

Team 13: eWaste Recycling Portal

1. Introduction

In recent years, the rapid development of electronic products has brought unprecedented convenience to people's lives. However, due to the short lifespan of many electronic devices and the continuous introduction of new technologies, the speed of electronic product updates is accelerating, and more and more electronic waste (e-waste) is generated, such as old smartphones, laptops, tablets, game consoles, and televisions. According to Bald é et al. (2017), the global generation of electronic waste reached 44.7 million tons in 2016. People often store these discarded electronic products, however, these devices often contain harmful substances, and improper handling can have serious impacts on the environment and human health (Lundgren, 2012). Furthermore, some outdated devices still contain valuable components that can be reused and recycled, which can help reduce the environmental impact caused by e-waste.

In addition to protecting the environment, timely disposal of electronic waste has other important benefits. Firstly, timely recycling and utilization of electronic waste can help improve the resource recovery of precious metals such as gold, silver, and copper, reducing the pressure on natural resource extraction. Secondly, the value of equipment decreases over time. The earlier the equipment is recycled or resold, the higher its value, which is conducive to the full utilization and circulation of resources. Finally, handing over the device to the next user as soon as possible can help extend its lifespan and reduce waste.

Although many countries and regions have established electronic waste recycling facilities and data cleaning services, these services often do not meet the needs of everyone. Some electronic waste disposal companies transfer the problem to other countries, resulting in environmental pollution and resource waste. At the same time, many people are concerned about data security issues on their devices, so they are unwilling to hand over their old devices to the recycling bin. Some people also save their devices as collectibles because they still need data stored on them and encounter challenges when attempting to transfer data from outdated electronic devices to advanced cloud-based storage systems. These issues pose many challenges to the treatment of electronic waste.

2. Project scope and objectives

In order to alleviate the increasingly serious issue of electronic waste and meet the need for sustainable electronic waste management, our client proposed this web application project to develop a website for recycling electronic waste. This project aims to develop a web application to provide various electronic waste disposal services for device owners, including data retrieval, data wiping, local recycling and reuse. The specific work is as follows:

1. Design and develop an easy-to-use website where users can register electronic waste, such as mobile phones, laptops, tablets, etc.
2. The website can accept detailed information about devices, classify them according to their condition, provide expected value and services concerning user requirements, charging corresponding service fees if necessary.
3. Administrators can manage user and employee accounts.
4. Employees can set detailed information about the device and obtain specific reports of the user's device.

The goal is to provide individuals with a convenient and secure way to recycle electronic waste. By doing this, we ease users' concerns about data security and minimize its negative impact on the environment. We hope that this project can raise public awareness and increase participation in electronic waste treatment, thereby promoting the sustainable development of the electronic waste treatment industry.

The project should be completed before May 11th, and it needs someone to design the website. Programmers need experience in developing websites and databases.

2.1 Work content that does not belong to this project:

1. The website is not responsible for the physical operation of actual recycling and disposal of electronic waste, which will be done by a third party.
2. The website is not responsible for data wiping work, and this work is entrusted to a third party.
3. In the process of designing a website, modifications should be made to the project according to the customer's requirements, but the website should not be put into the market for operation.

2.2 Measurement criteria for project success:

1. Design a complete electronic garbage collection web application and achieve most of the required functionality in the development period.
2. Communicate and have regular meetings with the client and the advisor to understand their needs and perspectives on the website, in order to develop and optimize the web app.
3. Submit the code, video and related documents of this web application project on time.

In summary, the scope and objectives of this project revolve around the development of an electronic waste recycling website, with a focus on website design, practical functionality development, and communication with clients to meet academic requirements.

3. Team name and list of members

This project is under the responsibility of Team13. The group consists of 6 members, namely Eylul Lara Cikis, Hongyu Pan, Kaijian Xie, Samar Musthafa, Zeyi Liu, and Zihao Yu.

4. Product backlog

The Team13 software team followed the standard Software Development Life Cycle process during the development of the software, which included requirements analysis, design, coding, testing, integration, and deployment.

Firstly, a requirement analysis is required, which analyzes the obtained function list and communicates with customers to clarify the software functional requirements and determine the user group who needs the software. Next, we can transform functional requirements into user stories from different roles, and analyze their importance. These user stories describe the needs and expectations of different roles (such as administrator, staff and device owner) when using the system. By providing a detailed description, acceptance criteria and priorities for each user story, we hope to provide correct and clear guidance to the development team.

4.1: User Registration and Login

4.1.1 Detailed description:

As a user, I hope to be able to register and log in using a standard account for easy access to the system.

4.1.2 Acceptance criteria:

1. Users can register by entering an email and password.
2. After successful registration, the user can log in using the same credentials.
3. After the user logs in, the system should redirect them to the appropriate role interface (such as administrator, employee, or device owner).

Priority: High

4.2 : Account Management

4.2.1 Detailed description:

As an administrator, it is difficult for me to do all the work alone. I hope to upgrade ordinary users to staff accounts so that they can access and execute specific functions. At the same time, I hope to have a dashboard for managing user accounts so that users can be easily managed and monitored.

4.2.2 Acceptance criteria:

1. The system should provide a user list for administrators to view the relevant information of all users (except for passwords).
2. Administrators should be able to upgrade regular users to staff accounts by selecting specific users in the user list and modifying their identities.
3. After upgrading to a staff account, users should be able to assist the administrator in completing the required tasks.

Priority: Medium

4.3: Information upload

4.3.1 Detailed description:

As a user, I hope to be able to submit my electronic device information and select the corresponding services, as well as view the processing status and resale prices of relevant electronic information, so that I can track the status of my device.

4.3.2 Acceptance criteria:

1. The system should allow users to select corresponding device information from a pre-filled list of available device information, such as device type, brand, model, and memory size.
2. The system should allow users to choose the type of recycling service based on their own needs.
3. If the user's device information is not listed in the available options, the system should allow the user to manually enter their device information.
4. The system should provide users with an estimated resale price based on device information and recycling status.
5. The system should allow users to view the current status of their devices, such as whether they have been extracted, received, or processed.

6. The system should provide users with notifications and updates on the progress of their device recycling

Priority: High

4.4: Equipment handling

4.4.1 Detailed description:

As a staff member, I would like to receive detailed information about user submitted devices and determine the condition and service type of electronic devices in order to take appropriate action. I also hope to be able to create and modify detailed information about the electronic device based on the information provided by the device owner.

4.4.2 Acceptance criteria:

1. The system should allow staff to view device details submitted by users.
2. Staff should be able to manually adjust the recognition status of devices, such as 'current', 'cycle', 'rare', and 'unknown'.
3. The staff should be able to modify and delete detailed equipment information based on the equipment situation.
4. The system should allow staff to adjust the device situation to be visible or hidden from users, in order to make modifications when the device information is incomplete or incorrect.

Priority: High

4.5: Device Classification and Third Party Interaction

4.5.1 Detailed description:

As an employee, I hope the system can interact with third parties based on the category of the device, in order to provide transaction, recycling, or sales advice to the device owner. In addition, I hope the system can generate QR codes containing rewards (such as coupons) to simplify the operation of device owners.

4.5.2 Acceptance criteria:

1. The system should display appropriate third-party transaction, recycling, or sales suggestions based on the device category.
2. When the device belongs to the "current" or "rare" category, the system should display at least one third-party information, such as CeX or eBay.
3. The system should generate a QR code for the device owner that includes rewards (such as coupons).

