Project specification

Web Application Development Team 1
Xin Wang, Kylie Hsieh, Yuxuan Wu, Yanyu Chen

1. Product Owner:

Xin Wang Andrew ID: xinwang3

2. Product Backlog

- Register function
  New customers can register a new account to use this Yummy application. Also, users may also choose to log in using their Google account. To register, users need to type in their first name, last name, password, username, and phone number.
- Login function
  Customers can log in using their username and password. After logging in, users will be automatically directed to the menu page, where they can place an order or reserve a table or just browse the menu.
- Reservation function
  Customers can reserve a table after typing in the number of people in the group, reservation time, and date. After making a reservation, the admin user (host) will receive a message about the reservation details.
- Order function
  If the customers choose to order, there are two buttons where they can choose to take out or dine in. If the customer selected dine-in, the page will prompt the user to enter the table number. After submitting the table number, the website will redirect the user to the order page. Customers can add a dish to their order by clicking the “+” button and clicking the “favorite” button to add the dish to their favorite list under the logged-in user identity
- View dish detail
  Customers can check the detailed information about the specific dish. Users are allowed to make comments under the dishes, add the dishes to their order or use the “favorite” button to collect the dishes.
- Check out function
  Customers can view their order summary, including the quantity, unit price, and total price, and send the order to the restaurant. Users on this page could modify the quantity of dishes they ordered or remove dishes they do not want.

3. First Sprint Backlog

Task	People in charge of this task
Create models and forms	All four members
Register and login function	Yanyu Chen
UI - Home page and Menu page	Yuxuan Wu and Xin Wang
UI - Reserve page and Dish page	Yanyu Chen
UI - Option page and Summary page	Kylie Hsieh

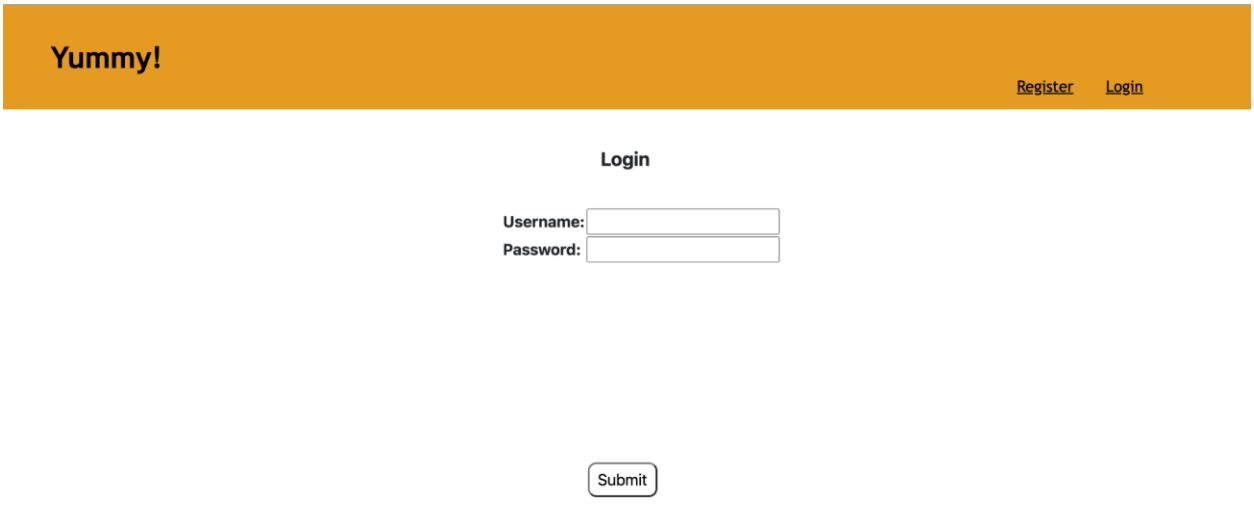
4. Implementation of Models

Please see the appendix screenshots for detailed implementations of Django models

Model	Description
Profile	This model is for saving customer information, including customer name and phone number.
Category	This model represents the category that a dish belongs to, including appetizers, meat, vegetable, soup, dessert, and snacks.
Food	This model saves all the dishes in a restaurant, including dish name, price, description, calories, is the dish spicy or not, and is the dish vegan food or not.
Comment	This model saves customers' comments for each dish.
FoodSet	This model saves the dish and its corresponding quantity for an order that a customer placed
Order	This model is for saving customers' orders.
Table	This model saves the table information in the restaurant, including the related orders, the occupied customer, and open time.
Reservation	This model saves the customers' reservation information.

