

NAME: PRANAV NAIR

COURSE: CSC 415 – 01

ASSIGNMENT 4: OPEN SOURCE SOFTWARE: ANALYSIS AND DESIGN

DATE: APRIL 6<sup>TH</sup> 2018

## 1. DESIGN

### *a. USE CASE DESCRIPTION*

#### USE CASE DIAGRAM

Pranav Nair | April 12, 2018



## IN – DEPTH USE CASE DESCRIPTIONS

### CONSUMER USER

- Primary Actor: User
- Preconditions: User has made a login account.
- Scenario: User logs into the web application and provides feedback. Once the user has provided some feedback, the next step is to logout.
- User Login System

- Normal:
  - Correct Login Details: Successful login to user Interface(Homepage)
- Exception: Login status
  - Incorrect Login Details: Failed login resulting in login again till successful
- User Interface
  - User enters feedback in feedback slot
- Contact Page
  - About user details
    - Name, Phone number, email address etc.

## ORGANIZATION

- Primary Actor: Organization
- Preconditions: Organization must have provided login details
- Scenario: Organization has logged in to enter Organization Interface. Organization receives feedbacks from user and sends it to restaurant. After that the organization logs out.
- Organization Login System
  - Normal:
    - Retrieve user feedback from system
      - Generate system output to be sent to restaurant
  - Exception: If user doesn't provide feedbacks.
- Organization Details
  - Name, phone number, email address etc.

