Final Report

Multiple Non-Profit Health and Wellness Web Platform
Group C

Client : Barb Marcolin

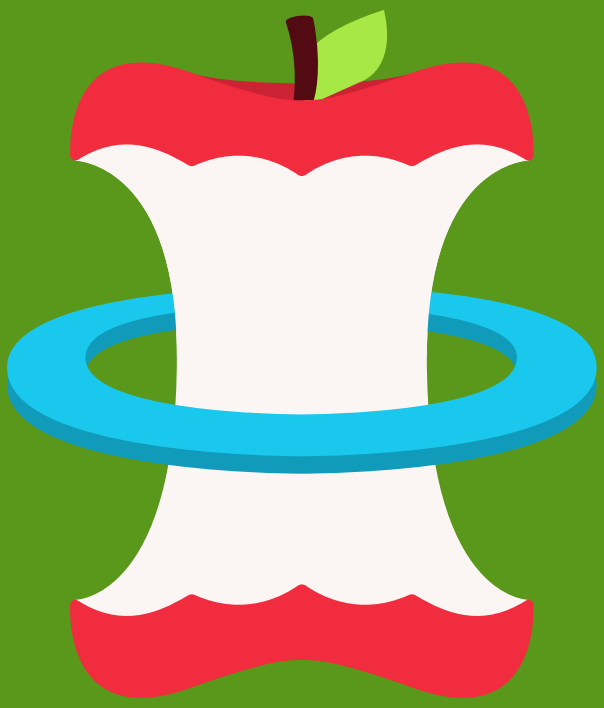
FoodSaviour

Team Members
Anshul Dhariwal
Mitch Hussack
Sydney Fang
Paul Gray

24 April 2023

Table of Contents

Project Description



FoodSaviour

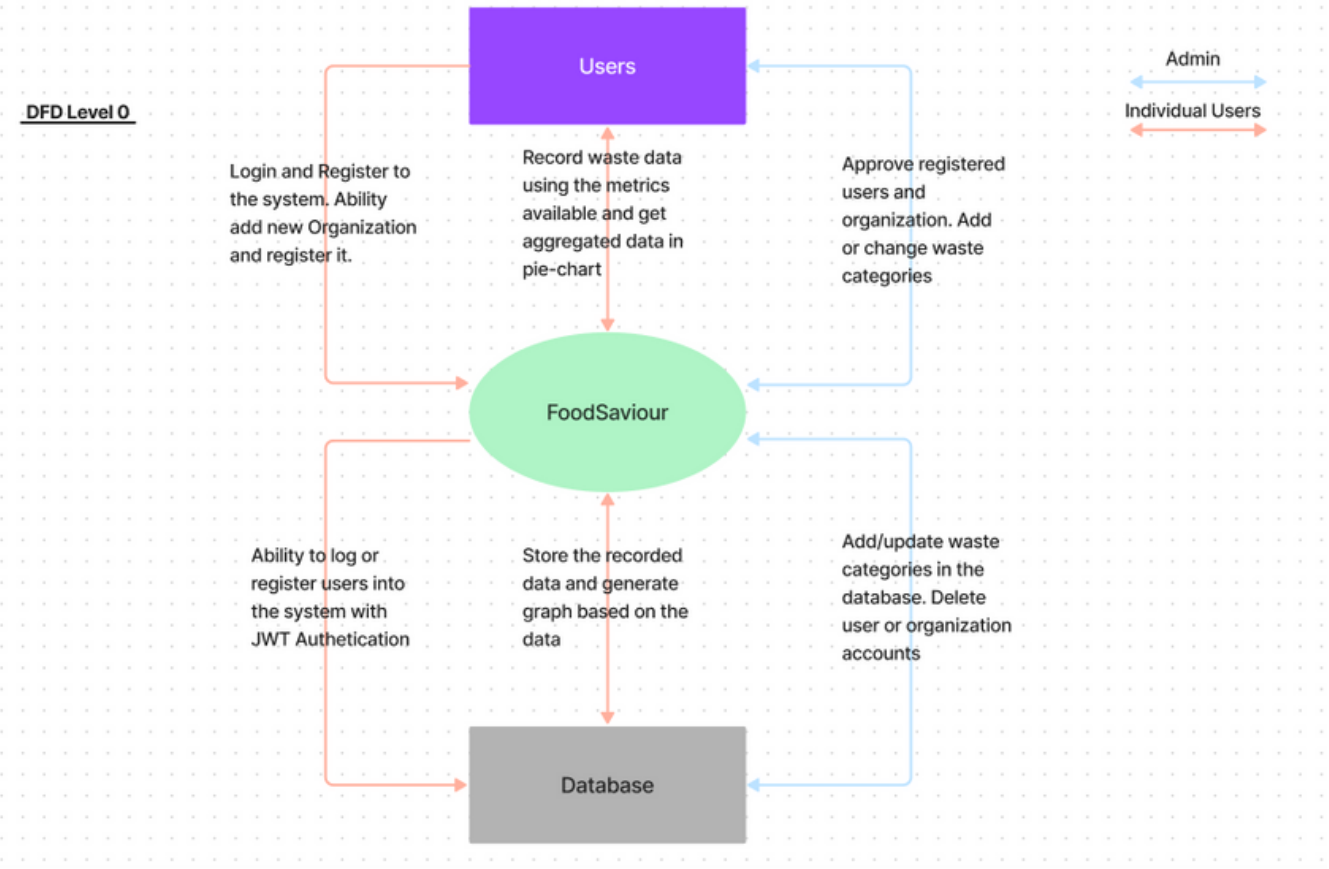
FoodSaviour is an innovative web platform designed to address the pressing issue of food waste in our food systems and industry. The platform provides a user-friendly dashboard that enables individuals and businesses to calculate and avoid food waste, and ultimately reduce the amount of food that ends up in landfills. By leveraging technology, FoodSaviour creates a network of individuals and businesses that can share surplus food and reduce food waste.

