



INFO30005
Web Information Technologies
PROJECT REPORT

Franklin Darmansa 1025392
Patricia Budiman 1012861
Vivian Gao 917035

Workshop: Martin Reinoso Wednesday 1:00 PM

1. INTRODUCTION

**“How might we connect people and event organisers
so that we can reduce food waste on campus?”**

Team Singularity, composed of Vivian Gao, Patricia Budiman, and Franklin Darmansa, aims to address the question: how might we connect people and event organisers so that we can reduce food waste on campus? Our solution, Unifood app is a platform that allows event organisers with excess food to post listings where users are notified of over-catered events based on their location. Unifood encompasses three key functionalities: first, an account management system for general users and event organisers; secondly, a form management system for organisers to post listing; and most importantly, a location-based notification system to increase usability and effectiveness. In the following report of our development process we discuss the background research to our problem; a description of the system; the ideation, design, and development process; the internal architecture of our application; and a discussion of its success.

2. BACKGROUND

2.1. CHOOSING THE PROBLEM: FOOD WASTE

When determining what problem to address for this project, there were two considerations that we kept in mind. First, we wanted to resolve an issue that impacted people in their everyday lives, and we wanted to build a solution that could genuinely solve the problem. Secondly, we wanted to create a product that we envisioned ourselves and our peers using. Our issue, food waste on campuses, was inspired by a challenge posed by the social impact platform, OpenIDEO, “How might we dramatically reduce waste by transforming our relationship with food?”.

2.2. RESEARCH PROCESS

Our research process can be broken down into four stages as we research the extent of food waste and how we, as university students, may be contributing to this problem. First, we aimed to comprehend the scope of food waste and the harmful impacts across the world. Secondly, we wanted to understand the motivations and systems in place that genuinely contributed to this problem. Third, we researched how we are unknowingly contributing to this devastating problem. Finally, we brainstormed the different ways we could tackle this problem. Summarised below are some key insights from our research process that helped shape the scope of our solution

2.2.1. How big of a problem is food waste?

Food waste has become a great issue in recent years attracting the attention of both the public and the government. The Food and Agriculture Organisation (FAO) of the United Nations stated that approximately 1.3 billion tonnes or one third of all the food produced for human consumption is wasted globally each year¹. According to the Foodbank organisation, Australia alone contributed to 7.3 million tonnes of lost or wasted food each year and the major contributors are households, primary productions and manufacturing². However, the amount of food being wasted keeps on increasing with

¹ Food Loss and Food Waste, Food and Agriculture Organization of the United Nations, accessed March 2020, <http://www.fao.org/food-loss-and-food-waste/en/>

² Food Waste Facts, Foodbank, accessed March 2020, <https://www.foodbank.org.au/food-waste-facts-in-australia/?state=vic>

more than \$10 million worth of food wasted in 2019 as reported on the year's Food Wasted Report by Rabobank which is an additional \$1.3 million worth of food compared to the previous year³.

According to Frischmann (2018), the FAO stated that if food waste were a country, it would come in third after the United States and China in the impact of global warming and the 30% of food wasted in the supply chain contributes to 8% of total global greenhouse gas emission⁴. Food which is disposed in a landfill becomes a significant source of greenhouse gases. They produce a large amount of methane which has 28 times the global warming potential of carbon dioxide which contributes greatly to climate change. In Melbourne alone, the yearly food waste is responsible for 2.5 tonnes of greenhouse gas emission each year⁵.

The process of food production requires significant amounts of energy and utilises a number of resources where unfortunately a large amount of food produced goes to waste thus causing wasted resources and harmful gases emitted as part of the process. When food is wasted, the resources used to process, package, transport and store the food is also wasted. According to the Stop Food Waste organisation, FAO reported that food waste directly contributes to food shortages, water stress, unnecessary biodiversity loss and increased gas emissions⁶. This includes the 1.4 billion hectares of land which is equivalent to one-third of the world's agricultural area which is used to grow food along with the water and oil that is used in food production⁷. Moreover, 70% of freshwater in the world is used for agricultural purposes which shows how much water is wasted due to food waste⁸.

2.2.2. Why is food wasted?

It is estimated that a large percentage of food is thrown away due to their irregular shape, size or colour. However, little farms waste these imperfect produce just because of their appearance, most of these "ugly produce" is thrown out by homes and businesses such as supermarkets and restaurants. Taber (2019) reported that homes and businesses generate 7.8 million tons of food waste per year⁹ and FoodWise estimates that 20 to 40% of fruits and vegetables are rejected before they reach supermarket shelves due to how they look.¹⁰

Moreover, customers often over-purchase food than required and their lack of knowledge to store the food properly causes the food to be thrown away. FoodWise estimates that Australians discard up to 20% of the food they purchase and almost 40% of the total rubbish in households is made up of food waste¹¹. A lot of people are unaware both of the amount of food waste and the impacts of the waste. This is worsened by the confusion created due to the unclear expiration dates and labelling by brands and supermarkets resulting in perfectly edible food to be thrown away long before it is expired

³ 2019 Australian's Food Waste Bill, Rabobank, accessed March 2020,

<https://www.rabobank.com.au/media-releases/2019/191126-aussies-annual-food-waste-bill-hits-ten-point-1-billion/>

⁴ Chad Frischmann, "The Climate Impact Of The Food In The Back Of Your Fridge", Washington Post, August 1, 2018,

<https://www.washingtonpost.com/news/theworldpost/wp/2018/07/31/food-waste/>

⁵ Jo Khan, "IPCC says we need to cut food waste to help fight climate change but how do we do it?", ABC News, August 14 2019,

<https://www.abc.net.au/news/science/2019-08-14/food-waste-climate-change-emissions/11399448>

⁶ Roff Smith, "Even Poor Countries End Up Wasting Tons Of Food", National Public Radio, September 28, 2015,

<https://www.npr.org/sections/goatsandsoda/2015/09/28/444188475/even-poor-countries-end-up-wasting-tons-of-food>

