

Sprint 1 Reflection Document

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 utilized 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 compute 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. These are often 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:** To be implemented in sprint 2, subject to change, but currently just contains the names of achievements, tiers and the date unlocked.

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 features 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

Majority of the iteration one testing has been 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 plan to continue to make these tests automated.

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

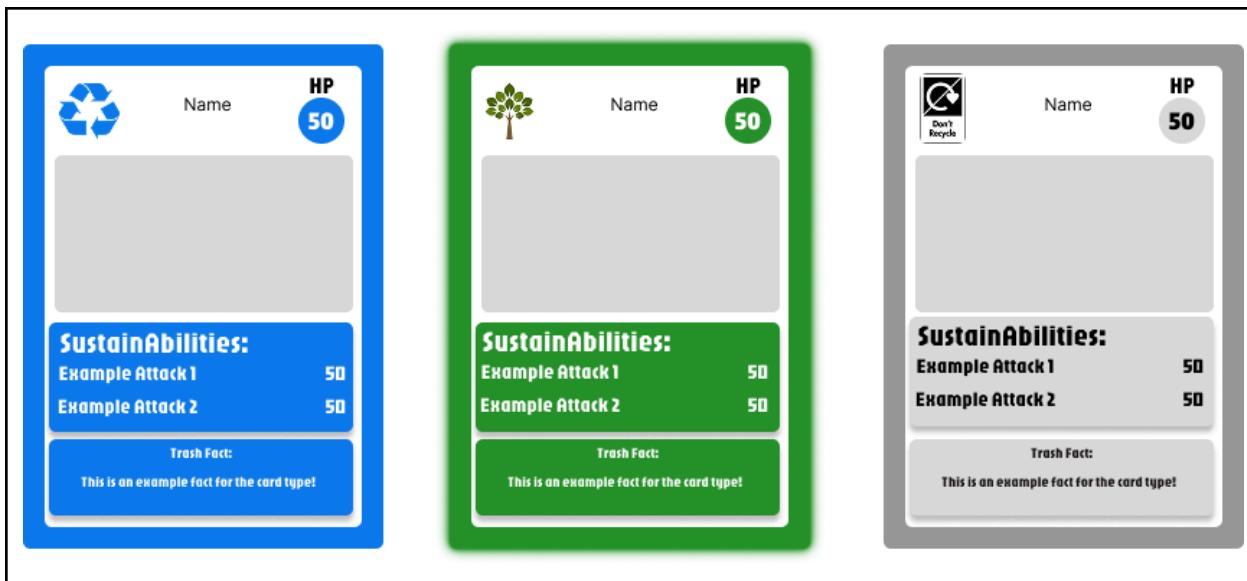
- ii. Battling against a gym
 - iii. Winning the gym
 - iv. Loosing against the gym
 - v. Wrapper recycle upon battle finished
- f. Scanner
- i. Scanning QR code
 - ii. Uploading QR code file

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, each 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.

