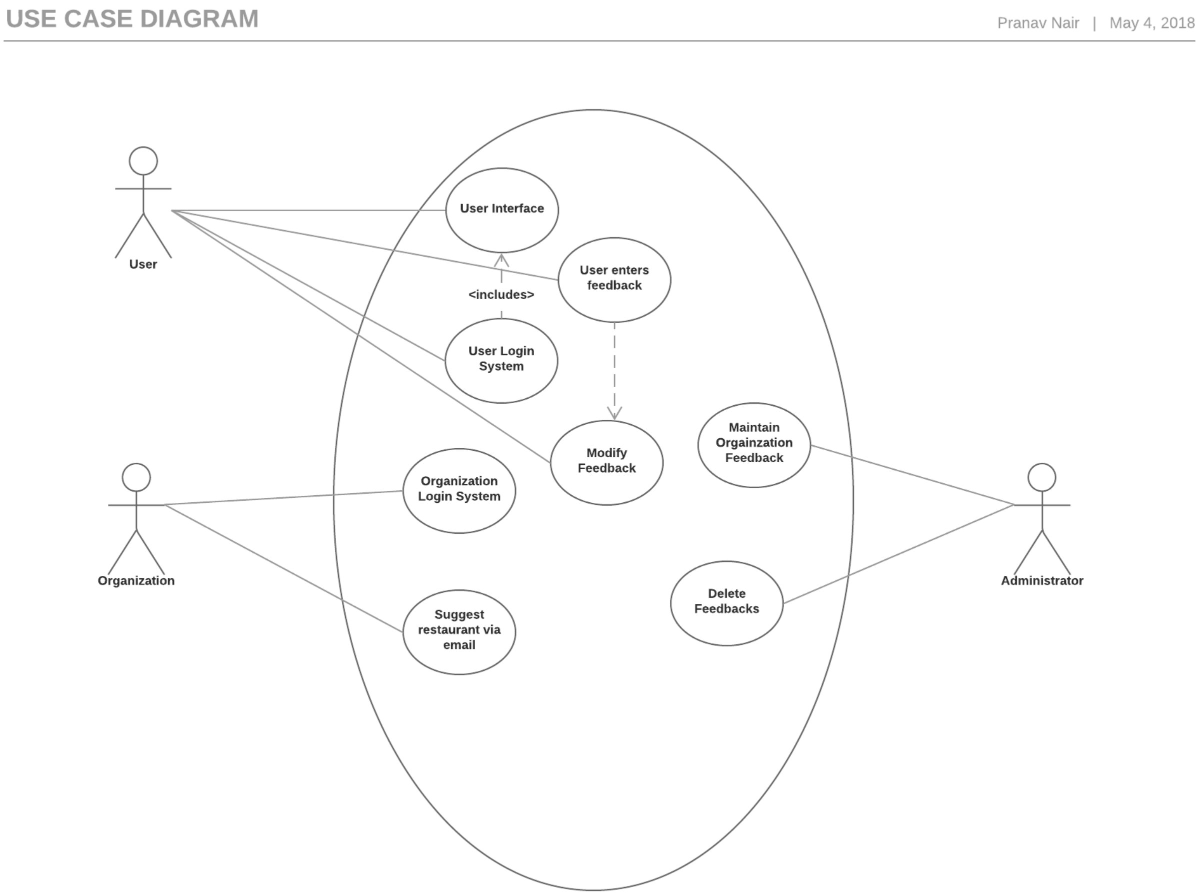
NAME: PRANAV NAIR
COURSE: CSC 415 – 01
ASSIGNMENT 4: OPEN SOURCE SOFTWARE: ANALYSIS AND DESIGN

1. DESIGN

a. USE CASE DESCRIPTION



IN – DEPTH USE CASE DESCRIPTIONS

CONSUMER USER

- Primary Actor: User
- Preconditions: User has made a login account.
- Scenario: User logins 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 form by clicking the feedback tab
    - User can edit feedback by clicking edit button
- Contact Page
    - About user details
        - Name, Phone number, email address etc.

