of the packs. This is to further add engagement and satisfaction to the user to encourage them to continue playing and therefore learn more about the effects of various items on the environment.

Wrapper Feature



Features:

This is a feature which embeds sustainability into the project. The feature works like so, the user cannot open another pack until they have recycled their “wrappers”. To recycle their wrappers they must visit a “bin” and battle against the reigning champion. In order to do this you will physically need to be near the bin to scan the QR code. This incentivises recycling as it is strongly tied to the location and action in game. This wrapper count is integrated into the profile screen so players can see if they need to battle and recycle their wrappers.

Assumptions:

There are no significant assumptions made for this feature.

Future Sprints:

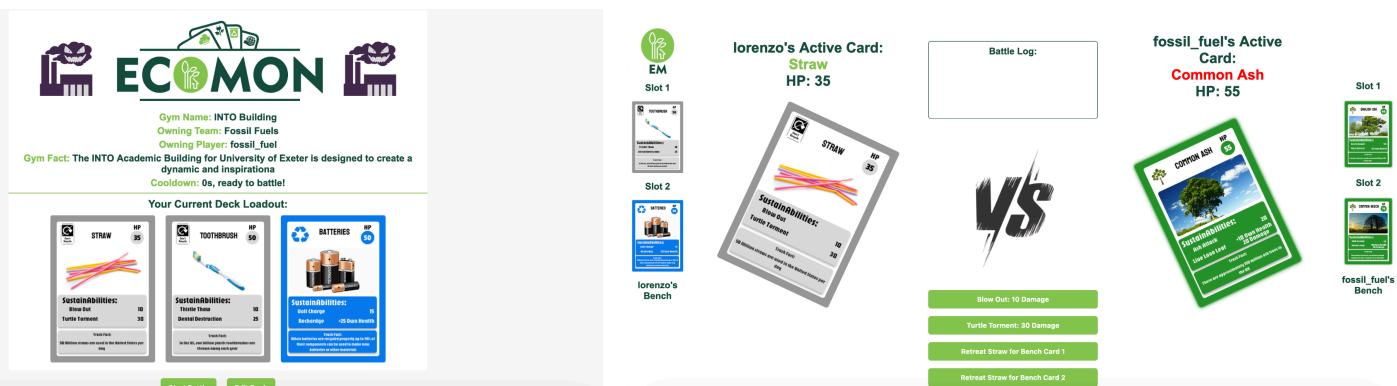
There are no additional tweaks that will be introduced to this feature in future sprints.

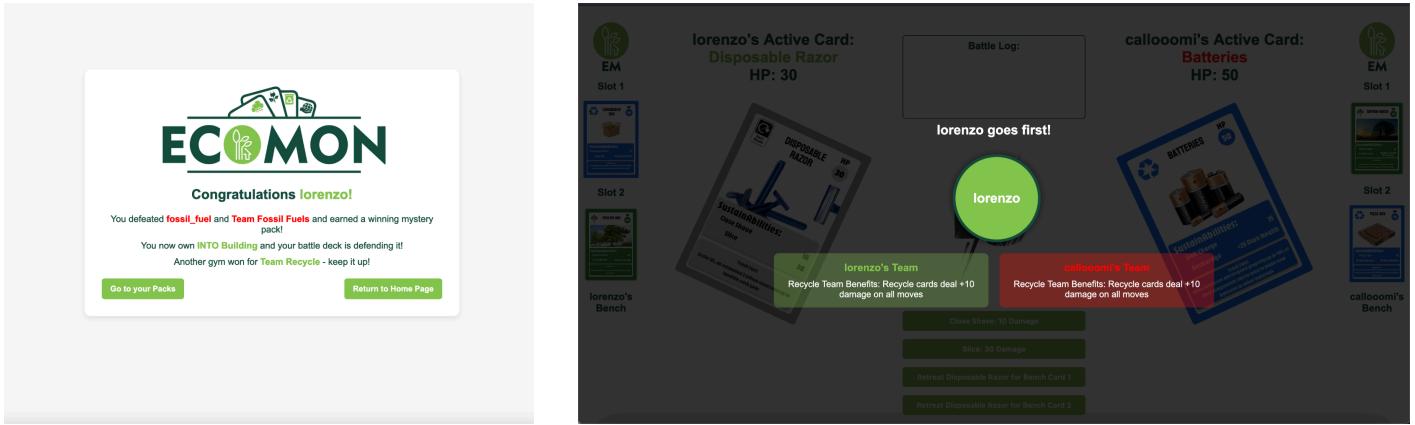
Sprint overview:

In the first sprint, we added the entirety of the wrapper feature.