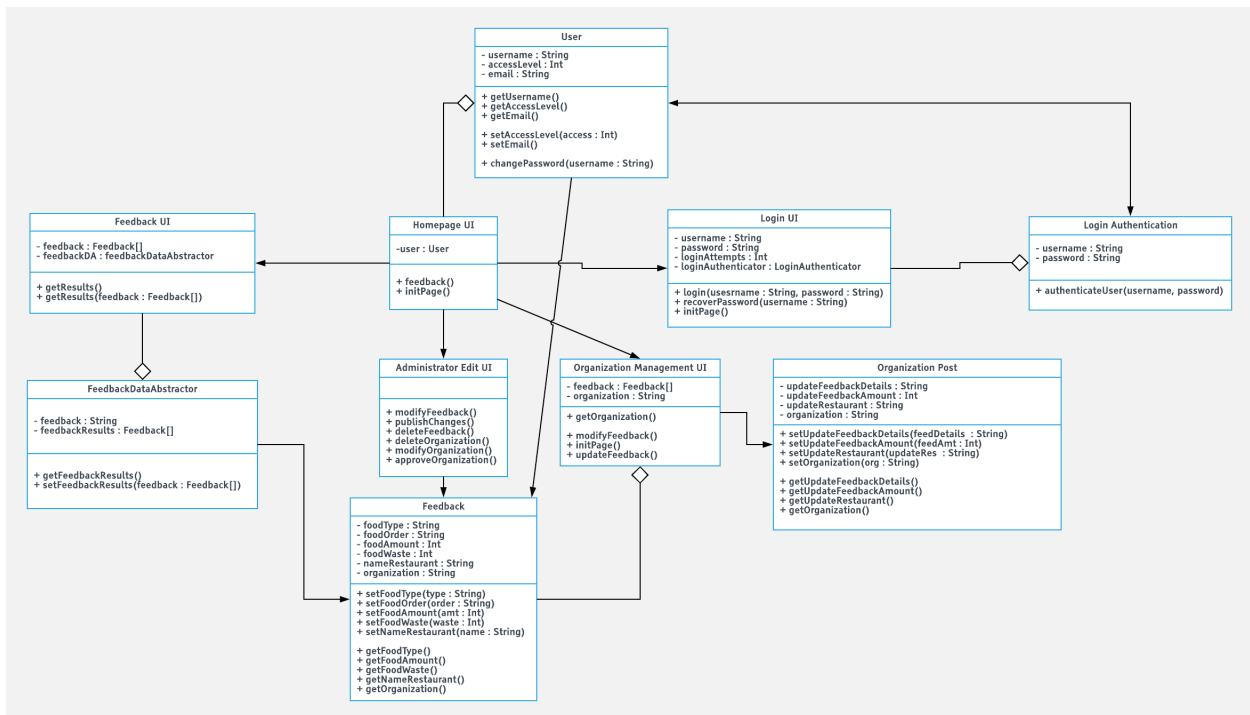
## SYSTEM ADMINISTRATOR

- Primary Actor: System Administrator
- Preconditions: Administrator should be able to open the website and have access to feedbacks and organization applications.
- Scenario: Admin opens web application and receive organizations application and approves it or not. Admin also delete feedbacks if necessary.
- Updates organization page
  - Normal:
    - Modifies organization applications and edits user feedbacks
  - Exception: If user doesn't provide feedbacks or organizations doesn't submit applications
- Perform system backup
  - Backup on database automatically
- Add/ Remove users from the system
  - Potentially automate spam detection

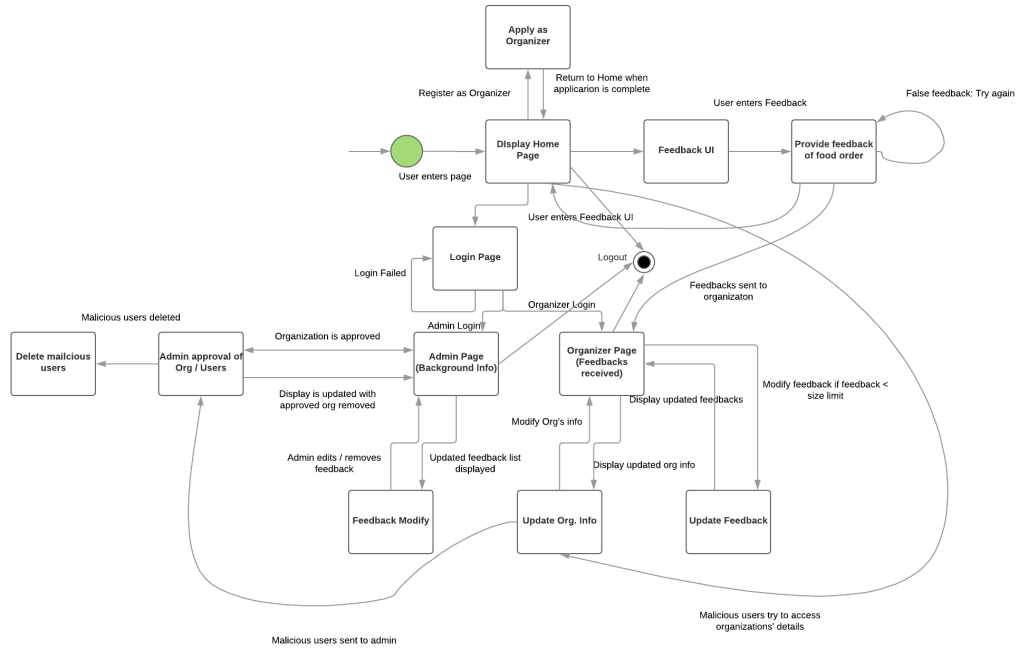
## b. DETAILED DESIGN CLASS DIAGRAM

Class Diagram Link for Better Readability:

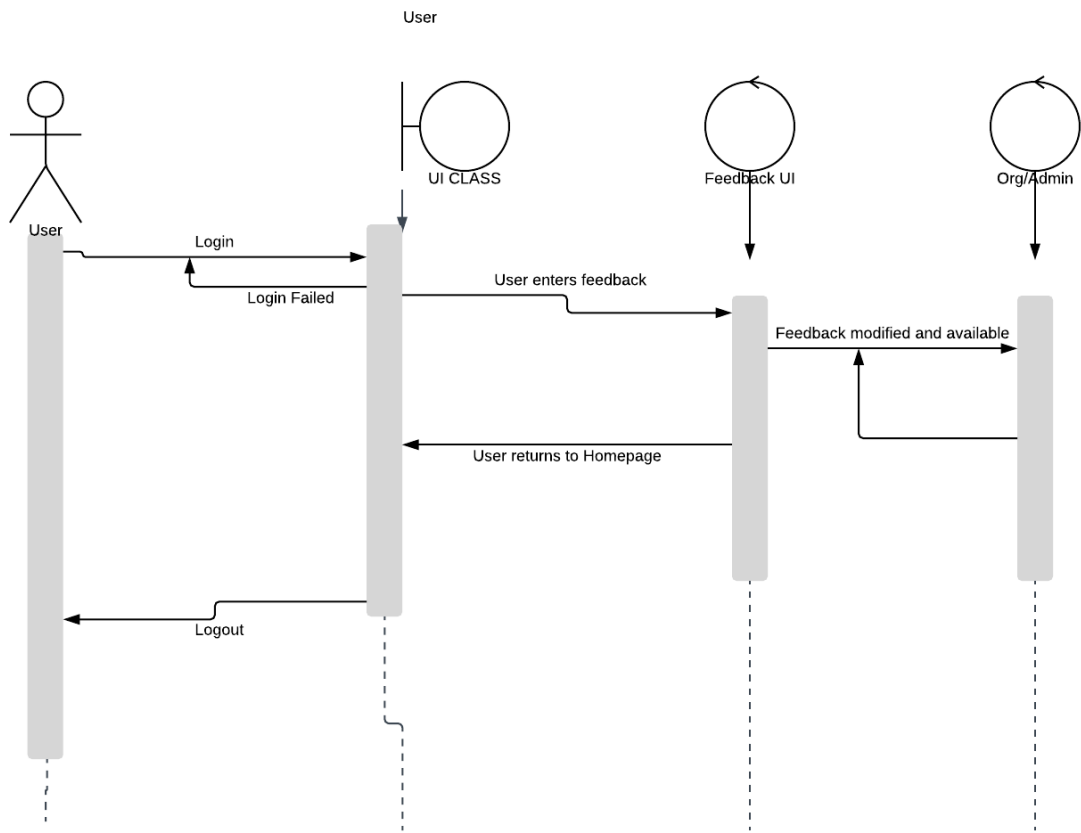
<https://www.lucidchart.com/documents/edit/ed4b8ee9-da59-42fb-9d28-4537e7207414/0>