<u>ORGANIZATION</u>

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

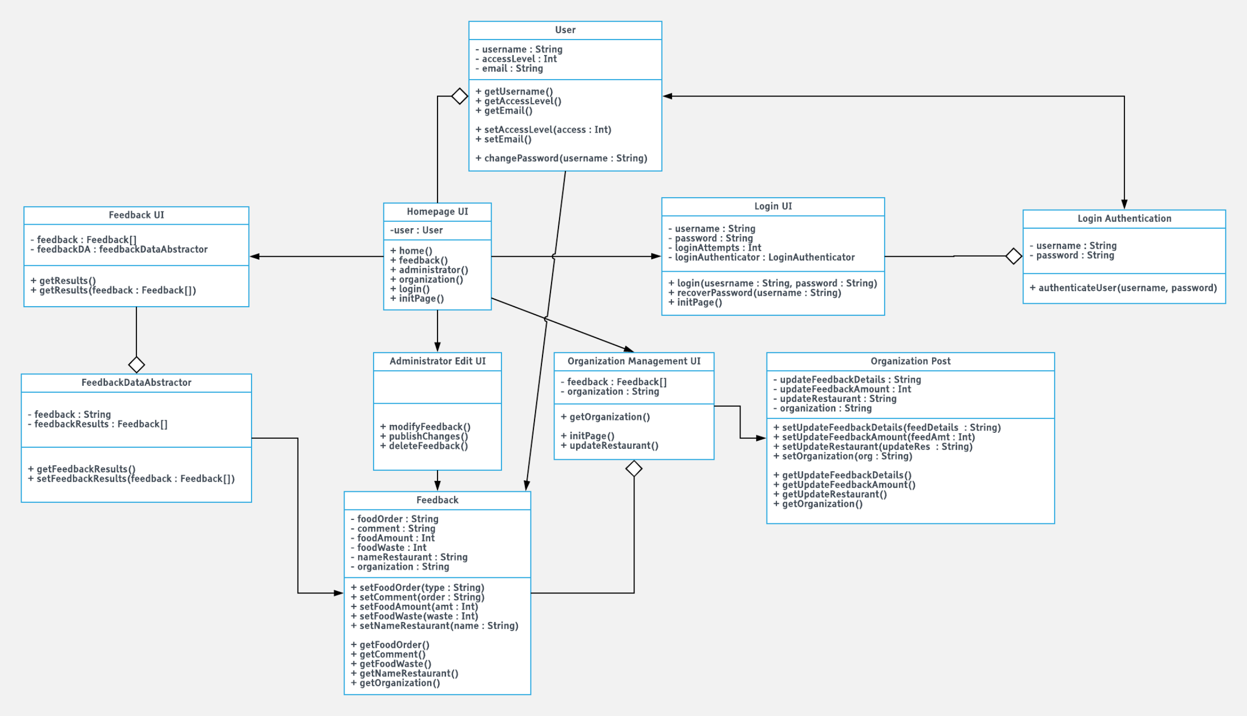
<u>SYSTEM ADMINISTRATOR</u>

- Primary Actor: System Administrator
- Preconditions: Administrator should be able to open the website and have access to user feedbacks and organization database.
- Scenario: Administrator opens web application and clicks the administrator tab to view organization Interface. Administrator also can modify or delete feedbacks if necessary.
- Updates organization page
    - Normal:
        - Modifies or deletes user feedbacks
    - Exception: If user doesn't provide feedbacks
- Perform system backup
    - Backup on database automatically
- Add/ Remove users from the system