In the second sprint, we thoroughly tested the feature.

Bin Battles





Features:

This is a core mechanic of the game which ties in both pack collection, recycling and gamification. The feature works like so, a user goes to the physical location of a bin (which are all marked on the map on the home page) and scans the qr code located on the bin, this activates the view battle screen. This will allow the user to see the details of the gym, e.g. the owning team or the fact. The user will have their deck loadout output to the screen and they can also edit their deck by clicking on the “edit deck” button.

Validation is enforced to ensure that when the “start battle” button is pressed:

1. A player cannot initiate a battle with a gym that is owned by the same team as them (this is done as in future sprints there will be a leaderboard system for each team, this will increase competitiveness and hunger to play more!).
2. A player cannot start the battle without having three cards in their loadout.
3. A user cannot initiate a gym battle if the gym has a cooldown (this cooldown is set to 30 mins after a gym is taken over, this is implemented to stop users from farming packs).
4. A player cannot start the battle if they are not within a certain radius of the gym (this is currently set to 200m to account for geographical limitations of getting the users location, but is subject to change if this is too large/small of a radius). If all of those conditions are valid, the user can start a battle!

In the battle the player will face the current “owners” of the bin (gym), this could be team fossil fuel or any previous players who have claimed the bin. The current owners will have previously selected three cards to defend the recycling bin. How it works is after a player defeats a gym, the cards that were used to defeat the gym are “left behind” to defend the gym. This means that the player cannot use those cards in other battles. This adds a more complex strategy for the players deciding which cards to leave and defend the bin (gym) and which cards they can use in other battles. The cards selected by the attacking user and the defending cards will battle using a variety of moves specific to each card. This is done through a battle simulation. Each card has 2

“sustainAbilities”, either dealing damage, healing its own HP, or a combination of both (for rarer cards). After taking over a bin the user gains a pack to open and furthers them in completing the collection of cards, this creates an incentive for players to be around and participate in recycling. However, in order to stop users abusing this mechanic to get an unlimited supply of packs, there is a cooldown on taking over gyms in place. The battle feature also contains a core sustainability feature actually embedded into our app. When the user opens a pack, they dispose of their wrapper in their recycling bin on their home page. This has a cap of three wrappers, after the player has three wrappers, the player needs to visit a bin/gym (which is represented as recycling bins spread around campus) and battle it. This will simulate the user visiting a recycling bin in real life, and in game. After the battle has finished, the player “empties” their wrapper bin and the count returns back to 0. This further promotes users to actually recycling in the game and in real life, while furthering their progression in the game (to open more packs).

Assumptions

The battle simulation occurs client side, this could cause potential problems if the client is a dishonest agent and reports the outcome of the battle incorrectly, therefore rewarding them free packs (and future points for their team). However, in the specification it states that actions can be user validated and therefore we have made the assumption that users are honest agents and therefore will not cheat. The user knows to set up their deck in a certain way, with deck card slot 1 in the view-gym page being the card that will be in the active slot, then deck card slot 2 and slot 3 being in the bench slot 1 and bench slot 2 respectively. It is assumed that the user also knows to strategically select these cards in an effective order, as if their active battle card “dies”, the card in slot 1 will automatically be brought in. If the card in slot 1 is also “dead” it will check slot 2.

Sprint Overview:

In Sprint 1, the initial implementation described in the features section was implemented. After a successful QR code scan, the view-gym page was initialised. This allowed the user to view all gym details including the owning player and team and gym cooldown. The user can also change and choose their deck cards to battle the gym. The battle constraints preventing a battle include: being owned by your team, not in the radius, if their deck has empty cards, and finally if the gym is on cooldown.

