



Test Cases

Wine Application: Wine-Know

Terroir Integrated Technology Specialists, T.I.T.S.

Shahrzad Gustafson	Project Manager
Collin Styles	Algorithm Specialist
Tim Chi	Software Architect
Janne Rapakko	Software Development Lead
Jagtej Sodhi	Software Development Lead
Jaysen Parmar	User Interface Specialist
Ankit Agarwal	Senior System Analyst
Ricardo Cruell	Subject Matter Expert
Kyle So	Quality Assurance Lead
Trent Russell	Database Specialist

Table of Contents

<u>Title (Test Case #)</u>	<u>Page</u>
Sip Test Cases	
Recommend Random Wine (TC 1)	6
Recommend Specific Wine: Red (TC 2)	8
Recommend Specific Wine: White (TC 3)	10
Recommend Specific Wine: Cabernet Sauvignon (TC 4.0)	12
Recommend Specific Wine: Chardonnay (TC 4.1)	14
Recommend Specific Wine: Merlot (TC 4.2)	16
Recommend Specific Wine: Pinot Noir (TC 4.3)	18
Recommend Specific Wine: Riesling (TC 4.4)	20
Recommend Specific Wine: Syrah/Shiraz (TC 4.5)	22
Recommend Specific Wine: By Descriptor (TC 5)	24
Eat Test Cases	
Wine Pairings (TC 6)	26
Recipes (TC 7)	29
Rating Test Cases	
Give Ratings (TC 8)	32
View Ratings (TC 9)	34
Terms Test Cases	
View Wine Terms (TC 10)	36
View Wine Term Definitions (TC 11)	37
Random Facts Test Cases	
View Random Fact (TC 12)	39

Assisting Test Cases

Get Help (TC 13)	41
Go Home (TC 14)	43

Test Case Key

Description: A summary of what that specific test case does

Desired Outcome: What is supposed to happen

User Goals: What the user wants to accomplish in this action

Dependent Test Cases: If the test case is dependent on any previous ones to work

Involved Requirements: The corresponding requirements for the Test Case

Details:

Priority - The urgency of the test case to the application

- 0 - Completed
- 1 - Highest Priority, Vital to Application
- 2 - High Priority
- 3 - Low Priority, Implemented if given time

Progress- The current phase the test case is in

- Planning - Test Case is currently being considered and planned
- Designing- Test Case is being laid-out and designed with team
- Developing- Test Case is currently being implemented
- Testing - Test Case is in test phase, being tested with the team and potential users
- Completed - Test Case is fully implemented into application and currently functioning
- Deferred- Development of test case is currently on hold

Test Phase - The current testing phase the test case is in

- Planned - Testing will commence when development is completed
- In Progress - Currently testing with team
- Complete - Test phase complete, Test Case currently functional
- Deferred - Development of test case is currently on hold, therefore testing is as well

Pre-Conditions: Actions that have to be true, before this Test Case begins

Post-Conditions: The outcome of the Test Case being completed

Trigger: How the Test Case begins (or how it is “triggered”)

Workflow: The user and system steps involved in the Test Case

Alternate Paths: Other paths the user can take if they decide not to complete the Test Case

Terms

Random Wine - A wine that is completely random. The system randomly selects a wine (no user input)

Specific Wine - A specific wine is one that is recommended by the system based off of some specifications, which were inputted by the user. The option the user has in selecting a specific wine is either by color, varietal or descriptor

Food Items - When the term 'Food Items' is used, it includes: Cured Meat, Red Meat, White Meat, Rich Fish, Fish, Bread, Roasted Vegetables, Vegetables, and Sweets. The pairings are based off a wine infographic (Winepairings.png). The wine returned is a varietal that tends to usually pair with that food item according to the info graphic

Type of Dish - When 'kind of dish' or 'dish' is used, it includes: Red Meat, White Meat, Fish, Starches and Vegetarian

Varietals - When the user is given the option to choose a wine by varietal this refers to 6 different varietals. These varietals include: Cabernet Sauvignon, Chardonnay, Merlot, Pinot Noir, Riesling and Syrah/Shiraz

Wine Glass Rating - A wine glass rating is the same thing as a 5 start rating, instead wine glasses are used. Below is an example of a '4 Wine Glasses'



Descriptor - The user is given 11 descriptors to choose from in one of the sip functionalities. The options are: Light, Fruity, Smooth, Supple, Earthy, Spicy, Big, Bold, Crisp, Rich, Creamy. The descriptors are from wine.com