⁷ The Environmental Impact of Food Waste, Stop Food Waste, accessed March 2020, <https://stopfoodwaste.ie/resource/the-environmental-impact-of-food-waste>

⁸ Water for Sustainable Food and Agriculture, Food and Agriculture Organization of the United Nations, accessed March 2020, <http://www.fao.org/3/a-i7959e.pdf>

⁹ Sarah Taber, "Farms Are Not Tossing Perfectly Good Produce. You Are.", Washington Post, March 8, 2019,

<https://www.washingtonpost.com/news/posteverything/wp/2019/03/08/feature/farms-arent-tossing-perfectly-good-produce-you-are/>

¹⁰ Food Waste Fast Facts, Food Wise, accessed March 2020, <http://www.foodwise.com.au/foodwaste/food-waste-fast-facts/>

¹¹ Damian Carrington, "One in Three Fish Caught Never Makes It To The Plate -UN Report", The Guardian, July 9 2018,

<https://www.theguardian.com/environment/2018/jul/09/one-in-three-fish-caught-never-makes-it-to-the-plate-un-report>

2.2.3. How are we, as students, contributing to food waste?

The University of Melbourne has identified food wastage on campus as a key concern to achieve sustainability targets for the entire city stating that, universities have a special role and responsibility in confronting the challenges of climate change and environmental stewardship¹². Within the institution, the Sustainable Campus initiative aims to expand on the University of Melbourne's position as a leader in campus sustainability and operates as a part of the Sustainability@Melbourne network¹³. The key concerns the institution aims to address range from green impact, biodiversity, the sustainability of buildings, and recycling & waste; however, the initiative does not directly combat the problem regarding food wastage on campus. With that aspect, the institution is yet to have implemented protocol and strategy to combat waste with catering on campus and amongst the student body. However, the preferred catering provider for university-led events is Blueprint Catering; this service provider enforces sustainability internally through better sourcing practices; by serving appropriate portion sizes and determining accurate catering numbers; and donating surplus food to local charities¹⁴. Therefore, we have shifted our contextual focus upon reducing food wastage on behalf of event organisers spanning the student body.

Current Initiatives at The University of Melbourne

The institution aims to deeply embed sustainability across all its activities: Research, Teaching & Learning, Governance, Operations and Engagement¹. With regard to operations, The University of Melbourne Student Union currently implements a number of services towards sustainability targets such as the Choose to Reuse Plate Program which provides a convenient waste-free option when purchasing food and drink from Union House, Parkville Campus¹⁵. However, the extent of policy regarding the reduction of food wastage for event organisers is limited to the following guidelines¹⁶:

- Send reminders to attendees prior to the event to ensure they RSVP/or cancel, ensuring accurate catering numbers.
- Under-cater events.
- Select food options that are easy to take away in containers.
- Find nearby students!

Upon further discussion with event organisers within the student body, they have identified difficulty with determining the suitable amount of food to cater to their expected attendance. An event organiser on behalf of the Healthy Eating and Nutrition Society has stated that they opt for a combination of perishable and non-perishable catering options and encourage the consumption of the perishable food first¹⁷. Another representative from Marketing Intelligence club has stated that the organisers do not have established contingency plans for excess food and instead offers to offload the leftovers to other students either in attendance or externally¹⁸.

As the university itself has addressed, the capacity to locate nearby students is a key contingency to limit the amount of food wastage for event organisers⁵. Within our analysis, we have determined that a platform to notify nearby students of excess food would be an effective solution. Not only is this a point of concern for the University of Melbourne but this would also reduce the amount of edible waste on campus, ease of organisation for event organisers, and sustainability within Melbourne.

¹² "Sustainable Campus," Sustainability at The University of Melbourne, accessed March 2020, <https://sustainablecampus.unimelb.edu.au/>

¹³ "Sustainability for Melbourne," City of Melbourne, accessed March 2020, <https://www.melbourne.vic.gov.au/about-melbourne/sustainability/Pages/sustainability-for-melbourne.aspx>.

¹⁴ "Sustainable Catering," Blueprint Catering, published 2010, <http://www.blueprintcatering.com.au/>.

¹⁵ "Choose to Reuse at Union House", Sustainable Campus, published 22 July 2019, <https://sustainablecampus.unimelb.edu.au/old-site/news/news/choose-to-reuse-at-union-house>.

¹⁶ "Frequently Asked Questions", Sustainable Campus, accessed March 2020, <https://sustainablecampus.unimelb.edu.au/a-z/f/faq>.

¹⁷ "Healthy Eating and Nutrition Society", Facebook, accessed March 2020, <https://www.facebook.com/HENS.unimelb/>.

¹⁸ "Marketing Intelligence MINT", Facebook, accessed March 2020, <https://www.facebook.com/unimelbmint/>.

2.3. UNIFOOD: THE SOLUTION TO FOOD WASTE ON CAMPUS

It is clear that food wastage is significant across the world and a key issue identified within the City of Melbourne. Our solution aims to address this problem at a local scale - amongst the student body at The University of Melbourne. Individual contributions to food waste comes down to three key reasons: first, over-purchasing food; second, poor systems in place to address this problem; and most importantly, a lack of awareness. We want to address all three of these reasons as the premise for evaluating the success of our application. Our vision is for Unifood to connect student event organisers with excess food to users who may be hungry nearby.

Our solution was inspired by a food wastage solution at MIT called Foodcam¹⁹. In summary, their approach was an automated system where notifications are sent out every time leftovers are placed in front of a camera at a set location. Through their research and experience, it is clear that notifying the student body of excess food is an effective approach to excess food for organisers, however, we aim to improve on their concept. Their solution requires users and event organisers to meet in one fixed location, whereas we aim to utilise location detection to notify surrounding students. Our intention is to make this interface more flexible for event organisers and bring students to them. Additionally, our application aims to make this experience more convenient for students by prioritising users who are closer and not overwhelming them with excess notifications. As for the effectiveness of MIT's Foodcam, it has been very successful, have been running the system for over 10 years and experience high levels of participation from its users. According to the users, just minutes after the notifications were sent out, people were coming for the food. This demonstrates that, on a local scale, food wastage can be solved if we can improve the communication between organisers and users.