Priority: Medium

4.6: Data Recovery and Erasure

4.6.1 Detailed description:

As the device owner, I hope to be able to choose data recovery and erasure services when I hand over the device to the electronic waste disposal system. For devices that require data recovery, I hope the data can be hosted in the cloud and sent to me through a secure link.

4.6.2 Acceptance criteria:

1. The system should provide options for data recovery and erasure services for device owners to choose from.
2. The data recovery service should be conducted after charging and the data should be hosted in the cloud.
3. The system should send notifications containing secure links to device owners via email to access managed data.
4. The secure link should be valid for 3 months, and can be accessed again by paying an additional fee within 3-6 months, after which the data will be deleted.
5. The data erasure service should be provided by a third party and ensure the data security of the device owner.

Priority: High

4.7: Payment Processing

4.7.1 Detailed description:

As the device owner, when I need to pay for data transmission fees, I hope to make the payment through PayPal and Stripe (sandbox environment only) to facilitate payment.

4.7.2 Acceptance criteria:

1. The system should provide PayPal and Stripe as payment options.
2. The payment process should be safe, smooth, and have a good user experience.
3. After successful payment, the user should receive a payment confirmation notification.
4. Payment information should be associated with user accounts for tracking and processing service requests.

Priority: Medium

4.8: Equipment Information

4.8.1 Detailed description:

As a staff member, I hope to be able to set up electronic device information for users to choose from, and I can modify and delete these electronic device information at any time.

4.8.2 Acceptance criteria:

1. The system should allow staff to add, modify, and delete electronic device information.
2. The device information should include model name, brand name, launch price, size, and memory.
3. The device information should be able to be displayed on the user page, and the user should be able to select this information when uploading their own device information.

Priority: Medium

This document describes the product backlog of the electronic waste system in detail, including the detailed description, acceptance criteria and priorities of each user story. These user stories are intended to guide the development members to realize the key functions of the system and finally meet the needs and expectations of different roles.

5. Analysis & Design

After completing the requirements analysis and user story, the next step is to develop the software architecture, select technical tools and design the page.

5.1. System architecture

This eWaste system consists of three main components in development: client, server, and database. The client is responsible for interacting with users, and the database is responsible for storing data. The server connects both the client and the database, transferring and processing data between each other. In order to meet personal skills and preferences, we choose the appropriate part among various technology stacks in system development, the results and reasons are as follows:

On the client side, we selected React, Tailwind CSS, and Material UI to implement all pages and functions. Compared with Bootstrap, Tailwind CSS can configure and implement responsive design to all types of devices at once, which better fits our needs so that we can implement the pages according to design. It emphasizes practicality and customization, allowing us to customize themes, colors, fonts, and other styles to easily meet various style and layout requirements. Material UI is an open-source UI component library based on React. Using MUI can quickly implement some components such as forms and dropdowns. We combine these three main libraries to implement the frontend.

On the server side, the team chose the Flask technology stack. Flask is a lightweight Python web framework that provides a simple and easy to understand API and minimalist design. This allows beginners to quickly understand the core concepts and principles of Flask when they are getting started. Two of the programmers responsible for the backend have no relevant web app development experience, but both have studied Python before. Therefore, Flask is a good place to get started, allowing programmers to quickly get started implementing project functionalities. Flask is suitable for building small to medium-sized web applications, as it allows developers to focus on implementing core functions without worrying about the complexity of the framework itself. Programmers have also considered trying Node.js and Spring Boot, but due to the high difficulty of learning the two, they eventually gave up.

In the communication between the client and server, the team members chose the RESTful API. RESTful follows the HTTP protocol and is easy to integrate with existing web technologies, making development and maintenance relatively simple. At the same time, RESTful APIs are language independent and can communicate with clients or servers of different programming languages and platforms, adapting to different scenarios and needs.

On the database, the developers choose MongoDB. The reason is that the team members have already learned about MongoDB in other courses. It is an extremely flexible, schema-free database that can store and access data in a very flexible and efficient way. Those features make MongoDB a popular choice for modern web and mobile applications. Choosing MongoDB saves a lot of learning time and also consolidates and practices the knowledge already mastered.

In summary, to ensure that the project can be completed efficiently within a limited time, we have chosen React, Tailwind CSS and Material UI as the client framework, Flask as the server framework, MongoDB as the database, and RESTful API as the communication method between the client and server. The selection of these takes into account learning time, development speed, usability, and the actual needs of the project, so that we can focus on implementing core functions without spending too much time building and learning.

5.2. Page and function design

With regard to website design, Hongyu Pan used Figma to design the prototype and UI of the webpage. Figma provides components and automatic layout functions for managers to design elements, improving design efficiency. Based on requirements provided, the following use case diagram is made:

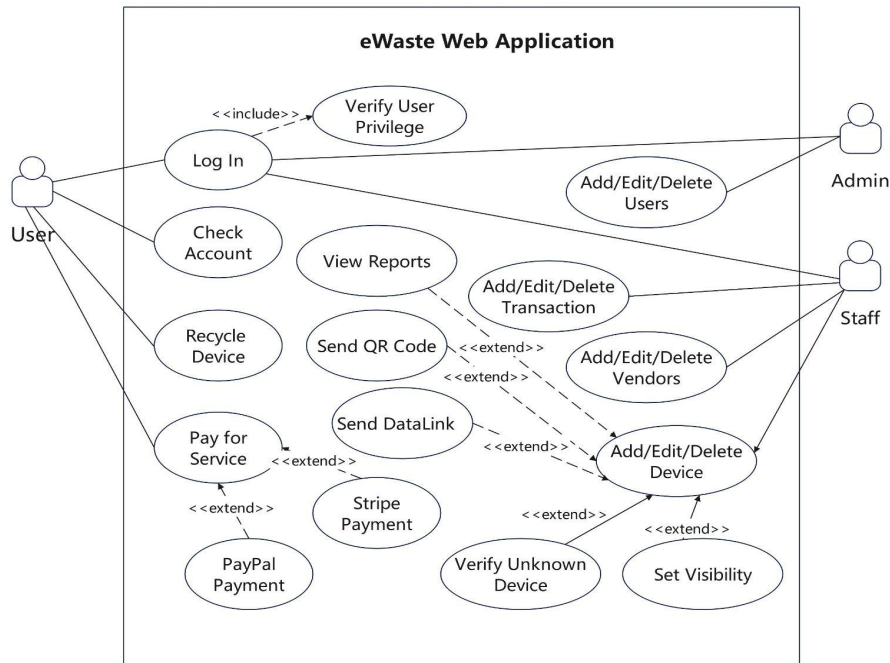


Fig 5.1 Use Case Diagram

Combined with the use case diagram, the first version of the page has been designed, which includes the login page, the recycling page, the user account, the staff dashboard, and so on. Since some components always show whenever the user interacts with the application, we create some fragments to store these things. Some pages and fragments are as follows:

5.2.1. Components

1. We create a login modal and embed it in the header so that the user can click the login button in the header no matter which page they are on. If the user is not logged in, they will not be able to jump to some pages such as the account page, or check their recycled device.
2. Notification is also a component, when user checks their notification box, they can see the new status of a device, and so on. There is also a button “Read All” to empty their mailbox.
3. We also integrated the form component of the staff framework so that we can reuse components for the staff and the admin dashboard.