Test Case #1: Recommend Random Wine

Description:

This test case outlines the ability for the user to be recommended a Random wine

Desired Outcome:

The user shall be prompted with a random wine that was recommended by the system

User Goals:

The user does not know what type of wine they want to be recommend so they would like to see what random wine the app will recommended

Dependent Test Cases:

None

Involved Requirements:

SR 01 (Recommend Random Wine)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is a random wine generator, we cannot predict the exact information that will be displayed but an example of the Information that will be display appears as so:

Name: Franciscan Cabernet Sauvignon 2010

Varietal: Cabernet Sauvignon

Color: Red

Year: 2010

Trigger:

The user shall select the “Random Wine” button from the Sip page

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**

3. From the Sip page, the user shall click on: **Random**
4. The system shall display to the user: The Wine name, the varietal, the color of the wine and the year of the wine. (See Expect Results for an example)
5. The system shall also display a button that says: **Next**
6. The user shall click on: **Next**
7. The system shall display to the user: the wine name, the varietal, the color of the wine, and the year of the wine (This information should be different than from Step 4)

Expected Results:

The test case passes if the following is true:

Because this is a random wine generator, we cannot predict the exact information that will be displayed. There could be a possibility that when the user presses the next button, it could display the same wine again) but an example of the information that will be displayed appears as so:

Name: Franciscan Cabernet Sauvignon 2010

Varietal: Cabernet Sauvignon

Color: Red

Year: 2010

Alternate Path:

None

Test Case #2: Recommend Specific Wine: Red

Description:

This test case outlines the ability for the user to be recommended a Specific Red Wine. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific color in mind, Red.

Desired Outcome:

The user shall be recommended a red wine.

User Goals:

The user wants to be recommended a red wine

Dependent Test Cases:

None

Involved Requirements:

SR 02 (Recommend Specific Wine: Red)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine (which should be Red), and year. Because this is still random but filtered by color, we cannot predict the exact information that will be displayed but an example of the information that will be displayed appears as so:

Name: Franciscan Cabernet Sauvignon 2010

Varietal: Cabernet Sauvignon

Color: Red

Year: 2010

Trigger:

The user wants a specific red wine and they shall select 'Red' under the 'Color' category

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Color**
6. The system shall take you to the color page that will display two buttons: **Red and White**
7. From the color page, the user shall click on: **Red**
8. The system shall display to the user: The Wine name, the varietal, the color of the wine (which in this case is **Red**), and the year of the wine. (See Expected Results for an example)
9. The system shall also display a button that says: **Next**
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal, the color of the wine (which in this case is **Red**), and the year of the wine (This information should be different than from Step 8)

Expected Results:

The test case passes if the following is true:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a red wine is returned. Because the user only specified that they want a red wine there is still some random element to this. Therefore we cannot predict which red wine it will display but an example would be

Name: Franciscan Cabernet Sauvignon 2010

Varietal: Cabernet Sauvignon

Color: Red

Year: 2010

Alternate Path:

None

Test Case #3: Recommend Specific Wine: White

Description:

This use case outlines the ability for the user to be recommended a Specific White Wine. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific color in mind, White

Desired Outcome:

The user shall be recommended a white wine

User Goals:

The user wants to be recommended a white wine

Dependent Test Cases:

None

Involved Requirements:

SR 03 (Recommend Specific Wine: White)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine (which should be White), and year. Because this is still random but filtered by color, we cannot predict the exact information that will be displayed but an example of the information that will be displayed appears as so:

Name: Rombauer Chardonnay 2011

Varietal: Chardonnay

Color: White

Year: 2011

Trigger:

The user wants a specific white wine and they shall select 'White' under the 'Color' category

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Color**
6. The system shall take you to the color page that will display two buttons: **Red and White**
7. From the color page, the user shall click on: **White**
8. The system shall display to the user: The Wine name, the varietal, the color of the wine (which in this case is **White**), and the year of the wine. (See Expected Result for example)
9. The system shall also display a button that says: **Next**
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal, the color of the wine (which in this case is **White**), and the year of the wine (This information should be different than from Step 8)

Expected Results:

The test case passes if the following is true:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a white wine is returned. Because the user only specified that they want a white wine, there is still some random element to this. Therefore we cannot predict which white wine it will display but an example would be

Name: Rombauer Chardonnay 2011

Varietal: Chardonnay

Color: White

Year: 2011

Alternate Path:

None

Test Case #4.0: Recommend Specific Wine: Cabernet Sauvignon

Description:

This use case outlines the ability for the user to be recommended a Specific Wine by Varietal. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific Varietal in mind (Cabernet Sauvignon)

Desired Outcome:

The user shall be recommended a wine based off of the varietal they specify, Cabernet Sauvignon

User Goals:

The user wants to be recommended a wine based off of a specific varietal, Cabernet Sauvignon

Dependent Test Cases:

None

Involved Requirements:

SR 04 (Recommend Specific Wine: Varietal)

Details:

Priority – 0

Progress Status – Completed

Test Phase – Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is still random but filtered by varietal, we cannot predict the exact information that will be displayed but the varietal should be Cabernet Sauvignon. An example of the Information that will be display appears as so:

Name: Franciscan Cabernet Sauvignon 2010

Varietal: Cabernet Sauvignon

Color: Red

Year: 2010

Trigger:

The user wants a specific wine by varietal and they will select 'Cabernet Sauvignon' from the Varietals page

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Varietal**
6. The system shall take you to the varietal page that will display 6 buttons: **Cabernet Sauvignon, Chardonnay, Merlot, Pinot Noir, Riesling and Syrah/Shiraz**
7. From the varietal page, the user shall click on: **Cabernet Sauvignon**
8. The system shall display to the user: The wine name, the varietal (which in this case is **Cabernet Sauvignon**), the color of the wine, and the year of the wine.
9. The system shall also display a button that says: **Next**
(If the user does not like the recommended Cabernet Sauvignon they can continue to be recommended different Cabernet Sauvignon wines)
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal (which in this case is **Cabernet Sauvignon**), the color of the wine, and the year of the wine (This information should be different than from Step 8)

Expected Results:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a Cabernet Sauvignon wine is returned. Because the user only specified that they want a Cabernet Sauvignon, there is still some random element to this. Therefore we cannot predict which Cabernet Sauvignon will display but an example would be

Name: Franciscan Cabernet Sauvignon 2010

Varietal: Cabernet Sauvignon

Color: Red

Year: 2010

Alternate Path:

None

Test Case #4.1: Recommend Specific Wine: Chardonnay

Description:

This use case outlines the ability for the user to be recommended a Specific Wine by Varietal. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific Varietal in mind (Chardonnay)

Desired Outcome:

The user shall be recommended a wine based off of the varietal they specify, Chardonnay

User Goals:

The user wants to be recommended a wine based off of a specific varietal, Chardonnay

Dependent Test Cases:

None

Involved Requirements:

SR 04 (Recommend Specific Wine: Varietal)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is still random but filtered by varietal, we cannot predict the exact information that will be displayed but the varietal should be Chardonnay. An example of the Information that will be display appears as so:

Name: Rombauer Chardonnay 2011

Varietal: Chardonnay

Color: White

Year: 2011

Trigger:

The user wants a specific wine by varietal and they will select 'Chardonnay' from the Varietals page

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Varietal**
6. The system shall take you to the varietal page that will display 6 buttons: **Cabernet Sauvignon, Chardonnay, Merlot, Pinot Noir, Riesling and Syrah/Shiraz**
7. From the varietal page, the user shall click on: **Chardonnay**
8. The system shall display to the user: The wine name, the varietal (which in this case is **Chardonnay**), the color of the wine, and the year of the wine. (View Expect Results for an example)
9. The system shall also display a button that says: **Next**
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal (which in this case is **Chardonnay**), the color of the wine, and the year of the wine (This information should be different than from Step 8)

Expected Results:

The test case passes if the following is true:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a Chardonnay wine is returned. Because the user only specified that they want a Chardonnay, there is still some random element to this. Therefore we cannot predict which Chardonnay will display but an example would be

Name: Rombauer Chardonnay 2011

Varietal: Chardonnay

Color: White

Year: 2011

Alternate Path:

None

Test Case #4.2: Recommend Specific Wine: Merlot

Description:

This use case outlines the ability for the user to be recommended a Specific Wine by Varietal. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific Varietal in mind (Merlot)

Desired Outcome:

The user shall be recommended a wine based off of the varietal they specify, Merlot

User Goals:

The user wants to be recommended a wine based off of a specific varietal, Merlot

Dependent Test Cases:

None

Involved Requirements:

SR 04 (Recommend Specific Wine: Varietal)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is still random but filtered by varietal, we cannot predict the exact information that will be displayed but the varietal should be Merlot. An example of the Information that will be display appears as so:

Name: Tilia Merlot 2011

Varietal: Merlot

Color: Red

Year: 2011

Trigger:

The user wants a specific wine by varietal and they will select 'Merlot' from the Varietals page

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Varietal**
6. The system shall take you to the varietal page that will display 6 buttons: **Cabernet Sauvignon, Chardonnay, Merlot, Pinot Noir, Riesling and Syrah/Shiraz**
7. From the varietal page, the user shall click on: **Merlot**
8. The system shall display to the user: The wine name, the varietal (which in this case is **Merlot**), the color of the wine, and the year of the wine.
9. The system shall also display a button that says: **Next**
(If the user does not like the recommended Merlot they can continue to be recommended different Merlot wines)
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal (which in this case is Merlot), the color of the wine, and the year of the wine (This information should be different than from Step 8)

Expected Results:

The test case passes if the following is true:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a Merlot wine is returned. Because the user only specified that they want a Merlot, there is still some random element to this. Therefore we cannot predict which Merlot will display but an example would be

Name: Tilia Merlot 2011

Varietal: Merlot

Color: Red

Year: 2011

Alternate Paths:

None

Test Case #4.3: Recommend Specific Wine: Pinot Noir

Description:

This use case outlines the ability for the user to be recommended a Specific Wine by Varietal. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific Varietal in mind (Pinot Noir)

Desired Outcome:

The user shall be recommended a wine based off of the varietal they specify, Pinot Noir

User Goals:

The user wants to be recommended a wine based off of a specific varietal, Pinot Noir

Dependent Test Cases:

None

Involved Requirements:

SR 04 (Recommend Specific Wine: Varietal)

Details:

Priority – 0

Progress Status – Completed

Test Phase – Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is still random but filtered by varietal, we cannot predict the exact information that will be displayed but the varietal should be Pinot Noir. An example of the Information that will be display appears as so:

Name: Belle Glos Meiomi Pinot Noir 2012

Varietal: Pinot Noir

Color: Red

Year: 2012

Trigger:

The user wants a specific wine by varietal and they will select 'Pinot Noir' from the Varietals page

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Varietal**
6. The system shall take you to the varietal page that will display 6 buttons: **Cabernet Sauvignon, Chardonnay, Merlot, Pinot Noir, Riesling and Syrah/Shiraz**
7. From the varietal page, the user shall click on: **Pinot Noir**
8. The system shall display to the user: The wine name, the varietal (which in this case is **Pinot Noir**), the color of the wine, and the year of the wine.
9. The system shall also display a button that says: **Next**
(If the user does not like the recommended Pinot Noir they can continue to be recommended different Pinot Noir wines)
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal (which in this case is Pinot Noir), the color of the wine, and the year of the wine (This information should be different than from Step 8)

Expected Results:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a Pinot Noir wine is returned. Because the user only specified that they want a Pinot Noir, there is still some random element to this. Therefore we cannot predict which Pinot Noir will display but an example would be

Name: Belle Glos Meiomi Pinot Noir 2012

Varietal: Pinot Noir

Color: Red

Year: 2012

Alternate Paths:

None

Test Case #4.4: Recommend Specific Wine: Riesling

Description:

This use case outlines the ability for the user to be recommended a Specific Wine by Varietal. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific Varietal in mind (Riesling)

Desired Outcome:

The user shall be recommended a wine based off of the varietal they specify, Riesling

User Goals:

The user wants to be recommended a wine based off of a specific varietal, Riesling

Dependent Test Cases:

None

Involved Requirements:

SR 04 (Recommend Specific Wine: Varietal)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is still random but filtered by varietal, we cannot predict the exact information that will be displayed but the varietal should be Riesling. An example of the Information that will be display appears as so:

Name: Pacific Rim Organic 2011

Varietal: Riesling

Color: White

Year: 2011

Trigger:

The user wants a specific wine by varietal and they will select 'Riesling' from the Varietals page

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Varietal**
6. The system shall take you to the varietal page that will display 6 buttons: **Cabernet Sauvignon, Chardonnay, Merlot, Pinot Noir, Riesling and Syrah/Shiraz**
7. From the varietal page, the user shall click on: **Riesling**
8. The system shall display to the user: The wine name, the varietal (which in this case is **Riesling**), the color of the wine, and the year of the wine.
9. The system shall also display a button that says: **Next**
(If the user does not like the recommended Riesling they can continue to be recommended different Riesling wines)
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal (which in this case is Riesling), the color of the wine, and the year of the wine (This information should be different than from Step 8)

Expected Results:

The test case passes if the following is true:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a Riesling wine is returned. Because the user only specified that they want a Riesling, there is still some random element to this. Therefore we cannot predict which Riesling will display but an example would be