Through the platform, individuals and businesses can track their food waste and generate graphs using the graphs. By preventing food waste, FoodSaviour helps to conserve natural resources and reduce greenhouse gas emissions. Overall, FoodSaviour is a powerful tool that empowers individuals and businesses to take action against food waste and build a more sustainable food system.

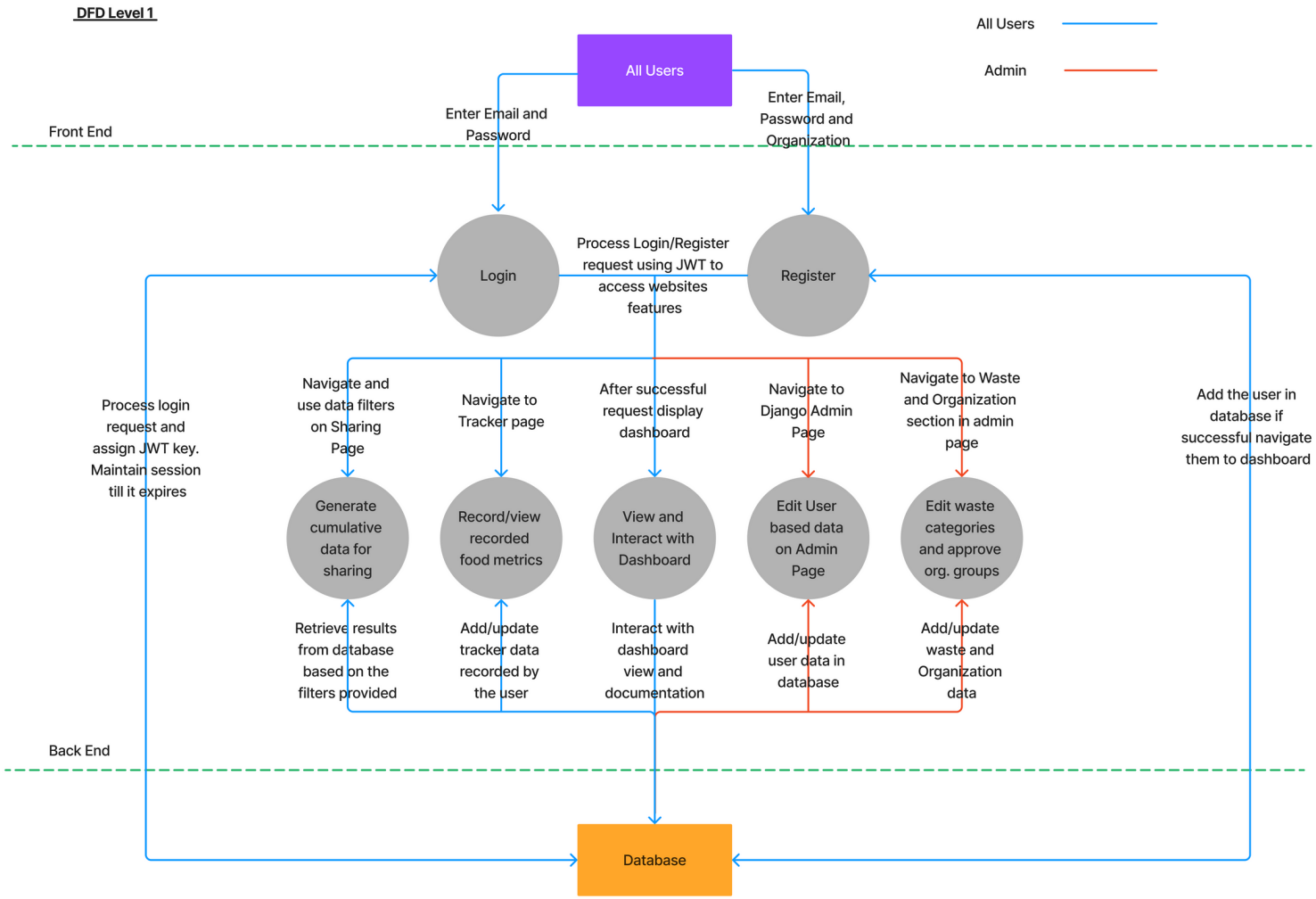
List of **User Groups**

User Groups	Description
Guest/Organization Members	Guest or Organization members will be able to log into the platform or register. They will have access to Tracker Data and Sharing page. Users will be able to record data based on different waste categories. Users will also be able to share their generated data in form of Pie-Chart.
Admin	Admin user will have the authoritative access to the website. This will allow them to add new waste categories, approve/disprove registered users and change Organization

DFD
Level 0


DFD Level 0
Users
Admin
Individual Users
Login and Register to the system. Ability add new Organization and register it.
Record waste data using the metrics available and get aggregated data in pie-chart
Approve registered users and organization. Add or change waste categories
