University of Sheffield

Department of Computer Science

May 17, 2022

COM6103 - Team Software Project Report

Team03 - Dark Horse

By

Kaihang Zhu

Die Hu

Brendon Rodrigues

Xiaolei Xu

Yetong Han

Tong Chen

1 Introduction

The ability to work well in a group is critical to personal and professional success. Collaboration toward a common goal of completing a functional and finished product can increase productivity. When teammates encourage and assist one another in completing a project, production time is reduced and errors are avoided. Open communication and collaboration facilitate increased learning opportunities and creativity.

This report describes how our team collaborated on a software project. A team of six is formed to develop a software project using an Agile methodology. Agile software development is a collection of iterative software development methodologies in which requirements and solutions emerge from collaboration among teams. Our team follows the scrum framework where we work in sprints(iterations). Scrum is an agile development methodology that uses iterative and incremental processes to develop software. Scrum is an agile framework that is designed to deliver value to the customer throughout the project's development. It is adaptable, fast, flexible, and effective. This report also discusses the details of the software developed and explains the functionality of each and every feature developed.

Every year, millions of tons of food is wasted. One of the main causes of such a severe waste problem of food that is wasted in houses or food-stores is the lack of awareness of expiration dates. Furthermore, people always have difficulty dealing with food that is approaching its expiration date - they are unable to consume it on their own and find it difficult to give it to someone else who may probably require it. Thankfully, because of the widespread use of the Internet, we can now use this technology to reduce this food-waste by sharing items that are about to expire, with others. As a result, in order to reduce food waste, our project ‘SpareFoodShare’ aims to create a software that allows users to share food that is about to expire or is no longer needed by them.

This web application aims to make it simple for people in different parts of the UK to share leftover food. Users can post food items they'd like to share using this web application. They can also use this website to browse, search, collect, and buy nearby sparefood. Further to that, the application divides users into Individual Users and Business Users based on their needs, and each user can use the software as both a buyer and a seller, making it more user-friendly.

2 Team name and list of members

Our team's name is Black Horse, which refers to a previously unknown ordinary person who emerges victorious in a competition. Likewise, we want to be the dark horse in this course because, in comparison to other teams, we lack web development experience. We want to use team and software management techniques that we learnt throughout this module and emerge as the course's black horse.

Team Members: Kaihang Zhu, Die Hu, Brendon Rodrigues, Xiaolei Xu, Yetong Han, and Tong Chen.

3 Project scope and objectives

The app's goal is to make it easier for users to share their impending and no longer needed groceries. The app is aimed at all people or groups who want to share food with others and allows them to claim a certain amount of money in return, thereby driving more people to use it. In addition, adding project functions, such as search and filtering, is convenient for users to use to maintain user stickiness.

There are three types of users: Individual Users, Business Users and Administrators.

A visitor to the website can browse and search for food items, but if they wish to post or buy any of the products they want, they must first register/create an account and log in. The web also provides users with a chat function, which makes it very simple for the buyer and seller to interact before making a deal. The buyer and seller must agree on a collection point for the food. The order will be completed once the collection is completed.

The administrators have the privileges of managing users, products and orders, including adding, deleting, editing and searching for a specific record. Besides, administrators can view the statistics of relevant metrics of the website. Also, some PDF reports can be downloaded from the website.

4 Product backlog

According to the project scope analysis in the previous section, we managed to turn the requirement document into a list of user stories. Particularly, the stories has been divided into three parts as follows:

- **Guest - the User who hasn't been registered in the app**

As a Guest, I want to view products and their details without registering, and if I seem to be interested in a specific product or if I feel I would benefit from this software, I am able to register to it and am able to use the other functionalities.

As a Guest, I want to have a user-friendly registration form, so that I can register conveniently and quickly.
As a Guest, I want to have the option to register as a business user or an individual user so that I can have the option to either bulk import the products that I wish to share if I were a business user or share just one item at a time that is spare if I were a regular individual user.

- **User - the User who is registered into the app**

There are two types of users, which are individual and business users. For the stories below, a user refers to both an individual user and a business user, if not specified by "business user".

In order to ensure that the email user offered is valid, I as an administrator need to require the user to pass an email validation when they sign in for the first time, so that I can send notifications to the User by email.

As a user, I want to see the information of an item on the main page with a picture, the expiration date , price and the pickup location, etc, so that I can get enough information to choose what I want.

As a user, I want to have an option to register as a business user or an individual user, I can have the bulk import function if I were a business user.

As a User, I want to add a product with details such as images, the expiration date , price, description and location, so that the product would be more attractive.

As a user, I want to have a secure chat application to chat with the person who listed the item, so that I can get more details about the item and arrange a place to pick it up.

As a user, I want to share the listed items on social media platforms, so that I can let more people know about my items.

As a user, I want to search items via a search bar or filter items by categories, so that I can view fewer items and be more likely to find what I want.

As a user, I want to see the seller's location on a map application(Google Map), so that I can figure out whether it is convenient to get this item.

As a user, I want to allow requesting a fee, so that I can get some charge with some items.

As a user, I want to list some items privately and get notifications if the expiration date is close, so that I have time to decide whether to sell these items before their expiration date .

As a user, I want to be able to refund for orders which are under certain circumstances.

As a user, I want to be able to use the app on a mobile device, so that I can use this application more conveniently everywhere.

As a user, I want to see an equally good web page on my mobile device, so that I can use it on my mobile just like I use it on my computer.

As a user, I want to see my order history, so that I can know clearly what I have bought or collected on the webpage.

As a user, I want to see my product list, so that I can easily know about what item I have pushed on this webpage for sharing.

As a user, I want to change the display status of my items from private to public or from public to private, so that I can hide or show the items by requirement.

As a user, I want to get my receipt after finishing an order, so that I can easily manage my budget.

As a business user, I have an additional function to bulk upload items via a CSV file, so that I can save hundreds of time when uploading large numbers of items.

- **Administrator**

In order to enable the administrator to CRUD user information via our web dashboard, we need to provide a CRUD interface that they can use to manage user information from their desktop or mobile devices.

In order to enable the administrator to CRUD products via our web dashboard, we need to provide a CRUD interface that they can use to manage various products from their desktop or mobile devices.

In order to enable the administrator to view order details via our web dashboard, we need to provide a table that they can use to view orders from their desktop or mobile devices.