Name: Pacific Rim Organic 2011

Varietal: Riesling

Color: White

Year: 2011

Alternate Paths:

None

Test Case #4.5: Recommend Specific Wine: Syrah/Shiraz

Description:

This use case outlines the ability for the user to be recommended a Specific Wine by Varietal. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific Varietal in mind (Syrah/Shiraz)

Desired Outcome:

The user shall be recommended a wine based off of the varietal they specify, Syrah/Shiraz

User Goals:

The user wants to be recommended a wine based off of a specific varietal, Syrah/Shiraz

Dependent Test Cases:

None

Involved Requirements:

SR 04 (Recommend Specific Wine: Varietal)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is still random but filtered by varietal, we cannot predict the exact information that will be displayed but the varietal should be Syrah/Shiraz. An example of the Information that will be display appears as so:

Name: Hogue Genesis Syrah 2009

Varietal: Syrah/Shiraz

Color: Red

Year: 2009

Trigger:

The user wants a specific wine by varietal and they will select 'Syrah/Shiraz' from the Varietals page

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Varietal**
6. The system shall take you to the varietal page that will display 6 buttons: **Cabernet Sauvignon, Chardonnay, Merlot, Pinot Noir, Riesling and Syrah/Shiraz**
7. From the varietal page, the user shall click on: **Syrah/Shiraz**
8. The system shall display to the user: The wine name, the varietal (which in this case is **Syrah/Shiraz**), the color of the wine, and the year of the wine.
9. The system shall also display a button that says: **Next**
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal (which in this case is Syrah/Shiraz), the color of the wine, and the year of the wine (This information should be different than from Step 8)

Expected Results:

The test case passes if the following is true:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a Riesling wine is returned. Because the user only specified that they want a Syrah/Shiraz, there is still some random element to this. Therefore we cannot predict which Syrah/Shiraz will display but an example would be

Name: Hogue Genesis Syrah 2009

Varietal: Syrah/Shiraz

Color: Red

Year: 2009

Alternate Paths:

None

Test Case #5: Recommend Specific Wine: Descriptor

Description:

This use case outlines the ability for the user to be recommended a Specific Wine by Descriptor. The user has in mind a wine they want based off of either: Color, Varietal or Descriptor and in this case they have a specific Descriptor in mind (The options of descriptors are stated in Terms). In this test case we will use the Descriptor **Fruity**

Desired Outcome:

The user shall be recommended a wine based off of the descriptor they specify, **Fruity**

User Goals:

The user wants to be recommended a wine based off of a specific descriptor, **Fruity**

Dependent Test Cases:

None

Involved Requirements:

SR 05 (Recommend Specific Wine: Descriptor)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is still random but filtered by description, we cannot predict the exact information that will be displayed. The wine displayed should be considered 'fruity'. An example of the Information that will be display appears as so:

Name: Innocent Bystander Pinot Noir 2011

Varietal: Pinot Noir

Color: Red

Year: 2011

Trigger:

The user wants a specific wine by descriptor and they will select 'Fruity' from the descriptor page

Workflow:

1. From the home page, the user shall click on: **Sip**
2. The system shall take you to the Sip page and display two buttons: **Random and Specific**
3. From the Sip page, the user shall click on: **Specific**
4. The system shall take you to the Specific Page and display three buttons: **Color, Varietal, and Descriptors**
5. From the Specific Page, the user shall click on: **Descriptors**
6. The system shall take you to the varietal page that will display 11 buttons: **Light, Fruity, Smooth, Supple, Earthy, Spicy, Big, Bold, Crisp, Rich, Creamy**
7. From the varietal page, the user shall click on: **Fruity**
8. The system shall display to the user: The wine name, the varietal, the color of the wine, and the year of the wine.
9. The system shall also display a button that says: **Next**
10. The user shall click on: **Next**
11. The system shall display to the user: the wine name, the varietal, the color of the wine, and the year of the wine (This information should be different than from Step 8)

Expected Results:

The test case passes if the following is true:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if a wine that is considered to meet that descriptor is returned. Because the user only specified that they want a fruity wine there is still some random element to this. Therefore we cannot predict which fruity wine will display but an example would be

Name: Innocent Bystander Pinot Noir 2011

Varietal: Pinot Noir

Color: Red

Year: 2011

Alternate Paths:

None

Test Case #6: Wine Pairings

Description:

This test case outlines the ability for the user to be recommended a wine that goes with a specific food item. (See Terms to see what these food items are). In this test case we will be seeing which wines pair nicely with Red Meat, Rich Fish, and Vegetables

Desired Outcome:

The user shall be suggested a wine that goes with a specific food item that they have indicated. The food items will be Red Meat, Rich Fish, and Vegetables.