5.2.2. Main Page

For the landing page, users can see various types of electronic waste accepted by the project specifically include the following content:

1. Photos and names of different types of devices, including four categories: phone, tablet, laptop, and others;
2. The brand and model of each device type, such as phone brands including Apple, Honor, Samsung, Sony, Oppo, etc. Among Apple-branded phones, there are different models of iPhone products.

3. Most popular devices and recent devices, showing the advantages using our website and user comments. There is a user recycle form on the left, which leads to the recycling page.
4. At the top of the page, there are "Login" and "Notification" buttons on the right.

5.2.3. Registration and User Pages:

1. From the login modal, users can jump to the registration page by clicking the register button, there is a form when creating a new user that requires filling in First Name, Last Name, password, confirmation password, and email address.
2. After registration, the user will automatically log in, and they can check their account to change their information.
3. Users can also see and search their recycling record and information. If staff send them a link for service or a QR Code, users can also see these links.

5.2.4. Recycling Page

The recycling page provides users with a way to submit device information, which can be jumped from the form in homepage. Some features are as follows:

1. When user choose a brand, corresponding options will appear in the model name, others are the same.
2. User can input brand, color and other basic information of device. They can also choose the service they want.
3. When they input necessary information, the expected value will calculate automatically in this page according to their device condition and quotes from different vendors. After they submit their uploading, the device will show in staff dashboard.
4. If users don't find options they want, they can upload their own device and it needs staff to verify.

5.2.5. Payment Page

Paypal and Stripe are available to users. When they choose to pay for their service, the system will detect what the service is and lead them to the payment page. If the user has not paid for the first retrieval yet but they want a longer one, they could pay for them together through one order.

5.2.6. Staff Dashboard and Admin Dashboard

1. For staff, they can view, edit, add, and delete devices, vendors, and transactions.
2. The staff can set the visibility of the device and verify unknown devices. When they set a device invisible, the user it belongs to will not see this device. They can also move a device from one type to another type and generate links or QR Codes for users.
3. For these several forms, the framework from Material UI can be used to configure them, so that we can use one structure to fit all different tables.
4. When some operations are done, notifications will be sent to users to remind them the updated status of the device.
5. Administrators manage all user accounts, they can add or modify users' information and permissions, such as upgrading users to staff.

For the above design, the client provided us quite useful suggestions during the iteration help us improve our system, as follows:

1. The original design of homepage was simple and cannot fully show the purpose of our application, with reminder from the client, we added “Recent Devices” and other components. In order to highlight the main functions of the website, we added a form on the homepage that allows users to directly enter information related to the brand, model, memory size, and color of the device they are searching for. Clicking the button will take them to the device's recycling page, allowing users to enter their device information on the homepage and obtain a rough quote.
2. The customer suggests separating the responsibilities of the administrator from those of the staff. The administrator is responsible for managing user authority, while the staff is responsible for other things besides people. A number of additional designs have also been added to the homepage to make it attractive.
3. Partial functionality missing: Add a field for filling in phone numbers during registration. When users upload information about recycling devices, the option of adding device conditions (almost new) is added to determine the recycling price. Due to the user uploading device information that has not been recorded in the database, a new interface has been added for users to upload new device data, which will be reviewed by the staff.
4. For the staff interface, it seems that the user pages managed by the staff are similar to the device pages and lack differentiation in design. we make changes to the button colors of different pages, making it easy for administrators to see the relevant content of the page at a glance.

The modified page has been recognized by clients and has been completed during the iteration.

5.3. Database design

Our MongoDB database contains three different collections: 'Device', 'Users', and 'Vendors'. We designed our database according to client's requirements and our own UI design, and also aimed to ensure that it is well organized and efficient in storing and retrieving different types of data related to user accounts, vendors and electronic devices. Although MongoDB is not a traditional relational database, there are still relations between our collections. We are storing user id as a field in a device document since user accounts are created beforehand and each user could upload several devices, we use this field to link devices to a specific user. Each device document also has vendor id as a field to get prices from vendors and we used that to get users a quote for recycling. Following are details of the three collections:

1. The Users collection contains all necessary information related to an user's account as following: *_id* (user ID, automatically generated by uuid4), email (user email), password (user password, using PBKDF2 algorithm and hash encryption through SHA-256), *first_name* (user's first name), *last_name* (user's last name), privilege (user's privilege level, divided into staff, admin, and user), *ts* (timestamp, used to record document created time), *ts_mod* (timestamp modified, used to record the last time when the document was modified), *is_deleted*: (used to identify whether the user has been soft deleted. When the user is deleted, its *is_deleted* field will be set to true, rather than deleting the record from the database. This approach preserves the history of the user and can restore the deleted user when needed.)
2. The Vendors collection contains specification and price information related to a electronic device from online vendors, including the following fields: *_id* (vendor ID, automatically generated by uuid), brand (electronic device brand), *model_name* (electronic device model), size (electronic device size), storage (electronic device memory), *sales_price* (market price of the electronic device),

ts (time stamp, used to record the creation time of the document), ts_mod (timestamp modified, used to record the time when the document was last modified), is_deleted: (Used to identify whether the vendor has been soft deleted.)

3. The Devices collection contains electronic device information uploaded by users and information related to following services and payments, including the following fields: *_id* (automatically generated by uuid), *user_id* (derived from the *_id* of the users collection), *vendor_id* (derived from the vendor set's *_id*), brand (electronic device brand), model (electronic device model), identification (classification of electronic devices and service types, including 'current', 'recycle', 'raw', or 'unknown'), status (current service status of electronic devices), operation_system (operating system used on electronic devices) memory_storage (memory size of electronic devices), color (color of electronic devices), type (type of electronic device), description (brief description of the customer's requirements for electronic devices and services), service (service type selected by the customer), datalink (datalink sent by the system to the electronic device owner to retrieve date), qr_code (QR code sent by the system to the electronic device owner for referral), device_ts (timestamp, used to record the creation time of uploaded information documents), device_ts_mod (timestamp modified, used to record the last modification time of the uploaded information document), payment_id (payment information ID), payment_amount (quantity of payment amount), payment_ts (timestamp), payment_ts_mod (timestamp modified), is_deleted: (check if the vendor has been soft-deleted.), verified (Equipment verification status), is_hidden (Whether device information is hidden from users)

5.4 Security Features

We have added certain security features to make accessing sensitive information in our database more secure.

- Password hashing and salting: Each time a user is registered their password goes through rounds of hashing and salting, only after this it is stored in the database so even if someone were to gain access to it no user password could be leaked.
- User privileges: There are 3 available user privileges, these are; “user”, “staff” and “admin”. Many of our api's are restricted by using these privileges wherever possible. For example: Only a staff or an admin account can access the user list, only a staff can modify a device or only an admin account can elevate a user into being a staff or admin account.
- API authorisation: Certain API's require session tokens to be present to be available. Not only that they require those session tokens to be from a privileged user. Such as getuserbyid will not return any user's information when called by a user and will only work when called by a staff or an admin.
- API information access culling: We have closed off certain information from being accessible from API's at all. For example no matter how high of a privilege someone has calling the API, no API can return user passwords or payment details.
- Input sanitisation: Though strictly not necessary since we are using MongoDB we have still sanitised every input and they are all bound to variables or objects before and call to the database is done.
- Privilege verification: Every front-end portal performs strict user-role verification to ensure unauthorized access is prevented.
- API access safe-guarding: None of the required pages can be interacted with from the front-end unless the user is logged in
- Input validation: All front-end device forms enforce valid values in the inputs to be able to submit the details

6. Evidence of Testing

Software testing is an important part of the software development process. Through testing, developers can identify errors in design and code, and correct them until the test passes. This can enable potential problems in the software to be discovered as soon as possible, improve the quality and stability of the software, avoid high costs caused by extensive modifications due to errors, and improve the user experience.