As an administrator, I want to remind users when their items are close to expiration date , so that the user will be aware of these items and make them visible to the public.

As an administrator, I want to have a data visualisation tool to help analyse the data, so that I can figure out how to improve the application.

As an administrator, I want to generate PDF reports for the various data stored, so that I can clearly view some information such as revenue, user proportion and product types.

As an administrator, I want to use the administration system on my mobile device, so that I can manage my website anytime, anywhere.

As an administrator, I want to delete the wrong items, so that I can eliminate error information which can mislead users.

5 Analysis & Design

In the early stage of the project, we conducted a detailed analysis of the project requirements, and finished the design of code architecture, database architecture and design of user flow.

5.1 Design of System Architecture

After discussion, we decided to use following framework for project development:

- Back-end：Django framework, sqlite3
- Front-end：Django framework, Bootstrap

5.1.1 Code Architecture

We divided the requirements into several relatively independent features and designed our code structure according to features. As Figure 5-1 shows, we create 5 applications (administration, chatbox, homePage, payment and userAuthentication) and each application implements one feature. Under the Django framework, applications can be imported as modules by each other. It is helpful for our later parallel development.

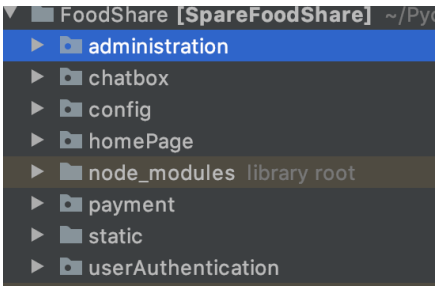


Figure 5-1 Code Architecture

In each application, the flowing of data between the front-end and back-end is completed through the MVT model (shown in Figure 5-2), which includes:

- Model: responsible for operating on databases through the ORM (Object Relational Mapping).
- Template: responsible for how to display the webpage to the user.
- View: responsible for business logic, complete data transfer between model and template and return HttpResponse to front-end.

In addition, each application also uses a different url controller which sends different url requests to corresponding view functions to process requests.