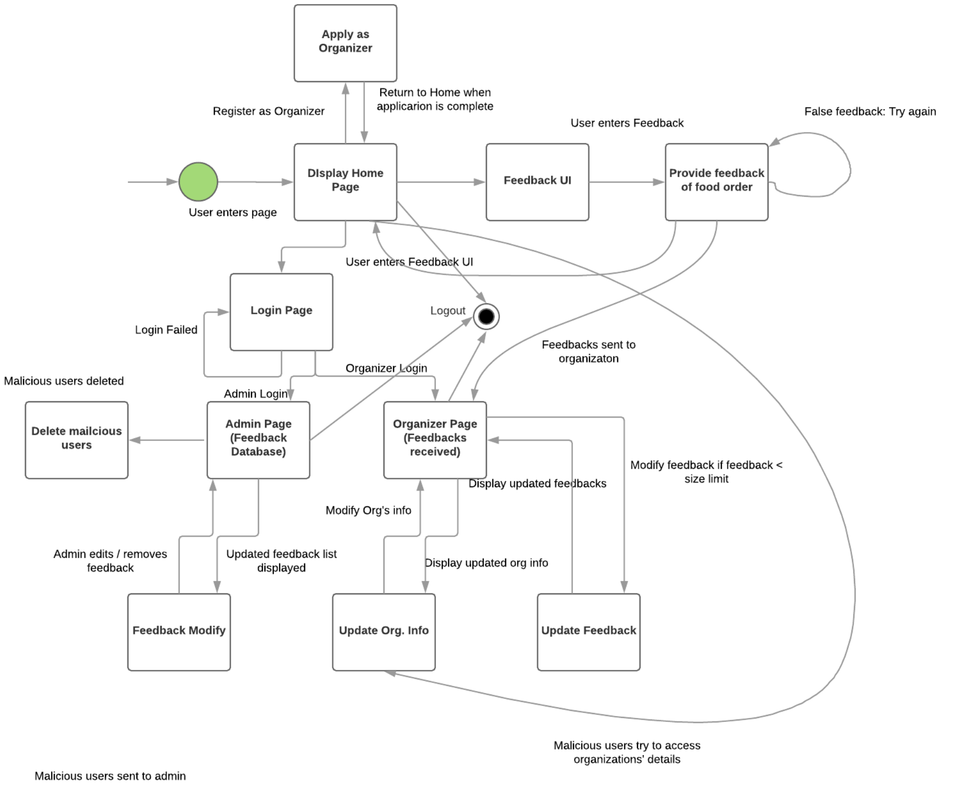
- Potentially automate spam detection

b. DETAILED DESIGN CLASS DIAGRAM

Class Diagram Link for Better Readability: https://www.lucidchart.com/documents/edit/ed4b8ee9-da59-42fb-9d28-4537e7207414/0