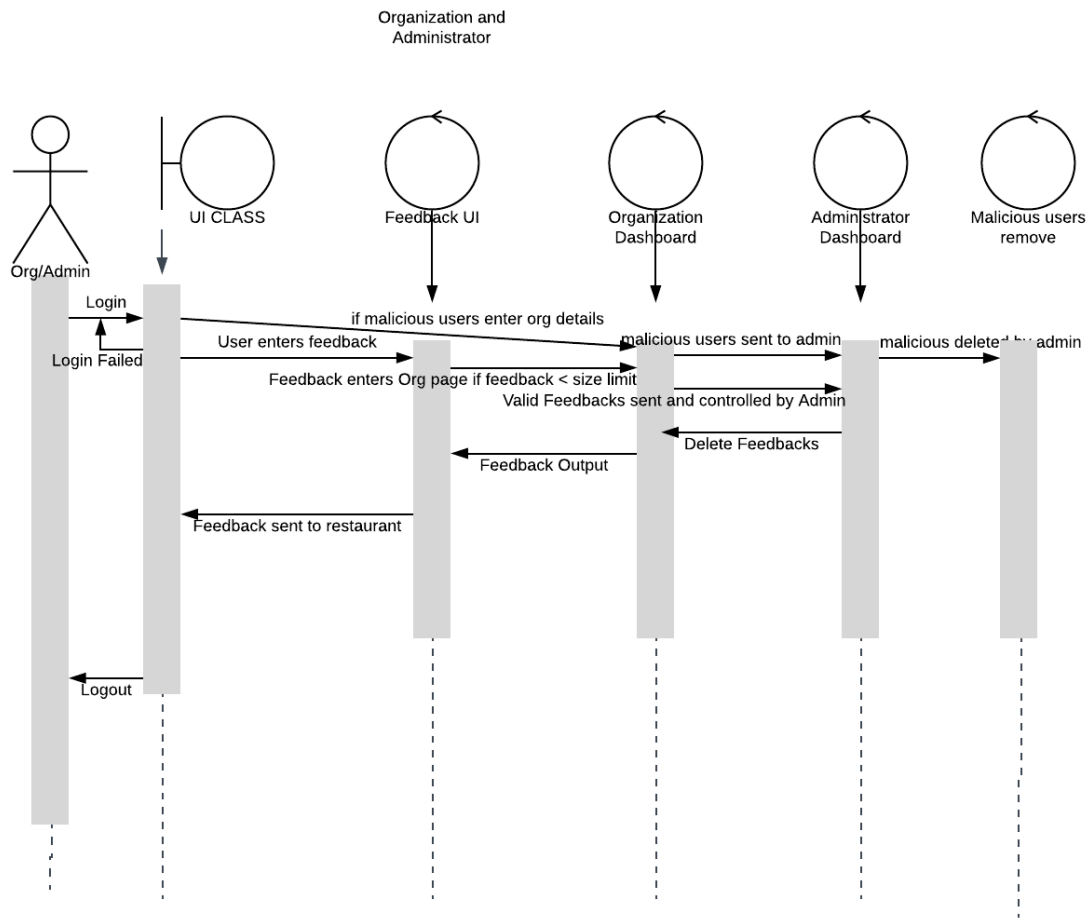


## c. STATE CHART



#### d. SYSTEM SEQUENCE DIAGRAM

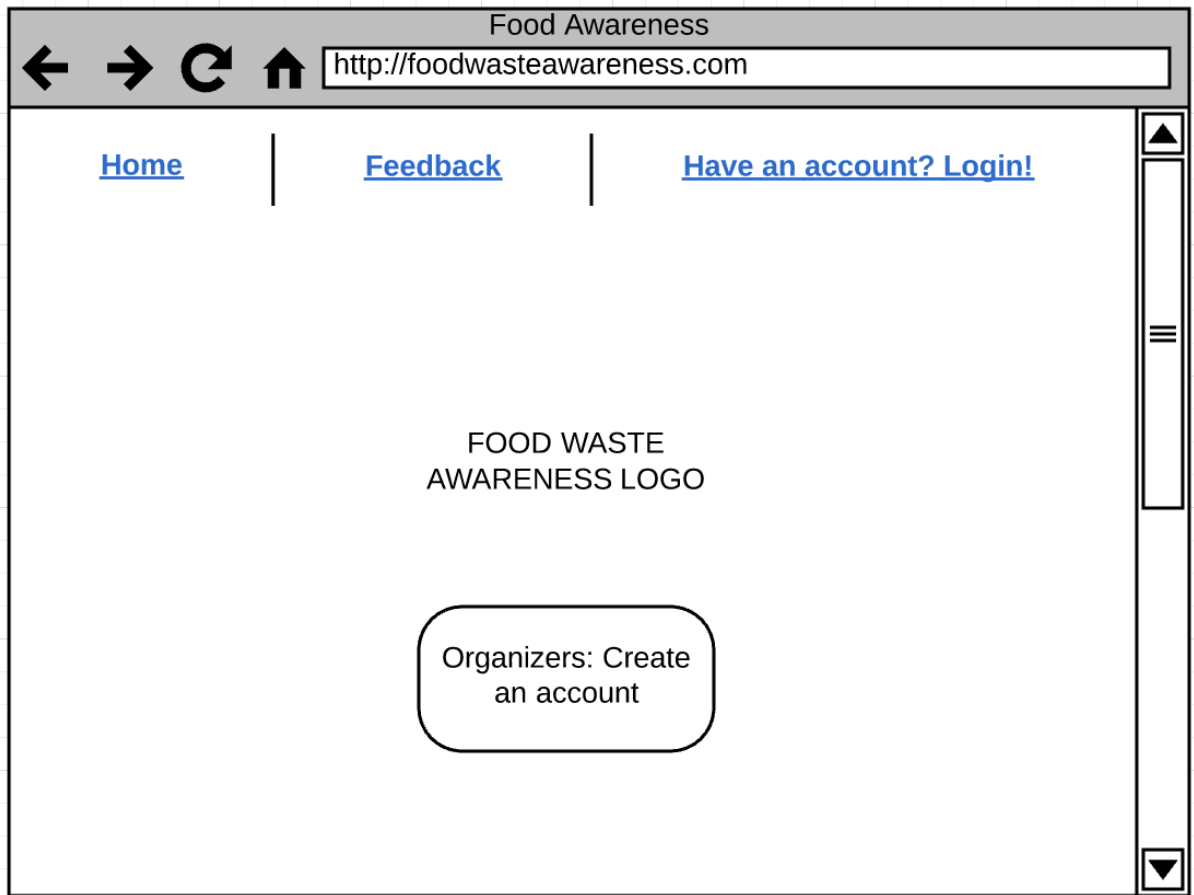




## 2. USER INTERFACES

### UI DESIGN:

Homepage



Feedback

Food Awareness

←

→

↺

🏠

http://foodwasteawareness.com/feedback/

[Home](#)

