CFGdegree Autumn 22: Software Stream

Urszula Kamińska, Keziah Belinda Malungu, Darya Staliarova, Aleksandra Grunt-Mejer

November 6, 2022

Figure 1: Web app logo

Contents

1 INTRODUCTION:

1.1 Aims and objectives of the project

Aims:

Food waste is a growing problem; therefore, our aim was to create a full–stack web application in which users can create an account, and upload announcements/adverts about food that they have available to share with fellow users in their vicinity. Users wishing to collect food will be able to see a list of food available to collect in their local area as well as in-depth descriptions about the food items.

Our objectives were to:

- Provide a web application with a user-friendly interface that is accessible and easy to use for all.
- Allow users to be able to search for food items.
- Create a fully functioning database that stores the data of the users and all the data about the food announcements/adverts so that this information could be later retrieved by the front end.
- Provide a web application with a user-friendly interface that is accessible and easy to use for all.
- Allow users to post announcements about available food and create accounts.
- Create a fully functioning database that stores the data of the users and all the data about the food announcements/adverts so that this information could be later retrieved by the front end.
- Provide a point of communication between users so that they can arrange pick-ups via a contact form.

1.2 Roadmap of the report

1. Introduction – An introduction into the project's purpose, the main aims, and the objectives we had for the functionalities of our web application.

2. Background – Insight into the motivations to create Sherfood and what rules/requirements govern the use of Sherfood.

3. Specifications and Design – An overview of the technical and non–technical requirements, as well as the architecture of our application.

4. Implementation and Execution – Insight into how we worked as a team to build and execute our application.

5. Testing and Evaluation – Details on how we were able to test our application and ensure that it was of a high standard.

6. Conclusion – A summary of the project and final remarks.

2 BACKGROUND

2.1 The Problem: Food Waste

It is estimated that **1.2 billion** tonnes of food are wasted, a substantial amount that could be given to the estimated **134 million** people that face severe hunger.[**1,2**] Food is lost or wasted in a variety of ways across the supply chain, some of which are avoidable. Some examples include when consumers waste food by overestimating how much they will need, or when retailers stop stocking unpopular food items meaning that produce is left wasted. Food waste is a growing concern as it has an impact on our planet's other precious resources, as 1/4 of our water supply is used to create our food, as well as 28 percent of our agricultural areas that were created by driving out species from their natural habitat and indigenous populations from their homes, to eventually make food that is wasted.[**3**] Apart from that, all the steps required to produce food e.g. transportation, the growing of food, packaging and its production are said to have a carbon footprint of **3.3 billion** tonnes of carbon dioxide each year.[**3**]

2.2 Our Solution: Sherfood

As mentioned above food waste is a global crisis, it would be difficult to try and eradicate this problem entirely, but we believe that our web application will help users to be more conscious of how they consume food. Our application allows users to create accounts and adverts/announcements of the food they have available, with detailed descriptions of their food group and food allergens. Users who wish to collect food can do so by inserting their address on our search page this will then be converted into coordinates with the use of a geolocation API, the back end logic will then check the SQL database and the user will be presented with a set of results within a 10km radius of their input. The user can then select an announcement that they are interested in and contact the advertiser of the food via a contact form on our site. A nice feature of our web app is that all food items that are out-of-date are periodically removed from our site with the use of an inbuilt function that compares the expiry date of the food and the current date. Once the user is complete with either selecting a food advert or posting one, they can then log out of the session.

3 SPECIFICATIONS AND DESIGN

3.1 Requirements technical and non-technical

In the first week of our project, we got together and created user and admin stories with the use of a Miro board. We all inputted our various ideas for what we thought would be necessary to ensure that we would fulfil our aims/objectives while providing a quality service for all users. We then narrowed down our ideas to features that were necessary to reach our objectives and additional features that could be implemented only if all necessary ones were completed.

Required features included in MVP:

- Users must be able to create an account and log in with a username and password.
- Users must be able to create/post announcements of the food that they have available alongside detailed information, such as the expiry date and food allergens.
- Users should be able to search using their address and find announcements of food within a 10km radius.
- Users should be able to contact the user that is advertising the food.
- The web – application should have a user-friendly interface.
- The front-end and back-end should connect to each other seamlessly and provide the correct information for the requests on the front.
- Login details should be able to be verified via connection with the MYSQL database.
- All usernames and email addresses should be unique for each user.

Additional features for future sprints:

- The user should be able to search via food categories.
- The user should be able to contact another user via a secure chat-bot to inquire about their food announcements/adverts.
- Adding an educational feature – to inform/update users on the importance of reducing food waste.
- Users should be able to see a visual map with pins of users that have food available.