6.1 Frontend Manual Testing

Due to the nature of the project and the usage of an agile model, the majority of testing done in the front-end was Manual UI and Workflow testing. This involved the front-end team physically testing out the specific functionalities in their local systems under specific edge-cases capable of making or breaking the system. We will take a look at the kinds of testing carried out in the front-end. Front-end also has an automated testing framework integrated into the project which has not been extensively developed. The following Manual tests were performed:

1. Test Case 1: Correct Registration - Verify whether the user can create a new account
2. Test Case 2: Partial information is empty during registration - Verify whether the user can successfully respond to the missing information during registration
3. Test case 3: Inconsistent password and confirmation password during registration - Verify whether users can successfully respond by entering different passwords twice during registration
4. Test case 4: Use registered email when registering - Verify whether users can successfully respond using registered emails during registration
5. Test Case 5: Use the wrong type of email during registration - Verify if the user can successfully respond with the wrong type of email (excluding @) during registration
6. Test Case 6: User Login - Users can log in to the website
7. Test Case 7: Invalid Email or Password - Verify whether the system can respond correctly when a user enters an invalid email or password
8. Test Case 8: User Logout - Users can log out from the website
9. Test Case 9: Go to the device upload page - Click on the "Get Quote!" button on the user's homepage to enter the device upload page
10. Test Case 10: Return to Homepage - Click on the circle containing the less than sign in the device upload page, a prompt will appear, confirm, and then return to the homepage
11. Test Case 11: Upload device information correctly - Verify whether the user can successfully upload device information after selecting or filling in all information
12. Test Case 12: Partial information is empty when uploading device information - Verify whether users can upload device information without missing some information
13. Test Case 13: View the user's device list - Verify whether the device status can be viewed in the device list after the user uploads device information
14. Test Case 14: Search Device - Verify whether the search box in the Device List can search for device information by device name
15. Test Case 15: Modifying Account Information - Verify if account information can be modified
16. Test Case 16: Inconsistent password and confirmation password during modifying account information - Verify whether users can successfully respond by entering different passwords twice during modifying account information
17. Test Case 17: Delete user account - Verify whether the user can still log in after deleting the user account by admin

18. Test Case 18:Hide device-Verify whether users can see device information when staff hides the device
19. Test Case 19:Send QR code - Verify whether the user can receive the QR code sent by staff
20. Test Case 20:Payment - Verify whether users can make PayPal and Stripe payments

6.2 Backend Unit and Integration Testing

While early in the development most of our backend testing was done using postman and individually writing GET and POST requests to check if our database integration and our calls were working properly because we did not have the frontend and backend systems connected, this was necessary for early development and incredibly helpful.

After using that system for a while, we have decided to write both unit and integration tests using pytest. For unit tests we are testing each api endpoint and their functionality individually. For example what happens when the login api is called, what would the return be if the email was incorrect, what would the return be if a password is incorrect. We are checking each output with assertions.

Integration testing the backend api is a bit more involved because it required us to call certain api's in certain orders and is better shown in an example steps:

1. Call registration api to create a new user and log in
2. Call postdevice api to create a new device for this specific user
3. Call updatedevice api with staff privileges to verify that specific device
4. Call addpayment api to add a new payment to this specific device

We assert if the output of each of these are correct by checking the database in each step automatically by our written tests. This procedure helped us a lot in working better as a team since someone working specifically in the front-end can look at our test cases and see that if they call those functions in that order the result will be expected.

Additionally a few times during development this helped us catch bugs even without thinking about it when we have changed something but broke something else unrelated. The best example of this was when we were adding authentication and security features, a lot of our back end test cases broke which allowed us to see where we might have fixed certain things while adding a new feature.

For example when we have added user privileges such as staff and admin and closed off certain responses without authorisation certain parts of our webpage broke but we didn't even have to go through and check since our test cases when ran alerted us that certain tests have failed so we could fix even before merging to our main branch.

7. Team management & communication

7.1. Team management:

Our team consists of 6 members, Their specific responsibilities are as follows:

- As the project leader, Eylul Lara Cikis is responsible for overall project management, assigning and accepting tasks for each stage to ensure the smooth progress of the project. Due to her language advantage, she is responsible for communicating with clients and providing periodic reports to ensure requirements are clear. She has been implementing and modifying all kinds of apis, writing

tests, writing scripts to build the database, fixing bugs, finding and cleansing data sources and loading them into our database.

- Hongyu Pan first worked on designing website prototypes to achieve a user-friendly interface and experience while making modifications to the page design based on customers' feedback. She implemented some pages for users in the frontend and made them responsive to mobile phones. She also created the Payment Sandbox, Notification System, and other components.
- Samar Musthafa coordinated the front-end team by creating a requirement list, assigning issues on Gitlab, creating appropriate labels and tasks and formatting the report. He created the Staff and Admin dashboard as well as the landing page, device pages along with several other pages and reusable components like the dashboard, Listing, etc. He also incorporated a unit-test system into the project for some of the appropriate components and Toast notification service, Layouts for different user types and Router Structure.
- Kaijian Xie implemented the login system and the header. He also worked on making the website responsive to mobile phones and implementing the individual edit/add pages of each of the dashboard components. He solved the problem of cross-origin cookies by Nginx, and wrote the correct Nginx configuration. Except for this, he helped others fix bugs and optimized code in frontend to keep consistency.
- Zeyi Liu worked as a backend developer. He worked on implementing user, device, transaction, and vendor apis, testing them and fixing problems in the backend. He adjusted existing apis according to the changes on the frontend. Besides, his work involves finding datasource, writing api documents, and commenting on backend code.
- Zihao Yu is dedicated to backend development, responsible for writing, testing, and modifying some apis to meet front-end requirements. In addition, his work includes taking meeting minutes, writing document apis and conducting requirements analysis. He was also responsible for writing documents. He searched for literature, collected background information and relevant data on electronic device recycling, collected team members' opinions and gains on the project, and completed the document work. At the same time, after exchanging opinions with team members, polish and modify the document.

7.2. Team communication:

Team members mainly hold daily meetings in the Diamond to discuss progress and the next steps. The team chat tool is Discord, which allows members to communicate in real-time without missing important information. If the discussion topic is not completed in the class, we will arrange relevant meetings after class, usually in Diamond. Zeyi Liu and Zihao Yu, who are responsible for the backend, use WeChat to communicate directly. They work together on coding and testing in Diamond during their free time. Eylul Lara Cakis, Hongyu Pan, Kaijian Xie, and Samar Musthafa have a dedicated channel on Discord where they communicate the progress of the project. If they encountered difficult problems, they discussed them in person.

In the team communication, we encountered two challenges. Firstly, everyone has different modules and schedules, and not everyone has time to participate in meetings. To address this issue, we scheduled important meetings at a time when all members could attend, such as weekends. We shared minutes and arrangements on discord as a backup and notice board for people who had not attended the meetings. However, the ability to use English was another problem for some team members during meetings. In most cases, they relied on their own attempts to solve the problem, and sometimes relied on others' translations.

