System Requirements Specification (SRS)

By Runtime Terror

Date: September 2022

1 Table of content

2 Problem Statement

Food wastage has been an incessant global issue for a long time now. It is estimated that one-third of all global foods are wasted annually [1]. With some parts of the world in extreme hunger and some wasting such colossal amounts of food, food management is definitely a necessity in today's world. Especially in Singapore, each Singaporean household is found to contribute an average of 1.5kg of food waste everyday [2]. Furthermore, a study of food wastage behavior in Singapore discovered that about 40% of food wastage happens at the final consumer level and that production level or retail level food wastages were not as major a contributor to food wastage [3]. With knowledge on the extent of food wastage in Singapore and its main contributor, working towards addressing this issue would be very crucial. Singaporeans also share the same sentiment as the above-mentioned study on food wastage behaviour in Singapore also found through surveying that a wide number of participants who expressed a desire to reduce their food wastage in the future hoped to do so by minimizing the discarding of unconsumed food [3]. As such, to aid contributors of food wastage to work towards reducing food wastage in the easiest and most desired way by them, an accessible and efficient food management system would be the perfect solution.

Existing online platforms used for food management are not able to effectively encourage users to prevent the wastage of food incurred by them due to expiration of food. Food management applications need to go beyond simply reminding users to consume their food items before expiry to motivating them to prevent such wastage by providing them a plan or suggestion on different ways they could consume these food items on time. Doing so will reduce users' efforts to prevent wastage and that will increase the likelihood of users engaging in activities to mitigate food wastage due to expiration.

3 Overview

3.1 Background

Food wastage is a dire problem in today’s world. Food waste problems may come from different causes. According to [4], on the customer side, food waste comes from three sources: food scraps, expiring goods, and uneaten food. Our application is intended to solve the problem of expiring and uneaten food. According to [5], S$257 worth of food is wasted per household annually and one-third of the respondents throw away more than 10% of food weekly. Due to our survey conducted with 71 NTU students, 66.2% of students say they are not mindful of the expiry dates, 88.7% students wants to exchange or give their food if they think they cannot finish by the expiry date and 83% students would like to have an application to maintain the expiry dates.

Understanding the need of an application for food item management, Foodify is an all-in-one application that helps users to manage the food expiry date and find better ways to consume, by either consuming themselves, exchanging or giving to others in need. The application will not only help to reduce food resources being wasted but also help to reduce the money wasted on food and the effort to remember and manage them.

3.2 Overall Description

The new online food management system to be created, Foodify. This will be an all-in-one application for food management. The application is available to all users who speak English. To broaden the usage of the application, the team will focus on building an easily accessible, scalable, maintainable and highly user-friendly application. Foodify will have the basic function of tracking and monitoring users’ registered food expiry dates and reminding them accordingly. It will also include additional features such as recipe recommendation to encourage consumption of the food items about to expire, a forum to exchange food items between users in order to prevent it from expiring and going to waste. The team will put in every effort to ensure a well-rounded application that is extremely feature-rich and easy to use with a few taps of the fingers.

4 Investigation & Analysis methodology

4.1 System Investigation

Foodify follows the industry-standard and practices for the entire flow of the application, so that the user can have an enriching experience. When a user accesses the Foodify page, he/she has to log in to his/her account. The authentication is ensured to be completely secure as the sign-in process and authentication is managed by Google servers in the backend. Upon successful authentication, the user is led to the home page. There are three categories on the homepage for the user to choose: My Profile, My Items, and Marketplace.

In My Profile page, users may update personal details like name, phone number, password. These details will be stored secured in Google Firebase that we can retrieve later when needed.

In the My Items page, the user may click on the add button to add a new item and provide the basic information of the new item such as name, expiry date, description to add the new item. The input items will be stored on Google Firebase. A list of existing items with corresponding expiry dates are shown in the My Items page for the user to keep track of the expiry dates of their items. Similarly, the user may choose to delete an item when he/she has consumed the food item. The deleted item will be deleted accordingly in Google Firebase.

Based on the existing expiring food items, the system will recommend some recipes that could be made using these food items. The recommendation logic is handled in the backend by using an external food recipe API, and displays the resulting recipes to the users.

In the Marketplace page, the user may post to sell their items. The post would include post name, quantity, image, description, price, and will be stored on Firebase. These posts will be retrieved to display in other users' marketplace. If the user is interested in an item, he/she could click on the chat button to negotiate with the seller. The buyer may click on the Pay button to pay and the payment is handled by the external third party.

4.2 Analysis Methodology

4.2.1 Feasibility study and requirements elicitation

To make Foodify as feasible and relevant as possible, detailed research and surveys will be carried out. The Back-end team will be tasked to research for the capacity of the cloud database, scalability, the recipes API for food recommendation. The Front-end team will be tasked to test the beta version of our application with some users to have their feedback.