FoodSaviour
Ability to log or register users into the system with JWT Authetication
Store the recorded data and generate graph based on the data
Add/update waste categories in the database. Delete user or organization accounts
Database

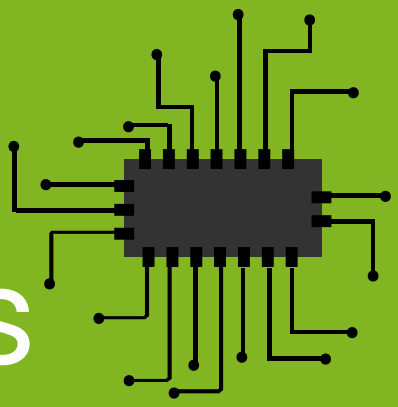
DFD
Level 1


DFD Level 1
All Users
Admin
All Users
Front End
Enter Email and Password
Enter Email, Password and Organization
Login
Process Login/Register request using JWT to access websites features
Register
Process login request and assign JWT key. Maintain session till it expires
Navigate and use data filters on Sharing Page
Navigate to Tracker page
After successful request display dashboard
Navigate to Django Admin Page
Navigate to Waste and Organization section in admin page
Add the user in database if successful navigate them to dashboard
Generate cumulative data for sharing
Record/view recorded food metrics
View and Interact with Dashboard
Edit User based data on Admin Page
Edit waste categories and approve org. groups
Retrieve results from database based on the filters provided
Add/update tracker data recorded by the user
Interact with dashboard view and documentation
Add/update user data in database
Add/update waste and Organization data
Back End
Database