7.3. Meeting minutes:

7.3.1 March 4th, 2023

Location: The Diamond

Participants: Eylul Lara Cikis, Hongyu Pan, Kaijian Xie, Samar Musthafa, Zeyi Liu, and Zihao Yu

Agenda: First round iteration plan and task allocation

Discussion points:

- Hongyu Pan shared the preliminary interface design plan and relevant opinions exchanged with customers for discussion.
- Samar Musthafa, Eylul Lara Cikis, and Kaijian Xie discussed the modification plan for the page and told Hongyu Pan how to execute it.
- Samar Musthafa listed a table containing the required APIs for the front-end and arranged it for Zeyi Liu, and Zihao Yu.
- Zeyi Liu and Zihao Yu discuss how to allocate tasks for API settings.
- Eylul Lara Cikis stated that mongodb will be set up as a new branch of the job.
- Eylul Lara Cikis and Samar Musthafa discussed whether to use cookie sessions to determine if the user is already logged in, but no results were discussed.
- Zeyi Liu asked what can be done for database settings if mongodb does not need to define a schema in advance.
- Eylul Lara Cikis requested him to write a bash script that can drop and set up the DB without having to go through everything manually, and provided a related example.
- Zeyi Liu and Zihao Yu asked others about the Mongodb database settings due to their lack of understanding, and Lara provided guidance and recommended corresponding learning links.

Decision and Action Plan:

- Determine the task allocation and deadline for the first iteration.
- Hongyu Pan will revise the design of the page and present it to customers on March 9th.
- Zeyi Liu and Zihao Yu wrote the corresponding API according to the requirements.
- Samar Musthafa continues to write README and document APIs.
- Eylul Lara Cikis, Hongyu Pan, Kaijian Xie, and Samar Musthafa began writing client code.

7.3.2 March 22, 2023

Location: The Diamond

Participants: Zeyi Liu, Zihao Yu

Agenda: Discussion on API Testing

Discussion points:

- The two discussed how to conduct effective API testing, and Zeyi Liu believed that Postman should be used to test the API. He explained the general process to Zihao Yu and demonstrated it.
- The two of them tested some APIs and found that there was a problem with the API used to obtain specific information. When the acquisition failed, there would be a "response: error" message, while when the acquisition succeeded, only the queried information would be serialized into a JSON string and returned, with a prompt of "response: success" added.
- The two of them assigned each person the project for testing. If Zihao Yu encounters any difficulties during the testing, they can discuss with Zeyi Liu or entrust it to Zeyi Liu for resolution.

7.3.3 March 28, 2023

Location: Classroom for Team Software Project Classroom

Participants: Eylul Lara Cikis, Hongyu Pan, Kaijian Xie, Samar Musthafa, Zeyi Liu, and Zihao Yu

Agenda: Second round iteration review and third round iteration plan

Discussion points:

- The server API settings have been basically completed and the testing phase has entered.
- Team members shared the results, issues, and lessons learned from the second round of iteration.
- Lara summarized her work over the past week: in the previous iteration plan, the backend API settings were basically completed and some testing was conducted. Devices and datalinks also have database capabilities, but do not yet connect to endpoints.
- Hongyu Pan summarized the unresolved issues for the second round of iteration: some APIs did not respond properly in Postman, possibly due to configuration issues.
- After observing, Samar Musthafa believed that all error requests were related to post requests and decided to fix two of the 500 errors.
- Zeyi Liu and Zihao Yu discuss the upcoming API testing plan.
- Eylul Lara Cikis, Hongyu Pan, Kaijian Xie, and Samar Mustafa discussed and decided to focus on The Diamond to complete the next tasks during Easter.
- The task for the next iteration of Eylul Lara Cikis is to complete and test all APIs so that all responsive pages can work. Find a suitable database containing various mobile phone information and upload the information. Continue to improve the user interface as expected, including functions such as payment, authentication, pricing algorithms, data link/qr codes.

Decision and Action Plan:

- Summarize the feedback from the second iteration and make corresponding adjustments for the third iteration.
- Task allocation was carried out for the third iteration, but due to the different arrangements for each person during the Easter holiday, no corresponding deadline was established.
- Eylul Lara Cikis, Hongyu Pan, Kaijian Xie, Samar Musthafa continue to develop and optimize user interface functionality.
- Zeyi Liu and Zihao Yu test all APIs and promptly report any abnormal situations.
- Team members use their vacation time to review and consolidate relevant knowledge, preparing for the upcoming work.

8. Planned & Completed Features

The project was developed by setting objectives from the feature list provided in the briefing material. Though this gave us a solid objective list to pursue initially, there were several design elements and features incorporated into the project following client meetings. Since we used an agile development model, it was rather easy to expand the feature set on the fly while keeping the project structure intact. The following Feature list is implemented in the Project:

1. Standard Account Registration and Login: The project features a completed User login and management system. This ensures that a user cannot access pages that they normally should not unless logged in. Additionally, since the project was dealing with three different types of dashboards, a user-access verifier is incorporated into the layout.
2. Admin system to upgrade/downgrade users to staff: The project features an admin-dashboard page that deals exclusively with user account manipulation. This can only be accessed by an admin account and they can edit individual user account properties.
3. Accept device details and identify the device type: This is done by using the device details page where the user can select from a massive dataset of well known devices. Additionally, if the user device is not a part of the provided list, they can enter information manually. Upon automatic

selection, the backend device identification algorithm tags the device with an appropriate identifier, whereas in the latter case, the device will remain unknown until a staff member manually verifies the device.

4. Current devices' price will be down in the portal. We have obtained the device cost from CeX and Argos using the dataset. The price will be balanced by weighing in the device condition along with this data.
5. Devices submitted for recycling can choose to get their data retrieved. Once this data is obtained, the staff member will upload it to a cloud storage of their choice and provide the URL from the dashboard. The user will then be notified of this change.
6. Rare devices are handled similar to current devices.
7. If a device is unknown, it is up-to the staff to update its identification and verify it.
8. Appropriate role assignment and management system: Admins and Staff members have their own respective dashboards and do not interfere with the user dashboards and vice versa.
9. Staff can edit or add new devices, transaction details as well as vendor list to increase the dataset. Additionally, staff can one-click verify devices without having to open the editor.
10. Devices can be hidden by the staff members by selecting the Hide Device option.
11. Device classification can be edited easily by the staff members. The portal supports three device types; viz, phones, laptops and tablets.
12. Payments have been integrated into the system Using Paypal and Stripe. This is done in Sandbox mode as of now.
13. Staff reports are provided in the dashboard. Along with other options like generating a QR code, sending it to the user and thus obtaining Coupon codes.
14. The landing page shows some of the recent devices manufactured and popular choices. Additionally, it also breaks down the entire recycling process.
15. Proper Notification system that notifies the user of any changes to device state and details
16. User dashboard displaying the devices the user has already manufactured.
17. User account details page for the user to modify details of their account.
18. Mobile-first web application so that it works easily on mobile devices.

9. Uncompleted Features

Due to some unforeseen circumstances, some features were not completed and thus not incorporated into the project. The following features are yet to be completed.