Implemented that if a successful gym battle is initiated, the user is taken to the battle screen where the user always starts first (this is changed in sprint 2). The user takes turns with the AI to make moves based on their card sustainAbilities. The win condition for the battles is that after a player/opponent loses all 3 cards (all cards die), they lose the game. If a player wins the game, their current deck defends that gym and they

cannot use those cards in future battles until that gym is taken over by someone else. The player also gets awarded a pack for winning and their wrapper count bin is emptied.

In Sprint 2, further enhancements were introduced to decorate the battles page and provide enhanced user interactions and feedback. By including animations it just makes the battles look cleaner and makes them more interactive. The battles in sprint 1 were designed to always allow the user to make the first move, however this was unfair and could make the game boring for the user (too easy to defeat a gym). To combat this, a coin-flip would give random and fairness to who goes first. Before the battle begins, an overlay covers the battle screen and simulates a coin flip with the user's username, and the opponents username. The coin flip is simulated using CSS keyframe queries. This overlay also allowed us to integrate our team benefits to battles.

Team benefits were a brand new addition to Sprint 2 development, and it allows users to choose their team more wisely as each team has a certain in-game benefit over the others. The overlay showed the user their own team benefit, and their opponents team benefit. This allows the user to strategically plan their attacks, as they can know what benefit (power ups) the opposing team has. This is further explained in detail in its own dedicated section, where we speak about brand new implementations in Sprint 2.

We also normalised battle interactions and made them more fair. In the previous sprint, players could use their “heal move” to heal health points above their max (starting) HP. This allowed users to “spam” healing moves and either cause stalemates, or make overpowered cards. To combat this, an HP constraint was added that doesn't allow users to heal above max HP. For example, if a card has a max HP of 40 and their current card has 35 HP, and the user heals with their sustainAbility, only 5 net health points would be added to their active cards healthpoints.

To enhance battle playability, animations were added to provide better user feedback. Each used move causes the battle log to flash. Each damage move “attacks” the opponent using CSS keyframes to change the position of the cards and make it smooth. The card comes out of the active spot and enlarges while swiping to the opposing card, while the opposing card gets smaller which shows damage being inflicted. While if a heal move is used, the card also enlarges and gets a colorful greenish overlay to show the heal being processed. Furthermore, damage and heal tags showcase how much damage and health points were processed for each move. On top of this, the battle feature was mobilised in Sprint 2 to provide modularity and increased playability across devices.

Finally, to provide further sustainability embedded into our application, card use count was implemented, where cards degrade over time from the amount of uses in battle. This will be further explained in its own section, however all that is needed to know is that after each battle the application checks for any cards that have ‘degraded’ after battle (if their current use count exceeds their max card use count), and displays the card/s to the user. This provides enhanced user feedback for players, as they can be made aware of cards that have decomposed and therefore are no longer usable in further battles until they have to re-pack the card/s. This is checked after every completed gym battle, regardless of whether or not the user won the battle.

Future Development and Improvements:

Making the AI mimicking opponent choose smarter moves (i.e. choosing the more powerful moves), as currently the move which an AI opponent does is randomly generated (50/50 between the 2 sustainAbilities). This was not implemented in Sprint 2 as it proved to be a lot more complex than initially thought out. The most optimal move is very subjective and context specific to which card is currently active, based on what the card type is and what the move sets include. This was also not implemented as it does not make the game unplayable, battles are still fair and the opponent can still win battles, especially with the HP constraint being implemented where a card cannot use a heal move if their HP is already at max.

Resetting the Bins/Gyms each day

Features:

We have decided that it is important to include mechanics that encourage players to consistently engage. One of the ways in which we have done this is by resetting all gyms at the start of every day. By “resetting” we mean that team fossil fuel will take over each gym so that players can battle over gyms throughout the day and harder to reach bins are not left to one player for extended periods of time.

This is done by creating a custom cli command to reset the gyms “python3 manage.py reset_gyms” and running a cron job (django-crontab) to run the custom command which resets all gyms.