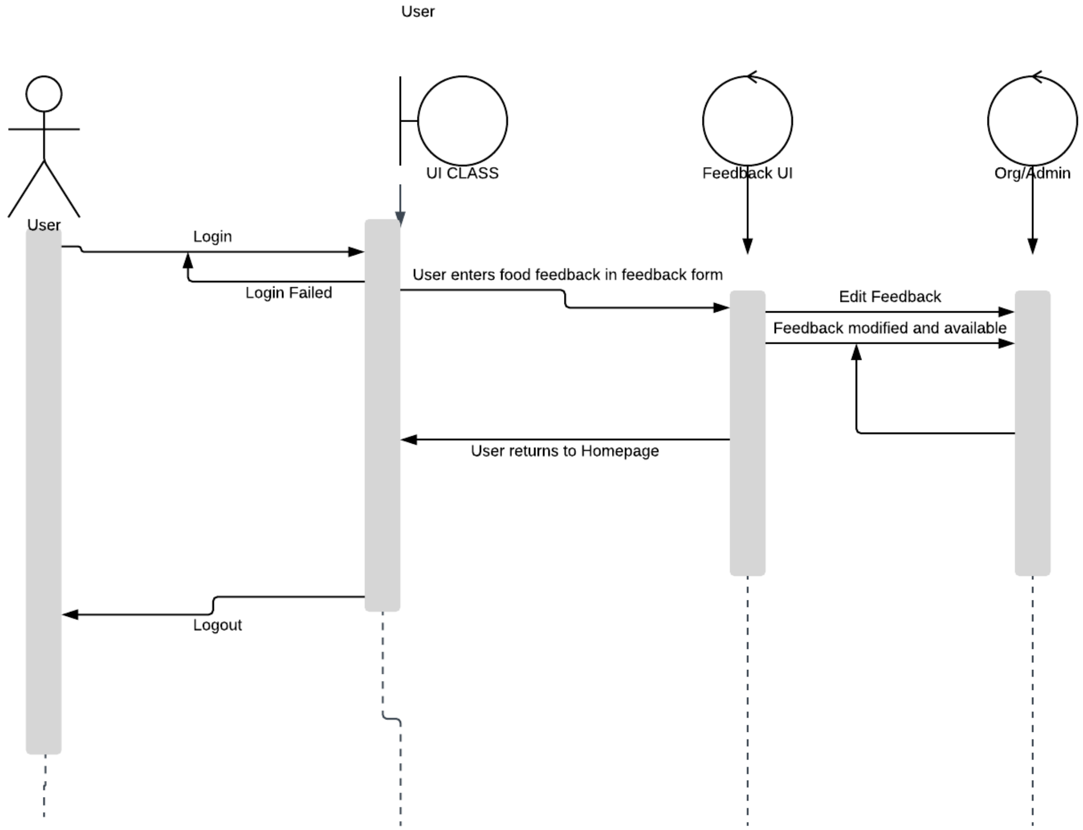


c. STATE CHART

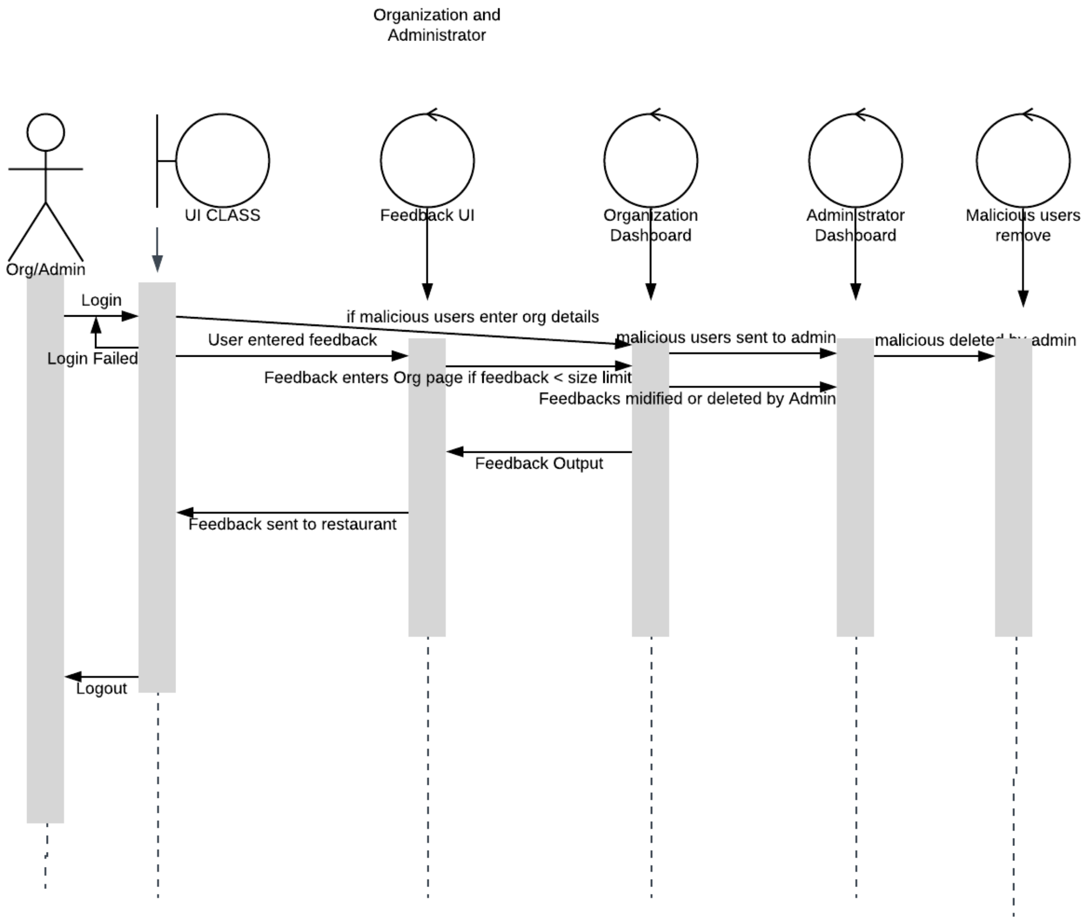

Apply as Organizer
Register as Organizer
Return to Home when applicarion is complete
User enters Feedback
False feedback: Try again
DIsplay Home Page
Feedback UI
Provide feedback of food order
User enters page
User enters Feedback UI
Login Page
Logout
Login Failed
Feedbacks sent to organizaton
Organizer Login
Malicious users deleted
Admin Login
Delete mailcious users
Admin Page (Feedback Database)
Organizer Page (Feedbacks received)
Modify feedback if feedback < size limit
Display updated feedbacks
Modify Org's info
Admin edits / removes feedback
Updated feedback list displayed
Display updated org info
Feedback Modify
Update Org. Info
Update Feedback
Malicious users try to access organizations' details
Malicious users sent to admin

