**System and Unit Test Report**

*Product:* $h0pLyf3 (Shopping List)

*Team:* Fantastic Five

*Date:* Sunday, December 3rd, 2017

Link to system & unit test template pdf (remember to **remove** later):

<https://piazza-resources.s3.amazonaws.com/j85ih2df9ug4gx/jal6w8eb4fs7je/SystemUnitTestTemplate.pdf?AWSAccessKeyId=AKIAIEDNRLJ4AZKBW6HA&Expires=1512350377&Signature=8AS9o%2FYvEZQJSNxglykqZunxtjs%3D>

We planned to extensively unit test all of the functions that modified data on our website. Unfortunately we ran into an error so large that we were not able to implement those tests. Naturally, to unit test our functions we need to be able to import the functions to be tested into our test file. Unfortunately, we were unable to import anything from that class because of this error: “ImportError: attempted relative import with no known parent package”. This error was peculiar but prevented us from even running tests from within the file that housed the functions themselves. That is because Django allows those functions to run when the application is running itself, but the import functions do not work correctly when run in isolation. We planned to extensively test each of the functions with the Django Testing environment, but because of this error, we were not able to implement our planned tests.

**System Tests**

Sprint 1:

A: As a shopper, I want to login to my account.

1. Navigate to[http://54.241.142.236/](http://54.241.142.236/signup)signup in a web browser. This will allow you to create an account. For testing purposes, use the following:

1. Username = ‘StudentJoe’
2. First name = ‘Joe’
3. Email = ‘[testemail@Email.com](mailto:testemail@Email.com)’
4. Password = test888case

2. The user should be redirected to [http://54.241.142.236/](http://54.241.142.236/signup)[login](http://www.shoplyfe.net/signup).

3. Entering the correct credentials will then redirect to [http://54.241.142.236/](http://54.241.142.236/signup)home for the user.

4. The user should now be logged in.

B: As a developer, I want a database set up.

1. Django comes packaged with sqlit3.

2. To create a database in sqlit3

a. Create a model in models.py

b. Run makemigrations and migrate in the shell

c. A database should now exist based on the model specified

C: As a developer, I want to learn Django. N/A

**Sprint 2:**

A: As a shopper, I want to be able to login to my account on a publicly accessible server so that I can use the application from anywhere.

1. See Sprint 1, story 1, however the site is now accessible from any machine.

**Sprint 3:**

A: As a shopper, I want to be able to keep track of what I’m buying at the grocery store.

1. The user should login using his credentials at h[ttp://54.241.142.236/](http://54.241.142.236/signup)[login](http://www.shoplyfe.net/signup).

2. By navigating to<http://54.241.142.236/shopping_list>, a or clicking on the shopping list link on the left hand side, the user will be brought to his personalized shopping list. If no recipes have been added, it will appear empty.

3. If recipes have been added, then:

1. Check boxes should appear next to each ingredient.
2. Clicking on a box will mark the ingredient as bought. Clicking on it again will revert the change.
3. This allows users to keep track of which items have been purchased or picked up at the store.

B: As a shopper, I would like a shopping list to be generated from the recipes I have chosen so that I can save time creating a grocery list.

1. After adding a recipe at<http://54.241.142.236/recipes>, a user will have the ability to add them to his list. Clicking the plus sign to the right of the recipe will add one instance of it to the shopping list.

2. The user should then be able to proceed to [http://54.241.142.236/](http://54.241.142.236/recipes)shopping\_list, or click on the shopping list link to see a list of ingredients generated. The list will display:

1. Correct amount of ingredients to buy based on user input
2. Correct tags
3. Correct ingredient type

3. Note also that clicking on a recipe in the recipes page will redirect to a page containing the details of said recipe.

C: As a shopper, I would like to be able to remove ingredients I have from my shopping list, so that I don’t buy extra stuff I don’t need.

1. After adding a recipe at<http://54.241.142.236/recipes>, a user will have the ability to remove items from his list. Clicking the minus sign will remove one instance of a recipe from the list, and clicking the rubbish bin will permanently delete the recipe from the user’s database.

2. The user should then be able to proceed to [http://54.241.142.236/](http://54.241.142.236/recipes)shopping\_list, or click on the shopping list link to see a list of ingredients generated. Any changes will be reflected on this page.

D: As a shopper, I want to be able to add my own recipes as well as remove them.

1. At<http://54.241.142.236/recipes> there is a link, Add Recipes+ that will redirect to<http://54.241.142.236/home/recipes/form/>.

2. A form will be displayed where a recipe can be added. For test purposes, a recipe has been supplied.

A. Name = ‘Salad’

B. Prep time = ‘15’

C. Cook time = ‘5’

D. Tag = ‘healthy’

3. After adding recipe details, an ingredient should now be added. Two ingredients have been supplied.

A. Name = ‘Lettuce’

B. Quantity = ‘1’

C. Type = Head

4. Clicking Done will save the recipe and redirect to <http://54.241.142.236/recipes>. Clicking add ingredient will redirect to [http://54.241.142.236/](http://54.241.142.236/recipes)ingredient\_form.

A. Name = ‘Cherry Tomato’

B. Quantity = ‘2’

C. Type = Cups