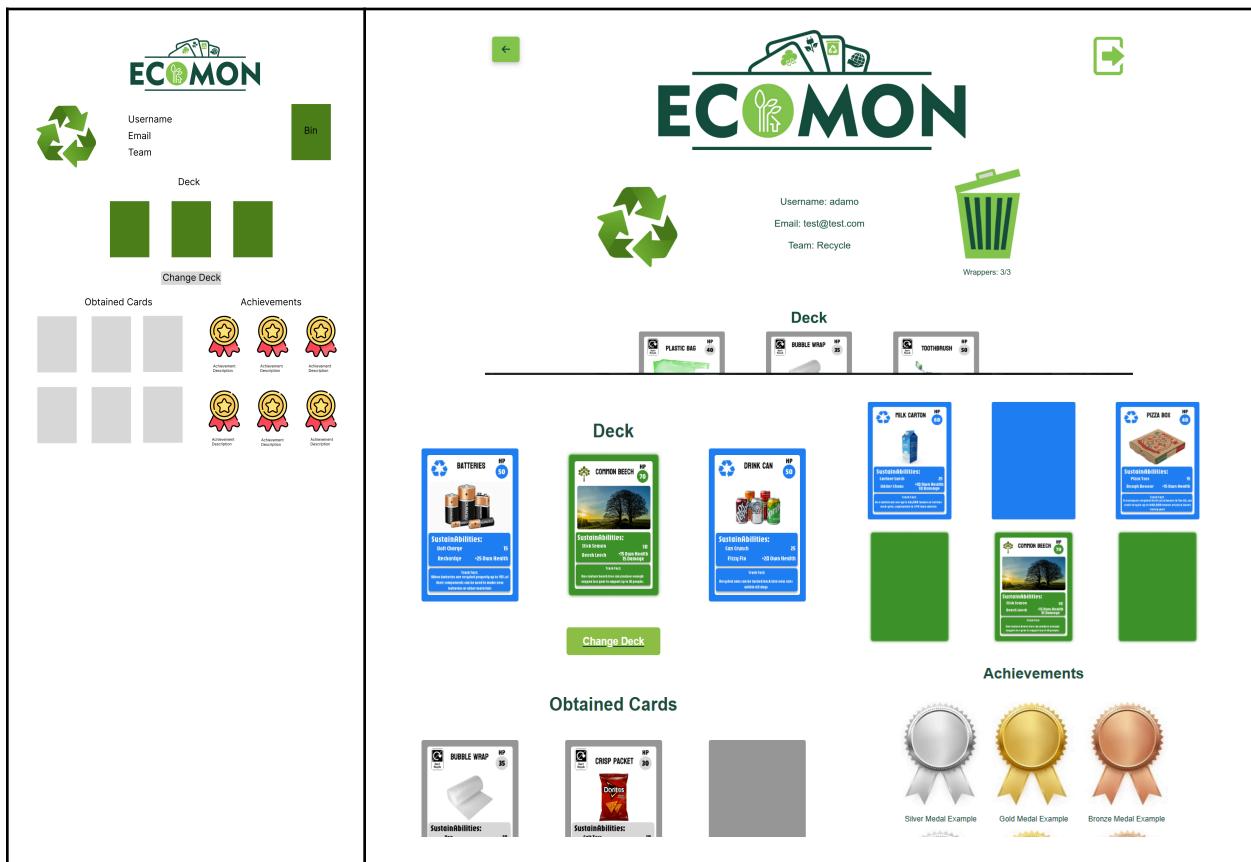
Assumptions:

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.

Future Sprints:

For Sprint One, only 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, we aim to introduce more cards. This works on improving gamification, as it makes more of a unique experience battling with and unlocking new cards, whilst also building on the sustainability point as it allows for more sustainability facts to be introduced and read by the users.

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 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. Similarly, underneath the cards obtained, the template for the ‘achievements’ section is displayed. This is an unfinished feature coming in Sprint Two as described below.

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.

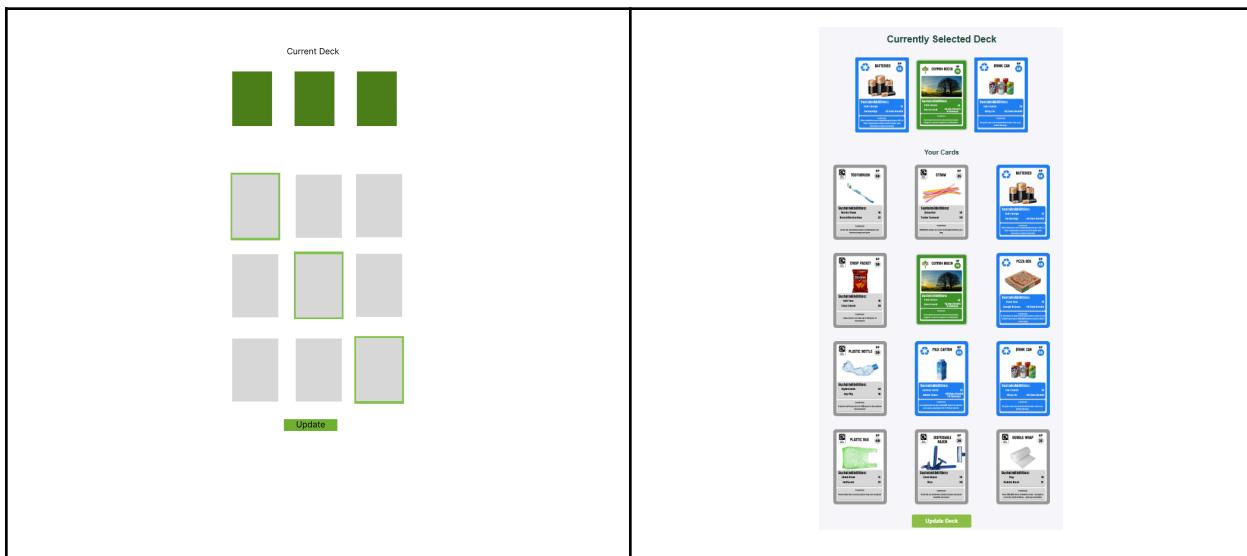
Future Sprints:

One component of Ecomon that will be introduced in Sprint Two is the achievements section. As designed in the UI mockup, we plan to create a new level of gamification enticing users to play for the long term and achieve continuing goals. The achievements section will reward players for completing tasks that take a long time such as logging in for an extended period of days or winning a certain number of bins. This feature will be introduced in the main release as it will keep users drawn to the game once they have started playing. However, it was excluded from prototype one as it isn’t crucial to the core aspect of the game but instead a further level to keep users playing.

Another feature to be built upon in Sprint Two is following on from the ‘Use Count’ attribute described above in the ‘Player Cards’ table. Recycle and plant cards will only have a certain number of uses representing their biodegrading factor and promoting sustainability. The cards obtained section will constantly change for users, not displaying cards they had once owned as they have now expired. We plan to introduce a ‘greyed-out feature’ that still shows users’ cards they had once collected that have

now expired. This will allow them to work towards obtaining a complete collection of cards whilst also keeping a core environmental point we are making with the cards expiring.

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

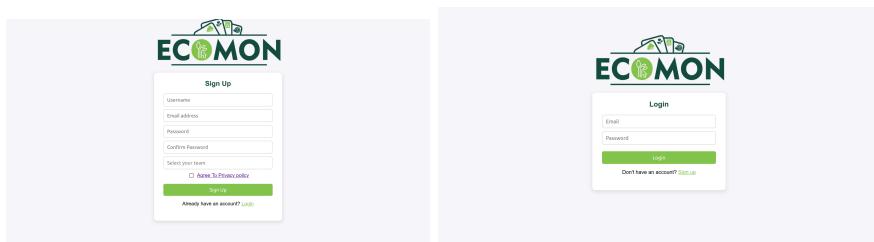
Assumptions:

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 3 different cards in the initialisation.

Future Sprints:

Similar to the profile page, the Change Deck page will likely build on the use_count feature of cards coming in Sprint Two. As explained above certain rarer cards will only have a limited use count to convey biodegradability in recyclable objects and plants. We plan to introduce a small feature where under each card you own, it displays a small number of how many uses that card has left before it expires. This will help the user tactically plan their decks for battle, possibly saving strong cards for only tougher battles where they require them.

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.

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

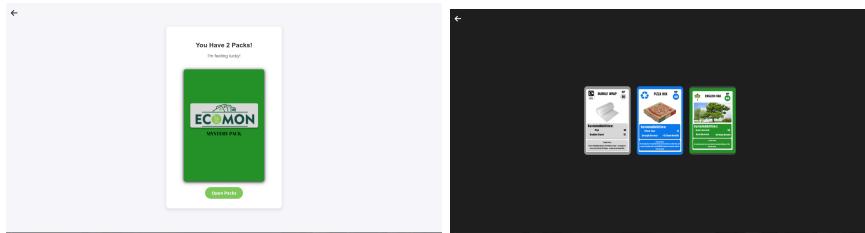
One assumption we have made is that user emails will be trusted as “verifiable emails” this is because we cannot verify every email entered due to email server and domain limitations.