In terms of requirement elicitation, the team will conduct a user survey to understand their habit of keeping track of food items and their need for a food management application. Research on papers and articles are also made to inspect some user needs for the application

4.2.2 System analysis and requirements specification

4.2.2.1 Perform an analysis of the problem using object-oriented techniques

A big picture analysis of Foodify and its different features will be developed using Unified Modeling Language (UML). This System Requirement Specifications document will form part of the documentation for the project. Some of the new features that can be expected in Foodify include:

- A forum where users can post and interact to sell away and minimize wastage of food items
- A item pages to show all items, add items, delete items and recipe recommendation generator
- A methodical reminder system for the registerested food items

4.2.2.2 Scope and Limitations

Analysis methodology will involve skill analysis, budget analysis, requirement analysis, data analysis, process analysis, (web) and application architecture:

- Skill analysis - Check and track application development team's capability to follow deadlines and quality of work
- Budget analysis – Project budget requirement breakdown, Method specification for budget tracking
- Requirement analysis – System I/O description, user requirement definition, functional requirement
- Data analysis – Involve data collection process, data validation, data storage, manipulation and retrieval
- Process analysis – Data/process flow analysis, process decomposition and system interfaces
- Application architecture – Analyze application information structure, usability, user interface design, interaction and application implementation.

4.2.3 Object-oriented design using UML

A detailed object-oriented design for Foodify's user registration or login process will be developed. The graphical representation of that design will be shown using UML. For the registration or login process, an user will key in their unique particulars such as username and password, afterwhich the request will be processed by Firebase which handles the backend process of the system. Upon successful registration or login, data from the Cloud Firestore will be accessed and the system will make all features available to the user allowing them to use Foodify. The system will be secured with a user's username and password.