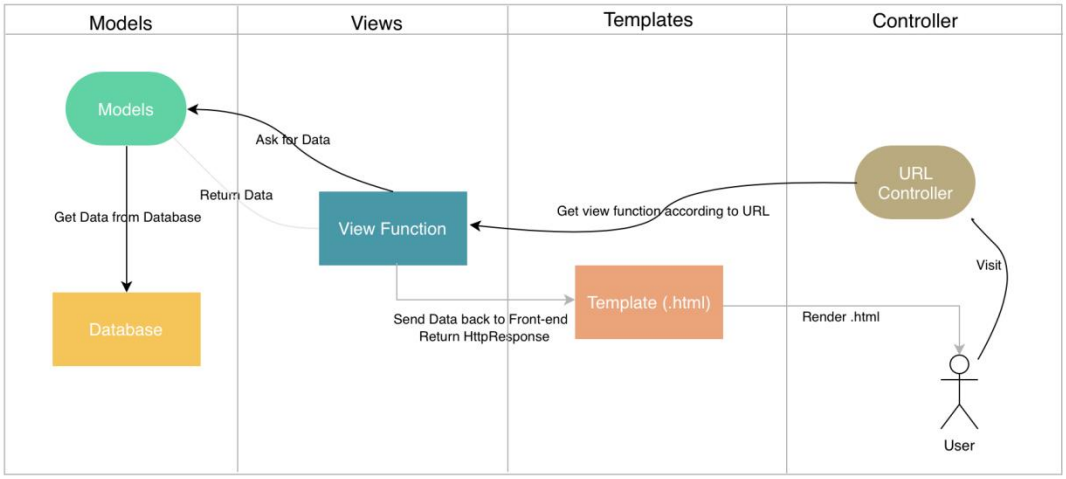


Figure 5-2 MVT Model

5.2 Database Architecture

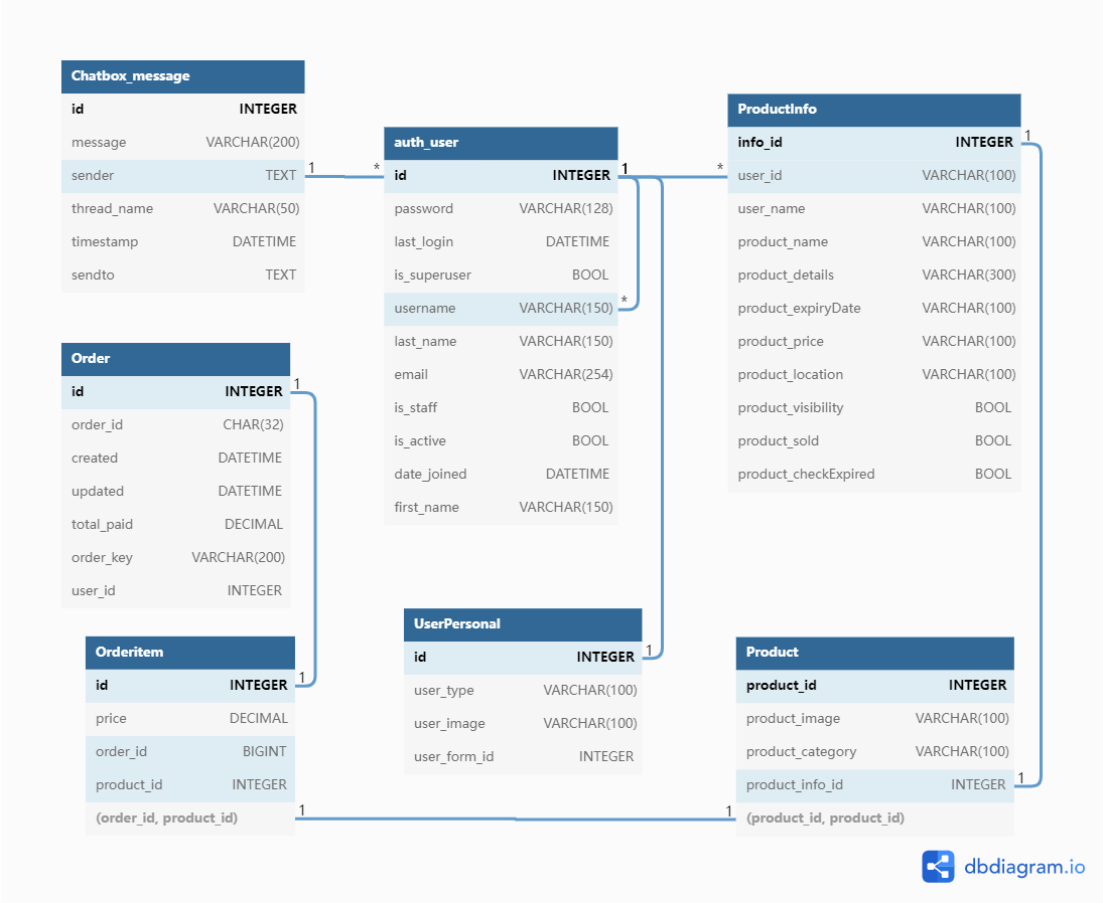


Figure 5-3 Entity Relationship Diagram

It is the core function of FoodShare that enables the user to share food that is about to expire or is no longer needed by them. Users may query the food information and their order records. According to this core requirement, we intend to store the following three independent categories of information:

1. The User information, including first name, last name, email address, user type,etc.

2. The Product(food) information, including images, expiration date , descriptions, locations.
3. The Order record for each payment. With this order record, customers can check where to collect the food or request a refund if there are problems.

It is noticeable that our application should contain at least three entities - User, Product and Order. Around these three entities, User has a one-to-many relationship with both Product and Order. Given these clear relationships, the relational database is very suitable for this project. We finally chose Sqlite which is the default database of django framework as our database.

The ERD(Entity Relationship Diagram) shows the database design for our project (Figure 5-3), which contains both the fields for each entity and the relationships between entities. Basically, three entities(User, Product, OrderItem) have been created. For the additional information of these entities, we create three respective entities - UserPersonal, ProductInfo and Order. Additionally, an entity called Message is set up to store the messages between seller and customer.

User Databases have two entities: User and UserPersonal. The primary key of the User Table is User Id. We intend to query for the additional user information with respect to User Id. As well, we can search for the items listed by a specific user and his/her order history. The detailed design of some important databases are as follows:

UserPersonal Table

Field Name	Description	Data Type
user_form_id (Primary Key)	Foreign key linked to 'id' of 'User'.	Int
user_type	The user type of a user. Just have 2 type: Individual, Business	Varchar
user_image	The profile photo of a user.	Image

Product Table

Field Name	Description	Data Type
product_id (Primary Key)	The id of 'Product'	Int
product_info	Foreign key linked to 'ProductInfo'.	Object of 'ProductInfo'
product_image	The image of the product.	Image
product_category	The category of product.	Varchar

OrderItem

Field Name	Description	Data Type
id(Primary key)	Automatically created by sqlite	Int
price	How much did the buyer pay	Decimal

order_id	Foreign key linked to the Order entity	Bigint
Product id	Foreign key linked to one product.Unique id for each product, here each OrderItem have a one-to-one relationship with Product	Int

5.3 Design of User Flow

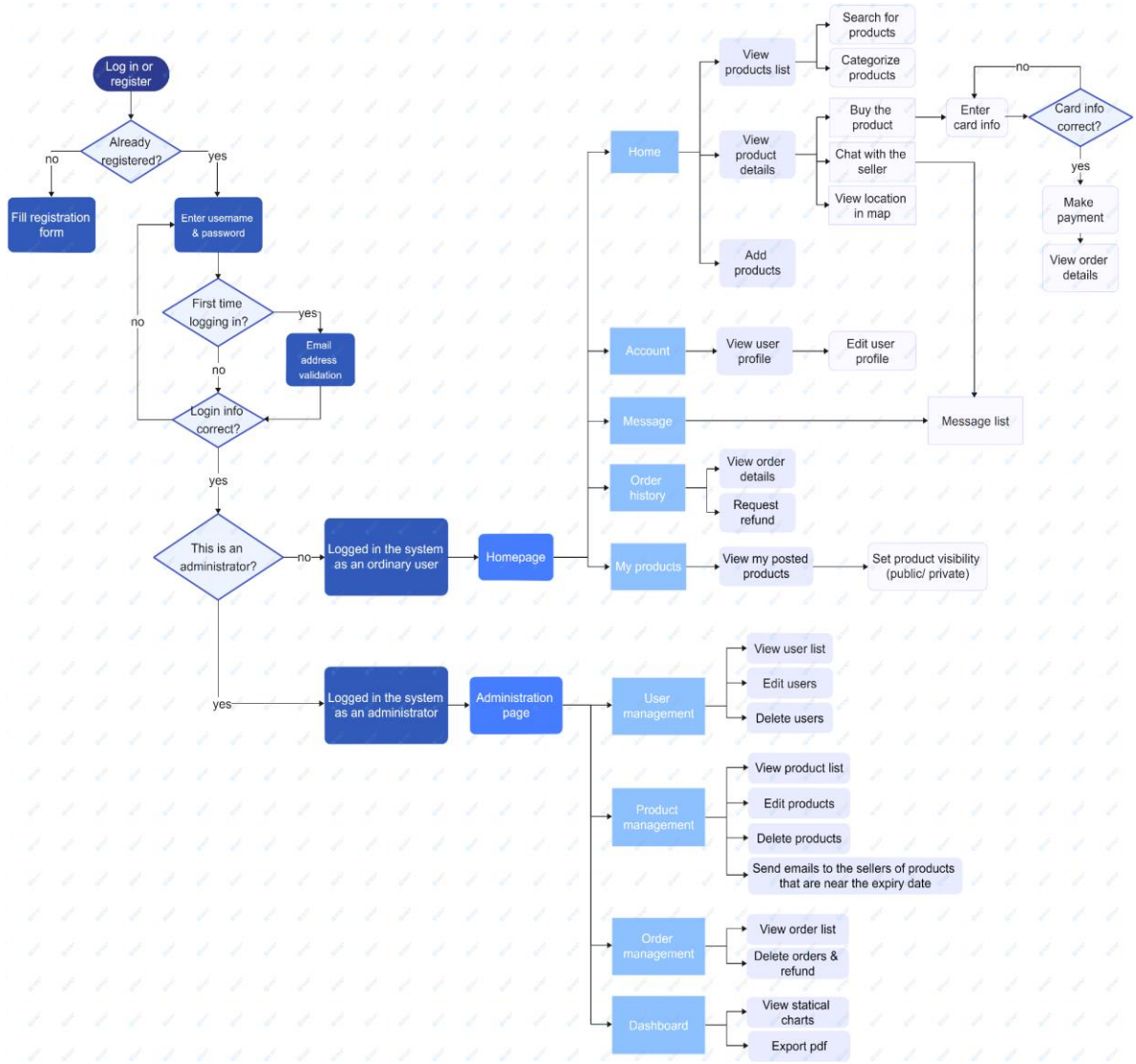


Figure 5-4 User flow (Generated using EdrawMax)

During the stage of project design, we created the user flow diagram(Figure 5-4). We divided the system into four main modules - login & registration, viewing & uploading products, chatting & payment, administration system.