Functional **Requirements**

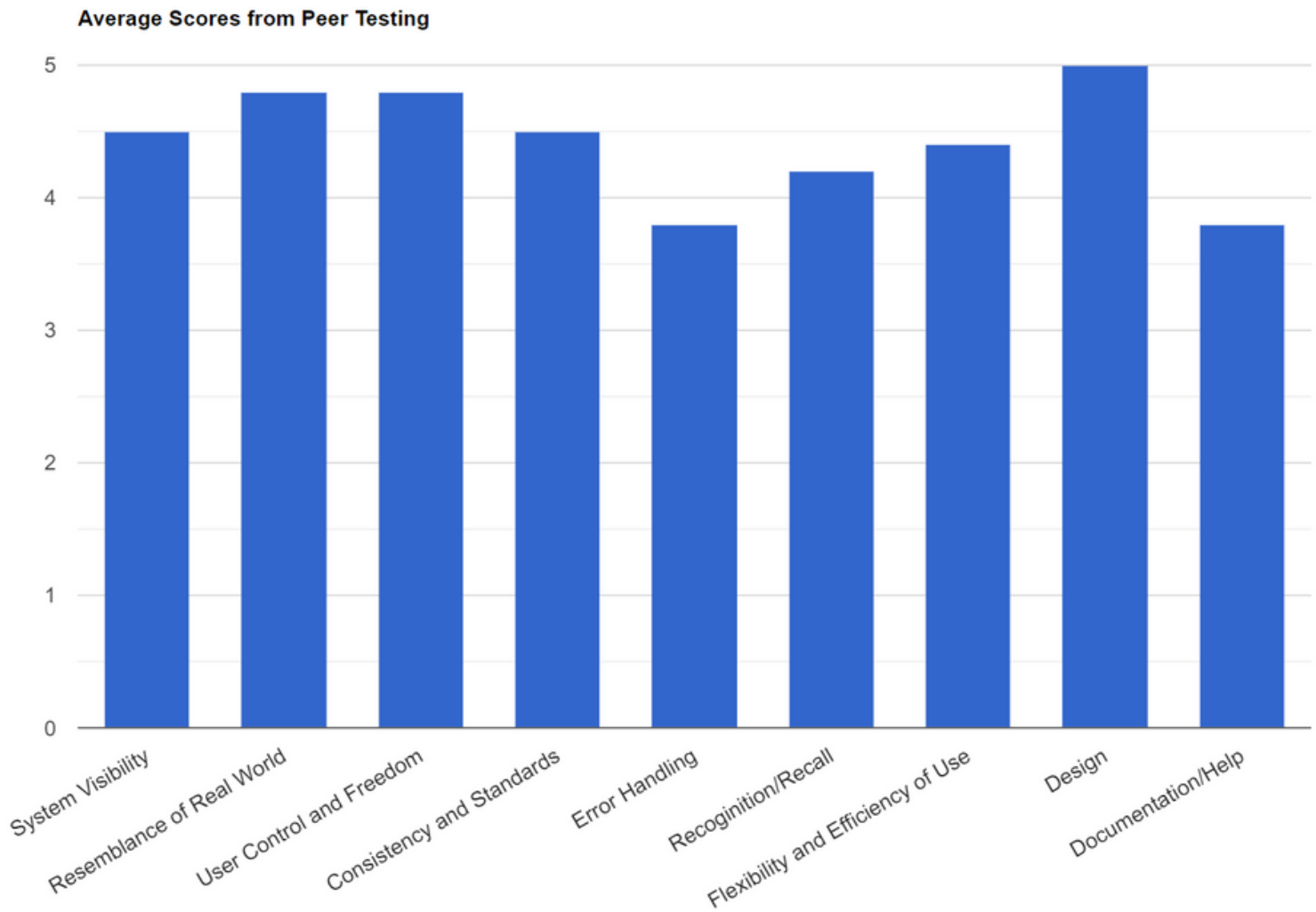
Login/Register Request	All the guest/existing users should be able to Register/Login into the system. While registering a new account User should have the option to join an already existing organization or add a new organisation. The system should be stateless and require JWT for user authentication.
Simplistic Dashboard Page	All the users should be able to access a simplistic, yet effective homepage/dashboard after login. The users from there should be able to navigate to different pages and documentation. Users should be able to access Navbar and process logout request.
Tracker Page	All the users should be able to select the waste category and record data. The data should be recorded through a form which should support form validation. Users should be able to see previously recorded data and edit if nescessary.
Sharing Page	All the users should be able to use filters provided to query cumulative data. This data should be displayed in Pie Chart with clear headings and values labeled beside the chart. The users should be able to export the chart in PNG format which can be shared accordingly.
Admin Page	Admin users should be able to access admin page to add/update necessary fields like User information, Waste Categories and Organization data