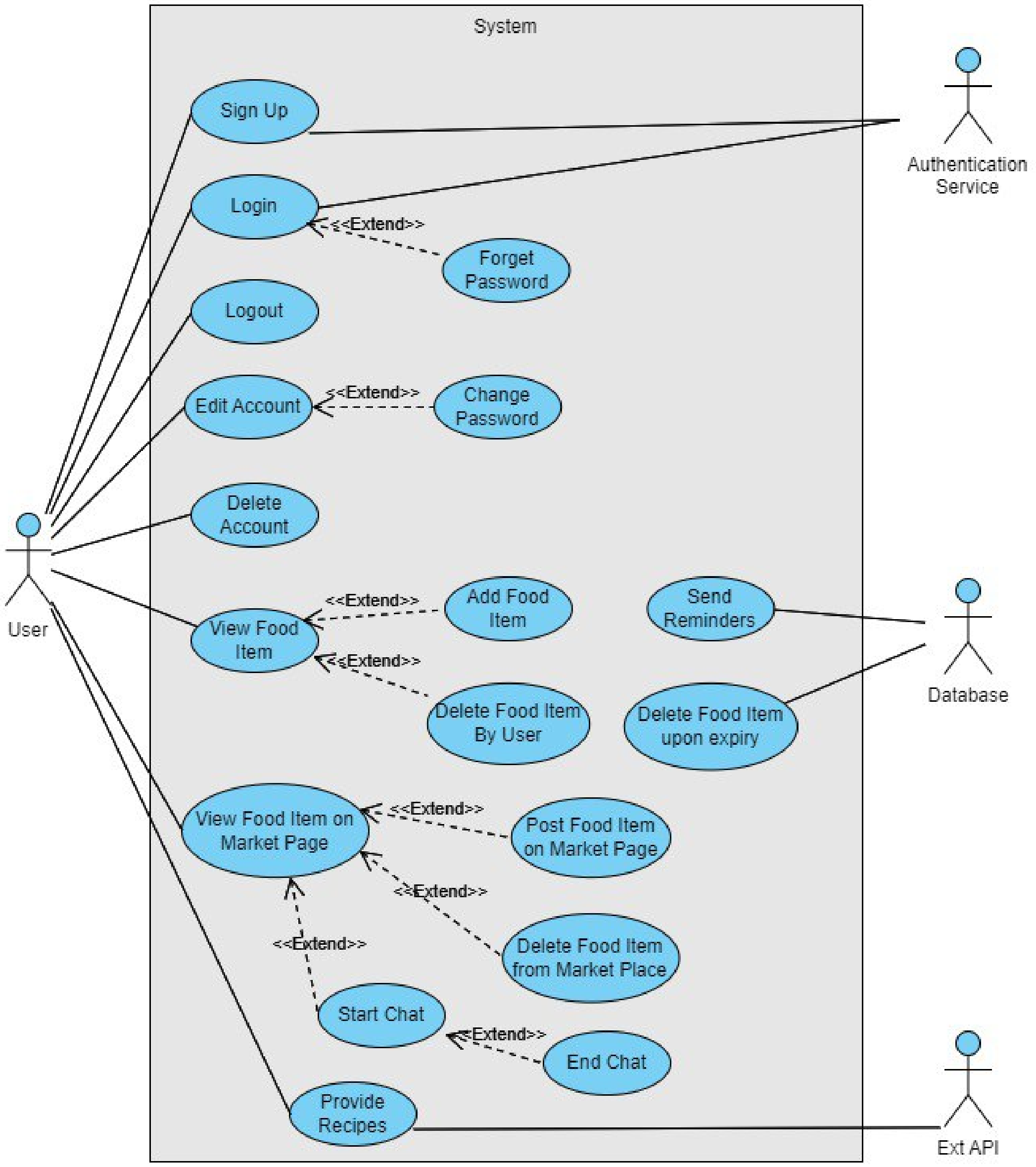
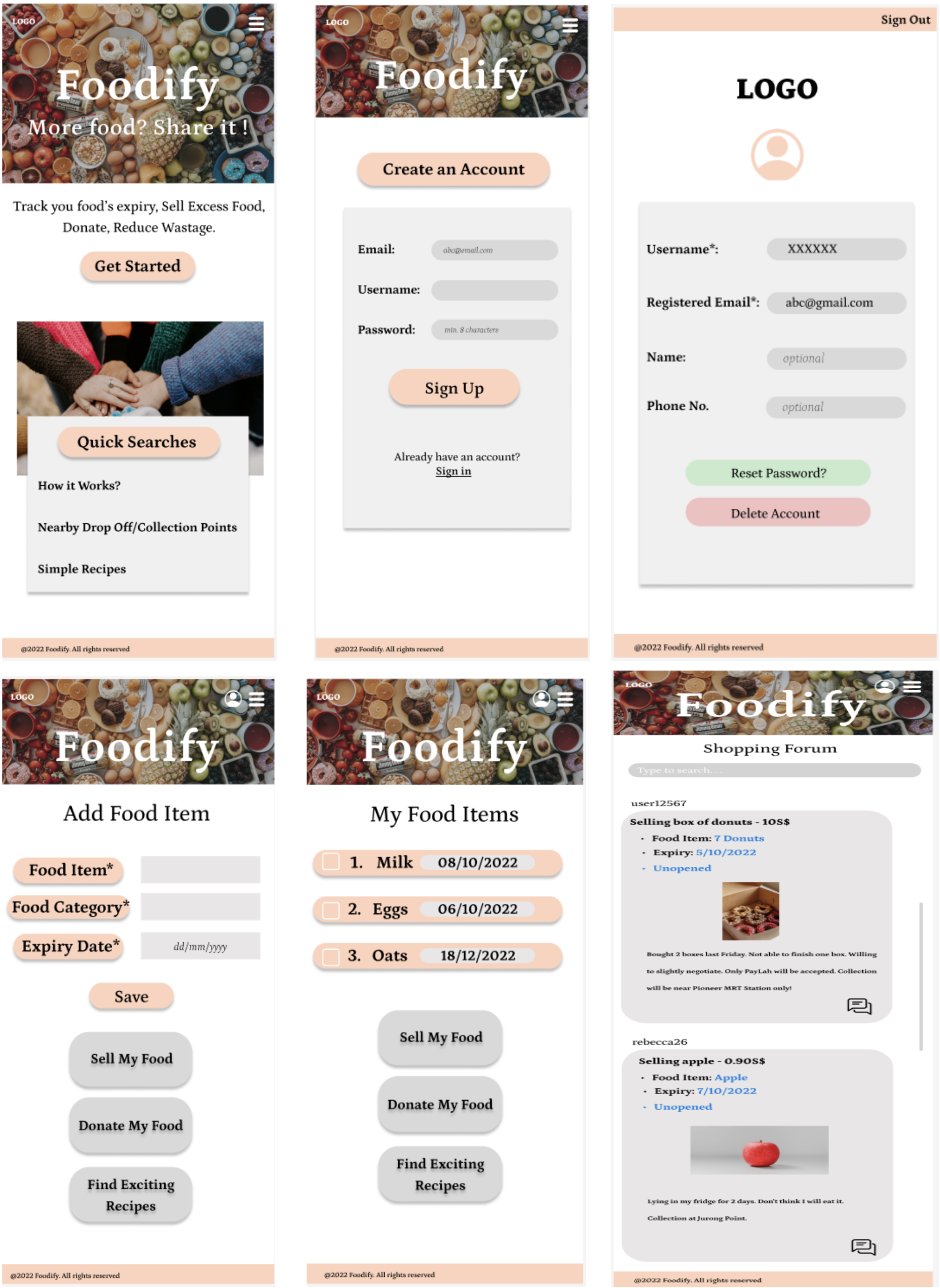


Figure 1: Foodify use case diagram

4.2.4 Prototyping



Track you food's expiry, Sell Excess Food, Donate, Reduce Wastage.

Username*: XXXXXX

Registered Email*: abc@gmail.com

□ 1. Milk 08/10/2022

□ 2. Eggs 06/10/2022

□ 3. Oats 18/12/2022

user12567

Selling box of donuts - 10S$

- **Food Item:** 7 Donuts
- **Expiry:** 5/10/2022
- Unopened

Bought 2 boxes last Friday. Not able to finish one box. Willing to slightly negotiate. Only PayLah will be accepted. Collection will be near Pioneer MRT Station only!

rebecca26

Selling apple - 0.90S$

- **Food Item:** Apple
- **Expiry:** 7/10/2022
- Unopened

Lying in my fridge for 2 days. Don't think I will eat it. Collection at Jurong Point.

