Overall Test Plan

First, all individual components are tested as they are developed. They are tested for functionality, as well as how to deal with incorrect input data. Once they work individually against the functional requirements, they are integrated in with the other parts of the project if not already done so. After verifying that the new additions did not hinder any prior workings of the project they are deemed as functionally done. Then comes user testing. Another user, that is not the developer, will then use the new functionality to verify if it does work.

Test Case Descriptions

D1.1 **Database Test 1**
D1.2 Create a new User for the system
D1.3 This test inputs a known dataset into the database of users. Then upon log in, this information is queried from the database and verified as a valid user for log in.
D1.4 Input: The input will be the fields used in the user creation.
D1.5 Output: A new User will be added into the database.
D1. 6 Normal
D1. 7 Blackbox
D1. 8 Functional
D1. 9 Unit

D2.1 **Database Test 2**
D2.2 View recipe information
D2.3 This test will query the database based off of a link. The query will then pull the correct corresponding information and present it on an html page.
D2.4 Input: A link to a html page that shows a certain recipe's information
D2.5 Output: A html page that shows all information gotten from the database for a recipe.
D2.6 Normal
D2.7 Blackbox
D2.8 Functional
D2.9 Unit

D3.1 **Database Test 3**
D3.2 Delete Recipe
D3.3 This test will remove a specified entry from the database.
D3.4 Input: A recipe that already exists in the database
D3.5 Output: The recipe entry will no longer exist in the database
D3.6 Normal
D3.7 Blackbox
D3.8 Functional
D3.9 Unit

I1.1 **Input Test 1**
I1.2 Edit Recipe Information

I1.3 This test will query the database for the current information of the recipe. The information will be presented in a modifiable format. Any changes will be saved and the database entry will be updated with any new information.
I1.4 Input: A recipe that already exists in the database
I1.5 Output: The recipe information in the database will be updated with new information.
I1.6 Normal
I1.7 Blackbox
I1.8 Functional
I1.9 Integration

I2.1 **Input Test 2**
I2.2 Add new recipe manually
I2.3 This test will add a recipe to the database using the manual format. The user will be able to input a varied size of step and ingredient fields, and the data will be verified for correct input type before entering into the database as an entry.
I2.4 Input: User entered data for all necessary fields to create a recipe entry.
I2.5 Output: A new recipe entry will be added into the database.
I2.6 Normal
I2.7 Blackbox
I2.8 Functional
I2.9 Integration

I3.1 **Input Test 3**
I3.2 Manually change password for user
I3.3 This test will verify that an existing user can modify their own password.
I3.4 Input: A string that does not match the current password for the user.
I3.5 Output: The password will be updated for the user in the database.
I3.6 Normal
I3.7 Blackbox
I3.8 Functional
I3.9 Integration

S1.1 **Server Test 1**
S1.2 Parse recipe portions from url
S1.3 This test will verify that the server will pull the correct information for the recipe from a given url.
S1.4 Input: A url to a webpage that has a known recipe on it.
S1.5 Output: The server will parse the recipe information from the webpage for the user to view.
S1.6 Normal
S1.7 Whitebox
S1.8 Functional
S1.9 Unit

UI1.1 **User Interface Test 1**
UI1.2 An existing user can log in
UI1.3 This test will verify that an existing user in the database can log in.
UI1.4 Input: The user credentials of an existing user in the database
UI1.5 Output: The user will be directed to a home page upon successful log in.
UI1.6 Normal
UI1.7 Blackbox
UI1.8 Functional
UI1.9 Integration

UI2.1 **User Interface Test 2**
UI2.2 User can view a list of all recipes in the database
UI2.3 This test will verify that a logged-in user is able to view all recipes in the database as links.
UI2.4 Input: A logged-in user
UI2.5 Output: A webpage that lists links to all known recipe entries in the database.
UI2.6 Normal
UI2.7 Blackbox
UI2.8 Functional
UI2.9 Unit

UI3.1 **Interface Test 3**
UI3.2 Add recipe via url
UI3.3 This test will verify that a user can add a recipe into the database from a url.
UI3.4 Input: A url to a webpage that has a known recipe on it.
UI3.5 Output: A new recipe entry that reflects the information parsed from the url.
UI3.6 Normal
UI3.7 Blackbox
UI3.8 Functional
UI3.9 Integration

Test Case Matrix

	Normal/Abnormal	**Blackbox/White box**	**Functional/Performance**	**Unit/Integration**
D1	**Normal**	**Blackbox**	**Functional**	**Unit**
D2	**Normal**	**Blackbox**	**Functional**	**Unit**
D3	**Normal**	**Blackbox**	**Functional**	**Unit**

I1	**Normal**	**Blackbox**	**Functional**	**Integration**
I2	**Normal**	**Blackbox**	**Functional**	**Integration**
I3	**Normal**	**Blackbox**	**Functional**	**Integration**
S1	**Normal**	**Whitebox**	**Functional**	**Unit**
UI1	**Normal**	**Blackbox**	**Functional**	**Integration**
UI2	**Normal**	**Blackbox**	**Functional**	**Unit**
UI3	**Normal**	**Blackbox**	**Functional**	**Integration**