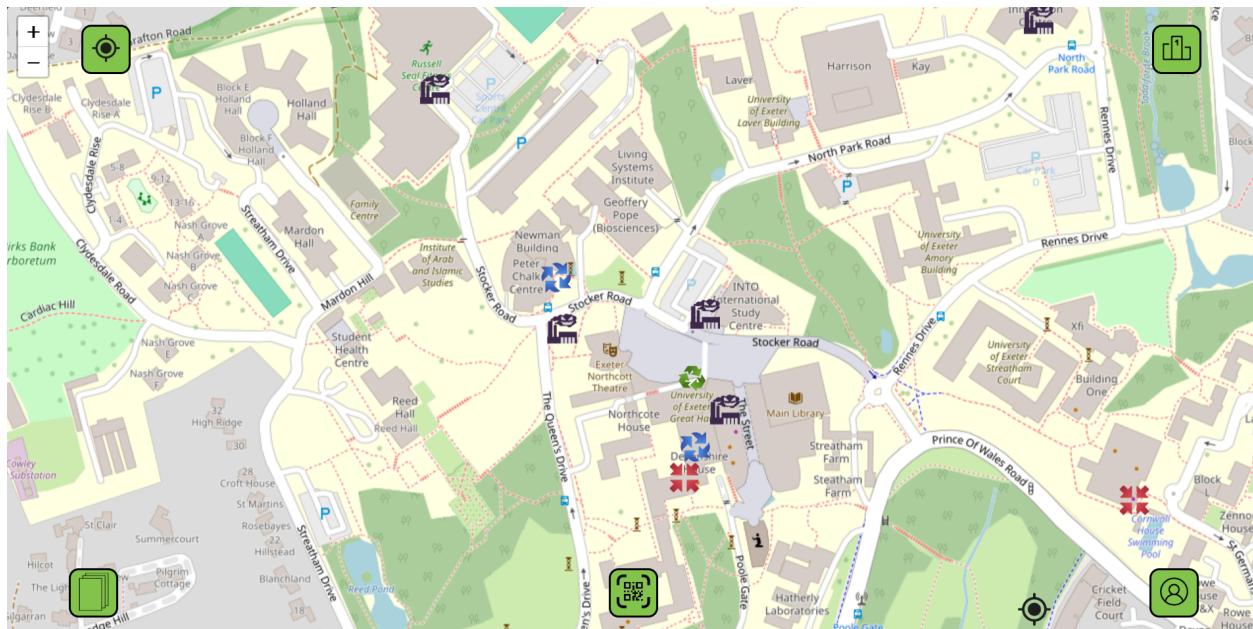
Assumptions

For this feature we have assumed that there will be a semi significant player base as if there are not enough players to take over all gyms near daily there would be no point resetting the gyms.

Future Sprints:

There are no future developments for this feature in following the sprints.

Home page



Features:

The home page is the main page of the app which you are directed to once you login, it contains a host of links to other pages on the app. These other pages include, the pack opening screen, the QR code scanning screen, the leaderboard screen and the profile page screen.

Furthermore, the background contains a dynamic map which contains the user location and the location of all bins you can scan, if these bin location markers are clicked on it displays the name of that specific bin. If the player icon is clicked on it says “this is you!” and tells you the radius accuracy of the location marker of the player (which will be different depending on the device, location and internet connection being used).

Additionally, there are three buttons, two in the top left and one in the top right. In the top left there is a button with + and - respectively. These allow you to zoom in and out if

you don't have a touch pad. Additionally, there is also a button in the top right which centers the user on the map in case they have moved away.

As an additional note, each of the pins for both the player location and the bins are customised. In the case of the player it's a crosshair icon and for the gyms the icon is the team that is currently owning the gym this makes it clear to users which gyms have already been battled and may be a tougher fight.

Finally, the entire page also changes dimensions when accessed on mobile in order to make the app more flexible and convenient for all users across platforms which encourages additional recycling.

Assumptions:

The biggest assumption made for the home page is that the user is using a device and in a location that allows for semi accurate geolocation, this is because there is no more accurate way to locate the player without using other more expensive solutions.