Figure 2: Foodify UI mockup

Multiple prototypes of Foodify with the above-mentioned features will be made. These prototypes will be tested in a Chromium Browser through which the web application will run. The prototypes will see how effectively the prototypes run with the use of Firebase. The prototypes will be presented to the sponsors to get their constant feedback on the product design.

5 Constraints

5.1 Scalability

Foodify is a web-based application. The front end of the application is made via ReactJS, which is a JavaScript library for building user interfaces. The back-end of the application is powered by Firebase, a cloud-hosted NoSQL database that is intuitive to use.

On the platform aspect, as our application is a web-based application, people who mainly use smartphones and find retrieving information through web browsers on mobile phones hard may not prefer using our application.

On the storage aspect, Firebase still has some limitations, such as limited querying capabilities. As Firebase is a NoSQL real-time database, traditional relational data models that do not apply to NoSQL will not be able to transfer the data. Therefore, it may be a constraint if you want to migrate between different database systems.

On the customer aspect, as we will mainly develop our application in English, we could not target non-English countries.

5.2 Data and Function Mapping

All data used in the application, such as personal information, food items information, posts, interests, and chat messages are stored in Firebase, a cloud-hosted NoSQL database. Any user's personal information, food item, post, interest, or message can be added remotely from the cloud system. Furthermore, once a user is logged in to Foodify, all permissible data can be readily accessed from any device. A source code change is required if there is a change in the data model such as addition/deletion/updating of new data properties or the way data is displayed in the front-end.

Foodify does not run on a mainframe system. Instead, the backend is deployed on the cloud, with a cluster of low-cost servers, operated by a third-party vendor, Google, spread out globally across different data centers for data redundancy. Any new function added to the cloud system is deployed across all users remotely, whereas any change in a function pertaining to the business logic or front-end requires a change and recompilation of the source-code of the main application.

5.3 Proprietary hardware and software

Modern browsers with access to websites can run the application. We recommend:

- Chrome version 58 or later
- Microsoft Edge version 14 or later
- Firefox version 54 or later
- Safari version 10 or later
- Opera version 55 or later
- Other chromium-based browsers

Older web browsers may experience unexpected behaviors due to unsupported features.

The application requires internet connectivity, either Wi-fi and/or mobile data. The application is currently not available as mobile phone application.

5.4 Batch updates and (close) real-time updates

Regardless of any manual backups, the system is configured to perform a daily backup of the database, assigned to a specific hour each day. The data transferred to our Google Cloud Storage bucket will be time stamped in ISO 8601 standard with the specific naming convention via a scheduled job.

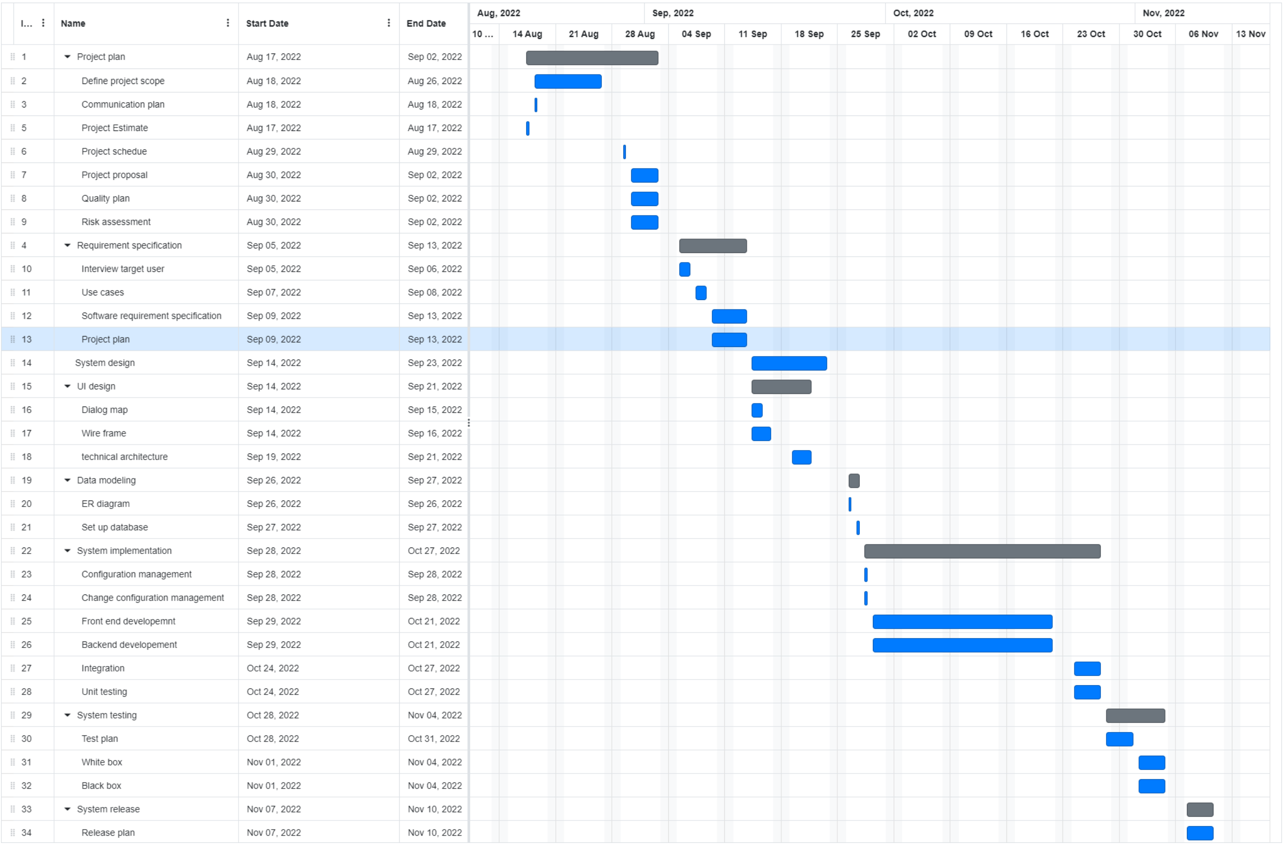
Foodify requires registration of all users for authentication purposes. All registered user data is synchronized via database system on a close to real-time basis.