[Feedback](#)

[Create Account](#)

[Login](#)

FEEDBACK FORM

Food Order:

Enter order

Food Waste Amount:

Enter food waste

Food Amount:

Enter amount

Restaurant Name:

Enter name

Feedback:

Provide feedback of food ordered from restaurant

Submit

Register



Food Awareness

← → ↻ 🏠

http://foodwasteawareness.com/register/

[Home](#)

[Feedback](#)

[Create Account](#)

[Login](#)

Registration

Username:

Enter order

Organization:

organization

Password:

Enter amount

Contact Info:

email

Verify Password:

Enter amount

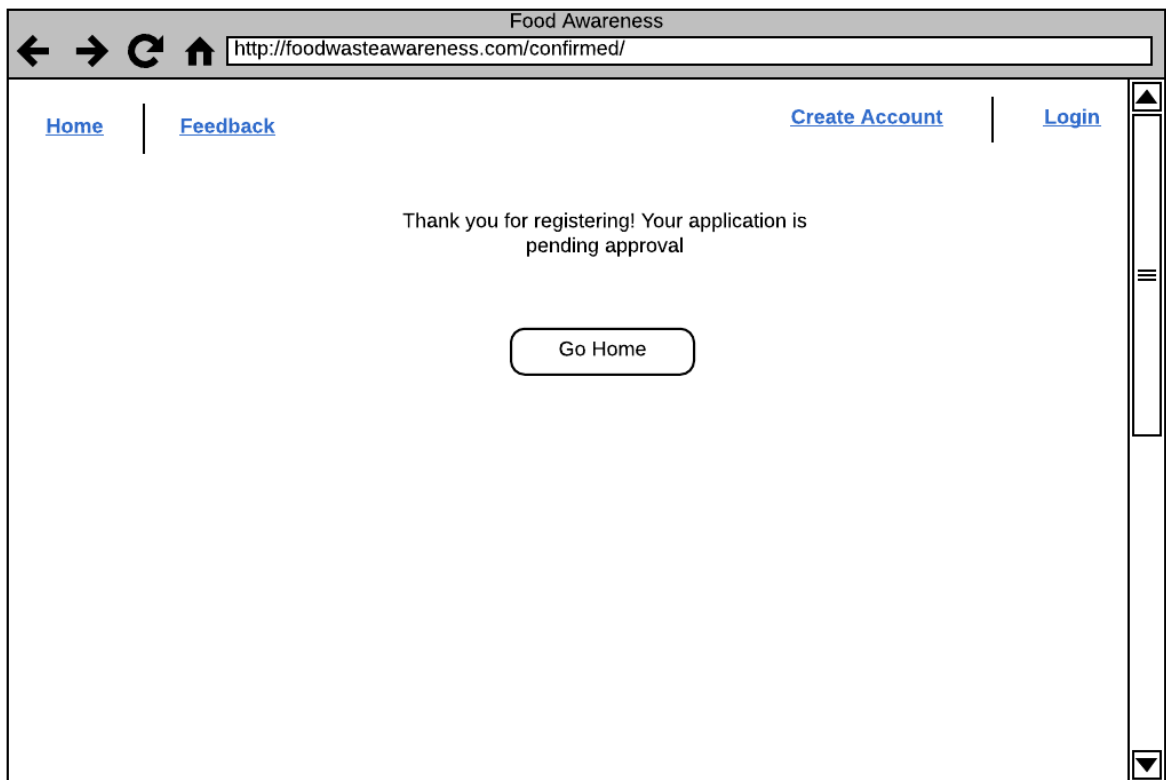
phone number

About:

Write a short description about organization

Submit

Register Confirmation



Edit and Submit Feedback(Organization)

Food Awareness

←

→

↻

🏠

http://foodwasteawareness.com/editfeedback/

[Home](#)

[Feedback](#)

[Edit Feedback](#)

[Dashboard](#)