2.3.1. System Evaluation: How does Unifood address food waste?

Upon launch, we expect that our app will have a positive impact on the community. Primarily, the app should help reduce food wastage from the university and student body. This impact would be brought about in two ways. Primarily, we intend to help connect hungry people to food from events as the application's primary purpose. Secondly, we want to help raise awareness about the issue of food wastage. Although our team prefers it if there are more users, it is best if organisers planned adequately and are more aware of unnecessary edible waste. This application aims to act as a back-up plan for organisers when they over-plan, or an unforeseen event occurs. Furthermore, we hope that our solution will inspire users to take action to reduce their own food wastage and, to make it easier for people to take part in helping our cause. Just by using the application and getting to the leftovers, people are already involved with tackling the issue.

Aside from the broader intended impact for the community, we had some personal goals for the Unifood application as well. Firstly, we aimed to make the interface as user-friendly as possible. This will allow more people to use our app and further reduce waste. Additionally, we want to make the process of connecting organisers and users as smooth as possible. More specifically, it should not be too difficult for organisers to send out requests that there is food available and for users to respond to them. Overall, our goal is to develop an application that is user-friendly and efficient in connecting people while reducing edible food waste.

3. SYSTEM DESCRIPTION

Unifood works for two target stakeholders, event organisers and users. Event organisers are defined as anyone on campus that organises catered events; these can include clubs, societies and

¹⁹ "MIT's Foodcam automatically alerts people when food is up for grabs", BusinessInsider, published April 20 2016, <https://www.businessinsider.com/mit-makes-a-food-cam-for-its-kitchen-2016-4?r=AU&IR=T>.

individual organisers. Users include anyone that is registered with Unifood and is near campus locations. For event organisers, Unifood functions as a contingency plan whenever they have over-catered their events. Aforementioned in our research process, the University of Melbourne does not have sufficient guidelines for event organisers with excess food, and we want to be the solution to that scenario. On the other hand, many students, and perhaps people in general, are very responsive to opportunities for food! A key reason why good food is wasted and people may still be hungry is that, on a higher level, there is often a disconnect between excess supply and latent demand. In other words, hungry people are not there when there is surplus food.

We have grouped our application into three key functionalities: account management, location-based notifications, and the form management. Outlined in this section are the three critical scenarios demonstrating these functionalities for the two different account types, the user and the event organiser.

- **Account Management:** Allows someone to create and manage their account, separated into User and Organiser. Both User and Organiser account types can sign up, log in, update their account details, and delete their account.
- **Location-Based Notifications:** User accounts can access their dashboard where all new event and food listings are available. On their dashboard, the user can share their current location and be notified of new events that are close to their proximity.
- **Form Management:** Organiser accounts are able to post a new listing with their location to be advertised to all users. In addition, organisers are able to manage their forms by either updating the details or deleting a listing.

3.1. UNIFOOD FUNCTIONALITY

3.1.1. Creating an account for the first time

1. Choose user or organiser account type
2. Sign up with details and then log in
 - a. User sign up: email, first name, last name, username, and password
 - b. Organiser sign up: organisation name, own name, contact number, email, and password
3. Access to user or organiser settings:
 - a. Update their account details
 - b. Delete their account

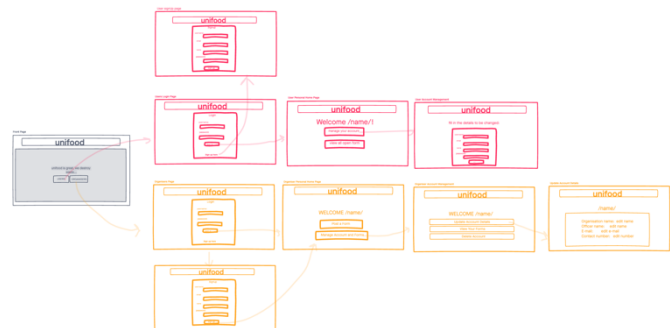


Figure 1. Page Layout for Account Management

User Scenario: Alice, a student at the University of Melbourne, finds out about the Unifood app from a friend who recommended it to connect them with events and be notified of food on campus. Alice decides to open themselves up to more opportunities to meet people at events and is passionate about reducing food waste as well, she decides to sign up with an account to get notifications from event organisers on Unifood. All Alice has to do is sign up and afterwards she has access to all the listings on Unifood and notifications of new events whenever she is on the app.

Organiser Scenario: Bob, events coordinator of a university club, always struggles with estimating how much food he needs for his weekly club BBQ. He doesn't want any club members to go hungry so he either ends up having too little food or having too much food that he needs to dispose of. Bob hates wasting good food and decides to create an organisation account with Unifood so that he has a backup plan in case he has too much. Bob accesses Unifood for the first time and signs up for an organiser account where he now has access to post Unifood listings whenever he has excess food for his BBQs.

3.1.2. Unifood for Users: viewing listings and receiving notifications

1. After a user has logged in, they have to share their location to get notifications of food that is close to them.
2. On the user dashboard they can get live notifications and view all food and event listings across campus.

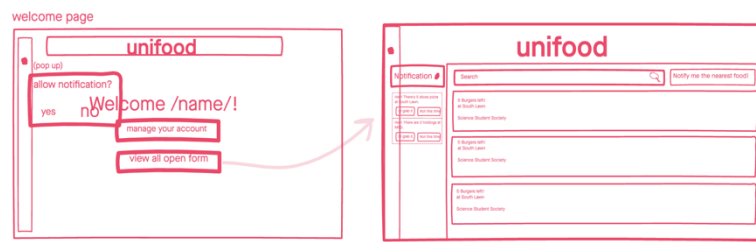


Figure 2. Page Layout for User Notification

3.1.3. Unifood for Organisers: managing forms

1. After an organiser is signed up and logged into their account, they can access their personal home page where they can post a new form or manage existing accounts and forms
 - a. Posting a new form allows them to submit a new listing for an event where there is excess food
 - i. Form details: email, organisation or event name, description, location, share live coordinates, time, quantity of food.
2. The account management page allows the organiser to update their account details, view their existing forms containing previous listings for events and if they wish to, to delete their account.
 - a. their view and edit form page contain all the forms that they have created before and they can edit the details, or delete any of their listings.