All user operations, such as adding new food records, posting new food exchange posts will be captured and all data are sent or updated in real-time in our Firebase cloud storage.

5.5 Project Schedule

There is a four-month timeframe to implement a production system for Foodify: a food management system, starting from August 2022 to November 2022.

We use a Waterfall model for our SDLC where we gather customer requirements and come up with a prototype and design our project. Followed by design of use case, then Front-End and Back end Development of the product by the development team based on the use case prototype. Finally, we can come up with planning for change management and product release, and testing of the application by QA before the final release.



I...	Name	Start Date	End Date
1	▾ Project plan	Aug 17, 2022	Sep 02, 2022
2	Define project scope	Aug 18, 2022	Aug 26, 2022
3	Communication plan	Aug 18, 2022	Aug 18, 2022
5	Project Estimate	Aug 17, 2022	Aug 17, 2022
6	Project schedue	Aug 29, 2022	Aug 29, 2022
7	Project proposal	Aug 30, 2022	Sep 02, 2022
8	Quality plan	Aug 30, 2022	Sep 02, 2022
9	Risk assessment	Aug 30, 2022	Sep 02, 2022
4	▾ Requirement specification	Sep 05, 2022	Sep 13, 2022
10	Interview target user	Sep 05, 2022	Sep 06, 2022
11	Use cases	Sep 07, 2022	Sep 08, 2022
12	Software requirement specification	Sep 09, 2022	Sep 13, 2022
13	Project plan	Sep 09, 2022	Sep 13, 2022
14	System design	Sep 14, 2022	Sep 23, 2022
15	▾ UI design	Sep 14, 2022	Sep 21, 2022
16	Dialog map	Sep 14, 2022	Sep 15, 2022
17	Wire frame	Sep 14, 2022	Sep 16, 2022
18	technical architecture	Sep 19, 2022	Sep 21, 2022
19	▾ Data modeling	Sep 26, 2022	Sep 27, 2022
20	ER diagram	Sep 26, 2022	Sep 26, 2022
21	Set up database	Sep 27, 2022	Sep 27, 2022
22	▾ System implementation	Sep 28, 2022	Oct 27, 2022
23	Configuration management	Sep 28, 2022	Sep 28, 2022
24	Change configuration management	Sep 28, 2022	Sep 28, 2022
25	Front end developemnt	Sep 29, 2022	Oct 21, 2022
26	Backend developement	Sep 29, 2022	Oct 21, 2022
27	Integration	Oct 24, 2022	Oct 27, 2022
28	Unit testing	Oct 24, 2022	Oct 27, 2022
29	▾ System testing	Oct 28, 2022	Nov 04, 2022
30	Test plan	Oct 28, 2022	Oct 31, 2022
31	White box	Nov 01, 2022	Nov 04, 2022
32	Black box	Nov 01, 2022	Nov 04, 2022
33	▾ System release	Nov 07, 2022	Nov 10, 2022
34	Release plan	Nov 07, 2022	Nov 10, 2022

Figure 3: Proposed Gantt Chart for Project Development Timeline

6 Operational requirements

6.1 Help desk support

The system must provide email and phone numbers for technical and customer support when in need. Questions sent to the email must be answered by the official support team at the earliest.

There must be moderators for the community functions that help filter the posts, avoid inappropriate post information, and answer questions when raised.

There must be a FAQ section to answer some common questions from users. The FAQ must be updated according to the updates from the application. The FAQ information must be clear and precise for the users to follow.

6.2 Application services and technical support

In community food exchange, users shall report the posts with unappropriate information to admins. Admins must proceed to consider and take appropriate actions.

In the technical aspect, the application must provide contact information for users to report any bugs or technical issues caught when used. The system must periodically provide surveys to ask for users' review of the system and suggestions for improvements. Developers must have access to the source code to address either these or any bugs or enhancements they come across on their own. Any issue with a necessary code fix will be duly recorded and a ticket will be created for the bug fix.