d. SYSTEM SEQUENCE DIAGRAM

User



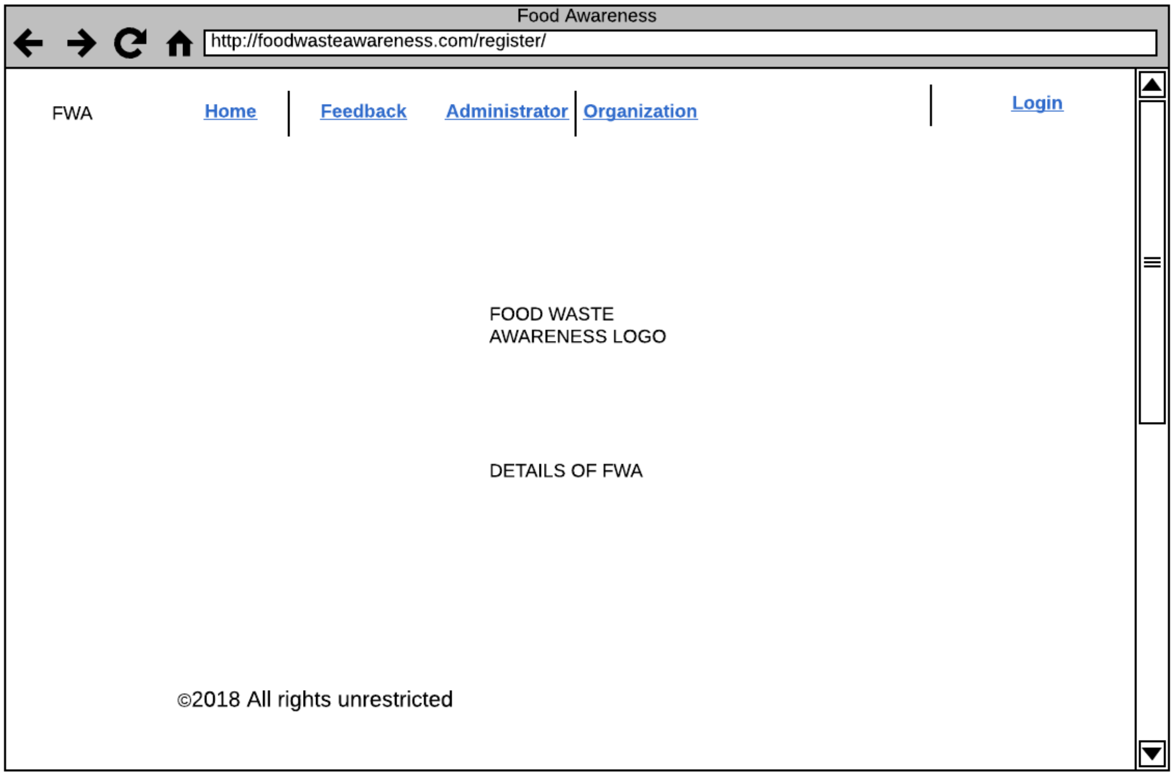
Organization/Administrator


Organization and Administrator
Org/Admin
UI CLASS
Feedback UI
Organization Dashboard
Administrator Dashboard
Malicious users remove
Login
Login Failed
if malicious users enter org details
User entered feedback
Feedback enters Org page if feedback < size limit
malicious users sent to admin
Feedbacks midified or deleted by Admin
malicious deleted by admin
Feedback Output
Feedback sent to restaurant
Logout

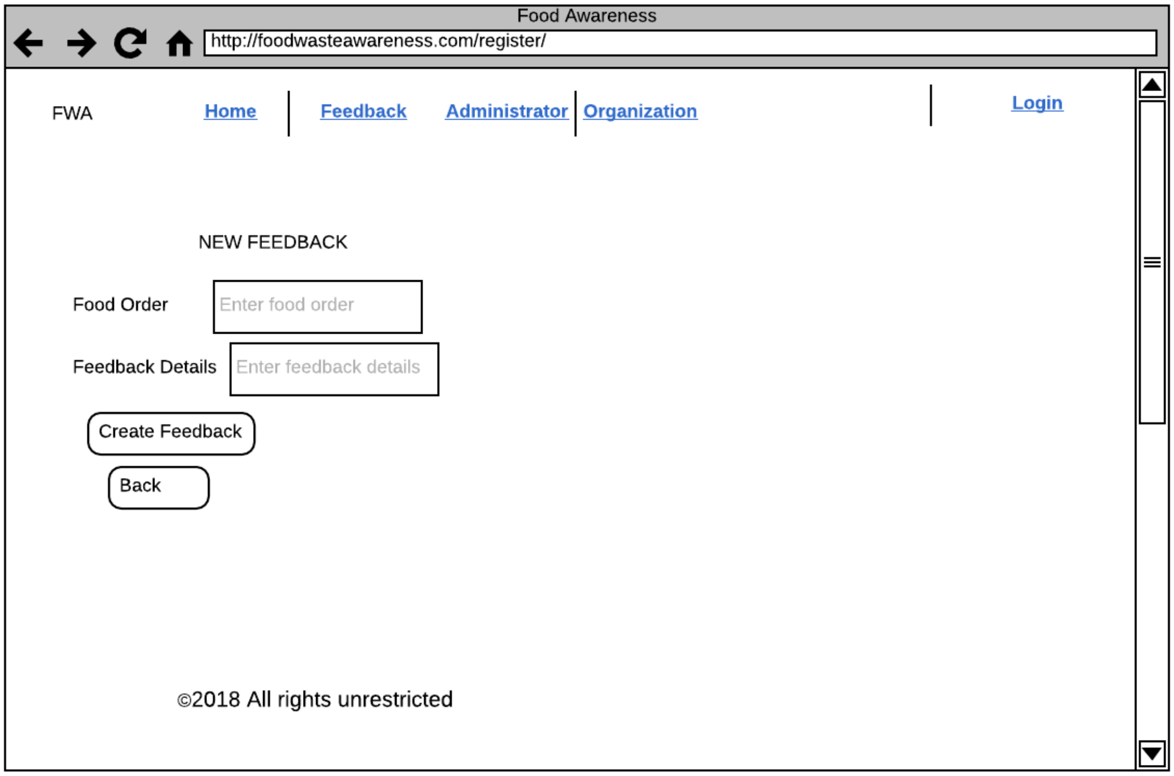
2. USER INTERFACES

UI DESIGN:

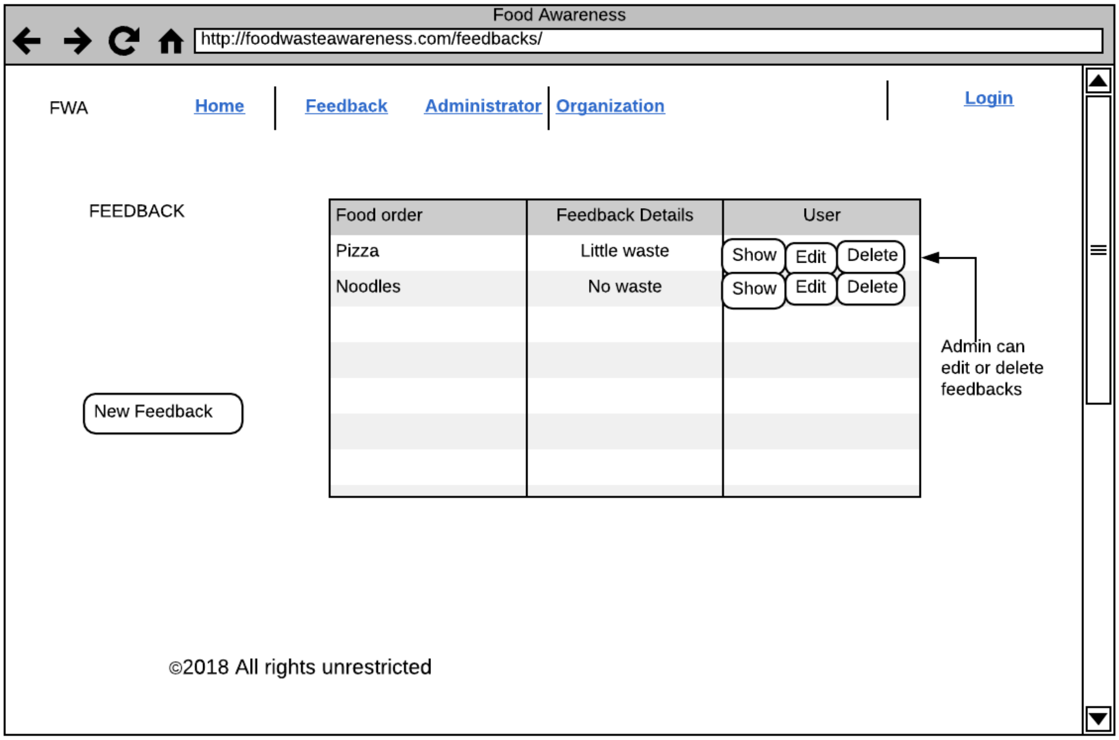
Homepage – http://foodwasteawareness.com/pages/Home