1. 3rd party login services like Google, Microsoft had to be abandoned due to lack of development time.
2. Though the device page shows the resale price of the device, it does not redirect to the respective pages. We could not figure out how to directly connect with the services.
3. Draft saving was not implemented due to lack of development time.
4. There are no automated test suites in the front-end and the individual tests themselves were performed manually. However, there is a testing framework integrated into the system for adding test-cases in the future.

10. Screenshots of relevant pages

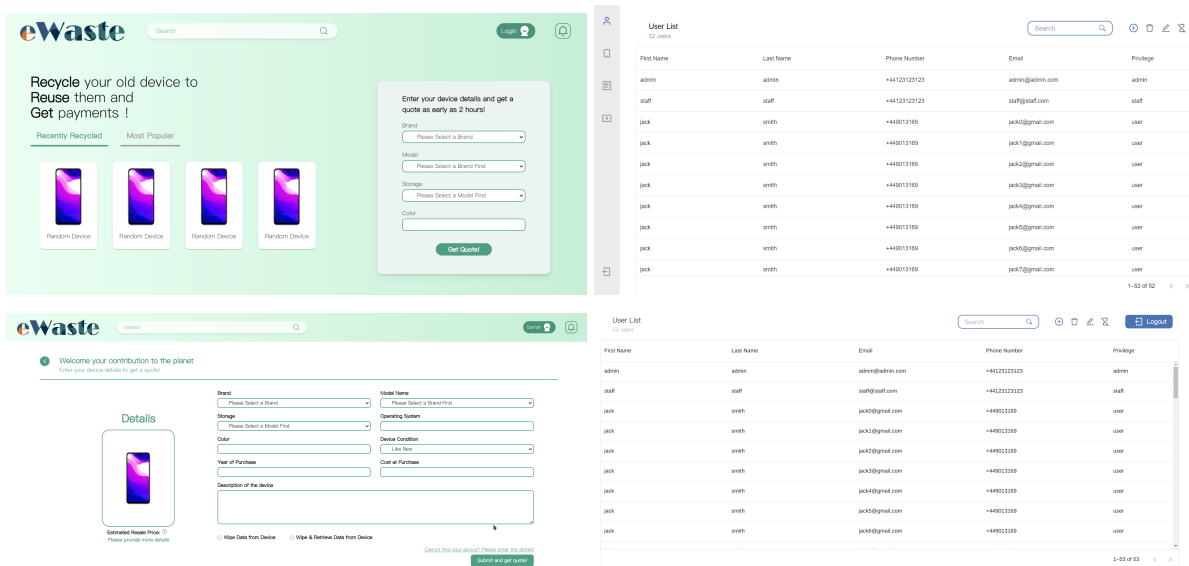


Fig 10.1 Landing Page, Staff Dashboard, New Device Page, Admin Dashboard

11. Conclusion

In this project, we have gained a lot of valuable experiences and insights. Some members have extensive programming experience, but for others, this is their first time participating in a team project. The experience of this team project has taught all of us a lot of teamwork experience and improved programming skills as well.

Firstly, we learned how to collaborate as a team. Each person in the team has different programming abilities and different strengths and weaknesses. Through the collaboration, we have learned how to divide and allocate tasks to maximize each member's contribution. At the same time, using English as the language of communication has improved their English proficiency for people who are not native English speakers.

Secondly, team members have been taught how to conduct requirement analysis and system design in practice. In the past learning process, we usually received a complete specification manual, which includes detailed guidance and requirements. However, in this project, the client had only provided basic requirements, and we had to investigate some potential requirements from projectwise. This process not only involved understanding and analyzing requirements, but also involved selecting appropriate programming languages, code structures, and system architectures. This places higher demands on our technical and problem-solving abilities.

Finally, in our past programming experience, we typically approach writing efficient and optimized code from a programmer's perspective. However, the experience of this project made us realize that programming not only needs to consider technology, but also needs to think from the perspective of users and products. This requires us not only to have a solid technical foundation, but also to have a deep understanding of user experience and human-computer interaction.

Finally, we also learned how to use GitLab. GitLab is of great help to our team's work. We learned how to create branches, commit changes, resolve merge conflicts, and conduct code review. This tool enables effective team collaboration on a unified platform, while also ensuring code quality and consistency.

However, we also encountered some challenges in this project. Firstly, due to poor communication, team members may not fully understand each other's intentions, resulting in repetitive work and some members not being able to communicate well in their work. This made our progress very slow in the early stages of the project. Time management is also the main problem we face. Due to the unclear division of tasks in each stage of the project, we experienced a phenomenon of rushing to work before the task ended.

Overall, although we encountered challenges during the project process, we ultimately successfully completed the entire project. I believe this experience will have a positive impact on our future learning and work

12. References

- [Baldé, C. P., Forti, V., Gray, V., Kuehr, R., & Stegmann, P. \(2017\). The Global E-waste Monitor 2017: Quantities, Flows, and Resources. United Nations University \(UNU\), International Telecommunication Union \(ITU\), and International Solid Waste Association \(ISWA\).](#)
- [Lundgren, K. \(2012\). The global impact of e-waste: Addressing the challenge. International Labour Organization](#)
- [React Code Documentation Guidelines](#)
- [React Testing Library](#)

13. Appendix

13.1 User Guide:

13.1.1 Account registration and login logout:

1. On the homepage, click the log-in button and then enter email and password if you already have one or click register to create a new account.
2. If you are creating a new account, provide required information such as name, email, password, phone number. Make sure your email is in valid format and your password is strong.
3. Once you successfully create your account, you will be automatically logged in.
4. Your user specific account pages will be available when you are logged in.
5. Click the user count button in the upper right corner and click My Account, you will be directed to your account information page where you can modify your personal information and account details. Click the logout button to log out.

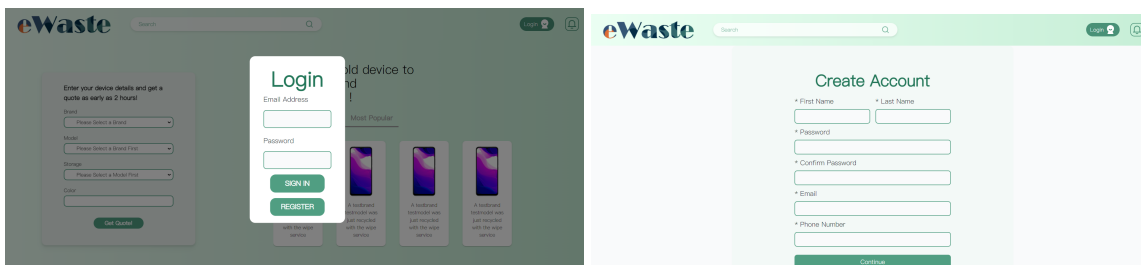


Fig. 13.1 Account Registration and Login pages

13.1.2 Recycling devices and get quotes:

1. On the home page, select your device's brand, model name and color from the provided list, then click on the Get Quote button, it will direct you to another page to provide more detailed information for the device so we can calculate a valid price for it. If your device spec is not listed, please enter its details manually. Choose your service from "wipe data from device" or "wipe & retrieve data from device". Click the 'submit and get quote' button and you are all set. Later you can find the status of the recycling process of all your devices in the "my recycle" page.