Considerations:

- We needed to consider that an announcement may be on the site for a while and may eventually go out of date so implementing a function that can periodically delete out-of-date announcements was important.
- Another consideration was that users may have various dietary requirements, so this information needs to be clearly stated in all announcements.
- We also needed to ensure that a user's privacy is protected and that they only share information with other users that they are comfortable with, via the contact form.

3.2 Design and architecture

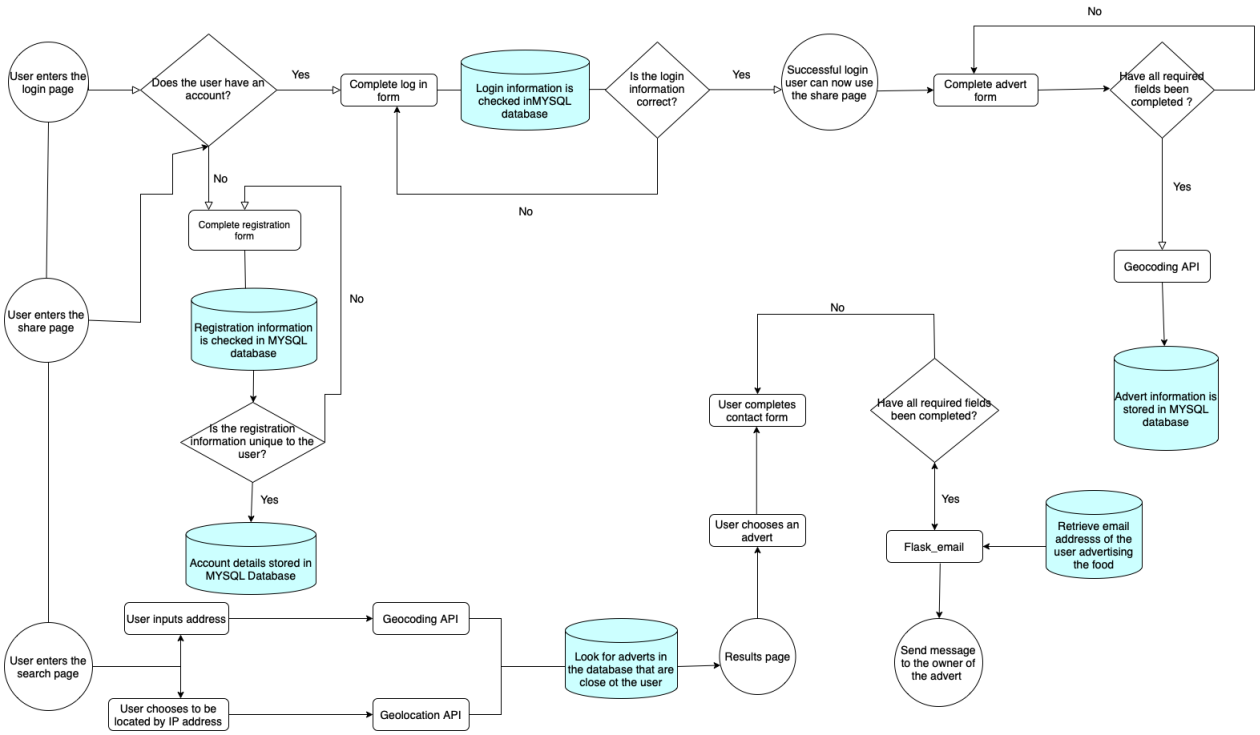

Figure 2: Architecture diagram of the flow of our system

Our web application utilises the features of an **MySQL** database, python on the backend and a frontend implemented by **Flask**. The front end of our application was created with the utilisation of these various tools: **HTML5, SCSS, Bootstrap, JavaScript, and jQuery**.

Our web application works by the user reaching the main page where they can choose to be located via IP address with the use of a **Geolocation API** or by entering their address details (which will be converted into coordinates via the use of a **Geocoding API**). The data from the APIs will then be sent to the functions in the backend, and a query will be made to the MYSQL database so that announcements/adverts that are within a 10km radius of the coordinates received, can be returned. The user on the front end will then be directed to an HTML page with all the relevant results presented to them. The user can then pick an announcement that is of interest to them, and they will be directed to an HTML page that prompts them to fill out a contact form so

that an email can be sent to the owner of the announcement, with the use of the flask-mail module.

Our web application also allows users to create accounts and log into their accounts. All login data is stored in an MYSQL database that is used to verify details for login as well as to ensure that new users have unique email addresses and usernames. Once a user has logged in successfully, they can post announcements about the available food that they have. All the information about their announcement will be stored in a MYSQL database. A diagram of the program flow and the main features of the system architecture is displayed above.