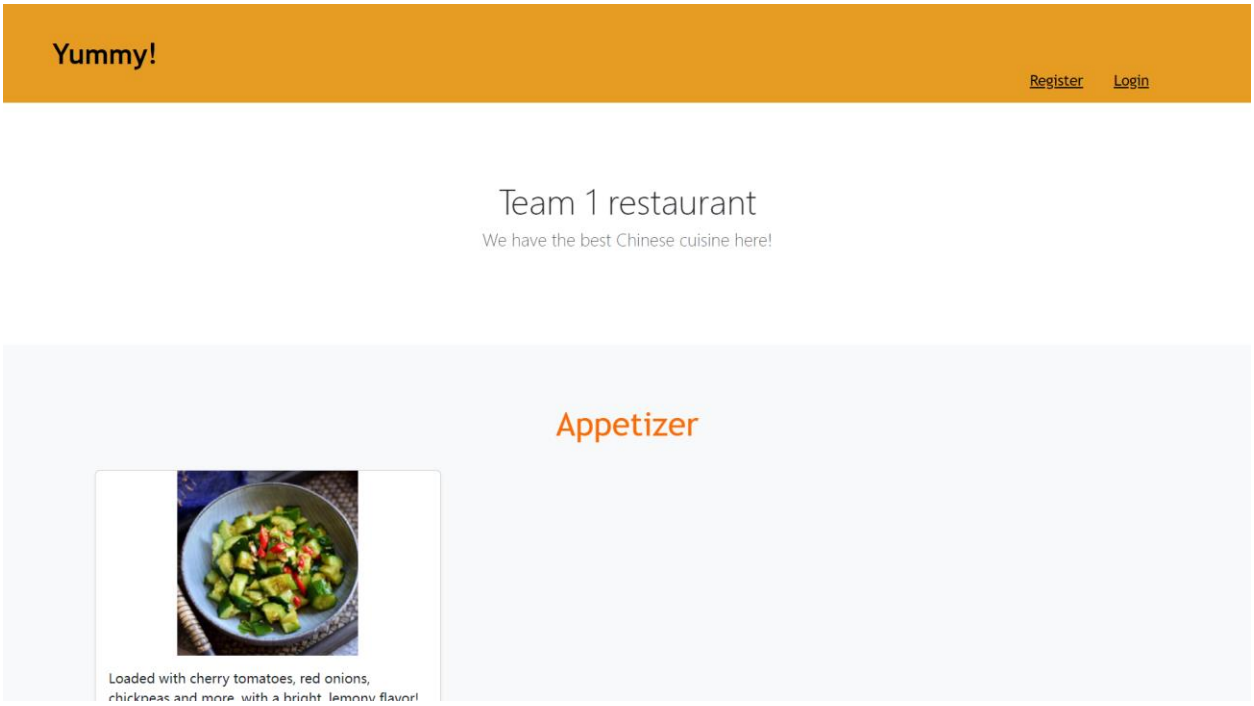
4. HTML mockups

Login Page

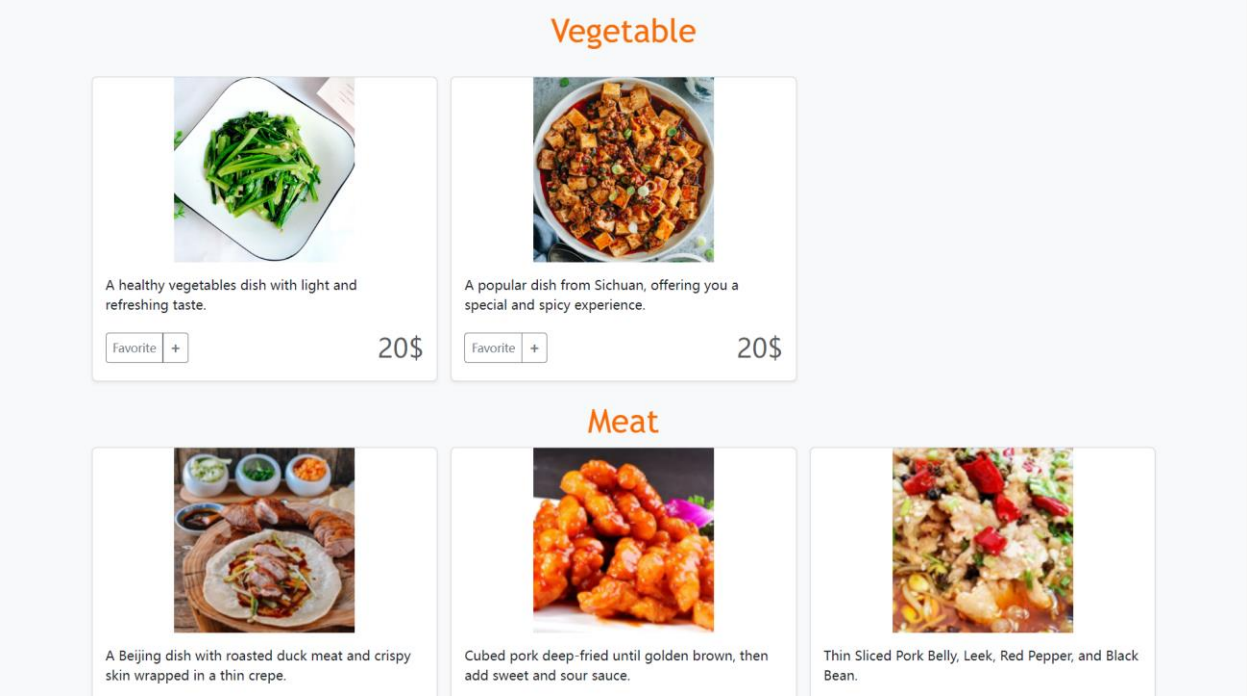


Register Page

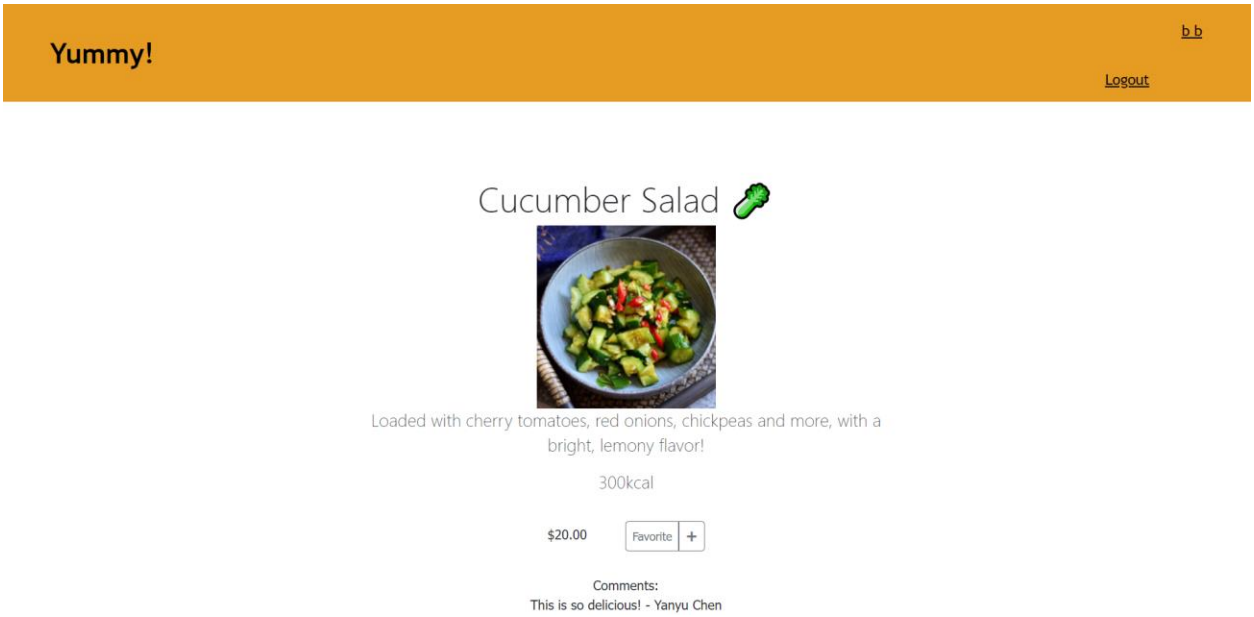
Home page



Menu page



Dish detail page



Reservation page

Yummy!

b b

Logout

Reserve Your Table

Enjoy your food

Step 1: find a table

Date:

Time:

#people:

Find

Step 2: your details

First Name:

Last Name:

Phone Number:

Special Comment:

Reserve

Option page

Yummy!

b b

Logout

Team 1 restaurant

Is this order take-out or dine in?

Take Out | Dine In

Please enter your table number if you are dining in: Submit

Order summary page

Yummy!

b b

Logout

Cart

Menu

Table 1

Image	Dish Name	Quantity	Price	
	Fried Pork Dumplings	- +	$10	Remove
	Scallion Pancakes	- +	$15	Remove

Appendix: Django Models

```python
from django.db import models
from django.contrib.auth.models import User


Wangxin555
class Profile(models.Model):
    user = models.OneToOneField(User, default=None, on_delete=models.PROTECT)
    phone_number = models.CharField(max_length=200, editable=True, blank=True)