Future Development:

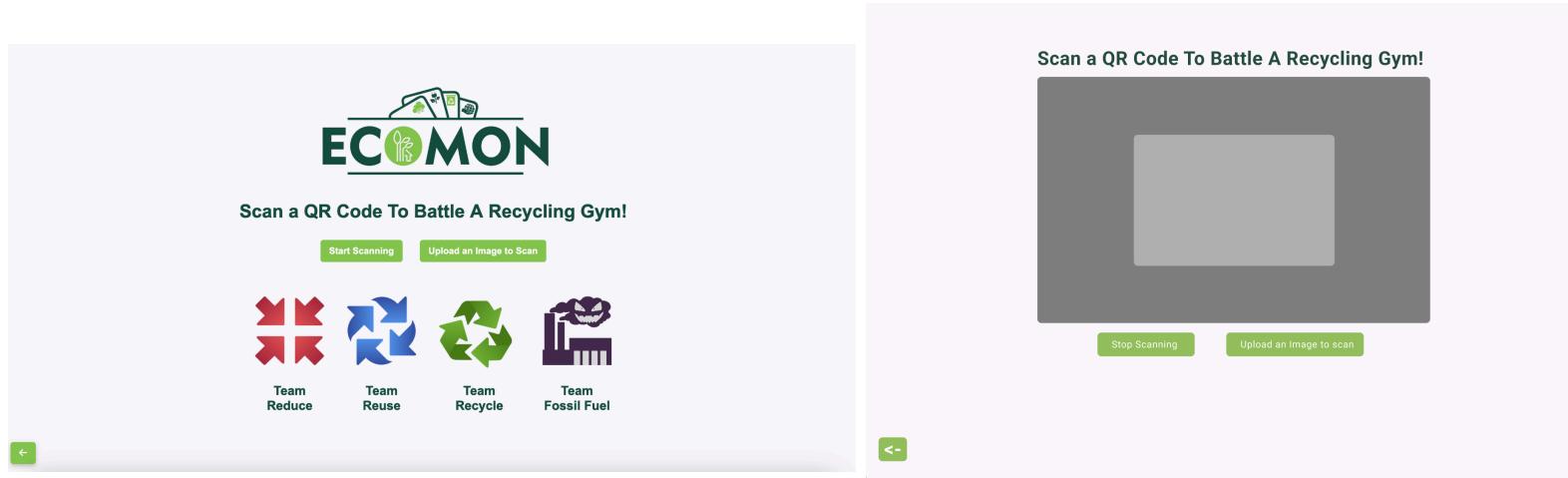
In the future, there is the possibility of adding automatic player location updating as opposed to updating on player click of locate button or reload of page. Additionally, we could also add shadow pins to the bins and player to clean the aesthetic a little more. Finally, we could consider moving to a different map provider such as google maps API which may provide a better geolocation service than a open source system such as leaflet, however this may not happen due to the costs and time constraints.

Sprint Overview:

In sprint one, we added the packs, scanner, profile, player locate and + - map buttons. We also added the map itself as well as "testing" bin pins and a pin on the player's current location.

In sprint two, we added new markers for the gyms based on the current player team that owns the gym. Additionally, we also added the leaderboard button and radius' around players and gyms when clicked.

Scanner Page



Features:

The scanner page is the main way for the user to interact with a recycling bin (gym) to fight them, earn rewards and empty their wrapper bin. The page can be navigated to by pressing on the scanner icon on the homepage of Ecomon. Upon opening the scanner page the user is greeted with 2 button options, either to scan a physical QR code or upload an image to scan. The “Start Scanning” button will initiate the QR code scanner and the “Upload an Image to Scan” button will not initiate the QR code scanner however will scan an uploaded version of a QR code. The QR code has input validation when uploading an image, and will only accept png, jpeg, jpg and gif files. The desired QR code will contain a URL corresponding to that particular gym’s id, for instance “127.0.0.1:8000/view-gym/1/” or “www.ecomon.org.uk/view-gym/1/” if the gym had an id of 1. The scanner page also features a back button to return the user to the home page, this helps ensure user navigation across the application.