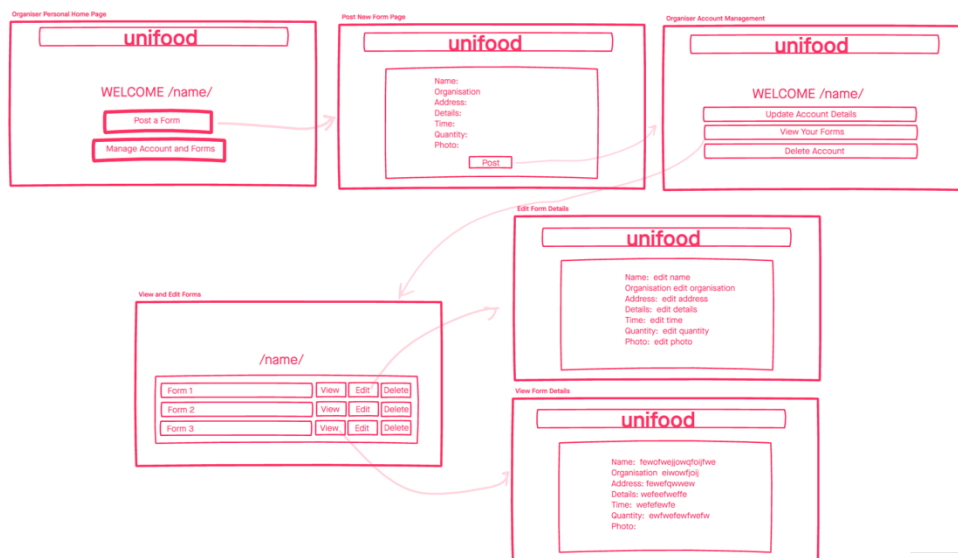


Figure 3. Page Layout for Form Management

4. DESIGN PROCESS

Our design process started with drawing inspiration from similar web applications and brainstorming together for how we wanted our interface to look. Afterwards, we designed a mood board to showcase the colour palette and went through the process of creating a logo, tagline, and additional visuals for the application. The user interface was a key component for the design process

as we wanted the application to be as user-friendly and visually engaging as possible, and we utilised Material-UI²⁰ components to achieve this.

4.1. MOOD BOARD



Figure 4. Unifood Design Mood Board

Displayed below are three key sources of visual inspiration for our design process. A vital component of the success of our application was to make it appealing to students as users. Therefore, we started with food-based applications that we used ourselves, Liven and Zomato. Our red colour palette is drawn from colour theory psychology²¹, suggesting that red is associated with emotion and passion, hence why most food apps and fast-food chains utilise red in their colour palettes. Another appealing component of the design process was to convey the fact that working together to reduce food waste is fun. Thus, we chose to stick with a monochromatic colour palette and bold yet simplistic designs which are key trends in web design²².

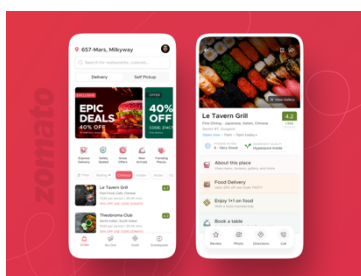


Figure 5. Zomato app by vijay verma

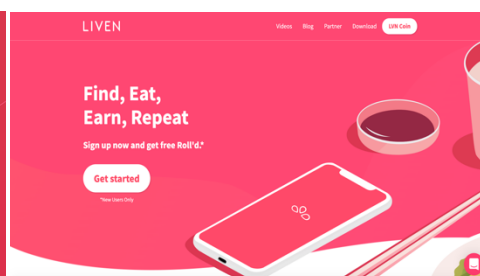


Figure 6. Liven Webpage

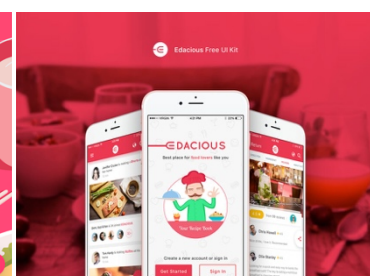


Figure 7. Edacious – Food App UI Kit

²⁰ Material-UI <https://material-ui.com/>

²¹ "Colour Psychology in Food Marketing", AWG Sales Services, published April 21 2016, <https://awgsaleservices.com/2016/04/21/color-psychology-in-food-marketing/>

²² "20 web design trends for 2019", Webflow Blog, published December 13, 2018, <https://webflow.com/blog/20-web-design-trends-for-2019>

4.2. VISUAL DESIGN PROCESS

The main visual design is the banner of the website created using Photoshop and Canva, drawing inspiration from the site of Liven. We utilised a vector for the banner created by Macrovector²³ and designed the Unifood logo using the fonts Arista Pro²⁴ and Glacial Indifference²⁵



Figure 5. Unifood Banner



Figure 6. Macrovector Vector

4.3. USER INTERFACE AND USER EXPERIENCE

A vital component of the front-end design was creating the UI and UX for both desktop and mobile. Although the application is predominantly optimised for desktop, we wanted to create an enjoyable and convenient mobile experience. To assist with the design process, we employed our own peers as test subjects and iterated the design based off of their feedback. Some fundamental changes we implemented from the reviews were first, ensuring that the application is dynamic to increase its appeal. Second, we changed the typography to be bolder and engaging, and third, we added popup alerts with animations.



