Models Design

- User - default auth model provided by django
  - First Name
  - Last name
  - Email
  - Password
  - Username
- Profile
  - user (One to one field related to user model)
  - Avatar
  - Phone number
- Recipe
  - Base recipe (id of another recipe)
  - Name
  - Cuisine
  - Total prep time (hours)
  - Total prep time (minutes)
  - Total cook time (hours)
  - Total cook time (minutes)
  - User (owner of the recipe) - foreign key to User
  - diets - many to many field related to Diet model
  - servings
- Steps
  - type (preparation or cooking)
  - Steo number
  - Hours
  - Minutes
  - Description
  - Recipe - foreign key to Recipe
- RecipeFile
  - File
  - Recipe - foreign key to Recipe
- StepFile
  - File
  - step - foreign key to Step
- CommentFile - recipe
  - File
  - comment - foreign key to Comment
- Ingredient
  - Name
  - Unit
  - Amount per number of serving in recipe
  - Recipe - foreign key to Recipe
- Diet

    - Name
    - Recipe - foreign key to Recipe
- Comment
    - Recipe - foreign key to Recipe
    - User - foreign key to User
    - DateTime
    - Text
- Like
    - Recipe - foreign key to Recipe
    - User - foreign key to User
- Favorite
    - Recipe - foreign key to Recipe
    - User - foreign key to User
- Rating
    - Recipe - foreign key to Recipe
    - User - foreign key to User
    - stars
- ShoppingListRecipe
    - Recipe - foreign key to Recipe
    - user - foreign key to User
    - ServingSize - the amount of servings chosen in shopping list

TEST IN P2_TESTS ENVIRONMENT
Accounts:

- Register
    - http://127.0.0.1:8000/accounts/api/register/
    - Provide users username, first name, last name, email and password (user.<field>, since these fields belong to the user objects), avatar and phone_number
    - This method registers a new user
- Log in
    - http://127.0.0.1:8000/accounts/api/login/
    - Provide existing username and password
    - This method logs the user in (given that the username and password are correct) and generates a jwt token which is saved into the p2_tests environment
- Update Profile
    - http://127.0.0.1:8000/accounts/api/update/<id>
    - Provide first name, last name, email and password (user.<field>, since these fields belong to the user objects), avatar and phone_number
    - Headers require to submit the jwt token generated in log in
    - This method updates the profile of the user with the kwargs['id']. It only allows the logged in user to update their profile (the logged in user is not allowed to update another profile)

Social Media:

- Get all Like/Favourites/Rating/Comments
    - http://127.0.0.1:8000/recipes/user/<likes/favorites/ratings/comments>/
    - Headers require to submit the jwt token generated in log in
    - This method return all current likes/favorites/ratings
- Add Like/Favourites/Rating/Comment for recipe <id>
    - http://127.0.0.1:8000/recipes/<id>/<likes/favorites/ratings/comment>/
    - Provide the id of user and recipe (for rating provide the number of stars)
      This method adds a likes/favorites/ratings for given recipe and user

- Get Like/Favourites/Rating for recipe <recipe-id>
  - http://127.0.0.1:8000/recipes/<recipe-id>/like/
  - Headers require to submit the jwt token generated in log in
  - Returns the like for the currently logged in user and recipe with the given id
  - Additionally 'get user like for recipe 1000' checks if 404 is correctly returned
- Delete Like/Favourites/Rating for recipe <id>
  - http://127.0.0.1:8000/recipes/<id>/<likes/favorites/ratings>/
  - This method deletes a likes/favorites/ratings for given recipe and currently logged in user
- Get number of Like/Favourites/Rating for recipe <recipe-id>
  - http://127.0.0.1:8000/recipes/<recipe-id>/likes/
  - Headers require to submit the jwt token generated in log in
  - Returns the number of likes for the recipe with the given id

Recipe:

- Create Recipe
  - http://127.0.0.1:8000/recipes/create/
  - Headers require to submit the jwt token generated in log in
  - Create recipe with diets provides a template for creating a recipe with diets
  - Create recipe with ingredients provides a template for creating a recipe with ingredients
  - Create recipe with steps provides a template for creating a recipe with steps
- Update Recipe steps
  - http://127.0.0.1:8000/recipes/update/<recipe-id>/
  - Headers require to submit the jwt token generated in log in
  - This method updates the details and steps of the given recipe
- Update Recipe ingredients
  - http://127.0.0.1:8000/recipes/update/<recipe-id>/
  - Headers require to submit the jwt token generated in log in
  - This method updates the details and ingredients of the given recipe
- Get Recipe By id
  - http://127.0.0.1:8000/recipes/details/<recipe-id>/
  - Headers require to submit the jwt token generated in log in
  - Returns a recipe with the given id
- Get All Diets/Cuisines
  - http://127.0.0.1:8000/recipes/<cuisine/diet>/all/
  - Headers require to submit the jwt token generated in log in
  - Returns all diets/cuisines
- Delete Recipe
  - http://127.0.0.1:8000/recipes/delete/<recipe-id>/
  - Headers require to submit the jwt token generated in log in
  - Deletes the recipe with the given id
- Get My Recipes
  - http://127.0.0.1:8000/recipes/my-recipes/
  - Get request

ShoppingList:

- Add Recipe to shopping list
  - http://127.0.0.1:8000/recipes/shoppingList/<id>/
  - Headers require to submit the jwt token generated in log in
  - Provide user id whose shopping list will be updated, recipe id which is supposed to be added and the serving size
  - This method adds a recipe to the shopping list
- Delete Recipe from shopping list

    - http://127.0.0.1:8000/recipes/shoppingList/<id>/
    - Headers require to submit the jwt token generated in log in
    - This method deletes the recipe from the shopping list of the user (only user who added that recipe is allowed to delete it)
- Get shopping list info
    - http://127.0.0.1:8000/recipes/shoppingList/info/
    - Headers require to submit the jwt token generated in log in
    - Returns the shopping list for the currently logged in user and sums the amounts for the same ingredients
- Edit serving size
    - http://127.0.0.1:8000/recipes/shoppingList/1/
    - Headers require to submit the jwt token generated in log in
    - Lets the logged in user edit the servings for a recipe in their shopping list

Search:

- Search recipe by name
    - http://127.0.0.1:8000/recipes/api/?search=<name>
    - Headers require to submit the jwt token generated in log in
    - Replace name with the name of the recipe (works with partial searches)
    - Searches through the recipes by the recipes name and their creators name
- Search recipe by diet/cuisine
    - http://127.0.0.1:8000/recipes/api/<cuisine/diet>/?search=<name>
    - Headers require to submit the jwt token generated in log in
    - Replace name with the name of diet/cuisine (does not work with partial searches)
    - Searches through the recipes by the name of the cuisine or diet