Sprint 1 Overview:

The initial scanner page was implemented, where a user could scan a physical image of a QR code or upload an image of a QR code. The main assumption here is that the user would only use the QR code scanner on QR codes generated by us, as the user could scan any QR code. For instance, if a QR code of “www.youtube.com” was scanned, this would redirect the user to youtube.com. The user should only ever scan or upload an image to scan of QR codes generated by us to represent each respective gym. This ensures that the user can interact with our created recycling gyms (bins) to earn rewards and empty their wrapper bins.

Sprint 2 Overview:

In Sprint 1 it was assumed that the user should only scan valid QR codes generated by us, however users could still scan other QR codes and it would redirect the user to

potential threat websites. This was mitigated in sprint 2 as QR code validation ensures that a scanned QR code matches the “www.ecomon.org.uk/view-gym/1”, where 1 represents the gym ID of a certain gym. If any other QR code is scanned or uploaded, it will not redirect the user and ensure that only valid urls are scanned. Mobile development was also integrated into sprint 2 and made sure that the user’s back facing camera was selected for mobile development to scan QR codes.

Creating Bins

It's important that new bins can be created for new locations. Navigate to /admin and login using the admin credentials. Press gyms and create a new gym with random cards and the owning player being fossil_fuels. Ensure the latitude and longitude match the location required.

Then create a qr code via a service like <https://www.qrcode-monkey.com/>. The url is [https://\[host\]/gym-battle/\[gym_id\]](https://[host]/gym-battle/[gym_id]). This can be scanned then they can battle the gym provided their location is within the required radius

Leaderboard Page(s)

The image contains two screenshots of the ECOMON application. The left screenshot shows the 'Player Leaderboard' page with the title 'ECOMON' at the top. Below the title is a table with the following data:

Username	Team	Owning Gyms	Battles Won	Bins Emptied	Packs Opened	Card Types Owned	Collection Total
calloconi	Recycle	1	57	13	29	2 8 7	17
adamo	Recycle	0	0	0	3	3 15 7	23
Jake	Reduce	0	0	0	45	5 18 8	31
lorenzo	Recycle	0	0	0	0	1 7 4	12

The right screenshot shows the 'Team Leaderboard' page with the title 'ECOMON' at the top. It features four green buttons labeled 'Reduce', 'Reuse', and 'Recycle' with corresponding icons. Below the buttons is a table with the following data:

Category	Value
Currently Owned Gyms	0
Recycle Cards in use	0
Plant Cards in use	0
Plastic Cards in use	0

Features:

The leaderboard pages provide the users a way to interact with other users. There is a player leaderboard and a team leaderboard. On the player leaderboard it ranks the players with the most gyms owned currently. Along with displaying other stats like total gyms won and information about that player's cards. The team leaderboard shows stats about what type of cards are in the gyms and information about currently owned bins. It can be seen at 127.0.0.1:8000/player_leaderboard & team_leaderboard. The same back button on other pages is also used here.

Assumptions:

The bins get reset each night. So an assumption is that players will retake these bins each day. Otherwise the leaderboards will be mundane.

Sprint Overview:

This feature was created and started in the second sprint. An easy way to view and compete with the top Ecomon players. Adding to the game element, while enabling players to compete with each other.

Sprint Two Features:

As we moved into Sprint 2, we aimed to expand Ecomon with new features and refinements that benefitted the overall game and not just individual pages as detailed below. We wanted to ensure the project continued to grow as a representation of Exeter's sustainability goals. By incorporating our internal feedback and testing, we implemented key improvements to enhance user engagement and overall functionality.

Our first major introduction was the leaderboard, as detailed above. It collated the teams data and displayed stats like which team owned the most gyms and the most cards, similarly a personal leaderboard was created where players could compare their stats to individual opponents. The feature increases motivation and engagement, encouraging players to take part in sustainable actions like recycling to increase their team stats but also increases gamification as they battle to top the leaderboard.

Another feature introduced in Sprint 2 was team benefits. Through playing and testing, we wanted to introduce a purpose for the teams (reduce, reuse and recycle) that would increase gamification and offer a chance to embed sustainability into the games. We settled on creating team benefits, a unique power move creating an advantage for each team. Team Reduce would receive less damage in battle, Team Recycle would gain bonus attack stats on their recycle cards and Team Reuse could use their cards for longer before they degrade. These implemented features allowed for a more fun experience and introduced a purpose for the teams.