Figure 7. Unifood Homepage on Mobile and Desktop

²³ "Set of season holidays birth day halloween bbq party thanksgiving and venice carnival isolated illustration Free Vector" by macrovector, Freepik, https://www.freepik.com/free-vector/set-season-holidays-birth-day-halloween-bbq-party-thanksgiving-venice-carnival-isolated-illustration_7251909.htm

²⁴ "Arist Pro Font", 10001fonts.com, <https://www.1001fonts.com/arista-pro-font.html>

²⁵ "Glacial Indifference", Fontquirrel, <https://www.fontquirrel.com/fonts/glacial-indifference>

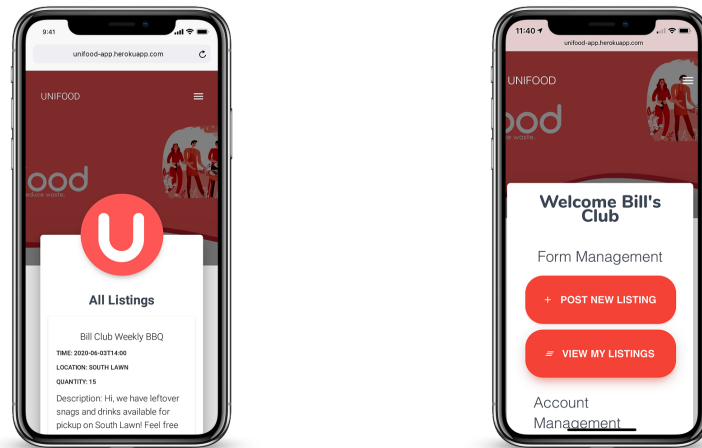


Figure 8. Listings Page and Organiser Form Management on Mobile

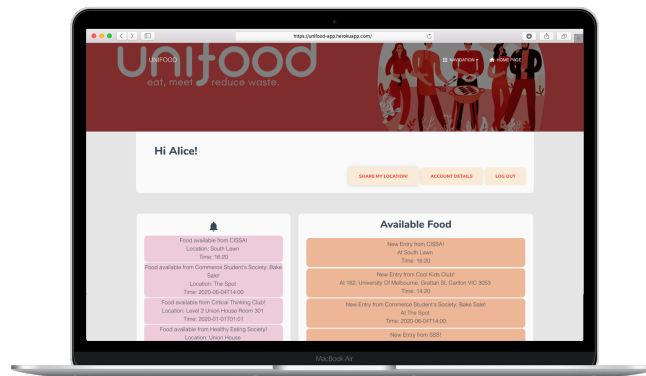


Figure 9. User Dashboard on Desktop

5. DEVELOPMENT PROCESS

Team Singularity consists of three web developers, Patricia, Franklin, and Vivian; throughout the entire development process, we divided the work evenly amongst us based on availability and skillset. Although none of us have previous experience with web development nor JavaScript, we were able to leverage our existing computer science, programming, and design abilities to create Unifood.

5.1. TOOLS AND FRAMEWORK

We used several tools and frameworks to collaborate together to successfully create Unifood remotely from each other. In terms of internal communication processes, Team Singularity utilised Zoom and Discord for conference calling as well as Messenger as the preferred form of contact. In addition, we employed Google Drive to collaborate on the report and a document to track our individual progress and weekly task allocation. In terms of the repository, we utilised Git for version control by hosting the code on Github and by working on our own respective branches using our preferred JavaScript IDE, WebStorm. Our website host is Heroku, and our database is hosted on MongoDB, team members also used CodeSandbox for prototyping the front-end interface and tested the backend API using Advanced REST Client.

5.2. INTERNAL DEVELOPMENT PROCESS

Team Singularity has a weekly meeting on Tuesday over a conference call where we gave an update on our overall progress and planned the tasks we had to complete for the week. After determining which tasks needed to be completed to stay on schedule, we allocated the work amongst us as outlined below in the task allocation section. We also had a weekly checkup on Fridays where we reported our progress and all team members were highly responsive when another member needed assistance. Additionally, we managed the Git repository where we worked our own branches and would update each other whenever we completed a component and pushed our branch to the master.

5.2.1. Task Allocation

1. **Project Proposal:** Patricia completed the background research on food wastage, Franklin completed the proposal outline, and Vivian researched into the context of food waste at The University of Melbourne and how Unifood provides a solution for food waste.
2. **Mockup App Server:** The Singularity team set up MongoDB, Heroku, and the mainframe of the server together. When implementing RESTful routes, we modularised the model-view-controller where Patricia completed the user functionalities, Franklin finished the organiser functionalities, and Vivian was tasked with the form functionalities.
3. **API Server with Front-End:** When creating the client-side component, the entire team worked together to establish the framework and system architecture using React. Vivian focussed on the front-end design by implementing Material-UI components and created the home page as well as the user interface design and form functionalities. Franklin and Patricia worked together to design the user functionalities, including location-based notifications. Patricia was able to successfully deploy the application to Heroku and have the website live for a demonstration.
4. **Report, Live App, and Repository:** For the final deliverable and report, the Singularity team collaborated together, particularly when difficulties occurred. Patricia focussed on the user, Franklin focussed on the organiser, and Vivian concentrated on the forms although, the whole team helped each other throughout the entire process.

6. SYSTEM ARCHITECTURE

The overarching architecture of Unifood is based on the MERN open-source JavaScript software stack for building dynamic web sites and web applications²⁶. Outlined below is a diagram of this system where the data is stored in the cloud-based MongoDB database, and the application is hosted on Heroku, available for mobile and desktop users. To give a higher-level overview of this architecture, our data is stored in MongoDB where it interacts with our back-end through the Mongoose document-oriented data model mapper. The structure of our back-end is composed of Node.js as the runtime environment and the Express.js web application framework where we utilise the Model-View-Controller (MVC) pattern and RESTful routes for executing CRUD actions through HTTP requests. We utilise Axios²⁷ HTTP client to connect our front-end and to execute back-end requests from Node.JS. The foundations of our front-end framework were bootstrapped using React-Bootstrap²⁸ to implement user interfaces using the React library; for the

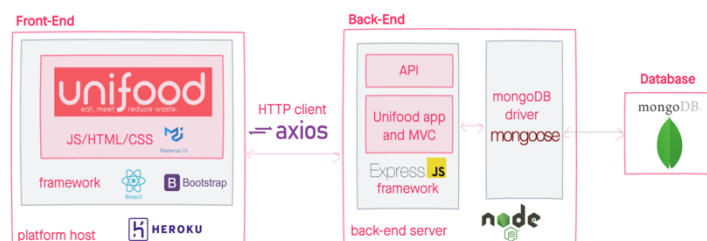


Figure 10. Unifood Architecture

²⁶ "What is MERN Stack?", Medium, published on January 15, 2020, <https://medium.com/@digimktg/what-is-mern-stack-9c867dbad302>

²⁷ Axios, <https://github.com/axios/axios>

²⁸ React-Bootstrap, <https://react-bootstrap.github.io/>

client design, we use components and templates from Material-UI²⁹ on top of HTML, CSS and JS. In addition, our packages are managed through Node Package Manager (NPM).

