UCook Project Details

By Team25

Team Members:

Kaidi Zhang (kaidiz)

Minyi Dai (minyid)

Tuo Shi (tuos)

Xiaodi Tao (xiaodit)

Yuyan Jiang (yuyanj)

Sprint #1 Product Owner(s):

Minyi Dai & Xiaodi Tao

1. Product Description:

UCook is a web application that provides a meal-sharing platform for users within a specific community. UCook allows users to either request a meal, in which case the user becomes the guest, or invites others for a meal at home, in which case the user becomes the host. To enjoy a "free meal" by the host, the guest is usually required to bring the ingredients :) The guest is welcomed to tip the host for the meal event.

2. Product Backlog:

- Anonymous Users' Actions

1. View welcome page

An anonymous user can view the welcome page of the app. However, to navigate for other functionalities, the user needs to be logged in.

2. Register

An anonymous user can register for our app by providing the username, email, password, and so on.

3. Login

An anonymous user can login to our app.

4. Link to Google / Facebook account

An anonymous user can choose to register using their Gmail or Facebook account.

5. Forget password

If the anonymous user forgets his/her password, we will send a link to the user's email for resetting the password.

- Guest Actions:

1. View user profile

A logged in user has a dashboard page where he/she can view the profile. The profile includes other information for that user including current addresses, avatar, self-introduction, and meal preferences.

2. Favorite / Follow another user

A logged in user can follow other users. A followers/following list will be displayed on their dashboard page.

3. Chat with other users

A logged in user can send a chat request to the other user. Our app supports a basic one-on-one chatting functionality!

4. Post a request for meal

A logged-in user can specify whether she/he wants to play the role of a host or guest on the welcome page. The user will be redirected to different pages based on the role she/he chooses upon visiting our web app. The user can be both a guest and a host at the same time, which depends on what page (host / guest) the user is currently visiting.

If the user chooses "I am a Guest" option on the welcome page, then the user is redirected to the Guest Page. The guest can post a request for a meal, specifying which community (address) she/he wishes to be hosted, what type of cuisine she/he wants, etc.

5. Accept a host's invitation

The guest can accept a host's invitation for a meal event, if any. When the guest accepts the host's invitation, the guest will be added to the meal event's guest list.

6. Search for nearby hosts

The guest can search for nearby hosts based on his/her current location. This will require the use of Google Map API.

7. Request to join a host's meal event

When the guest finds a host's post for a meal event, the guest can request to join that meal event. The guest will be added to the meal event's guest list, unless the host manually removes the guest from the list.

8. Leave a review for a meal event

After completing a meal event, the guest can leave a review for the meal event that she/he participates. The review will be accumulated on the host user.

9. Pay tipping to the host

When leaving a review for the meal event, the guest can choose to tip the host. This will require the use of Venmo API or other payment systems.

10. View meal history

The users can view specifically their meal history, sorted from the most recent ones to the least recent ones. Other sorting can be provided as well.

11. Cancel the request

The guest can cancel the request for a meal, if she/he doesn't want it anymore. After canceling the request, no other hosts can view the request.

12. Get meal event recommendations

Our app provides each guest with customized meal event recommendations based on the user's profile, cuisine preferences, and meal history.

- Host Actions:

1. View user profile

The same as guest.

2. Favorite / Follow another user

The same as guest.

3. Chat with other users

The same as guest.

4. Post a meal event

If the user chooses "I am a Host" option on the welcome page, then the user is redirected to the Host Page. The host can post a meal event, specifying where and when she/he is going to host, what type of cuisine she/he is hosting, what ingredients the host wants the guests to bring home, etc.

5. Accept a guest's request to join the meal event

If a guest sends a request to the host for joining the meal event, the host can either accept the guest's request or decline it.

6. Search for nearby guests

The host can search for nearby guests who wants a meal. This will require the use of Google Map API.

7. Send an invitation to a guest to join the meal event

The host can send an invitation to a guest to join his/her meal event via the guest's profile page.

8. Leave a review for a guest

The host can leave a review for the guest who attended his/her meal event. The review/rating will be accumulated on the guest user.

9. View host history

The hosts can view specifically their host history, sorted from the most recent ones to the least recent ones. Other sorting can be provided as well.

10. Cancel the meal event

The host can cancel the meal history, if she/he doesn't want to host anymore. After cancelling the meal event, no other guests can view the meal event, and the guest list will automatically be emptied. Guests who were in the guest list will get notified of this cancellation.

3. Sprint #1 Backlog:

As of the 1st Sprint, we try to achieve the following functionalities:

1. Front-End
    - Complete all templates, stylesheets, and other static assets.
    - Complete basic event handling
2. Back-End
    - Configure basic app routes
    - Set up models and forms on-demand
    - Prepare dummy data
    - Login/Register functionality
    - Host/Guest post form submission
    - External recommendation system

4. Data Models:

We currently have 6 models. Further models will be added as we proceed to more advanced functionalities. The draft models and their initial fields are listed below:

User

- Django default

Profile

- username
- email
- avatar

- bio
- sex
- addr
    - street, apt
    - city
    - state
    - country
    - zip
- fav_cuisine

Review

- event_id
- author
- datetime
- comment
- rating
- user_fk
- image (optional)
- tipping (optional)

GuestPost (for guest)

- guest_id
- content (intro, what I need)
- cuisine_type (chinese, korean, italian)
- create_datetime
- event_datetime
- status (if a host has accepted the request)
- host_id
- event_id (hostpost_id)

HostPost (for host)

- host_id
- content (what guest needs to bring)
- sample_img (optional)
- cuisine_type
- create_datetime
- event_datetime
- addr
    - street, apt
    - city
    - state
    - country
    - zip
- status (if len(guest_list) == num_guests)
- num_guests

- guest_list

Transaction/MealHistory

- host_id
- guests_list
- event_id

5. HTMLs:

We currently have 11 templates. Further templates will be added as we go.

- base.html
- welcome.html
- register.html
- login.html
- profile.html
- mypost.html
- posthost.html
- postguest.html
- detail.html
- explore-host.html
- explore-guest.html