A user should log into the system to use all the functionalities. The validation of email addresses has also been taken into consideration.

As for the module of viewing & uploading products, in addition to viewing all the products in the homepage, users can also use the search box and filter to search for particular products or view products by category. Besides that, users can upload products and can also view products posted by themselves.

As for the module of chatting & payment, users are able to chat with the sellers about the products, so that they can make a deal at a reasonable price. After the payment is completed, users can view their order history and details. It is also possible for buyers to request refunds for their orders on this system.

The administrators part is divided into four modules - the management of users, products and orders, and the dashboard. Especially, administrators can send emails to the sellers of products that are near the expiration date . The dashboard contains four statistical graphs of relevant metrics, such as the number of products by category and sales amount by month.

6 Testing

6.1 Test Plan

- **Test Environment:** Windows 10 x64
- **Test Method:** Integration Testing.
- **Test Objectives/Focus:** If the main functions are realized, and the fault tolerance of functions.

6.2 Test Documentation

No.	Test Case	Expected Result	Actual Result
1	Click the “Sign up”, “Account”, “Message”, “Order History”, “My Product” buttons or plus icon without logging in.	Enter into the login page.	As expected.
2	Click the “Sign up”, “Account”, “Message”, “Order History”, “My Product” buttons or plus icon after logging in.	Enter into the corresponding interface.	As expected.
3	Enter the wrong format of price and/or address when adding items.	Add failed and a hint message appears.	As expected.
4	Log in for the first time.	Receive a validation code by email.	As expected.
5	Users see the detail pages of their own uploading items from the Home page.	There are no “Buy me” and “Chat with seller” buttons, only “See the map”button.	As expected.
6	Use the search bar to find items by entering keywords.	Items with keywords either on their names or descriptions will be shown.	As expected.
7	Sellers change their own items from “Public” to “Private” or from “Private” to “Public”	The items will be hidden or be shown on the Home page.	As expected.
8	Enter the wrong card number in the	The card number and icon will	As expected.

	purchase interface.	be red and cannot enter the next step.	
9	Use a .csv file to bulk upload.	The items in the file are all added into the app.	As expected.
10	Change expiration date of items from not expired to expired on the administration page.	The item will be hidden on the main page of the app.	As expected.
11	Simulate chatting on the chat page.	Both sellers and buyers can receive the messages even if they are offline.	As expected.
12	Refund by delete a certain order on the administration page.	The order will disappear, the orders & Sales record on the dashboard will delete the refund one.	As expected.
13	Share items using social media.	The chosen item can be shared by social media.	Share failed.

As indicated in the above table, test case 13 failed. It's at a lower level on the priority list and time-consuming to implement. So we didn't implement it finally.

7 Team management & communication

7.1 Work plan

7.1.1 Gantt Chart

At the beginning of this project, we created a google online document for the report, which is good for collaborative editing. We made a Gantt chart (Figure 7-1) of work plans and put it online too, so that team members can be more specific about the uncompleted tasks and how much time is left. And in this way, we could make adjustments to the work plans in time as well.

From the perspective of the entire project development, we divided the whole project into five tasks: analysis and design, coding for the front-end part, coding for the back-end part, testing and report writing. Before the implementation of this project, we first analysed the project requirements provided by the client, and discussed the project technology that is required. Then around the user needs, we divided the system into several modules and designed the mockups of all pages. We also did the design for the database and code structure. Then we started coding and implementing the front-end and back-end part.

	Feb 21-28	Mar 1-6	Mar 7-13	Mar 14-20	Mar 21-31	Apr 1-3	Apr 4-10	Apr 11-17	Apr 18-24	Apr 25-30	May 1-8	May 9-12
Task 1 [Analysis and design]												
Requirements analysis	X											
Prototype design	X	X										
Database design		X	X									
Weekly meeting	X	X	X	X	X	X	X	X	X	X	X	X
Task 2 [Coding for front-end]												
Registration & login pages			X	X								
Homepage			X	X	X	X						
Chat pages				X	X	X	X					
Payment pages					X	X	X					
Administration pages			X	X	X	X	X					
Adapting pages for mobile							X	X	X	X		
Task 3 [Coding for back-end]												
User models				X	X	X						
Product models				X	X	X	X					
Chat models					X	X	X	X				
Order models					X	X	X	X				
Views of administration					X	X	X	X	X			
Views of chatbox						X	X	X	X			
Views of homepage					X	X	X	X	X			
Views of user authentication					X	X	X	X				
Views of payment							X	X	X			
Code refactoring									X	X	X	X
Task 4 [Testing]												
Test plan			X	X								
Testing implementation				X	X	X	X	X	X	X	X	X
Debugging according to the test results				X	X	X	X	X	X	X	X	X
Aggregation of testing results										X	X	X
Task 5 [Report]												
Project introduction & user stories				X	X							
System analysis & design						X	X	X	X			
Team management & communication				X	X	X	X	X	X	X	X	X
Summary & video										X	X	X

Figure 7-1. Gantt Chart

As for the testing part, we designed some test plans, and then conducted testing as soon as one part of the project was finished in an iterative process. We did the testing throughout the whole implementation of this project, so that once a bug was detected, we could promptly fix it and optimise the code as soon as possible. Each person tested the part developed by them individually. After the whole project was completed, we selected one member as a tester to test the whole application and we aggregated all the testing results and wrote them in the report.

We began to work on the report once the mockups and database designs were finished. Around the analysis in task1, we wrote the project introduction and user stories, followed by the part of system analysis and design.

7.1.2 Trello

We used Trello to collaborate, communicate and coordinate on this project. It can record and display work progress on more specific projects. By using this tool, we could clearly see what was being worked on, who was working on what, and where something was in a process (Figure 7-2), which highly improved our cooperation efficiency.