6.1. BACK-END SERVER

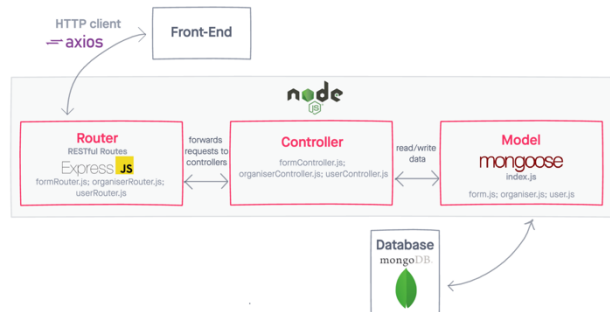


Figure 14. Back-End Architecture

The architecture of our server-side consists of MongoDB, Mongoose, Node.js, Express.js, RESTful routes and MVC pattern. Mongoose is the Object Data Modelling (ODM) library used to manage relationships between data from the schemaless NoSQL database, MongoDB³⁰. We utilise Mongoose to create the models of our data: user, organiser, and form. In addition, we created CRUD functions in the controller component, implemented using RESTful routes with the MVC pattern that permit HTTP requests from our front-end within the router. In the following sections are the three different models, their controller function names, the URL path, the HTTP method used, and a description of the function. In terms of the three distinct functionalities outlined above: account management relates to the User and Organiser models; form management uses the Form model, and the location functionality is implemented in both the Form and User models.

6.1.1. Organiser MCV

Organiser Model

```
const organiserSchema = new mongoose.Schema({
  organisation_name: {type: String, unique: true, require: true},
  officer_name: {type: String, require: true},
  contact_number: {type: String, require: true},
  email: {type: String, require: true, unique: true},
  password: {type: String, require: true}
});
```

Organiser Routes

NAME	PATH	METHOD	FUNCTION
index	/organisers	GET	The root of the organiser management paths
loginOrganiser	/organisers/logon	POST	Validates organiser logon details
organiserPreview	/organisers/update/:id	GET	Gets details of an organiser by id
getOrganisers	/organisers/all	GET	Gets all of the organisers
getOrganiserById	/organisers/:email	GET	Gets organiser details by email
addOrganiser	/organisers/signup	POST	Registers a new organiser account
updateOrganiser	/organisers/update/:id	POST	Updates details of organiser by id
deleteOrganiser	/organisers/delete/:id	GET	Deletes organiser account by id

6.1.2. User MCV

User Model

```
const userSchema = new mongoose.Schema({
  username: { type: String, required: true},
  email: { type: String, required: true},
  password: {type: String, required: true},
  first_name: { type: String, required: true},
  last_name: { type: String, required: true}
}, {collection: 'users'});
```

User Routes

NAME	PATH	METHOD	FUNCTION
index	/users	GET	The root of the user management paths
logIn	/users/login	POST	Allows a user to log in
addUser	/users/signup	POST	Allows a user to sign up
getDetails	/users/login/:username	GET	Validates user login details by username
updateUser	/users/login/update/:username	POST	Updates user details by username
deleteUser	/users/delete/:username	GET	Deletes user by username

²⁹ Material-UI, <https://material-ui.com/>

³⁰ "Introduction to Mongoose for MongoDB", freecodecamp, published on 11 January 2018, <https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>

6.1.3. Form MCV

Form Model

```
const formSchema = new mongoose.Schema({
  email: {type: String, required: true},
  name: {type: String, required: true},
  description: {type: String, required: true},
  address: {type: String, required: true},
  time: {type: String, required: true},
  quantity: String,
  photo: String,
  latitude: Number,
  longitude: Number
});
```

Form Routes

NAME	PATH	METHOD	FUNCTION
index	/forms	GET	The root of the form management paths
createForm	/forms/createForm	POST	Creates a new form/event listing
updateForm	/forms/updateForm	POST	Updates an existing form by form ID
deleteForm	/forms/deleteForm	POSTS	Deletes an existing form by form ID
getAllForms	/forms/formList	GET	Displays all forms
getAllFormsByEmail	/forms/formList/:email	GET	Displays all forms with specified email

6.2. FRONT-END CLIENT

The framework of our front-end for Unifood was created using React-Bootstrap and by implementing components and templates from Material-UI. We chose React as our Javascript library for our web framework as it is a highly UI-focused website with dynamic pages through the react-router-dom³¹. This can be exemplified through the [organiser login page](#) which dynamically directs organisers to their dashboard. More so, we implement the react-history tool³² to push data, such as the organiser's name and email, onto their dashboard and form management. These tools allow organisers to seamlessly access their own listings and update their account details (please refer to the repository readme for information on demonstration).

```
import {useHistory} from 'react-router-dom';
...
export default function OrganiserHomePage(props) {
  let history = useHistory();
  ...
  const viewForms = () => {
    let path = '/organisers/forms';
    history.push(path, {id:id, orgName:organisation_name, email_add:email_add});
  }
}
```