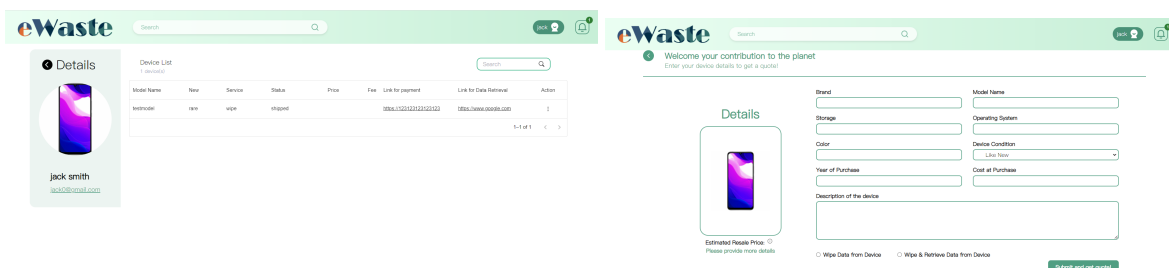


Fig 13.2 User Account Dashboard and Recycling lists.

13.1.3 Get your QR code for bonus:

1. If you have submitted a device, you may get a QR code for bonus.
2. You can find your QR code for a specific device of yours on 'my recycle' page. You will be notified when the QR code is ready.

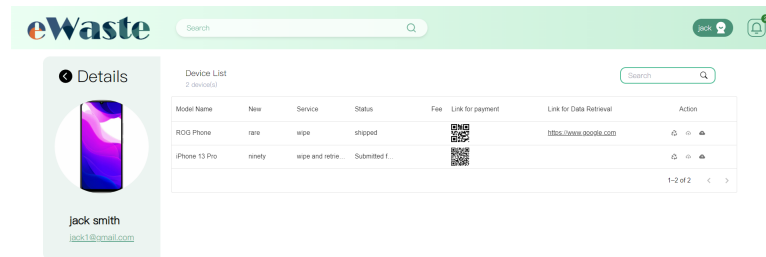


Fig 13.3 User Dashboard

13.1.4 Make payments for data retrieval and extend the retrieval deadline:

1. On 'my recycle page', you can make payments for your recycling devices.
2. Click the retrieval button to pay for your data retrieval service. You can choose either paypal or stripe to pay. After a successful payment, you can see the amount you have paid for that specific device on the same page.
3. Click the 'extend' button to pay for an extended data retrieval service. Follow the same payment process and you will find your service becomes "further retrieval" on the same page.

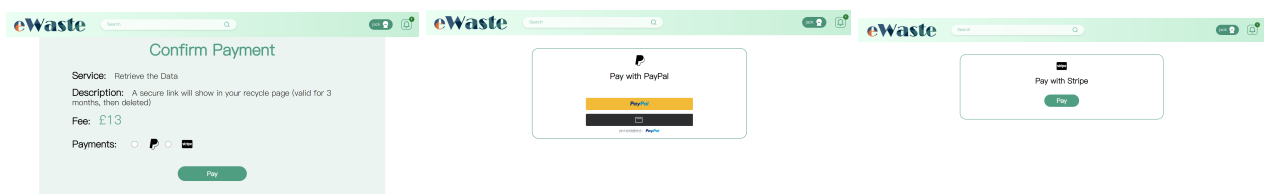


Fig 13.4 Payment pages (Sandboxed)

13.1.5 Data retrieval from recycle device:

1. If you have submitted a device for recycling service, you can opt to have your device's data retrieved.
2. A staff member will upload a url datalink for your data retrieval once it's ready. You can find the link for a specific device of yours on 'my recycle' page. You will be notified when the data retrieval is ready.

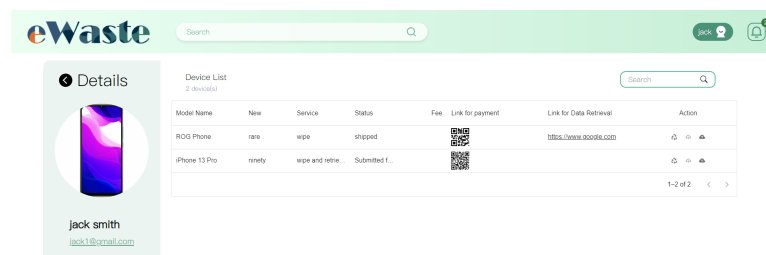


Fig 13.5 Data Retrieval Page

13.1.6 Check out your notifications:

1. You will receive notifications about changes to device status, qr code and datalink. Click on the notification button in the upper right corner to see all your notifications.

eWaste

Search

<

Welcome your contribution to the planet
Enter your device details to get a quote!

Last Successful Login
Last successful login at 2023-05-11 11:26:11 59029

Read All

Details

Estimated Resale Price: ⓘ
Please provide more details

Brand

Storage

Color

Year of Purchase

Description of the device

Model Name

Operating System

Device Condition

Cost at Purchase

☐ Wipe Data from Device

☐ Wipe & Retrieve Data from Device

Submit and get quote!

13.1.7 Staff Dashboard Actions

Device List

51 devices

Model Name	Brand Name	Color	Data Link	Description	Identif.	Memory	Operating	Service	Status	Type	Verified	Hidden
Y15	Vivo	red	www.google.com	this is a phone	raw	64	android	wipe	shipped	phone	<div></div>	false
Moto X (Gen 2)	Motorola	red	www.google.com	this is a phone	raw	32	android	wipe	shipped	phone	<div></div>	false
Pixel V2	Levix	red	www.google.com	this is a phone	raw	16	android	wipe	shipped	phone	<div></div>	false
Xperia Z2	Sony	www.google.com	this is a phone	raw	32	android	wipe	shipped	phone	<div></div>	false	
A7	Alcatel	red	www.google.com	this is a phone	raw	64	android	wipe	shipped	phone	<div></div>	false
Aqua Belle	Hibex	red	www.google.com	this is a phone	raw	32	android	wipe	shipped	phone	<div></div>	false
Redmi 2 Prime	Xiaomi	red	www.google.com	this is a phone	raw	16	android	wipe	shipped	phone	<div></div>	false
ZenFone 2 Deluxe	Asus	red	www.google.com	this is a phone	raw	64	android	wipe	shipped	phone	<div></div>	false
A2	Asus	red	www.google.com	this is a phone	raw	64	android	wipe	shipped	phone	<div></div>	false
Phantom 8	Techno	red	www.google.com	this is a phone	raw	128	android	wipe	shipped	phone	<div></div>	false
Marathon M5 Lite	Gionee	red	www.google.com	this is a phone	raw	32	android	wipe	shipped	phone	<div></div>	false

1 - 51 of 51

< >

Details

Y15

Vivo

Expected Value:

E UNKNOWN

(this is a phone)

* Brand

Vivo

* Model Name

Y15

* Identification(New)

Raw

* Operating System

Android

* Memory Storage

64GB

* Color

red

* Type

Phone

* Expected Value

E UNKNOWN

* Status

Shipped

* Verified

Yes

* Description (No more than 500 words)

this is a phone

Wipe Data from Device

Wipe & Pairdown Data from Device

13.1.8 Admin Dashboard Actions

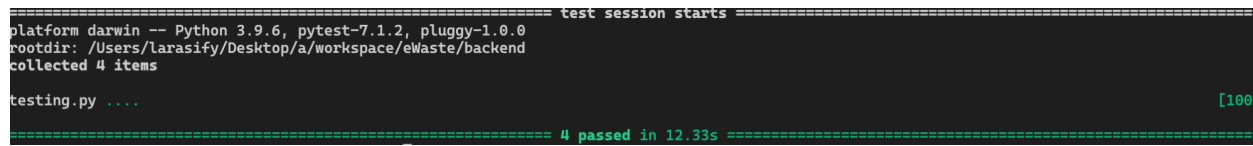
[illegible]