The agile development approach was used in this project. In the beginning, we listed the user stories according to user requirements, and prioritised them to create the product backlog. At our weekly meeting, we chose some user stories and subdivided them into smaller tasks, then put these task cards

on the ‘To Do’ module on Trello. With the help of Trello, tasks were arranged to specific team members with expected dates. As tasks were completed, team members could move the task cards to the ‘Doing’ and ‘Done’ module on Trello. In this way, task progress and project schedules can be clear to every team member.

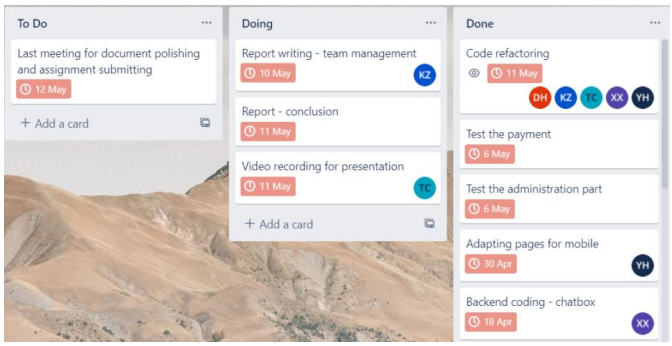


Figure 7-2. Screenshot of Trello

7.2 Team Management and Work Distribution

This project's development can be divided into four stages. We conducted a detailed analysis of the requirements in the first stage and divided them into several distinct features. Brendon jotted down all of the features we could think of for the project and cleared up any questions we had with the client about the requirements so that we were clear on everything that was required and expected before we could start working on it. We then created an excel sheet with all of the features and prioritised them based on the client's requirements, which helped us understand which features needed to be developed first, and we concentrated on those features first. Brendon then took notes on everyone's previous coding experiences, which helped us figure out who could be a part of a specific development process that they were knowledgeable about. Front end, back end, design, and so on.

After that, we decided to create some mockups and since Tong had some designing experience, the mockups were done by Tong. Other members of the team put forward some revisions to the mockup, and completed the design of user flow, code structure and database structure on this basis. We then discussed about the framework that we would use for the project and discovered that most of us had some to good experience with python and finally, we decided to use the Django framework and sqlite3 to complete the project coding since Django REST Framework (DRF) is a Python/Django library that aims to build sophisticated web APIs. It is open source, mature, and well supported. It is a versatile and feature-rich toolkit with a modular and customizable architecture that allows for the creation of both simple and complex REST constructs.

In the second stage, we learned that most of the members in the group are good at front-end development, so we decided to focus on the front-end of the project first. We divided the front end of the entire project into seven parts, including Login and Register, Homepage, My Account, My Product, Detail Page, Chat and Payment, and Administration System. The division of labour is as follows: Ye Tong is responsible for "My Account" and "Login and Register" because the layout of these pages is very similar. At the same time, the layout of Homepage and My Product and the logical business involved are also very similar, so Tong completed the front-end development of these two pages. ; Kaihang is responsible for the Detail Page, which displays the detailed information of the product; Xiaolei is responsible for the Chat and Payment parts, because these two parts are very closely linked; Die is responsible for the Administrator System, which enables system administrators to manage the platform uniformly. And Brendon, who is better at the back-end, completed the establishment of the User database, Product

database and the login and registration functions at this stage. He also structured the different applications of Django.

In the third stage, we focused on backend development. Brendon had to choose to return to India due to personal reasons, so he temporarily stopped his work. After knowing the news, the remaining team members met in time to discuss and completed a new division of labour. It was finally decided that Tong should complete the back-end code of Homepage, My Account, My Product, Detail Page and Administration System, Xiaolei to overcome the complex part of Chat and Payment, and Kaihang to complete map function in the Detail Page, the optimization of the front-end page is completed by Yetong, and the statistical chart and PDF export function are completed by Die. During this time, we have been in constant contact with Brendon on whatsapp and kept him updated on our progress and involved him in some of the group meetings where he would provide some feedback from his end.

The main task of the fourth stage is to test and optimise the page function. Die and Kaihang tested various functions of the website, and promptly reported bugs to the backend. At the same time, they began to write the introduction, testing and team management parts of the report; Tong and Xiaolei fixed bugs and optimised related codes after receiving their feedback. On this basis, Tong added the function of logging in to the project for the first time to verify the mailbox and the function of sending reminder emails to the background administrator, and Xiaolei added a refund function to the project. Yetong has adapted all mobile pages.

Finally, each team member participated in the work of writing the report. Tong and Xiaolei completed the code-related portion of the report. Tong finished recording the video. Xiaolei and Brendon revised the user story. Yetong completed the user guide. Die presents various visualisations for user flow and team progress management. Kaihang refactored the test documentation. Finally, we revised the team management and conclusion sections together. We reconvened after Brendon came back and he helped in writing a significant part of the report as well; for eg: Intro, conclusion, the challenges faced etc.

7.3 Communication as a Team

- Daliy: We use WhatsApp as a communication tool. After the team members push their own code on gitLab, they will also update their progress in the group chat to ensure that everyone can understand the progress of the project. And we will also ask the questions we encounter in the group, and other members will help to answer them. We would discuss in the group and would decide on when we could arrange our weekly meetings.
- Our group uses Trello to record and display the progress of work on more specific projects. Record this week's work and next week's plan to the corresponding To Do, Doing, and Done lists. At the same time, we also set deadlines for each small task. This allows each member of the team to easily understand the current work progress, and also make the next work plan clear.
- We have weekly in-person meetings which have played a vital role in our development process. First, each member will report on their work done for the week. The information is then compiled and a work plan for the next week is determined. Finally, members who encounter difficulties will raise their own problems (usually some bugs encountered in coding), and other members will help solve them together. At the same time, we will also raise some doubts, or express our own new ideas in the development process. We take every point of view seriously and discuss it. Often, after discussion, we find that there are some unreasonable places in the code structure or user flow. We'll explore how to improve.
- If due to special reasons, we cannot meet offline, we would also hold online meetings on zoom to ensure the normal progress of the project. And every time there is a meeting, Brendon would

be responsible for recording the content of the meeting where he notes down the minutes of the meeting, forming a formal document and sending it to the group chat.

