1.4 User profiles

When designing your campus exploration app there are three main types of users you need to consider:

1. Player users - These are the primary target audience for the app. The player users will primarily be students and staff and expect to use the app through a browser on a mobile device.

2. Game keepers - These users are responsible for creating and administrating the app and could be students or members of staff. The game keepers should be able to configure and update their instance and its resources. The game keeper users will be expected to access the app through a desktop or laptop browser.

3. Developers - Developers should be able to build, extend and redeploy the app for future and alternative uses. Developers will expect to be able to access the source code from an online repository and expect clear and concise documentation and deployment instructions.

Player users: These users interact with the website through making an account with the register page, and can interact with all of the public functions of the app (Scanning QR codes, doing quizzes, getting badges, customising profile pages, reading the info pages, seeing and appearing on leaderboards)

Game Keepers: These users interact with the website with an admin account, which has access to all the player user functions but also admin-only pages such as a page that allows them to add and remove restaurants, and a page that allows them to add and delete items.

Developers: Developers can access the code for the app as well as documentation for deploying it via the public github page.

1.5 Features

1.5.1 Sustainability Features

The app must promote sustainability goals in line with the University of Exeter's Sustainability Policy.

Users are encouraged to visit sustainable restaurants, which promotes eco-friendly habits. They can also find out information and learn about sustainability via the information page, which provides news articles, and by doing quizzes on the website.

1.5.2 Location-based engagement

Location should be a feature of the app to facilitate the campus aspect of the spec and location can be verified in multiple ways. Users will visit designated locations around campus and interact with sustainability-related challenges.

Users can scan QR codes that are located at sustainable restaurants to prove that they have indeed visited that location.

1.5.3 Gamification Features

The gamification elements should be dynamic and engaging, motivating users to interact with sustainability-related content.

Gamification elements include the leaderboard, to encourage competitiveness, the quizzes, which promotes engagement, and the personalisation options.

1.5.4 Technical Requirements

□ *Framework: Django must be used for the web application.*

□ *User Authentication: The app must support user login and manage different types of user profiles (player, game keeper, developer).*

□ *Data Privacy: The application must be GDPR compliant with secure handling of personal data.*

□ *Code Repository: The code must be hosted on GitHub, and version control must be evident throughout the development.*

This project uses Django for the framework to provide user authentication and website serving, as well as managing databases related to the project. It has a full data privacy policy and has options for users to download their data as well as delete their accounts. All of the code is stored on GitHub at https://github.com/IestynDP/ECM2434.

1.5.7 Gamification Design

The gamification design should be both persuasive and enjoyable, promoting engagement through intrinsic motivation.

The ways the project encourages engagements include the leaderboard to leverage competitiveness, the quizzes to make use of interactive education, and the personalisation options.

2.2.1 Process documents

The Kanban board should be a chart divided into at least seven columns: Backlog, Specification (doing and done), Implementation (doing and done) and Validation (doing and done). The board must clearly identify group members and the tasks they are undertaking (and have under- taken). Regular snapshots of the agile board should be taken to show the progression of tickets over the project. A time lapse of the Kanban board changes would be a useful way to capture the revision history. Records of meetings should be kept with a clear note of attendance and tasks. It should be clear which members have worked on which tickets and tasks and how those tickets link to technical contributions in the source code and technologies that are being used.

The Kanban board, hosted on Trello, can be found at https://trello.com/b/UZJKLl8G/runtime-errors-group-software. Snapshots of the board and the meeting notes can be found in the submission folder.

2.2.2 Technical documents

The source code snapshot is a link or file containing all the source code needed to build the product and instructions on deploying the artefact that clearly identifies which group members wrote which code. Professional and consistent coding conventions suitable for the chosen language should be followed. A testing strategy and the developer documentation should be included to demonstrate maturity, reliability and professionalism of the technical contributions.

All the technical documentation can be found on GitHub, including all code and tests. The github also includes instructions on how to deploy the product.

2.2.3 Product documents

The product documents are designed and published for the client to receive. These should include a public handle for the project where the product can be tested with branding and documentation that is suitable for the users.

The website is hosted on a public server, allowing clients to interact with it. It can be accessed at https://pixel87.pythonanywhere.com/accounts/login. There is also promotional media, such as a poster, contained in the submission file.