User Goals:

The user wants to be suggested a wine that pairs nicely with the food item they indicated

Dependent Test Cases:

None

Involved Requirements:

SR 06 (Wine Pairings)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The user is taken to the display wine page, which will be displaying: the name, the varietal, the color of the wine, and year. Because this is still random but filtered by varietal (each pairing has a list of varietals it is known to pair with), we cannot predict the exact information that will be displayed. The wine displayed should be a varietal that pairs with that food item. An example of the information that will be display appears as so:

Rich Fish Pairing: (According to the diagram Pinot Noir pairs with Rich Fish)

Name: Innocent Bystander Pinot Noir 2011

Varietal: Pinot Noir

Color: Red

Year: 2011

Trigger:

The user is seeking a wine that pairs with the specific food item they are eating so they shall select the "Wine Pairings" button from the eat page

Workflow:

1. From the home page, the user shall click on: **Eat**
2. The system shall take you to the Eat page and display two buttons: **Pairings and Recipes**
3. From the Eat page, the user shall click on: **Pairings**
4. The system shall take you to the Pairings Page and display the nine pairings: **Sweets, Cured Meat, Red Meat, White Meat, Rich Fish, Fish, Breads, Roasted Vegetables, Vegetables**
5. From the Pairings Page, the user shall click on: **Red Meat**
6. The system shall display to the user: The wine name, the varietal, the color of the wine, and the year of the wine.
(The varietal returned and the food item should be consistent with the info graphic, in this case Red Meat goes with any of the varietals listed in the medium red and bold red wines)
7. The system shall display to the user a back button
8. The user wants to view another wine pairing, the user shall click on the: **back button**
9. The system shall take you to the Pairings page and display the nine pairings: **Sweets, Cured Meat, Red Meat, White Meat, Rich Fish, Fish, Breads, Roasted Vegetables, Vegetables**
10. From the Pairings Page, the user shall click on: **Rich Fish**
11. The system shall display to the user: The wine name, the varietal, the color of the wine, and the year of the wine.
(The varietal returned and the food item should be consistent with the info graphic, in this case Rich Fish goes with any of the varietals listed in the light red and rich white wines)
12. The system shall display to the user a back button
13. The user wants to view another wine pairing, the user shall click on the: **back button**
14. The system shall take you to the Pairings page and display the nine pairings: **Sweets, Cured Meat, Red Meat, White Meat, Rich Fish, Fish, Breads, Roasted Vegetables, Vegetables**
15. From the Pairings Page, the user shall click on: **Vegetables**
16. The system shall display to the user: The wine name, the varietal, the color of the wine, and the year of the wine.
(The varietal returned and the food item should be consistent with the info graphic, in this case Vegetables goes with any of the varietals listed in dry white wines)

Expected Results:

The test case passes if the following is true:

The user is brought to a screen displaying a wine matching their specific characteristics. This functionality works correctly if the pairing selected and the varietal of the wine returned are corresponding in the wine info graphic. Because the wines are being filter by varietal, there is still a random aspect to it

In this case the varietals that match with Red Meat: Cabernet Sauvignon, Monastrell, Aglianico, Malbec, Syrah, Red Table Wine, Temprallino, Sangiovese, Zainfandel, Grenache, Merlot

In this case the varietals that match with Rich Fish: Viognier, St.Laurent, Pinot Noir, Zweigelt, Gamay, Chardonnay, Roussanne, Marsanne,

In this case the varietals that match with Vegetables: Sauvignon Blanc, Pinot Grigio, Gruner Veltliner, Albarino, White Table Wlne

This would be an example output of Red Meat Pairing:

Name: Vodegas Ostatu Rioja

Varietal: Tempranillo

Color: Red

Year: 2009

This would be an example output of Rich Fish Pairing:

Name: Innocent Bystander Pinot Noir 2011

Varietal: Pinot Noir

Color: Red

Year: 2011

This would be an example output of Vegetable Pairing:

Name: Baytsen Buitenvverwachting Sauvignon Blanc

Varietal: Sauvignon Blanc

Color: White

Year: 2012

Alternate Paths:

None

Test Case #7: Multiple Recipes

Description:

This test case outlines the ability for the user to view different recipes, based on a type of dish (See Terms), that uses wine as an ingredient

Desired Outcome:

The user shall be suggested a specific recipe that satisfies both the type of dish the user selected and the recipe incorporates wine as one of the ingredients

User Goals:

The user wants to view a recipe that satisfies the type of dish they want to cook and also that the recipe incorporates wine into the dish

Dependent Test Cases:

None

Involved Requirements:

SR 07 (Recipes)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The system shall display a recipe that satisfies the type of dish the user wanted and also uses wine as an ingredient. The system will display an image that consists of the ingredients, the directions for the recipe, and an image of the recipe.