4 IMPLEMENTATION AND EXECUTION

4.1 Development approach and team members roles

We decided to use the Agile methodology and considered this 4-week-process a single sprint, starting with a **Sprint Planning**, ending with a **Sprint Review**. Knowing the importance of effective planning, we decided beforehand to use several tools / methods popular within Agile projects in IT, e.g. **user stories, MoSCoW prioritisation method to decide on MVP and code review**.

Also, we highly valued principles presented in the Agile Manifesto and prioritised frequent interactions over processes and tools. Working software and responding to changes was what we wanted to emphasize.

Team Member Roles

Our team was self-managed and cross-functional. We didn't assign specific roles to our colleagues, each of us had an opportunity to gain experience on every step of our project. We were in contact on slack so we could communicate anytime and ask for guidance.

4.2 Tools and libraries

Tools:

- **GitHub** - we created a repository to keep the project code online, and to make collaboration and version control more efficient. Each team member worked on their own branch.
- **IDE** - MySQL Workbench, PyCharm, VisualStudioCode
- **Geolocation API** was used to identify user's location from their IP address if they prefer not to insert that data manually
- **Geocoding API** was used to convert a user address to coordinates
- **Collaboration tools** - Slack, Zoom, Google Sheets, Miro
- **Wireframing** - Moqups
- **System modeling** - StarUML, Miro
- **Graphic design** - Canva

Libraries:

- **collections** - namedtuple was used for easier data identification where multiple variables were passed to a function as *args; defaultdict used to ensure that the right amount of data is sent to the database even if a variable doesn't exist
- **datetime** - used to receive today's date to check whether an item has expired
- **flask** - used as a framework to connect our back-end logic with front-end
- **flask_mail** - interface to set up SMTP and send messages by a user
- **functools** - wraps used not to override the data of the functions, which are taken as an argument by decorators

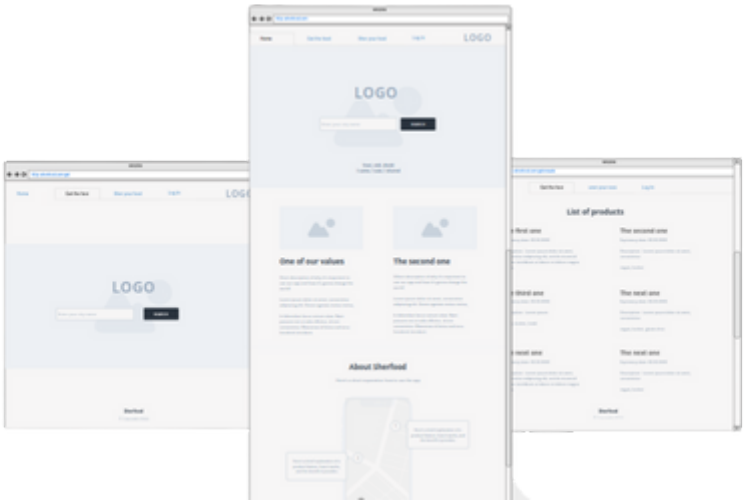

Figure 3: Page wireframe made in Moqups

- **haversine** - estimate the distance between the user and the place where an item can be picked up, based on the coordinates from Geocoding API
- **mysql.connector** - used to connect backend and MySQL database
- **re** - used for data validation to check whether a given data matches the regex expression for an email address
- **requests** - used to send HTTP requests to APIs
- **sys, os** - used to check the parent of the file to allow absolute imports in the flask app
- **unittest** - used for testing individual functions as well as a flask app

4.3 Implementation process

Figure 4: User stories with MoSCoW prioritisation after voting

We started work with the first, initiating meeting. During that meeting, we decided on the subject of our project, its functions and we discussed requirements. We created a **Persona** (person from a big city, in their 20s or 30s, caring about the environment and climate changes) to plan how our future users would use our app, which needs it would help fulfill and what features would be required. Creating and analyzing **user stories** helped us predict crucial aspects of our app. Next, we used the **MoSCoW** method to prioritise gathered requirements and as a result, we decided on the **MVP**. Based on a priority rank, we arranged requirements and divided them into smaller tasks and assigned them to ourselves. Then we created Activity **UML** diagrams, to see how our

functions will work, so we could plan the logic of our code better.

We had meetings 2-3 times a week and at each meeting, we referred to our work, presenting what we had done since the previous meeting, what difficulties we had encountered and what we were planning to work on till the next meeting. We did backlog refinement when needed and helped each other with code review. Progress was updated on branches and tested using unit tests. Every task was integrated one by one, then we created test cases for the application. We were refactoring our code on an ongoing basis to reduce the redundancy, maintain its readability and tried not to leave any technical debt.