Feedback - http://foodwasteawareness.com/feedbacks/new



Administrator - http://foodwasteawareness.com/feedbacks

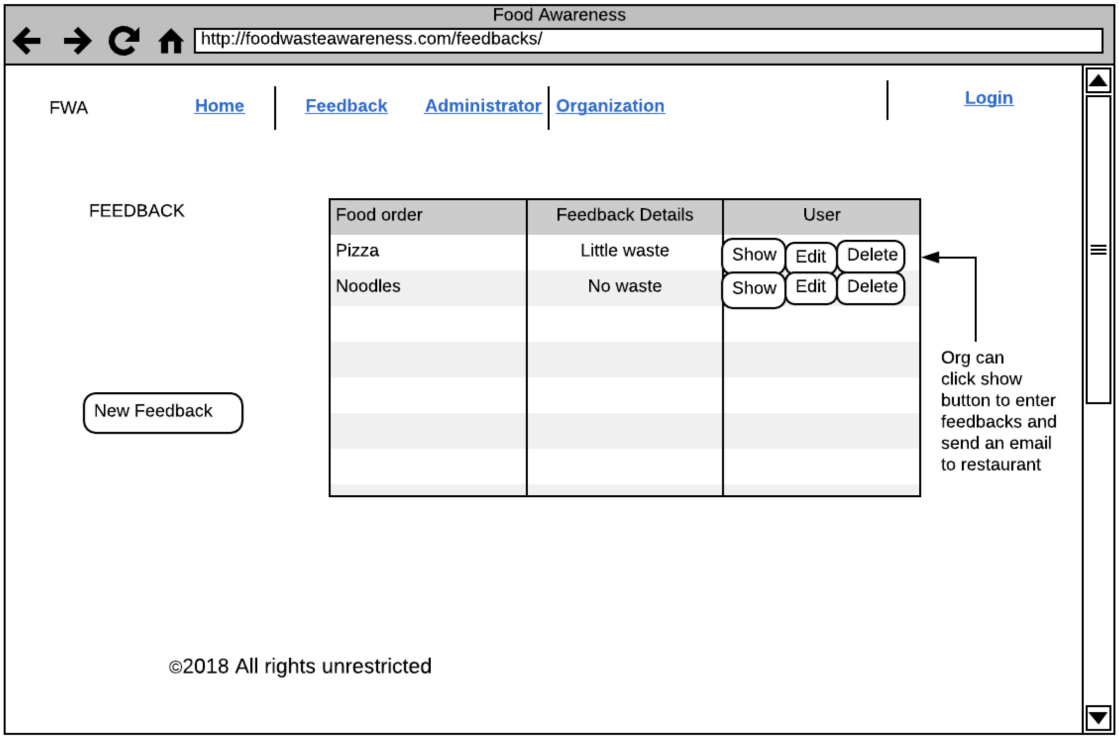


Food order	Feedback Details	User
Pizza	Little waste	Show Edit Delete
Noodles	No waste	Show Edit Delete

Admin can edit or delete feedbacks

©2018 All rights unrestricted

Organization - http://foodwasteawareness.com/feedbacks

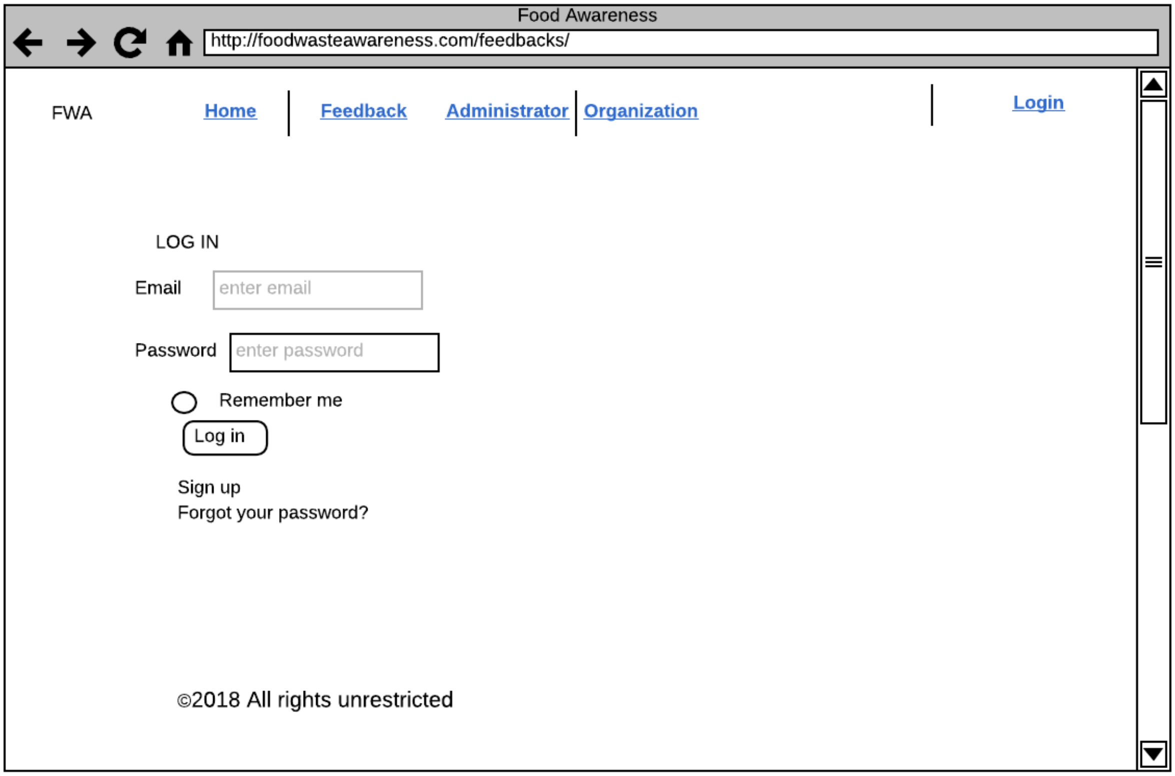


Food order	Feedback Details	User
Pizza	Little waste	Show Edit Delete
Noodles	No waste	Show Edit Delete