Technical Specifications



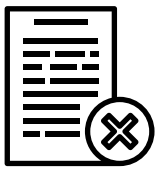
- Programming Languages:
  - Python
  - JavaScript
  - CSS
  - HTML
- Web Frameworks:
  - Django Rest Framework (DRF) - Backend
  - ReactJS - Frontend
  - MDB UI Kit - Styling
  - D3JS - Graphing
- Database Management & Hosting:
  - MySQL
  - Canada Compute

Peer Testing Results



Identified Issues

- Resolved Issues:
  - Improved Organization Registration View
  - The register Link on the Login page fixed
  - Improved form validation and error handling for Tracker Page
  - Fixed the issue of delayed input on Sharing Page
  - Adding error handling on Login and Register Page
  - Added "Documentation" link for further instructions about the website
- Unresolved Issues:
  - No server delay handling on Frontend



Unimplemented Requirements

- User Discussion Board for interacting with other users
- Advanced sharing features for users to share their data with specific user/user group
- Email activation for user accounts
- "Forgot your Password" functionality for the Login Page
- Functionality to filter out tracked metrics based on a particular date
- Known Bugs:
  - JWT crashes after prolonged exposure due to a deprecated React package
  - Label overlap with small values on chart

Step-By-Step **Project Instructions**

- **Where can the client find our code:**
  - All the code, for both the backend and frontend, is available in our GitHub repository. The link to our repository is provided later on with a link to the Prototype Video. The backend and frontend have separate repositories.
  - The repositories are all updated and ready to use.

- **Installation steps:**
  - For testing purposes and monitoring the Database, we recommend using MySQL workbench, an open-source software.
  - Python 3.9 and above
  - ReactJS 18.2 and above
  - Backend installation on VM:
    - Once in our directory on the VM run these commands:
      - cd backend
      - python install -r requirements.txt
      - python manage.py makemigrations
      - python manage.py migrate
      - python manage.py runserver
    - This is should install all the required packages for the backend and deploy the server
  - Frontend Installation on VM:
    - Once in our directory on the VM run these commands:
      - cd food-saviour
      - npm install
      - npm start
    - This should deploy the frontend server.
    - Note - Do make sure that you follow backend instructions first and successfully deploy it.

Additional Information

- The web application was designed to be robust but would still require additional maintenance at regular intervals to make sure everything works fine.
- There are automated tests implemented which can be used by future developers and testers.
- The tests would need updates if there is a major restructuring of the system.
- As mentioned previously some features were left out due to time constraints as listed below:
  - Discussion Board
  - Advanced Sharing Page
  - Email activation & Forgot Password

Project Links

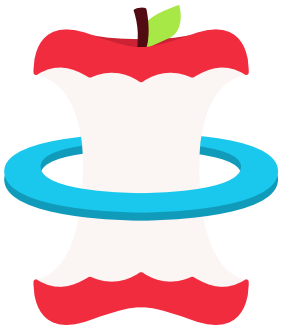
Promo Video

https://drive.google.com/file/d/1NDZqlS2_RkSyxmr9inZ0kW3Kpm1zIYR3/view?pli=1

GitHub Repository

- **Frontend:** https://github.com/COSC-499-Group-C/FoodSaviour-frontend
- **Backend:** https://github.com/COSC-499-Group-C/FoodSaviour-backend

Thank You



- Sydney Fang - Team lead
- Paul Gray - Quality Assurance lead
- Anshul Dhariwal - Technical lead
- Mitch Hussack - Client liaison