At the end of the sprint we've achieved our goal, the app is running and working as intended. However, we have ideas for making a few improvements that were mentioned in the first sprint but not included in our MVP due to lack of time. We plan to implement them in the next sprint. We would like to create a function that allows users to add photos to the announcements, the possibility of rating users and to introduce an internal communication system. The API was a challenge for us, but together we managed to launch the application and test its functions with success. At first, we were thinking about presenting the announcement as a pin on a map, but due to the short time, we changed the idea and decided to focus on a different approach.

4.4 Implementation challenges

In our project, we were using a whole-team approach style, because for us it is important that everyone is equally responsible for the project management, code quality and overall success of the project.

However, we faced with few challenges during our collaboration. For instance, we were having **problems with git**. Especially, we faced a merge conflict which ended up with loosing a lot of the code. Fortunately, we were able to cope with that quickly and this experience gave us better understanding of merge conflicts and commits management.

As well, we experienced some problems with **connecting backend and frontend**. The problem was that not all of us had a good representation how frontend communicate with backend. Due to experience of one member of our team we managed to connect them as well as gain more knowledge on this topic.

Initially, we received an **authentication error** during implementing sending emails feature but with the use of a one-time app password instead of using the password to our company's email address, it was able to work. We also connected to SMTP port 465, which is a port used for communication between the email client and server, where SSL encryption is started automatically.

We also had some problems with **connecting MySQL and Python**. Generally, most of problems with connection to database were related to missing some flask packages, environment problems or wrong interpreter. All those problems were solved by team cooperation and using Stack Overflow.

5 TESTING AND EVALUATION

5.1 Testing strategy

As our code has been written, it is essential to test it because we all can make mistakes and testing will ensure the quality of the project. Of course, our primary objective is offering the best satisfaction of the users, that's why testing is crucial aspect of our software project.

During implementing all features, on each stage of our project we were manually testing our web application e.g. simply print statements each return statement as well as testing different inputs in newly created features, functions and etc.

One of most used and important testings in our project was **unit testing**.We find it useful and efficient because it isolates written code to test and determine if it works as intended. Functionality of each feature was tested individually and independently scrutinized for proper operation.

5.2 Functional and user testing

We used **functional testing** in our project. We mainly concentrated on testing main functions and basic usability of the system to check whether a user can navigate through the web application without any problems. We also checked the usage for error conditions whether the user will get suitable error messages when errors occur.

What is about **user testing**, we adapted exception handling and flash messaging to help the user understand an error when it occurred. For instance, when a user is trying to log in but his username/password is not in the our database they will see the message "Sorry, username or/and password are invalid. Try again" above the search engine and the user will remain on the login page. Such a method is useful to stay user-friendly and not to show technical information to the user.

Additionally, we used **blackbox testing**. We asked few our acquaintance who do not have any experience in development software to use the web application. We took feedback from them what helped us to improve some features.

5.3 System limitations

There are limitations to the amount of testing we managed to complete, because we had only four weeks to implement the project. Due to the structure of the code (lack of OOP), we found it difficult to test all our features independently.

Other limitation of our system is that we are using local Workbench, not any external server, for example SQLAlchemy , which would streamline our workflow and more efficiently query the data. If we had more time, we could have been able to adopt this.

Our system is also depend on Google API, if something happens with Google Maps API, it will affect our web application as well.

6 CONCLUSION

In conclusion, ultimately our project was a success and we are extremely proud of the outcome, despite various technical issues and time constraints.

Our aim was to create a full stack web application to encourage users to be more conscious of the way that they consume food and we were able to achieve this. In the future we would consider using features, such as a Kanban board to keep track of the development of our tickets.

The skills that we've learnt from this project has given us a unique perspective into what working on a sprint with a team in industry will be like. We are looking forward to using our newly - acquired skills in the future.

Food waste is a globally serious problem, especially when there are people who suffer from hunger. Apart from hunger, it is also a waste of resources, products and work. We can try to change the situation, not only by lobbying for systemic changes, but also by our own little actions. Sharing food is in our opinion something that everyone can do and in addition, it can increase the level of sensitivity to the problem of both hunger and food waste. In Poland millions of tonnes of food is wasted, Sherfood is expected to bring a sense of responsibility among individuals for this global issue. Power to less waste.

References

1. https://www.bbc.com/future/bespoke/follow-the-food/how-to-use-food-waste-for-good/ Date accessed: 5/11/2022

2. https://www.wfpusa.org/articles/60-percent-of-the-worlds-hungry-live-in-just-8-countries-why/ Date accessed: 5/11/2022

3. https://www.trvst.world/waste-recycling/food-waste/why-is-food-waste-a-global-issue/ Date accessed: 5/11/2022