13.2 Setup Guide:

Developed using python3.9 and mongoDB

1. cd into backend folder

2. Create your python environment and install the required libraries using “pip install -r requirements.txt”
3. Run “pytest testing.py” to run test cases



```

===== test session starts =====
platform darwin -- Python 3.9.6, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/larasify/Desktop/a/workspace/eWaste/backend
collected 4 items

testing.py ....

===== 4 passed in 12.33s =====

```

Fig 13.9 Backend testing

4. If you wish to run with live data comment out line 56-57 in app.py (which are buildvendordatasource() and rebuildddb() functions which will rebuild the database to default configuration)
5. Run “flask run” to start the backend server

MongoDB Setup:

1. Project is setup to use mongodb run on mongodb://localhost:27017/
2. Installation and setup steps can be found for
 - a. Windows: <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/>
 - b. MacOS: <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-os-x/>

13.2.2 Frontend Setup

Developed using NodeJS 16, ReactJS and other components. The system uses; Mui, React-hot-toast, etc as its UI library. Additionally, it uses Jest and React-Testing-Library as its test generation library

A complete list of required modules are provided in the *package.json* file. This file deals with the setting up and execution of the front-end server. You can see that the *nginx* rules are configured to reroute front-end requests from port 8080 to 3000, this is because the front-end server occupies port 3000. To run the server, perform the following:

1. ``cd`` into *frontend/* folder
2. Run ``npm i`` to setup all the node_modules. Once this step has completed, all required packages will be fetched and installed.
3. Run ``npm start`` to execute the server. On successful compilation, the last log will indicate *webpack* having compiled; with or without *warnings*. Keep in mind that, even though the server thus launched is fired at *localhost:3000* as can be seen from the browser launched by the execution, the actual webserver will be at *localhost:8000*. Manually visiting *localhost:8080* will launch the actual web application
4. To test the system, run ``npm test``. Since it uses *react-testing-library*, the spec structure requires the developer to define a **.test.js* file for every corresponding **.js* file that is required to be tested.

13.2.3 Nginx

1. <https://www.nginx.com/resources/wiki/start/topics/tutorials/install/> download and install nginx from the instructions given from their wiki
2. Use ``nginx -c /archives/nginx.conf`` to point nginx to given config file which is set to forward any api connections to the right port
3. If any changes are made to the *nginx.conf* file, it can be tested using ``nginx -T``
4. Use ``nginx -s reload`` to reload nginx with the new configuration file

13.3 Additional Feature:

13.3.1 Stripe Payment

As we ran out of the 5 minute limit in our video we weren't able to show the stripe payment system working but it was shown previously in the client meeting, so we are adding screenshots of the system here only.

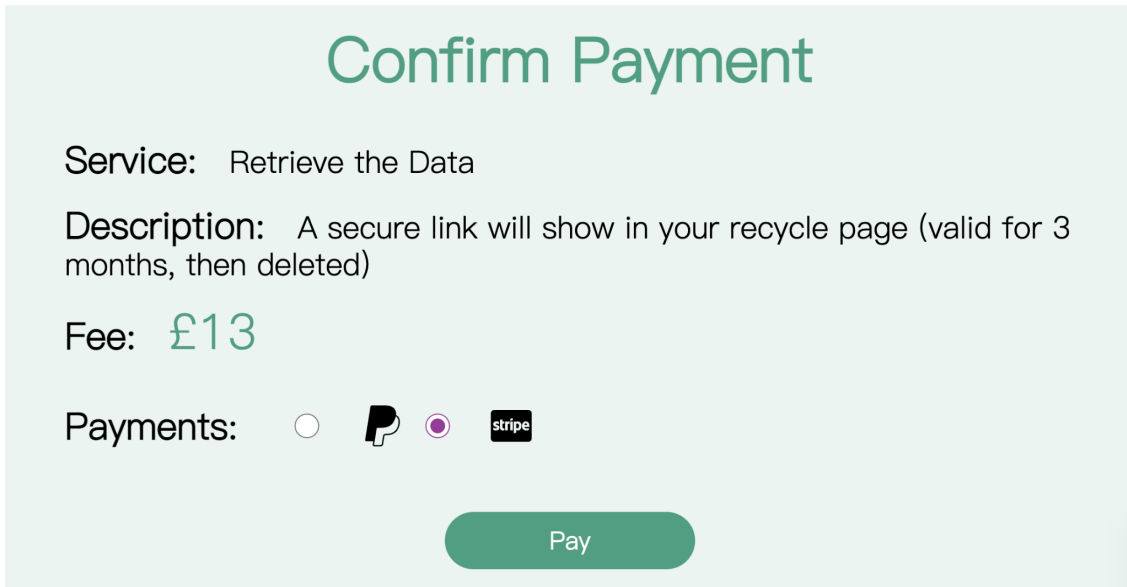


Fig 13.10 Stripe payment selection

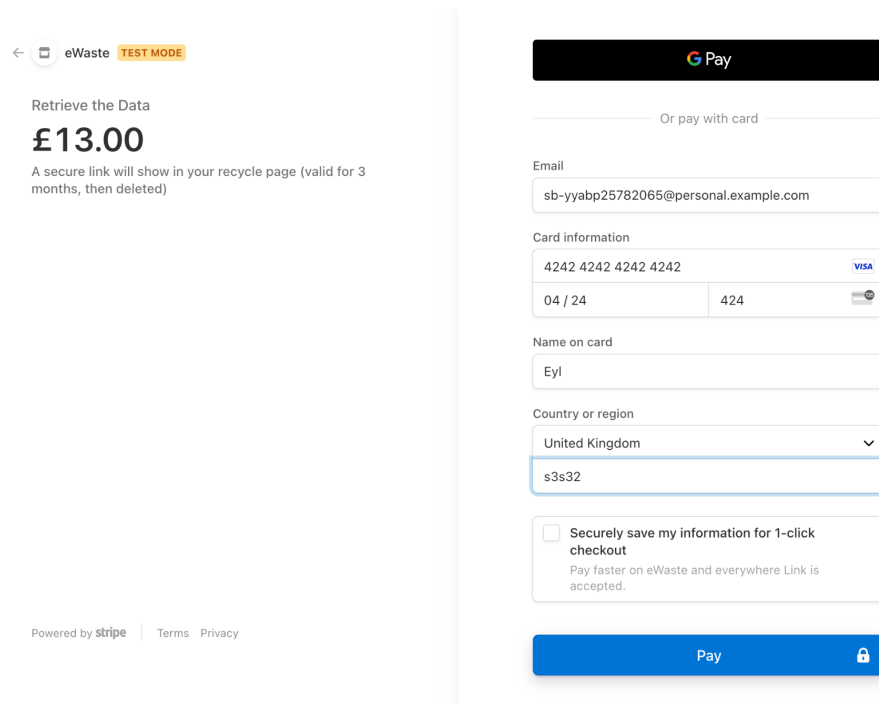


Fig 13.8 Stripe payment system sandbox for eWaste