We developed the user interface of Unifood upon Material-UI components based off of sample templates³³. These components allowed us to create a more engaging user experience through animated and interactive displays such as the visuals of the home page; the movement in the login screens; and the animation of alerts. In addition, React and Material-UI components are optimised for mobile devices, ensuring a user-friendly experience on all devices. To elaborate on some notable components of Material-UI, the card

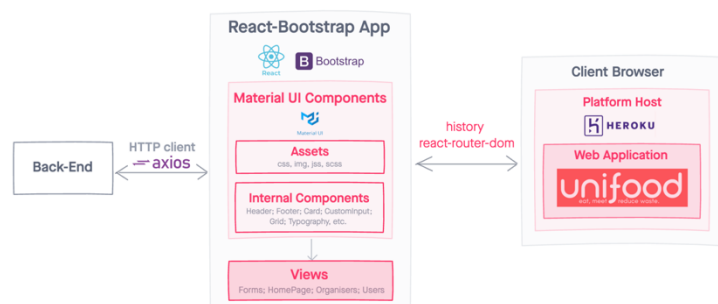


Figure 15. Front-End Architecture

³¹ React-Router-Dom, <https://www.npmjs.com/package/react-router-dom>

³² React-History, <https://www.npmjs.com/package/react-history>

³³ Material-UI Templates, <https://material-ui.com/getting-started/templates/>

component³⁴ (as seen in the log-in pages) adds interaction and allows Unifood to be a more engaging interface. More so, the header of the home page changes colour upon scroll to ensure a consistent and visible experience for users. By implementing Material-UI React components, we were able to effectively produce a highly appealing web application that directly aligns with our system evaluation goals.

6.3. LOCATION-BASED NOTIFICATION FUNCTIONALITY

As the ability to implement location-based notifications for Unifood users was a key component for the success of our application, we shall outline the internal architecture of this functionality.

1. For users, there is a button labelled "Share My Location!"; when this is clicked, there will be a pop-up warning, reminding users that their location will be collected.
 - a. As users agree, the pop up will tell the users that the location sharing succeeds or not. Otherwise, the user's position will not be retrieved.
 - b. Similar to user's location sharing, when creating a form, there is a "Share My Coordinates" button that has the same behaviour as "Share My Location!" button has.
2. The notification works once the user successfully shares their location.
 - a. The system will go through each form/event and get the distance between the user and the events and will only take the events that are distanced less than 500 meters (considerably near) from the user.
 - b. After getting the list of events that are near to the user, the system will only add new listings that are not entered to the notification list, and therefore, the notification list will not be spammed by the same events.
 - c. To keep the notifications real-time, we use socket.io to retrieve the forms every 2.5 seconds from the database.

As the ability to implement location-based notifications for Unifood users was a key component for the success of our application, we shall outline the internal architecture of this functionality. This functionality is displayed on the user's home page where we use CSS for the structure and styling and using some components

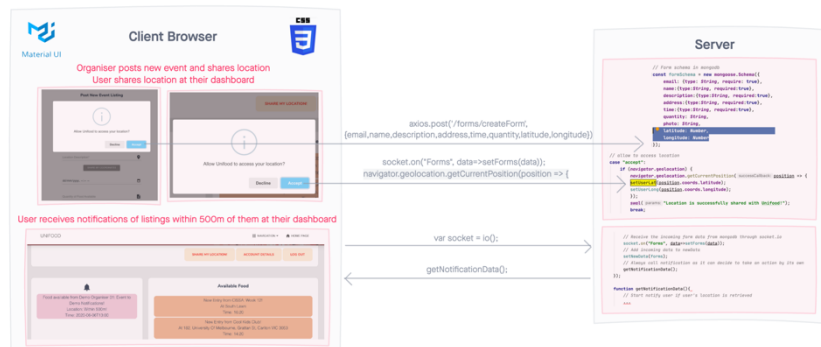


Figure 16. Location-Based Notifications Architecture

will be a pop-up warning, reminding users that their location will be collected. As users agree, the pop up will tell the users that the location sharing succeeds or not. Otherwise, the user's location will not be retrieved. Similar to user's location sharing, when creating a form, there is a "Share My Coordinates" button that has the same behaviour as "Share My Location!" button has. It retrieves location information that consists of latitude and longitude. This information is stored in the form's database at the backend. For users, the location information is only used at that moment to find the nearest events without being saved. However, the forms keep the location information to help users locate the events. The location data is stored with the other details of post information through Axios POST API to send the data to the backend.

³⁴ Material-UI Cards, <https://material-ui.com/components/cards/>


```

function getLocation(){
  // ask for permission
  swal({
    text:"Allow Unifood to access your location?",
    icon:"info",
    buttons:
    {
      cancel:"Decline",
      accept:
      {
        text:"Accept",
        value:"accept"
      },
    },
  }).then((value)=>{
    switch(value){
      // allow to access location
      case "accept":
        if (navigator.geolocation) {
          navigator.geolocation.getCurrentPosition(position => {
            setUserLat(position.coords.latitude);
            setUserLong(position.coords.longitude);
          });
          swal("Location is successfully shared with Unifood!");
          break;
        } else {
          swal("Geolocation is not supported in this browser!");
        }
        break;
      default:
        swal("Location is not shared with Unifood!")
    }
  });
};

```

As the user's location is successfully retrieved, the user will then receive the notifications of the events near them on the left container of the home page, which indicates the notification functionality works successfully. Under the hood of the notification, the system will go through each form/event and get the distance between the user and the events. The system will take the events that are distanced less than 500 meters (near) from the user. After getting the list of events that are near to the user, the system will only add new listings that have not been entered to the notification list. Therefore, the notification list will not have duplicated events. The removed forms/events will be removed on the notification container as well in real-time. To provide a good experience for users in getting notifications, we use Socket.IO library to achieve real-time updates of both the forms and notifications. On the backend, we set up the Socket.IO to listen for a connection (a user logs in or signs up) and will continuously check and send the forms to the client-side. Once the user logs in or signs up, the Socket.IO which has been set up on the client-side sends an event to the server-side signalling a "connection" and this connection ends when the user logs out of their account. Therefore, users could experience the real-time notification in our web application.

```

// using socket to listen income data from server in realtime
var socket = io();

export default function UserDashboard(props) {
  ...

  function getNotificationData(){
    // Start notify user if user's location is retrieved
    if (userLat) {
      // find the nearest leftover from user
      for (let i = 0; i < newData.length; i++) {
        const eventLat = newData[i].latitude;
        const eventLong = newData[i].longitude;
        const distance = getDistance(userLat,userLong,eventLat,eventLong);
        // considerably near (< 0.5km)
        if (distance < 0.5) {
          // initial data for notification
          if (notifyData.length === 0) {
            notifyData.push(newData[i]);
          }
          else {
            // check whether the data already in notification data
            var dataIsInNotif = false;
            var tempNotifyIndex = [];
            for (let j=0; j<notifyData.length; j++) {