Org can click show button to enter feedbacks and send an email to restaurant

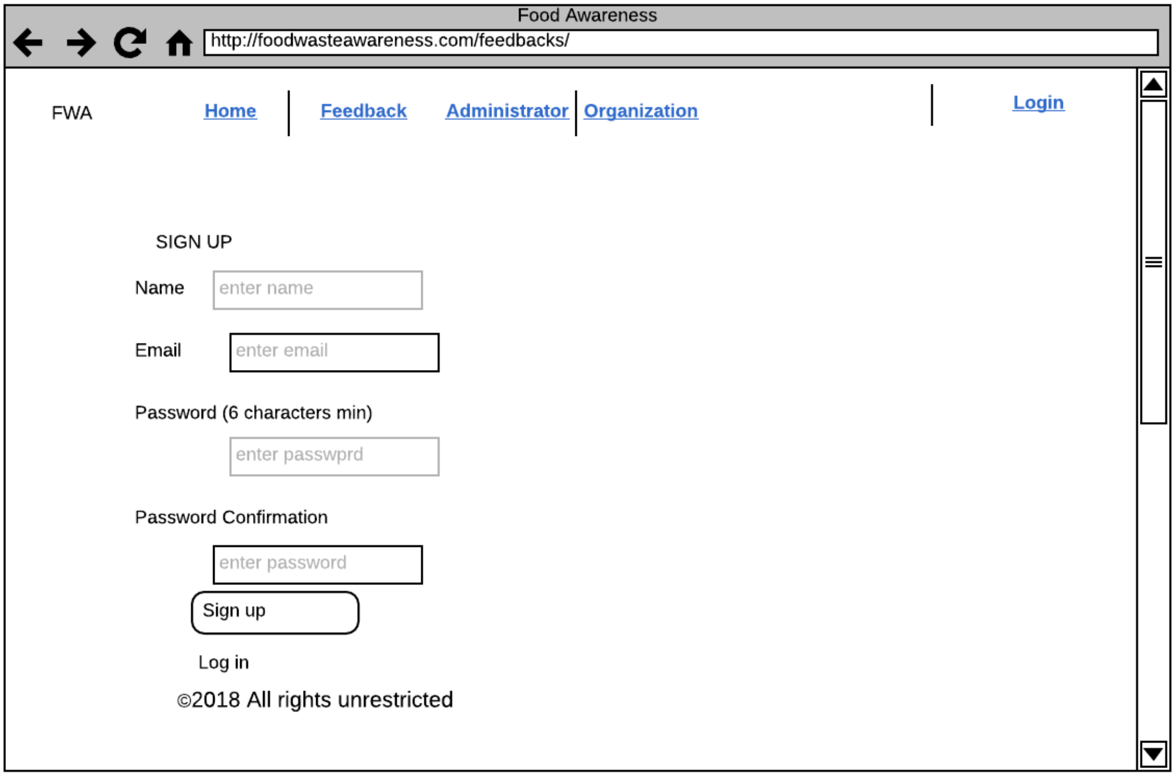
©2018 All rights unrestricted

Login(User/Organization) - http://foodwasteawareness.com/login



©2018 All rights unrestricted

Register(user/Organization) - http://foodwasteawareness.com/sign_up



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 logins 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 message in the redirected page when users enters feedback or user/organization successfully logged in

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 database 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 database where organizations and administrator can access.

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/webapp/code/prototype/student1`

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

Tools used

The rails program was manually tested on the server. I used terminal to access the server and login with my username and password. I created a directory where I implemented the web application. I run rails server on command line and open the browser to access the web application.

Test Cases

1) Login Page Authentication

Description: User/Organization login to their account

Steps:

1. User/Org selects login from homepage
2. User/Org enters login information
3. Login outcome:

Login fails for incorrect information and user/org is brought back to login page. Provides error feedback

Login is successful for correct details and can access feedback/organization tabs.

Reason: I implemented this way so that the users/organizations cannot access the tabs directly without logging in.

Expected Results: User/Org has the ability to access feedback/organization tabs

2) Entering Feedback

Description: User clicks feedback tab and enter food details

Steps:

1. User clicks feedback tab.
2. User enters food details

Reason: This allows users to enter their opinion on the food they ordered

Expected Results: User can enter feedback which is sent to the database

3) Administrator

Description: Admin clicks administrator tab and modify or delete feedbacks
Steps:

1. Admin clicks administrator tab
2. Admin selects delete to delete feedback or selects edit to modify feedback

Reason: Administrator has the most power to a system. Hence, the admin can access the database of user and organization.

Expected Results: Administrator has the ability to modify and delete feedbacks

4) Organization

Description: Organization clicks organization tab and can send suggestions to restaurant
Steps:

1. Organization clicks organization tab
2. Organization selects show and enters suggestion
3. Organization clicks send email

Reason: This allows organizations to receive user feedback and send suggestions to restaurant via email for future orders.

Expected Results: Organization provides suggestions to restaurant via email