Also in Sprint 2, degrading cards were properly introduced. As per the sustainability goals at Exeter, we wanted to raise awareness about degrading and how plastic trash items take longer to decompose than their recycled counterparts. Thinking about this, we came up with decomposing cards, a way to gamify awareness on decomposing and landfill. As plant cards are the most powerful, and plastic are the weakest, we needed a way to balance the playing fields and remove the power for someone to just always play

a plant card in battle. By introducing decomposing cards, Plant cards would expire after 3 uses in battle, recycle cards after 5 uses, and plastic after 8. This improved the Ecomon experience in many ways, firstly, players could no longer abuse powerful cards in battle creating a more challenging experience as they tactically choose when to play weaker cards. Secondly, it raised awareness on sustainability goals and how plastic takes the longest to degrade. Finally, it improved the long-term enjoyment of the game, meaning players constantly had to repack cards and keep playing as they would never ‘complete’ the game and pack all the cards.

Finally, a major improvement we wanted to make in Sprint 2 was the mobilisation of Ecomon. As the game revolved around scanning QR codes on different recycling bins around campus. We agreed, the game would work much better on a mobile counterpart compared to a laptop. That is why in Sprint Two, every page was completely mobilised by first designing the mobile pages in Figma and then using CSS to implement the design, to create a fun and all inclusive Ecomon experience on the phone. This means now that the Ecomon game is hosted, players can access it from their mobile device and scan QR codes around campus easier to further enjoy the game.



These improvements alongside all previously stated throughout helped create a more captivating and accessible platform, ultimately driving greater participation not only in Exeter’s sustainability initiatives but in sustainable practices as a whole.

Conclusion

Our initial two sprints for Ecomon have successfully concluded, resulting in a stable game that effectively gamified real world sustainability by encouraging users to visit actual recycling bins around campus. In addition we have successfully developed with sustainability in mind, incorporating environmentally conscious practices into the game features that will be used when the user isn't near a bin such as our card decomposition and bin wrapper usage. This aligns closely with the University of Exeter's commitment to sustainability goals that we set out to help achieve.

Being able to work towards the goals of reducing single use plastic through the acknowledgement of plastic decomposition, and getting the user to travel to the bin. Whilst also informing about the british wildlife we're aiming to save through the plant cards.

Waste & Recycling	
31	95% reduction in single use plastic packaging
32	85% recycling or all waste
33	Campus wide waste recycle / recycling programme

Biodiversity	
41	Establish net positive plan for Woodland, Wetland and Campus wide biodiversity

Throughout development, we remained closely aligned with the feedback from the respective PTA investors we talked to, ensuring we consistently met their expectations and incorporated their desired features. This included making sure the application "couldn't just be reskinned" and was truly unique in its sustainability efforts, which became the driving force for our development and improvement of ideas. By incorporating their suggestions during weekly reviews, we refined key features, improved user interface and made our approach to group work even better. This iterative approach allowed us to achieve all the desired results outlined and strengthened the games impact and appeal.

From a development perspective, the majority of our group taught themselves the Django framework while also working with HTML and CSS for the first time. Despite this, we ensured high-quality code and efficient collaboration by meeting at least 3-4 times a week as a full group, with additional sub-group meetings when personal schedules conflicted. We also kept in constant contact through Teams and WhatsApp to ensure communication remained clear and continuous

Our development was further streamlined through the use of Trello for Kanban management and a structured GitHub workflow, which allowed us to maintain a steady pace and ensure task distribution was both efficient and well-coordinated. This approach allowed us to focus heavily on core features in the first sprint, while in the second sprint, we balanced our attention across various improvements, especially as our schedules became more intense with other commitments.

As a group we are proud to have met all key milestones in the initial sprint and to have completed all additional features and improvements in the second sprint with ample time to deploy and reflect effectively.

As with any product, we recognise that there is always room for improvement. In future sprints, we would focus on quality of life updates and further enhancements, ensuring that both the user experience and environmental impact of Ecomon continue to evolve and improve. However, we are incredibly happy with the final result, having delivered a polished and fully functional game that effectively promotes sustainability in an engaging and meaningful way.