7.4 Challenges

1. **Different coding backgrounds:** The first challenge we encountered was the inability to divide labour reasonably. Since everyone comes from a different background, they specialise in different areas. During the initial meetings and while getting to know each other, we learned that among the six members in the group, only Xiaolei and Brendon have experience in back-end development, while the remaining four only have experience in front-end development. After discussion, we decided that it would be unrealistic for Xiaolei and Brendon to do all the backend work.

   However, learning takes time, and we didn't think it should delay the development of the project. So we settled on a plausible solution during our meeting. Xiaolei and Brendon worked on the back-end development first, while the remaining four focused on the front-end coding and learned the back-end code during this period. Just like the division of labour described in the 'Second Phase' above.

   At the same time, we also believed that work must be distributed flexibly according to each person's ability and experience. For example, Tong Chen has experience in designing web pages, so she undertakes the work of mockup, and she can help members with weaker abilities in this area to make suggestions on their web page styles in the coding stage. Xiaolei Xu is more experienced in back-end development, so he is responsible for the implementation of the difficult chat and payment functions. Brendon also has back-end development, so he is responsible for building the User database and implementing the login and registration part.

   Die Hu is very familiar with Bootstrap, so her part is to use Bootstrap to build the background administrator system. And her style does not have to be unified with other pages, because this is a completely independent system. Yetong Han doesn't have much coding experience, so he was assigned the front-end of the login and registration page at the beginning. But he gradually developed a strong interest in the front-end during the development process, and accumulated a lot of experience, so later he volunteered to undertake the mobile phone adaptation of all pages.

   In summary, we assigned tasks based on each team member's experience and expertise. This approach greatly improved our efficiency and to make full use of the time at each stage.

2. **Adapting to latest technology:** During our initial meetups and getting to know each other, we discovered that we all have different coding backgrounds, as well as experience with various frameworks as mentioned above. After which we tried to figure out which framework would be best for our project and which would be the most convenient and simple to learn for the team. We chose Django as our Python framework because we realised that everyone on the team had some extensive experience with python. It was one of the most difficult challenges we faced because none of us knew Django as it is one of the latest technologies. We set aside a few weeks to learn the framework while also designing, gathering requirements, and planning the project. We struggled at first, but once we understood how the framework worked, we found it to be very simple to use, and we believe we made the right decision in choosing Django as our project framework.
3. **Application Integration:** In a software project, working in a team entails working on various aspects of the project at the same time. We used GitLab to integrate and manage our entire

project because we worked on our own parts and pushed the code we developed to GitLab. When we tried to merge on GitLab a couple of times, we ran into a lot of merge conflicts and errors which would be very frustrating and a nightmare. Fortunately, we were able to resolve the merge conflicts and continue working on the project.

4. **Holding weekly meetings:** We decided to meet every week to check on the team's progress in addition to our regular weekly meetings with the advisor. We would meet before each advisory meeting to ensure that we were prepared. However, finding the right time for meetings was sometimes difficult due to our diverse schedules. If meeting in person proved difficult for various reasons, we would try to meet online via Zoom. Brendon had to take a LOA and return to India, making it difficult to hold a meeting with the entire group. As a result, the five of us would meet in person or online and then update Brendon on the meeting minutes.
5. **Unexpected circumstances:** It's very likely that members will fall out over time, especially in large groups. Members leaving can be disruptive, as the number of members available at any given time is an important factor in early planning. Each person has their own set of skills and abilities. With just one member absent, the project's unique abilities are lost. It can also be aggravating to pick up where another person left off, leaving the group to complete someone else's work.

   During the course of the project, Brendon had to return to India due to a personal unforeseen reason. He explained the situation to us right away and did his best to get as much work done as possible. Before leaving the UK, Brendon had a brief meeting with us to explain what he had done so that we could take over his work smoothly.

   Due to this unexpected situation, the work of each member became more onerous later. So we decided to reassign work. When re-assigning, we thought more about how to maximise the overall efficiency of the team. At this point, the members' learning process of the back-end had not yet been completed. However, in the absence of a member, this method would lead to a huge waste of learning costs. For the Django framework, the time it takes to learn and successfully operate the database for the first time is long, and after that, the time spent will be greatly shortened regardless of which database the learned knowledge is applied to.

   Therefore, we believed that it was unwise for everyone to take the time to complete the first step in such a sudden situation. After discussion, we decided that Xiaolei would continue to complete the difficult Chat and Payment functions, Tong who first learned the Django framework would do the rest of the back-end work, and Yetong would complete the mobile adaptation of all pages. Reporting and testing by Die and Kaihang.

   In this way, we saved a lot of learning cost and time cost. And we worked harder, spending more time writing code than ever before, and meeting more often than ever. We also made it a rule to update ourselves on the day's work in the group chat at 10am every morning. This not only achieves the role of mutual supervision, but also allows members of the group to keep abreast of the overall progress. At the same time, we often wrote code together late into the night in the school conference room.

   So, with the efforts and cooperation of all members, we finally completed almost all the functions during the presentation. This is something we did not expect before, and it is also the result of our joint efforts.

6. **Group Diversity:** Because the team's members came from various backgrounds and countries, it was sometimes difficult to understand what one was trying to say. However, we did our best

to consider each member's viewpoints and made it a point to ensure that no one's voice went unheard. If we ever had a problem, we would look it up on Google and make an effort to understand each other.

7.5 Meeting Records

We had weekly meetings. Most of the meetings are summaries and reports of daily work, so that members can understand the current work progress and make reasonable arrangements for the work plan for the next week.

