

Project Specification

By Andy(tianqiw2), Sacha (sbarthol), and Joshua(jcmathew)

1. Introduction

In the game, users will have a competition to design burgers according to a list generated by the program. According to a list generated by the server, they will need to make "correct" burgers, both in sequence and in ingredients. The correct burgers will get maximal points. Burgers with some materials in the list will receive partial points; those that try to spam the materials by excessive use will get negative points due to wasting. The highest scores will show on a leaderboard, which will be stored in a relational database.

The game screen consists of two hands, a conveyor belt(s), and a burger ready for building. The left hand is the hand the current player can control; the right hand belongs to the opponent. Therefore, when player A moves the left hand, it should reflect on player B's screen as the right hand.

2. The product backlog

Please note that some tech stacks might be subject to slight changes.

UI/UX/Frontend design: the upper navbar directs the user to two screens after login: a screen for starting burger-building games and a screen for the leaderboard. These features will be implemented using simple Django wiring techniques. To maintain a consistent and aesthetic website, the users use Bootstrap and possibly Figma to aid our design. The authors draw something from scratch or use images from websites for graphics and image design.

General backend design: given the abundance of Django resources, the authors decide to use Django as the backend. We will refactor previously written code and improve readability and efficiency.

Name recording feature and leaderboard: given the game's competitive nature, we will need to keep a leaderboard and update it when it ends. To prevent cheating, we decided to store these records with SQL and perform a semi-real-time update with AJAX. The leaderboard consists of three features: username, time of record, and scores. If duplicate names are present, the user will be notified to change their name to a new one.

Player screen feature: our current player screen is implemented with HTML canvas. One round of the game is playable for two people. To update efficiently for both slide players, we decided to use **WebSocket**. The upper screen will be a list, where the users need to plan accordingly to put ingredients from the conveyor belt onto the burger by clicking. The users then need to put the ingredients on burgers quickly to build a successful burger. All these features will be stored in a database (Redis/MySQL, undecided) to prevent cheating via changing canvas content.

Security Measures: Django will generate CSRF tokens to combat CSRF attacks.

3. The first sprint backlog

We plan to do the following steps for the first sprint:

- (1). Implement wiring for scoreboard/canvas game screen;
- (2). Implement simple wiring for updating the scoreboard;
- (3). Implement a simple game mechanism (movement of hands reflecting on both sides for Websocket, ingredient workflow) and dealing values concurrently for plans.

4. Implementation of data models used by the application(tentative)

```
class Score(models.Model):
    user = models.ForeignKey(User, default=None, on_delete=models.PROTECT)
    creation_time = models.DateTimeField()
    ip_addr = models.GenericIPAddressField() # for validation purposes
    Score = models.IntegerField()

class IngredientList(models.Model):
    class Ingredient(models.TextChoices):
        BACON = 'BACON', _('Bacon')
        MUSHROOM = 'MUSHROOM', _('Mushroom')
        LETTUCE = 'LETTUCE', _('Lettuce')
        PICKLE = 'PICKLE', _('Pickle')

    Ingredient_list = models.CharField(
        max_length=20,
        choices=Ingredient.choices,
        default=Ingredient.BACON,
    )
```

5. Tentative HTML mockups for your application

Scoreboard:

Navbar	Game	Scoreboard
--------	------	------------

Nickname	Score
Alan Turing	100000
Ada Lovelace	100000
Farnam Jahanian	10000

Player Screen:

Navbar	Game	Scoreboard
--------	------	------------

