- The **product backlog**: a complete list of all functionality (i.e., the actions) of your project, and an English description of each action. We strongly suggest that you organize these features into groups/modules based on related functionality.
- The **first sprint backlog**: a complete list of the functionality you will complete during your first sprint, and how that work is allocated among your team members.
- The **name and Andrew ID of the product owner** for the first sprint.
- A **complete implementation of the data models** used by your application. This may be written in as Django models, SQL, or some equivalent style implementation if you use another framework.
- A complete set of **drawn wireframes** or HTML mockups for your application, for all non-trivial views within the application.

Connect4 Product Backlog

1. Game flow and functionalities

Start/login page
Users can login through this page or go to the register page to create an account.
- Connect4 logo
- Text boxes for username and password
- Register button -> register page
- Login button -> login page

Register page
Users can register for an account on this page.
- Username
- Password
- Confirm password
- Register button -> global page
- Already have an account? Login -> login page

OAuth
User authentication will be implemented using OAuth for the final project. We plan to start with the default Django authentication and then transition to using OAuth.

Header template (all html pages besides login and register)
- Profile -> profile page
- Leaderboard -> leaderboard page
- Global -> global page

Global page
Users can choose to join an existing game room or create a new room to start a game. If they were to join an existing game room, they start the game with the room owner right away.

Otherwise, the new room will be added to the list on the global page and they'll wait for someone to join their room.

- If player has not opened a room yet: “Start new game” button -> gameplay page
- If player has an open room: “Resume game” button -> gameplay page
- Global list of all current open rooms with opponent's nickname
    - Player can click into any open rooms to start the game -> gameplay page
    - Open rooms are not shown if the user has already opened a game room

Profile page
Users can update their nickname, profile photo, and bio text here. They also log out from this page.

- Upload tokens
- Profile picture
- Nickname (defaulted to username)
- User bio
- Logout button -> start/login page

Leaderboard page
Users can see their game stats (# of wins) and position on the leaderboard if applicable. The top 10 players will be displayed on the leaderboard

- Top 10 users' # of wins (listed in order)
- Shows the current user # of wins at the top & position on the leaderboard if applicable

Gameplay page (when user start a new room or enter a room)
The gameplay page will have the game board in the center, and a status bar and a timer on top. The two players' information will be displayed on the left and the right. The player information will be highlighted when it's that player's turn to make a move. To make a move, the player will click on the drop arrow on top of a column to drop their token. If it was not a given player's turn, the drop button is disabled and they cannot make a move. If a player's turn times out, i.e. the player did not make a move within 60 seconds, the opponent wins automatically.

- If user is currently in a game:
    - Opponent nickname and profile picture (right column, highlighted when it's their turn)
    - User's nickname and profile picture (left column, highlighted when it's their turn)
    - Interactive game board (centered in the page, AJAX polling for updates)
        - Buttons to drop tokens for each column
    - Timer for each turn (60 seconds)
        - Lose if timer runs out
    - Status bar
        - Shows “waiting for opponent”, “[player]'s turn”, “[player] wins”
        - “Surrender” button for each player if they wish to quit the game -> global page. The opponent wins automatically
- If user is not in a game:

    - Show text that directs the user to the global page to join or start a new room
    - Exit button if the user wants to close the room before another player joins -> global page
    - Link to global page -> global page

Additional features (if time permits)

- Friend's list: Users can challenge their friends to a game
- Web sockets to optimize polling
- Room codes: Users can enter a room by the room code (global page)

2. Data models

- Profile model
    - user: foreign key that links to the user object
    - token_picture: token photo that the user has uploaded; or None
    - profile_picture
    - nickname
    - bio
    - wins: number of wins the user has
    - room: foreign key that links to the room the user is playing in; or None

- Room model
    - room_id: globally unique room id assigned to the room
    - player_1: foreign key that links to the player that created the room
    - player_2: foreign key that links to the player that joined the room
    - turn: integer value 1 or 2 indicating which player's turn it is
    - board: a string of 42 comma separated strings ("", "1", or "2") that denotes token locations
    - status: the status message of the room

models.py

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

class Room(models.Model):
  room_id = models.IntegerField(default=-1) # globally unique room id
  player_1 = models.ForeignKey(User, default=None, on_delete=models.PROTECT,
related_name="player_1")
  player_2 = models.ForeignKey(User, default=None, on_delete=models.PROTECT,
related_name="player_2", null=True)
  turn = models.IntegerField(default=1)
  board = models.CharField(max_length=2000)
  status = models.CharField(max_length=200, default="")
```

```
class Profile(models.Model):
  user = models.OneToOneField(User, on_delete=models.PROTECT)
  token_picture = models.FileField(blank=True)
  profile_picture = models.FileField(blank=True)
  nickname = models.CharField(max_length=50)
  bio = models.CharField(max_length=200)
  wins = models.IntegerField(default=0)
  room = models.ForeignKey(Room, default=None, on_delete=models.PROTECT,
related_name="room", null=True)
```

3. **First sprint backlog**
   a. HTML
      i. **Login and Register page**
         - Players can register and log into their account using basic Django authentication
         - Research into OAuth and explore how to incorporate it

      ii. **Global page** to create or enter a new room
         - join_room action
            - Go to the gameplay page
            - Update the room model to fill in the “player_2” field
            - Start game on player 1’s turn
         - create_room action
            - Go to the gameplay page
            - Create an instance of the room model and fill in the “player_1” field and other fields besides “player_2” field

      iii. **Gameplay page**
         - Players take turns to place their tokens (basic gameplay)
         - Default red and yellow dots for players’ tokens
         - Checks for winning conditions and displays proper status message
         - Allow players to exit a game and starts new games after a game ends

   b. Django python files
      i. Models.py for Profile and Room data models
      ii. Forms.py for register and login
      iii. Urls.py for routing
      iv. Views.py for defining actions and sending back HTTP responses in json format to AJAX

   c. AJAX (javascript)
      i. Polling to update gameplay page for each game
      ii. Polling to update global page

    d. Other static files
        i. CSS for style

4. **Product owner of sprint #1**
- Name: Lucy Wang
- AndrewID: keyingw