Below are the minutes of the two most important meetings:

Team Number/Name:		Team 3 / Dark Horse	
Meeting Name(format):		Group Meeting (In Person)	
Date of Meeting:	28th February 2022	Time:	3PM - 4PM
Meeting purpose:		Make a schedule of the project	

Meeting Notes and Decisions

- Choose Python as the primary language of the project, Django as the framework, and SQLite as the database.
- Divide the project into three main stages:
  - UI design and front-end implementation.
  - Database design and connected; Back-end implementation.
  - Optimization and improvement; Writ report.
- As for first stage:
  - Define the basic model of the project.
  - Determine the basic functions of the website
  - Divide the front-end into 7 main pages.
- Assign work according to page and specific functions.
- Responsibility:
  - Home Page & My List Page — Tong Chen & Brendon Rodrigues
  - Registration & Login and Account & Editing Page — Yetong Han
  - Chat and Payment Page — Xiaolei Xu
  - Administrator Page — Die Hu
  - Product Detail Page — Kaihang Zhu

Team Number/Name:		Team 3 / Dark Horse	
Meeting Name(format):		Group Meeting (In Person)	
Date of Meeting:	10th May 2022	Time:	2PM - 3PM
Meeting purpose:		Make a conclusion of the project and allocate the final work	

Meeting Notes and Decisions

- Summarise the whole project according to the client's feedback during the client meeting.
- According to the client's feedback, the problem are summarised into two parts, the page display part of the front-end and the function realisation part of the back-end:

- Front-end:
  - Mobile views adaptation does not meet client's requirements, the product information of the homepage can not display completely on small screen devices such as IPhone SE, IPhone 6/7/8.
  - The menu disappears on the mobile version homepage.
- Back-end:
  - Lack of refund function.
  - The x-axis appears a little shift in the second graph of the dashboard.

- Allocate the final parts work:
  - Front-end: Improve and optimise small screen device adaptation of web pages, rearrange the page layout on mobile device — Yetong Han
  - Back-end: improve the refund function and fix the bug where the administrator would refund users when deleting order records — Xiaolei Xu
  - Initialise the database file and fix bugs of back-end— Tong Chen
  - Fix the bug of x-axis's shift — Die Hu
  - Complete the remaining unfinished sections of the report — All members

8 Completed & Uncompleted Features

There are 8 sprints during our Software Development Life.

Week 1-4:

- Prototype design

Week 5：

- Database design
- Optimise the UI design based on client's feedback

Week 6-7：

- Complete the web pages based on the mockup

Week 8:

- Registration and login function
- Allow users to choose user type when they register
- Display item cards on the home page

Week 9:

- Display item details on the item details page
- Show seller's address on Google Maps
- Create a web page to uploading product with details
- User can view their own products

Easter Vacation：

- Chat application
- Search items via a search bar or filter items by categories
- Allow user to review his/her order history
- Change the visibility of my items
- Bulk upload items via a CSV file
- Enable the administrator to update or delete products
- Enable the administrator to view order details

Week 10:

- Get notifications if the items' expiration date is close
- Get a receipt after payment
- Administrator can send email to users to remind their items are close to expiration date
- A data visualisation tool to help analyse the data
- Adapting websites for mobile devices

Week 11:

- Refund for orders
- Email validation when sign in for the first time
- Administrator can generate PDF reports
- Bug fixes and functional improvements

8.2. Uncompleted features

- Sharing Function: Our app does not currently support sharing items on social platforms. Because it is at a lower level on the priority list and it is time-consuming to implement this function. So we didn't implement it.
- Refunding Function: When we conducted requirement analysis, we didn't take into account that users might want to request a refund. Later, during the last client meeting, the client pointed that we were missing the refund feature. So we implemented this functionality in the last iteration. Users can request a refund on the order history page. And the administrator can also make a refund for users directly in the administration system.
- Responsiveness Issue: We have finally finished adapting websites for mobile devices, but due to time constraints, there are still 2 minor problems. On the 'My Product' page, users can't click on the main menu icon. And on the 'detail page', the image of the products is so large that it covers some information.

9 Conclusion

During this module, we created the software application 'SpareFoodShare.' This web application allows users to share food that is nearing its expiration date that they no longer require or are unable to use for various reasons and would like to share with others who may require it. During the initial stages, the group was formed and we were assigned the project that we needed to develop. We then began by analysing the client's requirements, clarifying any ambiguities, and converting them into user stories. Following that, we created a prototype and database based on these stories. Finally, we used the Django framework to implement the majority of the required features. We communicated simultaneously on various platforms, documented all important specifications, and kept track of all meetings. The project was completed on time, with a slight delay due to unforeseen circumstances affecting one of the team members. We made every effort to keep the client informed of our progress and to complete each task on time.

One of the most important lessons we've taken away from this project is how to work in a group. First, we learned about Task Management, which involved breaking down tasks and assigning them to different team members. We were able to use a Gantt chart, which is a work schedule with clear deadlines for each task. We used Trello to create small tasks and assign them to appropriate team members to track task progress. Then we learned about collaboration, which taught us how to collaborate with context, transparency, and accountability. Then we learned about Resource

Management, which involved understanding and planning the team's capacity and resource management in order to assign tasks equally to the right person with the capacity to complete the task.

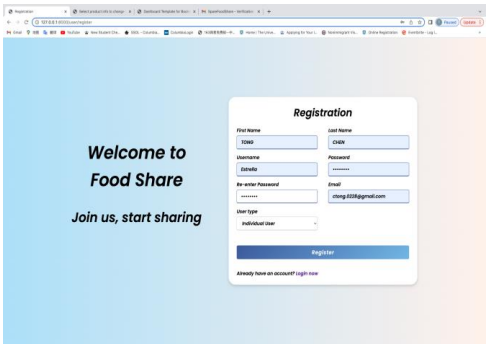
Working in a group can be difficult at times. In an ideal software team, it is always expected that every member of the team contributes equally in the development of the software; however, this usually is never the case. Software teams usually follow the 80/20 rule where 20% of the project is developed by 80% of the team. We had a few issues to deal with as a team of six during this module, such as different members of the team not being experts in certain technologies, communication issues and having to schedule meetings according to everyone's convenience, problems integrating the software together since everyone worked on it separately, and finally one of the most uncontrollable challenges of having a team member take a leave of absence. However, we overcame all of these obstacles and did our best to produce a well-functioning and satisfactory product that met almost all of the client's requirements. Apart from team management, we've learned the value of properly documenting the project by providing a detailed description of each and every project function, as well as properly commenting on the code to help other developers understand how it works. This benefits both the current development team and future developers who will work on the project in the future.

Overall, the project aided us in improving our teamwork and project management abilities. We can now confidently say that we have good teamwork experience, which is essential in software development. Working effectively in a team is an essential skill for any software developer. Coding is frequently thought of as a solitary activity involving hours spent alone in front of a computer screen. A developer, at any level, is worthless unless he or she can collaborate constructively with other developers to solve problems. In many ways, we've learned that group projects can be both rewarding and frustrating. Teams can always accomplish more than individuals working alone. People with different perspectives and levels of knowledge can learn from each other in teams.