Future Sprints:

There are some small bugs around logging in, signing up and signing out. An example of this may be that sometimes when you sign out it sends you to a page where it says

“sign in successful”. This will be polished out in future sprints in order to make the authentication of users more consistent and understandable for the user. This is not a priority for this sprint as it does not compromise core functionality needed.

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. 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.

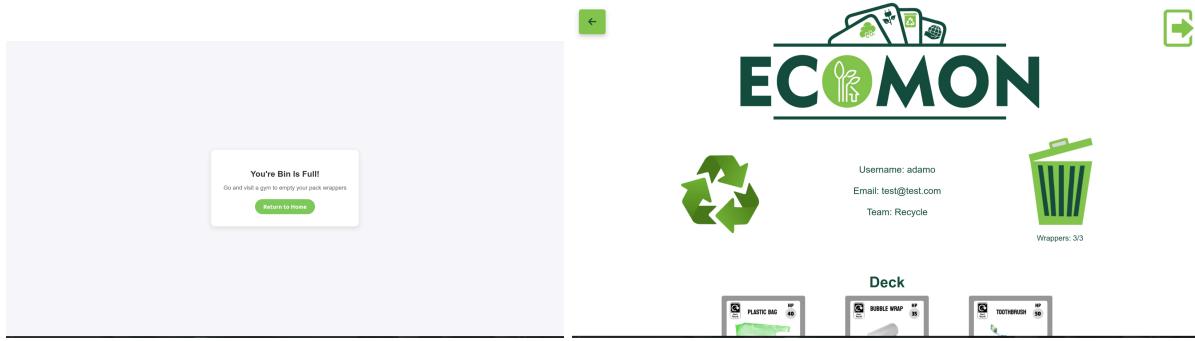
Assumptions

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

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.

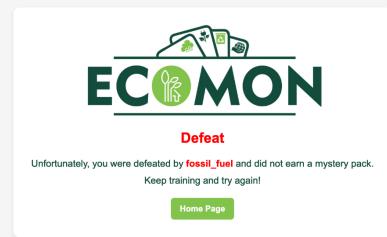
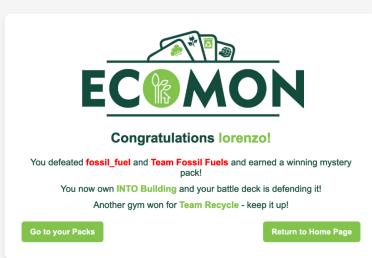
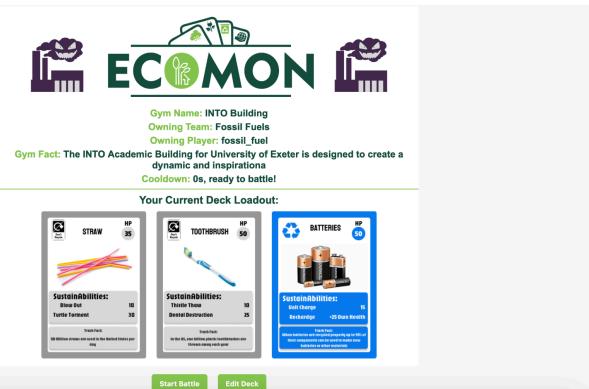
Assumptions

There is no significant assumptions made for this feature

Future Sprints:

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

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 3 cards in their loadout, 3. A player cannot start the battle if the gym still has a cooldown (this cooldown ensures users cannot farm packs or future points for the leaderboard), 4. A player cannot start the battle if they are not within a certain radius of the gym (this is currently sent 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 3 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 3 wrappers, after the player has 3 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.

Future Sprints:

In future sprints we will likely improve the battle animation “simulation” to engage players better and further incentivise recycling ideas, as right now the only user feedback is the battle log, specifying what happens on each turn. So by including animations it just makes the battles look cleaner and makes them more interactive. We will also improve the way the battles are played by:

1. Starting the battle with a “coin-flip” to give random and fairness to who goes first, as right now the mechanic is that the player always goes first.
2. 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).
3. Enhance the battle feature to ensure better mobile compatibility, a more intuitive design, and improved overall usability.
4. Introducing card “use count”, this will be a further sustainability feature embedded into the game. Each different card type will have a certain number of use counts before “wearing out” or “degrading”, in which after each use the respective card will even visually look more used/worn out, and after the max use count the card could be completed deteriorated and the user has to pack the card again to use it (we are not 100% sure if we want to make it so the player loses the card and has to repack it, or the card is just on a cooldown until it is “recycled” at a bin or just in a new feature that we will create).
5. Improving feature for substituting in a new active card after the old one has “died”. Currently if the active card dies, the backend will automatically check if there is an available card in bench slot 1, if not available, it will check if there is an available card in bench slot 2 and automatically substitute a card in the game. We aim to improve this feature by allowing the player to choose which new active card gets substituted in the playing field when an active card dies.

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.

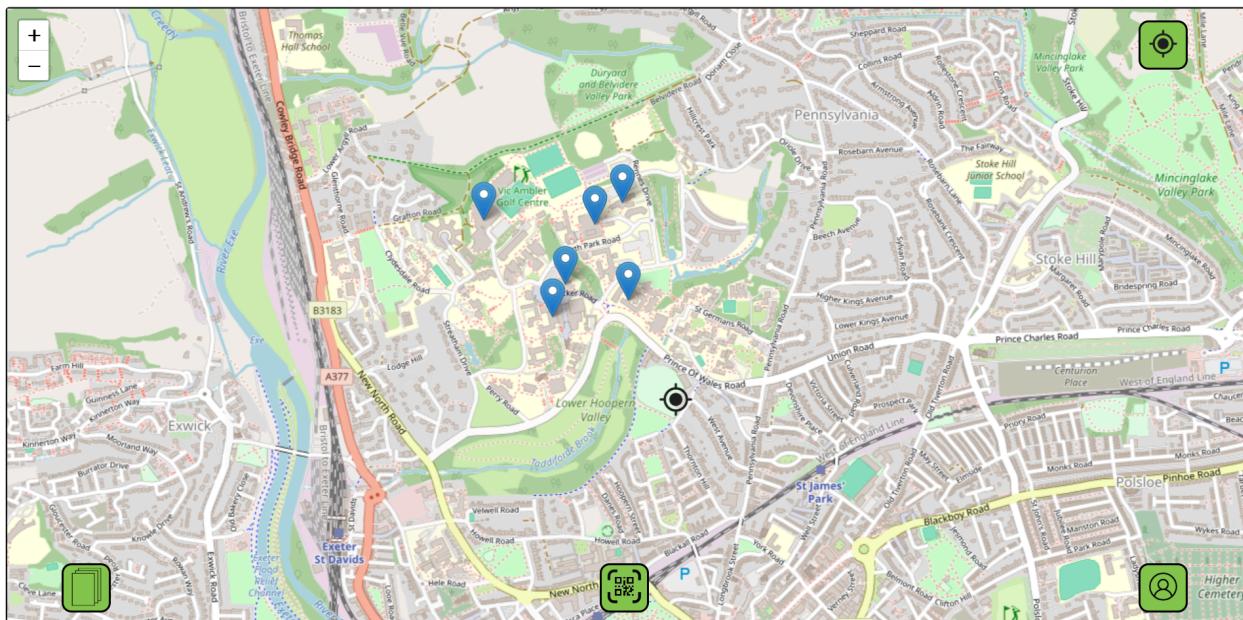
Assumptions

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 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).

Finally, 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.

Assumptions:

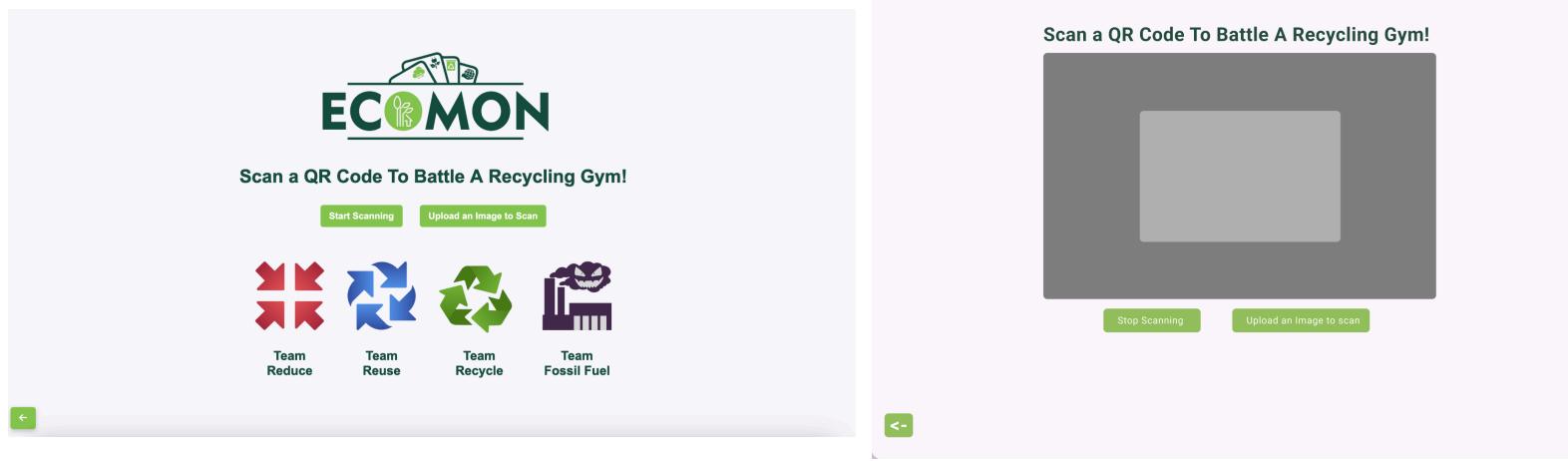
The biggest assumption made for the home page is that the user is using a device and in a location that allows for semi accurate geolocation, this is because there is no more accurate way to locate the player without using other more expensive solutions.

Future Sprints:

Later in development we plan to add custom icons for the bins (as well as adding more bins) on the map this will allow the user to more easily identify their use. Furthermore,

the map tiling will likely be changed in order to improve visibility of locations. Additionally, the user's location will automatically update every x seconds so there is less discrepancy between the map location and real player location. Finally, validate that the home page works on the phone as intended.

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/” 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.

Assumptions:

The main assumption here is that the user will 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.

Future Sprints:

Could try to ensure that if a QR code is scanned for a gym with id=1, it ensures that the user is within the radius of that gym. This can be a possibility however, there is validation for this when a user tries to start a battle on the “view-battle/1/” page anyways. We will also look into trying to improve the validation, as currently if there is no camera available, the scanner might not work if the “stop scanning” button is clicked after the “start scanning” button is clicked - currently errors are just logged to the console. Furthermore, future sprints could include making the scanner page look more appealing or have more to it in terms of design.

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

Conclusion

Our initial sprint for Ecomon has successfully ended with a stable game that lays down the key groundwork for combining gamification with real world sustainability efforts in its current state but has far more space for its impact to grow.

From a development perspective, we ensured high quality code and efficient collaboration by meeting at least 3-4 times a week as a whole group with additional meetings of subgroups when members' personal calendars didn't align. Utilising this alongside development methodologies such as Trello for the Kanban management and a structured GitHub workflow allowed us to maintain a steady development pace and efficiently distribute our core gameplay tasks.

As we move into Sprint 2, we aim to expand Ecomon with new features and refinements, ensuring the project continues to grow as a representation of Exeter's sustainability goals. By incorporating our internal feedback and testing, we will implement key improvements to enhance user engagement and overall functionality.

One of our main additions will be a team leaderboard, allowing members of team Reduce, Reuse, and Recycle to contribute independently while fostering friendly competition. This feature is designed to increase motivation and engagement, encouraging more players to take part in the sustainable actions.

Additionally, we will focus on optimising our web application for all device sizes, with a particular emphasis on mobile compatibility. Since we anticipate that mobile devices will be the primary platform for gameplay and QR scanning, ensuring smooth performance and usability is essential. Below are some of our initial wireframes and mockups, which will serve as the foundation for further development as we continuously iterate and improve.



These improvements alongside all previously stated throughout will help 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.