In order to ensure high availability and uptime, the system must have Network Administrator and Database Administrator support Database redundancy and network best practices must be put in place to allow adequate system performance and efficiency. Further, some users shall be selected as beta testers for the new features, which will allow them to find unidentified bugs or issues before major releases.

6.3 Administration features

There must be authorized system administrator(s) who have the right to delete or block any users from using the application should the need arise. The administration must check and filter out long-time inactive users to consider deleting the accounts. The administrators must preview, delete and take other actions depending on the severity of the inappropriate information posts. The administration must also monitor the posts to ensure complete adherence to the code of conduct and communication standards among members. Any members violating the code of conduct and policies will be blocked from using the application.

6.4 System hardware fail over and routine back up

The system data should have a reference on the users’ device for user access in any system down time. A computer operations center must be set up to carry scheduled data back up to avoid data loss and any unexpected system behaviors. The hardware must be serviced every quarter and any quality checks must be performed to ensure quality standards of the equipment.

The cloud provider must have adequate service level agreements to ensure proper uptime of the application. Moreover, complete procedures and guidelines must be followed for data storage in compliance with regional guidelines and data regulations.

6.5 Audit trail

System audit trail must be an inherent part of Foodify. All exchange actions in the system will be captured. The audit trail includes both the user access and modification as well as administration access. Proper Identity Access (IAM) Policies must be in place to ensure complete track of the access and modification history to check any unintended or unauthorized access of the application. Regular monthly audit log checks must be taken, and the audit trail must be verified on a fortnightly basis.

7 Functional Requirement

7.1 User sign up/log in

7.1.1 When users open the application, the system must direct the user to sign up/log in page

7.1.2 When the user chooses sign up option, the system must direct users to input their particulars

7.1.2.1 The registration particulars must include name

7.1.2.2 The registration particulars must include email

7.1.2.2.1 The email must be valid

7.1.2.2.2 The email must not be registered before

7.1.2.2.3 The registration particulars must include password

7.1.3 When user chooses log in option, the system must direct users to input log in information

7.1.3.1 The log in information must include email

7.1.3.2 The log in information must include password

7.1.3.3 The system must verify if the credentials are valid

7.1.3.3.1 If the credentials are valid, the user must be logged into the application and directed to the home page

7.1.3.3.2 If the credentials are valid, the user must be logged into the application and directed to the home page

7.1.4 There must be an option for forgot password

7.1.4.1 The user must be asked to enter their registered email address and a password recovery link must be generated

7.1.4.1.1 If the email address does not exist in the database, a message "No account found" must be displayed.

7.2 User information modification

7.2.1 Users must be able to change their name

7.2.2 Users must be able to change their email address

7.2.3 Users must be able to change their password

7.2.3.1 Users must be asked to enter their old password.

7.2.3.2 Users must be asked to enter their new password.

7.2.3.2.1 If the old password is incorrect, the system must display a pop up "incorrect password"

7.2.4 Users must be able to delete their accounts

7.2.4.1 The system must verify the action of deleting account again with the users.

7.2.5 Users must be able to log out

7.3 Food items management

7.3.1 Users must be able to add food items to their accounts

7.3.1.1 The food item must include the name

7.3.1.2 The food item must include quantity

7.3.1.3 The food item must include the expiry date

7.3.2 The system must notify the users 10 days, 5 days, and every day within 3 days to the expiry date

7.3.3 The system must delete the food items when they pass their expiry date

7.3.4 The user must be able to view existing food items.

7.3.4.1 The existing food items must include the corresponding expiry date.

7.4 Food exchange community

7.4.1 The users must be able to post the information about the food to exchange

7.4.1.1 The food exchange information must include name

7.4.1.2 The food exchange information must include open/unopened status

7.4.1.3 If the checkbox is sell, the food exchange information must include price

7.4.1.4 The food exchange information must include expiry date

7.4.1.5 The food exchange information shall include picture(s)

7.4.2 The users shall chat privately with other users to discuss about food exchange

7.4.3 The users shall post the used items to share with the community

7.4.3.1 The system must provide a drop-down list for the users to choose items from

7.5 Recipes recommendation

7.5.1 The users must be able to enter food items to search

7.5.1.1 The users shall input multiple item

7.5.2 The system must display recipes based on the input food items.

7.5.2.1 The recipe must include name

7.5.2.2 The recipe must include basic information on how to make the meal.

8 Input Requirements

8.1 User registration informtion

While creating account for Foodify, each user should have their email address. The email address is the unique ID for accessing information with respect to the user. The email address should be valid and active in order to use as the user email. The check for a valid email address would be done at the time of registration. A password is also required for the registration. After registering with the email address, the authentication service will be trigger to verify the registration and the user will be able to access the functionality of the application after successfully registered.

8.2 Food item information

To add a new food item, users are asked to enter the food item information, including name of the food item, food item category. Each food product has a corresponding Best Before/ Expiry Date on the side, usually in (DDMMYYYY) or possibly other formats. The user can easily read the date and must enter it into the system to register the Food Item.