Appendix

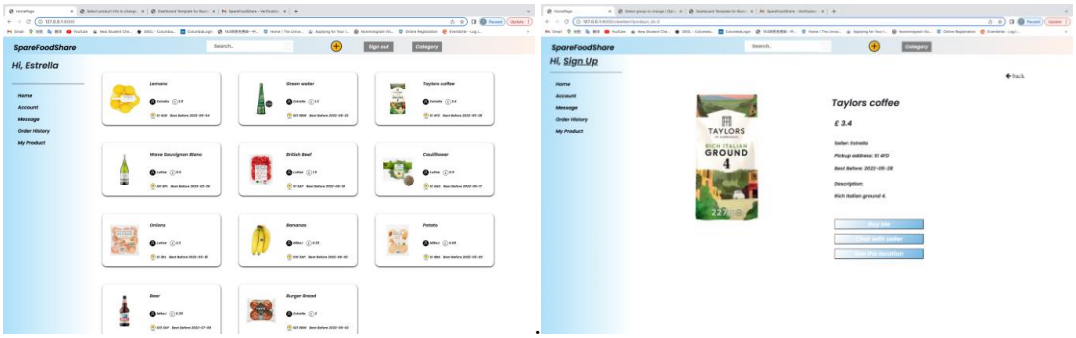
a) User guide

- **User Registration/Login**: After entering our website for the first time ever, users can browse all products, search, and filter functions as visitors. If users want to use other functions such as requesting to get a product or post a product on the system, they would need to log in. Click the "Sign Up" to enter the login page. Users can click 'Register now' to register as a new account user. During registration, the system will check three things: unique username, correct email format, and the same password entered twice. Users also need to choose to register as 'Individual User' or 'Business User'.

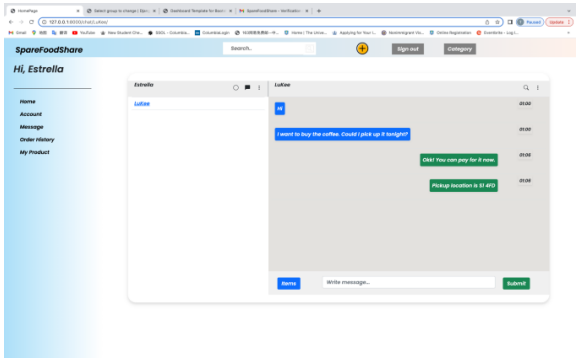


After that, the page will automatically jump to the "Login" page. Suppose a user logs in the first time, the user's email needs to be verified. The system will send an email containing a random 4-digit code to the user. Moreover, he/she types the correct username, password and code to enter the Homepage. Once the user is logged in, they would directly be redirected to the homepage every time they visit the webpage unless the user logs out.

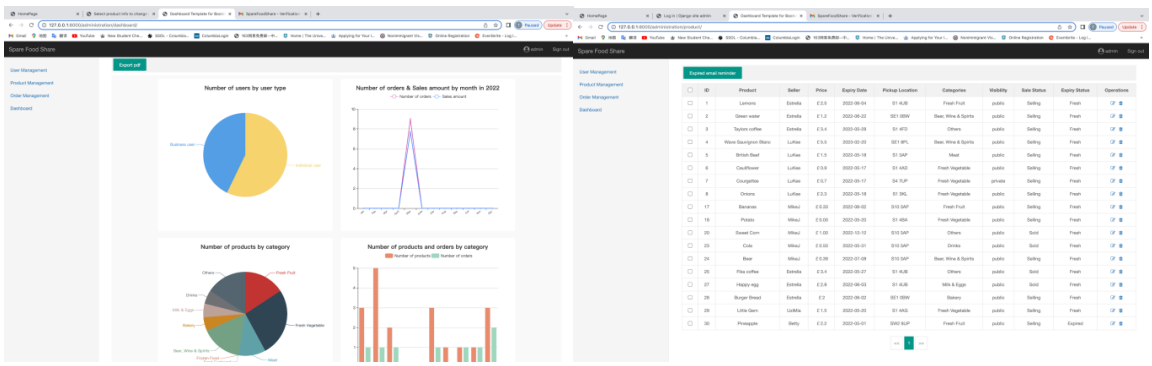
- **Add Item(s):** Users can click the top-right plus icon on the Homepage to add products. If the user is not logged in, they would not be able to add an Item and would be redirected to the sign-up page first. Once Logged in, they can either add a single item if they are an individual user or add bulk items with a CSV file if they are a business user.
- **Menu/Dashboard:** They can also choose a category to view products under different categories and search products by keyword. On the left menu, users are presented with various options to visit various pages. For example, click 'Account' to check and edit the user's personal information, click 'Message' to see all the message history, click 'Order History' to view all the orders' information and apply for a refund, and click 'My Product' to view uploaded products and set them to private or public.
- **View Products**: Users can click a product card to see detailed information of the specific product. The user can view the location of the product by clicking "See the map" to check the pickup location on the map. The user can also check its expiration date along with many other details of the product which can help the user know the product better.



- **Chat-System:** If the user is interested in a specific product and wishes to request it, they can start a chat with the seller of the product by clicking on 'Chat with seller' so as to decide on the pick up point and any other details the user wishes to know before getting the product. Finally, click 'Buy me' to complete the payment. Please use the card number shown in 'stripecards.txt' for demo purposes. After payment, the user can click the link to view the order and its receipt.



- **Administration:** Administrators of the system can use a particular account (written in readme.md) to enter the administration system. On this page, they are provided with various management functions. They can manage information about users, products and orders (edit or delete) and view the products which will expire in the next three days and send emails to remind users so that the users are aware of the soon expiring product. The dashboard can also show some database charts that can be exported as a PDF file.



b) **Setup guide**

1. To run this project, you can follow these steps: Download or use ‘git clone https://git.shefcompsci.org.uk/com6103-2021-22/team03/project.git’
2. Extract it in a folder and open a powershell/terminal
3. If you were a win user, please use the following commands directly;

   If you were a mac user, please change the second command to 'source venv/bin/activate'

   Commands:

   - python -m venv venv
   - venv\Scripts\activate
   - pip install -r requirements.txt
   - python manage.py runserver
4. In config /settings.py the stripe keys belong to one of the team members - just put your own details in here.

   Stripe Payment

   PUBLISHABLE_KEY = ''

   SECRET_KEY = ''
5. You can open this web application in a browser by url 'http://127.0.0.1:8000/'