Wangxin555
class Category(models.Model):
    name = models.CharField(max_length=500)


Wangxin555 +1
class Food(models.Model):
    name = models.CharField(max_length=500)
    price = models.FloatField()
    description = models.CharField(max_length=500)
    picture = models.ImageField()
    content_type = models.CharField(max_length=50)
    category = models.ForeignKey(Category, on_delete=models.PROTECT)
    calories = models.FloatField()
    # 1 for spicy and 0 for non-spicy
    is_spicy = models.BooleanField(default=False)
    is_vegetarian = models.BooleanField(default=False)


Wangxin555 +1
class Comment(models.Model):
    text = models.CharField(max_length=500)
    creation_time = models.DateTimeField()
    creator = models.ForeignKey(User, on_delete=models.PROTECT)
    post_under = models.ManyToManyField(Food, related_name="comments")


# handle quantities of different foods
Wangxin555
class FoodSet(models.Model):
    food = models.ForeignKey(Food, on_delete=models.PROTECT)
    quantity = models.IntegerField()


Wangxin555 +1
class Order(models.Model):
    foods = models.ManyToManyField(FoodSet, related_name="orders")
    customer = models.ForeignKey(User, on_delete=models.PROTECT)
    order_time = models.DateTimeField()
    # whether the food is take out or not
    is_takeout = models.BooleanField(default=False)
    is_paid = models.BooleanField(default=False)
    total_price = models.FloatField()


Wangxin555 +1
class Table(models.Model):
    orders = models.ManyToManyField(Order, related_name="table")
    customer = models.ForeignKey(User, on_delete=models.PROTECT)
    open_time = models.DateTimeField()

YanyuChen98
class Reservation(models.Model):
    num_customers = models.IntegerField(blank=False)
    table = models.ForeignKey(Table, on_delete=models.PROTECT)
    first_name = models.CharField(max_length=200, editable=True, blank=False)
    last_name = models.CharField(max_length=200, editable=True, blank=False)
    phone_number = models.CharField(max_length=200, editable=True, blank=False)
    comment = models.CharField(max_length=200, editable=True, blank=True)
```