8.3 Post of food item

To post another post about food item, user are asked to insert information about post, including post name, food item, open/unopen status, description, image and price.

- The food item is chosen from the current existing items from the user's food items.
- The description is an optional option to describe the current status of the item.
- Image is to be uploaded from the local device. The image is used to show more clearly the status of the food item.
- The price is the selling price from the seller. As currently, our application is developed in Singapore, the price unit is Singapore Dollar.

8.4 Items for recipes searching

The user may insert food items in the search box to search for recipes based on the input items. The input items are treat as the ingredients and the recipes based on the input items are displayed to the user. The user may insert multiple food items, separated by a comma.

9 Process Requirements

The following are among the inherent requirements that the food tracking system must be able to handle.

9.1 Database transaction

A database transaction symbolizes a unit of work performed within a database management system (or similar system) against a database and treated in a coherent and reliable way independent of other transactions.

A transaction generally represents any change in a database. Transactions in a database environment have two main purposes:

- To provide reliable units of work that allow correct recovery from failures and keep a database consistent even in cases of system failure, when execution stops (completely or partially) and many operations upon a database remain uncompleted, with unclear status.
- To provide isolation between programs accessing a database concurrently. If this isolation is not provided, the programs' outcomes are possibly erroneous.

Also, a database transaction, by definition, must be atomic (it must either complete in its entirety or have no effect whatsoever), consistent (it must conform to existing constraints in the database), isolated (it must not affect other transactions) and durable (it must get written to persistent storage).

For our system, the database system must be able to receive, and save information of the user, food items, food item posts. The information must be able to be retrieved or modified by external actions, including both from the users or from the main system.

9.2 Data integrity

Data integrity is the overall accuracy, completeness, and consistency of data. Data integrity also refers to the safety of data in regard to regulatory compliance — such as GDPR compliance — and security.

When the integrity of data is secure, the information stored in a database will remain complete, accurate, and reliable no matter how long it is stored or how often it is accessed. Data integrity also ensures that the data is safe from any outside forces.

There may be some factors that affect the integrity of data stored in the database, including human error, transfer error, bugs and viruses or compromised hardware. Make sure the database are periodically backed up, validated when data are transferred, and kept updated with the latest anti-virus software.

Additionally, in our system, transaction that are completed must be committed to the database. Unfinished or timed-out transactions must be rolled back and handled. Intermediate data security must be maintained

9.3 Data validation

Data validation is an essential part of any data handling task whether one is in the field collecting information, analysing data, or preparing to present data to stakeholders. Hence validating the accuracy, clarity, and details of data is necessary to mitigate any project defects.

In our system, data error from the user’s end and from the back-end database-processing end must be gracefully handled. There must be data validation and error-handling routines as part of the system. Lastly, login credentials must be validated before allowing access to the system

9.4 Performance

The application shall have a responsive look with different phone screen size. Long running operations or requests shall avoid blocking the UI. There should be minimal delays in accessing and to assist the over-all user experience. The system must resolve locking issues and handle concurrent use of the system on a 24x7 basis.

9.5 Data repository

The online registration system will maintain the existing Cloud Firestore database as the main repository of data. All information about user’s personal information, food items and food item posts are to be stored on the database as data repository.

10 Output Requirements

10.1 Exception reports

Each online registration user must have a view of summary of actions done for a particular session or a particular registration function. For example, a status of transaction (successful/unsuccessful) is to be displayed when te user sign up, log in, change password, post food item. A confirmation is needed when the user indicates deleting account to verify again the action.

10.2 Food items list

The food items are viewed as a list, the the name and corresponding expiry on the food item view. From the food item view, the user may further do ther actions, including add food items and delete food items.

10.3 Food item post

All the food item post are summarized with all post information and to be displayed to every user when the user click on the marketplace page.

10.4 Sales Reports and summaries

Foodify administrators must be able to extract summarized and rolled-up data of sales transactions conducted on Foodify into meaningful information. All records will be archived but accessible on demand.

11 Hardware requirements

11.1 Network

A stable and high-speed wireless connection is required for stable functioning of the application. We recommend a minimum speed of 10 Mbps to ensure smooth functioning of the application.

The bandwidth used by our application will be optimized for the best experience based on the participant's network. It will automatically adjust for 3G, 4G or Wi-Fi environments.

11.2 Client computers

	Windows	**Mac**	**Linux**
Operating system	Windows 8 or later	macOS Sierra 10.12 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
Propressor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Memory	2 GB minimum, 4GC recommended		
Screen	1280x1024 or larger		
Internet connection	required		

11.3 Client mobile phones

	Android	**iOS**
Operating system	Version 7.0 or later	Version 11.0 or later
Screen	480x800 or above	
Internet connection	Minimum 2Mb	

12 Software Requirements

12.1 Client Operating Systems

As our application is a web-based application, every operating system that support assessment web application is sufficient to use the application. The common Operating system for computer is Unix, Windows, MacOS and on mobile phone is iOS and Android.

