Milestone 5

Who: Musaab Al-Bakry, Meelan Patel, Victor Boynton, Jordan Goodridge, Pamela Wyatt

Title: PanMates

Automated Tests: We use automated tests using Django. The overarching idea of how we implement automated tests using the shell to examine the behavior of a certain method or checking if the data is being loaded into the application as expected for a smaller set of functional aspects. By running ./manage.py test, Django runs an automated test based on tests written in the tests.py file.

User Acceptance Tests: Our test plan includes test cases for our diet, recipe difficulty, and login features. The diet test case essentially verifies whether a specific element of a diet is properly allocated for a recipe. Django unit tests use Python's standard library module "unittest" using a class-oriented approach. The examples used below illustrates how we test to see if vegan and nut allergy dietary restrictions are included with the corresponding recipe. We feel that including the appropriate dietary restrictions are the most important features to add for the safety of our users. Additionally, similar logic is used with the recipe difficulty level ensuring that the difficulty feature is being properly allocated for the user to have the best possible PanMate experience. The login feature asserts if a user provides the correct login credentials.

Examples of three features:

```
from django.test import testCase

from django.contrib.auth import login

from .models import Diet, Level

class DietTestCase(TestCase):
    def setUp(self):
        Diet.objects.create(name="Vegan", disc="No animal products.", tags=["vegan", "soy
milk"])
        Diet.objects.create(name="Nut Allergic", disc="No nuts", tags=["no", "nuts"])
    def test_Diet(self):
        vegan = Diet.objects.get(name="Vegan")
        self.assertEqual(vegan.disc, "No animal products.")
        nut_allergic = Diet.objects.get(name="Nut Allergic")
        self.assertEqual(nut_allergic, "No nuts”)

class LoginTestCase(TestCase):
    def test_login(self):
        login(user="admin", password="theAteam")

class LevelTestCase(TestCase):
    def setUp(self):
        Level.objects.create(name="Beginner", difficulty=1, disc="Novice")
    def test_level(self):
        beginner = Level.objects.get(difficulty=1)
        self.assertEqual(beginner.name, "Beginner")
```