To:

Restaurant Name

Subject:

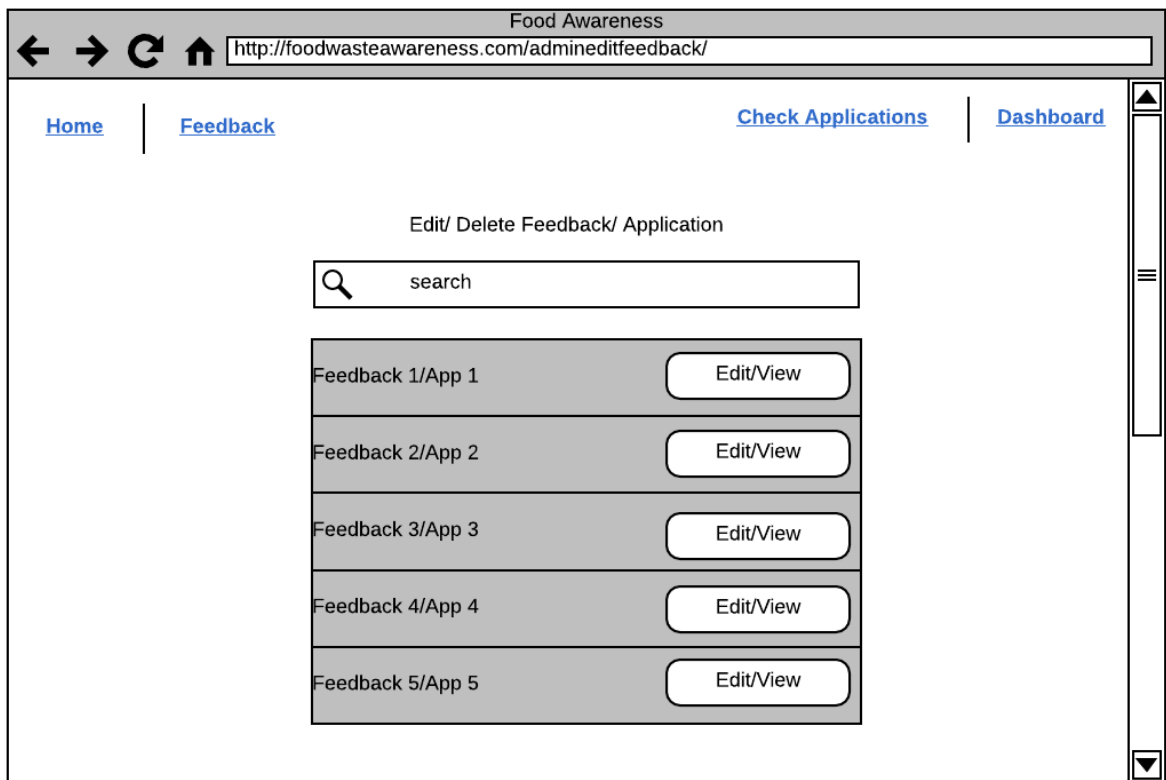
Order Feedback Summary by consumer

Summary:

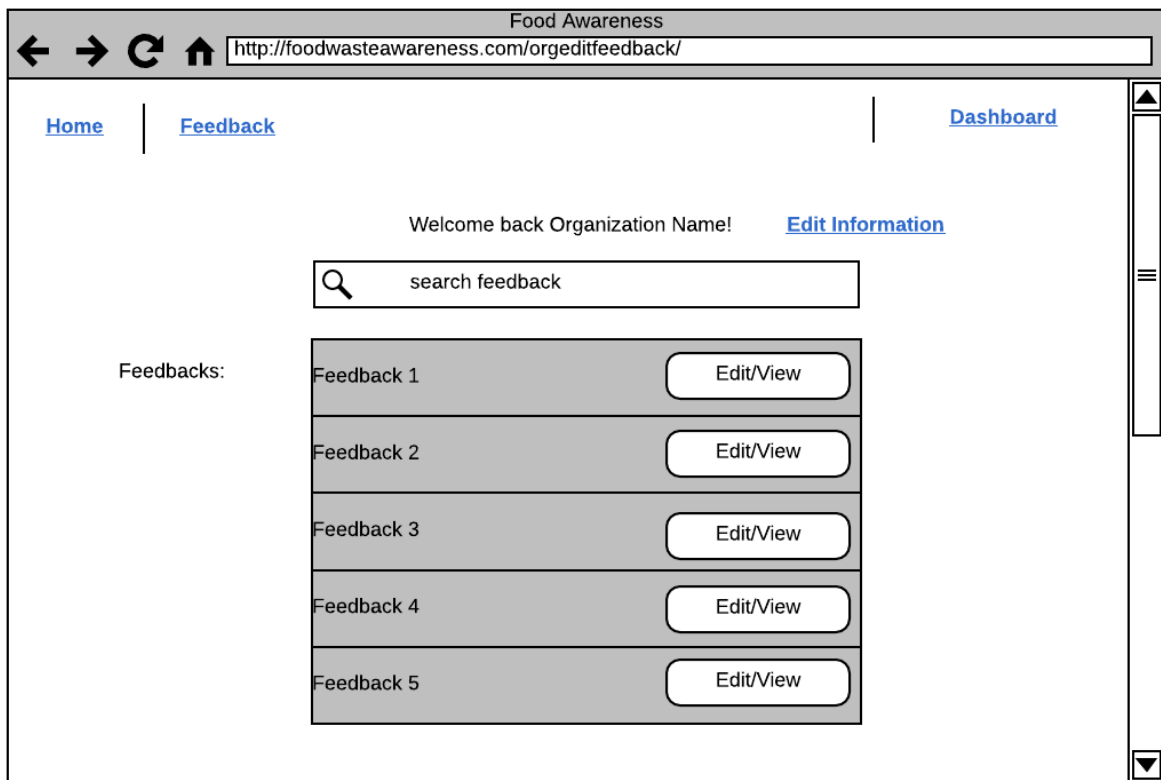
The consumer's food order is foodorder.  
The amount of food ordered is foodamount.  
The amount of food wasted by consumer: foodwaste.  
  
Additional Details:  
  
Thank you for your time  
Organization

Submit

Admin Dashboard



Organization Dashboard



### **Eight Golden Rules**

- 1) Strive for Consistency - The web application meets this requirement with consistency through various UI designs. The pages that require users to provide their feedback based on their order is similar for every user that logs in to this web application. The pages that require users and organizers to login is also consistent. The consistency in formatting reduces any ambiguity for the user trying to navigate the feedback form.
- 2) Enable Frequent Users to Use Shortcuts – The web application consists of a toolbar that allows the admin and organizers to jump to pages they would require to work on.

- 3) Offer Informative Feedback – This web app displays this rule by providing a confirmation alert for successful submission for organizations applying.
  - 4) Design Dialogs to Yield Closure – The requirement is met because the input is straightforward and accurately display what is to be expected. We also provide redirection for confirmation pages whenever a input is provided.
  - 5) Offer Simple Error Handlings – If a user doesn't enter any information in the feedback form field, it wont be submitted to the organization to process since it is blank.
  - 6) Permit Easy Reversal of Actions – Since all the input is listed on a single page, all actions are easily changed during input before submission. We also know that the feedbacks provided by the users are sent to the organizations and admins to modify.
  - 7) Support Internal Locus of Control – The user has full control of the system by creating a free account and providing a feedback based on the food they ordered from a restaurant.
  - 8) Reduce Short – Term Memory Load – Since the primary purpose of the user on the web application is to provide feedback in a single area, there is less memory to use. The details are presented explicitly and concise manner so that the user doesn't find it difficult to navigate through the application.
- Modularity and encapsulation of data to facilitate information hiding and reuse
    - The program uses modularity for each page which performs its own functions and algorithms for the user, organization and administrator. Most of the data in the application is hidden from the user to avoid ambiguity of navigation from one page to another.
  - Elegance and efficiency of the algorithms
    - The algorithms implemented enable the application to run smoothly without any lag and meets the functionality of the project.
  - Appropriateness of data structures used for the problem
    - The web application uses a stack where it stores the recent feedbacks from user in a queue(FIFO). If the stack is full, the oldest feedback is removed and the new feedbacks can be added to the stack.
- Full pathname to the project on the VM:  
/home/student1/csc415-ind-project/student1

- Private project repository on Github.  
nairp2/webapp(private)