```

```

        // data (event) is already notified
        if (newData[i]._id === notifyData[j]._id) {
            dataIsInNotif = true;
            notifyData[j] = newData[i];
            // Event no longer available, delete from notification
            if (!formContainsObject(newData, notifyData[j])){
                tempNotifyIndex.push(j);
                delete notifyData[j];
            }
            break;
        }
    }
}

...

// add new data (event) to notification board
if (!dataIsInNotif){
    notifyData.push(newData[i]);
    setNotifyData(notifyData);
}
}
}
}
}

// get distance using longitude and latitude
// retrieved from https://www.geodatasource.com/developers/javascript
function getDistance(lat1, long1, lat2, long2) {
    if ((lat1 === lat2) && (long1 === long2)) {
        return 0;
    } else {
        var radlat1 = Math.PI * lat1 / 180;
        var radlat2 = Math.PI * lat2 / 180;
        var theta = long1 - long2;
        var radtheta = Math.PI * theta / 180;
        var dist = Math.sin(radlat1) * Math.sin(radlat2) + Math.cos(radlat1) * Math.cos(radlat2) *
            Math.cos(radtheta);
        if (dist > 1) {
            dist = 1;
        }
        dist = Math.acos(dist);
        dist = dist * 180 / Math.PI;
        dist = dist * 60 * 1.1515;
        dist = dist * 1.609344;

        return dist;
    }
}

```

7. DISCUSSION AND CONCLUSION

7.1. EVALUATION

To evaluate the success of our system, it is imperative to restate the problem and critical goals we have outlined in our background research. First, we aimed to address individual contributions to food waste which comes down to three key reasons: first, over-purchasing food; second, inadequate systems in place to address this problem; and most importantly, a lack of awareness. By creating Unifood, we have directly solved the second problem by providing a contingency plan for over-catered event organisers. More so, as a user-friendly and student-oriented application, we believe that Unifood is effective in raising awareness towards food waste as it is a compelling narrative throughout our app. In addition, by raising awareness and providing a contingency for event organisers, we provide a solution to the excessive purchase of food. On these three criteria, we have evaluated Unifood to be successful at addressing the waste of edible food on campus.

Overall, our goal was to develop an application that is user-friendly and efficient in connecting people while reducing edible food waste. In terms of user experience and user interface, we aimed to create an appealing application to attract users. By gaining feedback on usage from our own peers, we were able to iterate through our development process and produce an application that is successful in its appeal to users. Our application is visually impactful while being simplistic in its design, reminiscent of similar popular food applications. Furthermore, the location-based notification feature ensures that users are not overwhelmed with listings and can see events that are within close proximity to them.

Aside from the user component, guaranteeing ease of use for organisers is also key to our success. By creating a simplistic interface and easy access to form management and posting new listings, event organisers will be able to quickly advertise any available food by merely accessing the application, posting a listing, and sharing their location. This will ensure quick response time and integrates seamlessly into the lives of our target users.

7.2. REFLECTION

Although we believe the premise and functionalities of Unifood directly address our initial problem, there are several aspects that can be improved upon. These aspects can be divided into increased mobile optimisation; a greater focus on drawing awareness to food waste; and opportunities for further development. In terms of mobile optimisation, Unifood is a well-functioning application on mobile devices; however, it has been initially designed for desktop users. This means that the Unifood app is not as impactful in mobile mode compared to desktop usage, although we predict that most users will be utilising its functionality on portable devices. Furthermore, we were unfortunately limited in time for the development of this application; with more time available, we would ideally implement more information throughout the app regarding food waste and its impact. Although this is briefly discussed in the home page and about page, our team believes that emphasising this key issue is crucial, and it would be ideal to further convey the impact of food waste to users.

In the future, there are a number of functionalities that Unifood can implement to increase its usability. Primarily, the notification system is currently hosted in-app and can only be accessed through the website. In our initial proposition, we wanted to implement instant notifications through Facebook Messenger, which would more effortlessly weave into students' lives. However, due to the limitations of Facebook Messenger API for commercial developers, we were unable to execute this idea. Furthermore, we believe that the premise of Unifood has the potential to expand beyond merely University of Melbourne students. As Unifood offers a platform for any individuals who have excess food to any individuals who are registered as users, our application can be applied on a grander scale due to its location-based notification system. Unifood has the capacity to serve as a contingency for anyone with extra food, whether it is individuals or perhaps, restaurants. By executing our interface across the city, or even further, Unifood has the potential to contribute to a more significant impact against food waste.

7.3. CONCLUSION

Throughout this project, Team Singularity has executed the web development of a solution to address food waste amongst their peers and on campus. By working together through these steps, from problem formulation to execution, we have successfully addressed the scope and provided a solution with Unifood. We believe that Unifood has the ability to make a positive impact on our student community and reduce our combined edible food waste. Even more so, during this project, the team has fine-tuned their web development skills and fostered an understanding of the fundamental components of web design. Overall, this project has provided Team Singularity with great value, and we hope that our application can give even more value to our student peers, the university, and the greater community in addressing food waste.

We want to thank Martin Reinoso for the direction and support he has offered us throughout this project and we would like to thank Ke Chong for his patience and skills to assist us with developing Unifood.

[5,784 words excluding references]