12.2 Client Application

Modern browsers with access to websites can run the application. We recommend:

- Chrome version 58 or later
- Microsoft Edge version 14 or later
- Firefox version 54 or later
- Safari version 10 or later
- Opera version 55 or later
- Other chromium-based browsers

12.3 Network system

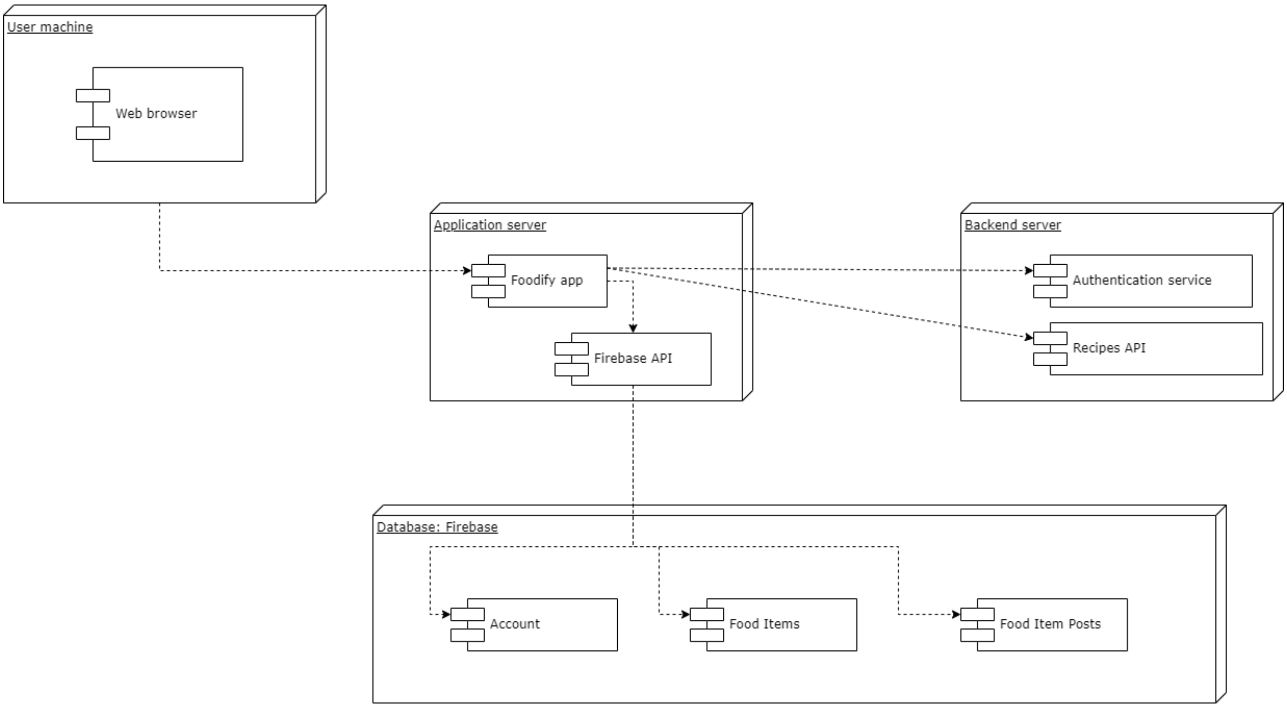
All data regardless of sensitivity, is sent to the Firebase backend system over the HTTPS protocol. Since some of the data sent over the web is sensitive, an encrypted network protocol such as WPA2/AES is highly encouraged to prevent any data leakage.

The following network protocol is used to establish communication with the backend service: TCP/IP (Port 443), HTTPS.

12.4 Licences Valid licences are required to run software from third party vendors:

- GitHub: MIT
- Figma: Organization License
- Google Firebase: Blaze Plan

13 Deployment Requirements



14 Reference

[1] B. J. H. Ng, Y. Mao, C.-L. Chen, R. Rajagopal, and J.-Y. Wang, “Municipal food waste management in Singapore: practices, challenges and recommendations,” *Journal of material cycles and waste management*, vol. 19, no. 1, pp. 560–569, 2015, doi: 10.1007/s10163-015-0405-8.

[2] *Food Waste - Food Waste - Electronic Waste*. (n.d.). Towards Zero Waste. Retrieved September 7, 2022, from https://www.towardszerowaste.gov.sg/foodwaste/

[3] B. Grandhi and J. Appaiah Singh, “What a Waste! A Study of Food Wastage Behavior in Singapore,” *Journal of food products marketing*, vol. 22, no. 4, pp. 471–485, 2016, doi: 10.1080/10454446.2014.885863.

[4] Cesaro, R. (2021, May 21). *Food waste in Singapore [Part 2] - Problems, and solutions - Zero Waste Consultant*. Zero Waste City. Retrieved September 7, 2022, from https://zerowastecity.com/food-waste-in-singapore-part-2-problems-and-solutions/

[5] Yusof, N. (2019, August 6). *Every Singaporean household throws away an average S$258 worth of food a year: Study*. TODAY. Retrieved September 7, 2022, from https://www.todayonline.com/singapore/every-singaporean-household-throws-away-average-s258-worth-food-year-study