Trigger:

The user is seeking a recipe that involves cooking with wine and they have in mind a type of dish they want to cook so they shall select the “Recipes” button from the Eat page

Workflow:

1. From the home page, the user shall click on: **Eat**
2. The system shall take you to the Eat page and display two buttons: **Pairings and Recipes**
3. From the Eat page, the user shall click on: **Recipes**

4. The system shall take you to the Recipes Page and display the five recipe options: **Red Meat, White Meat, Fish, Vegetables, and Starches**
5. The user shall click on: **Red Meat**
6. The system shall display a recipe that has ingredients (which includes a wine), the directions, and an image of the dish. The recipe should be a recipe for Flank Steak (a red meat)
7. The system shall display a button below the recipe that says: Back to Recipes
8. The user shall click on: **Back to Recipes**
9. The system shall take the user back to the Recipes page and display the five recipe options: **Red Meat, White Meat, Fish, Vegetables, and Starches**
10. The user shall click on: **White Meat**
11. The system shall display a recipe that has ingredients (which includes a wine), the directions, and an image of the dish. The recipe should be a recipe for Chicken Cacciatora (a white meat)
12. The system shall display a button below the recipe that says: Back to Recipes
13. The user shall click on: **Back to Recipes**
14. The system shall take the user back to the Recipes page and display the five recipe options: **Red Meat, White Meat, Fish, Vegetables, and Starches**
15. The user shall click on: **Fish**
16. The system shall display a recipe that has ingredients (which includes a wine), the directions, and an image of the dish. The recipe should be a recipe for Fried Shrimp Balls (a fish)
17. The system shall display a button below the recipe that says: Back to Recipes
18. The user shall click on: **Back to Recipes**
19. The system shall take the user back to the Recipes page and display the five recipe options: **Red Meat, White Meat, Fish, Vegetables, and Starches**
20. The user shall click on: **Vegetables**
21. The system shall display a recipe that has ingredients (which includes a wine), the directions, and an image of the dish. The recipe should be a recipe for French Onion Soup (a vegetarian dish)
22. The system shall display a button below the recipe that says: Back to Recipes
23. The user shall click on: **Back to Recipes**
24. The system shall take the user back to the Recipes page and display the five recipe options: **Red Meat, White Meat, Fish, Vegetables, and Starches**
25. The user shall click on: **Starches**

26. The system shall display a recipe that has ingredients (which includes a wine), the directions, and an image of the dish. The recipe should be a recipe for Garlic Wine Rice Pilaf (a starch dish)

Expected Results:

The test case passes if the following is true:

The corresponding recipe should be displayed for the type of dish

Red Meat	White Meat	Fish	Vegetables	Starches
FLANK STEAK IN WINE SAUCE INGREDIENTS Yield: 4 servings 1 lb Flank steak 2 tablespoons Butter Salt Pepper 1 lb Onion, thinly sliced 15 oz Tomato sauce 1 cup Dry white wine 1/8 teaspoon Garlic powder 1/4 cup Parsley, chopped 1/4 cup Chives 1/4 cup Basil 4 oz Mushrooms, sliced drained Cilantro, buttered noodles  <p>Flank Steak in Wine Sauce Instructions: 1. Cut flank steak into thin strips. Cut each piece into diagonal slices crosswise, about 1/4 inch thick. 2. In a large skillet, brown steaks a few at a time in oil. Season with salt and pepper. Remove last batch of meat and pour off all drippings. 3. Cover pan with aluminum foil. Let stand for 5 minutes. 4. Remove foil. Sprinkle remaining meat with oil and pour off drippings. 5. Add tomato sauce, white garlic powder, parsley, cilantro, and basil. Simmer for 10 minutes. 6. Bring to boil; reduce heat. Cover tightly and simmer for 1 hour, or until meat is very tender. 7. Add mushrooms and continue to simmer for 5 minutes. If sauce is too thick, add juice from mushrooms. 8. Mix lemon juice and cornstarch; stir into sauce. Thicken with salt and pepper. 9. Serve meat and sauce over hot buttered noodles.</p>	CHICKEN CACCIATORA INGREDIENTS Yield: 4 Servings 3 lb Chicken cut-up 1/4 cups Oil 1 Onion, chopped 2 can Tomato sauce (8oz each) 1 teaspoon Oregano 1/4 teaspoon Thyme 1/4 teaspoon Salt 1/4 teaspoon Pepper 1 Garlic clove, minced 2 tablespoon Red wine 1 can Sliced mushrooms 1 lb Spaghetti; cooked  <p>Chicken Cacciatore Instructions: In large skillet, brown chicken in oil. Combine chicken with onion, tomato sauce, oregano, thyme, salt, pepper, garlic, red wine and mushrooms. Cover and cook on LOW for 5 hours. Stir in drained mushrooms. Spoon over hot spaghetti. Your Chicken Cacciatore is ready. Bon appetit!</p>	FRIED SHRIMP BALL... INGREDIENTS: Yield: 4 Servings 3 lb Fresh shrimp 1/4 cups Bamboo shoots chopped fine 1 teaspoon Salt 2 teaspoon Corn starch 1 Deep frying oil 4 Water chestnuts chopped fine 1/4 teaspoon Fresh chopped ginger 1 tablespoon White wine 1 Egg white  <p>Fried Shrimp Balls Instructions: devon. Mince the shrimp. Combine all of the ingredients except oil. Use 1/2 cup of the mixture to make one ball. Heat oil in a deep pot or skillet and deep fry shrimp balls until Golden Brown. Remove from oil and place onto paper towels to drain excess oil. Serve Hot. Your Fried Shrimp Balls is ready. Bon appetit!</p>	FRENCH ONION SOUP INGREDIENTS: Yield: 6 servings 1 lb Butter 6 Medium cooking onion sliced 1/4 cups Sugar 1 tablespoon All purpose flour 1 cups Dry white wine 4 cups Beef broth 1/4 teaspoon Pepper 6 1/4 in day old french bread 2 cups Grated swiss cheese 1/4 cups Grated parmesan cheese Paprika  <p>French Onion Soup Instructions: In large Dutch oven over medium heat, melt butter. Add onions and sugar and cook until onions are translucent. Add flour and cook until onions are tender but not colored. Uncover pan, increase heat slightly and continue to cook, stirring regularly until onions are rich caramel colour (10-12 minutes). Gradually add wine, cook, stirring constantly, until mixture boils. Add beef broth and bring to a boil. Reduce heat, cover and simmer over low heat for 15 minutes. Ladle soup into 6, 1 1/2 cup oven proof bowls. Top each with bread slice; sprinkle with swiss cheese and parmesan cheese. Place bowls on jelly roll pan. Broil until cheese melts and bubbles. Your French Onion Soup is ready. Happy cooking!</p>	GARLIC-WINE RICE PILAF INGREDIENTS Yield: 4 servings 1 Rind Of 1 Lemon 8 Cloves Garlic, Peeled 1/4 cups Parsley 6 tablespoon Unsalted Butter 1 cups Regular Rice (Not instant) 1 1/4 cups Chicken Stock 1/4 cups Dry Vermouth 1 Salt & Pepper To Taste  <p>Garlic-Wine Rice Pilaf Instructions: Chop together the lemon rind, garlic and parsley. Heat the butter in heavy 2-3 qt. pot. Cook the garlic mixture very gently for 10 minutes. Strain the mixture. Add the rice, stock and vermouth. Add the lemon rind and wine in a saucempan. Heat until it begins to bubble at ten sides. Stir into rice; add salt and finely ground pepper. Cover tightly and simmer over low heat for 15 minutes. Turn off heat and let stand for 10 minutes. Fluff with a fork. Drizzle a towel over the pot and cover the bowl until it is time to serve. Serve hot or at room temperature. Your Garlic-Wine Rice Pilaf is ready. Bon appetit!</p>

Alternate Paths:

None

Test Case #8: Give Ratings

Description:

This test case outlines the ability for the user to give a rating of 1-5 'Wine Glasses' (5 being the best) of a random wine. In this release the wines displayed to review will be generated randomly each time. In future releases the user shall be able to search for a specific wine

Desired Outcome:

The user shall be able to rate their wine on a scale of 1-5 'Wine Glasses' and then be given confirmation that the wine they rated was updated with their rating

User Goals:

The user wants to rate a wine they have had before

Dependent Test Cases:

TC 9: View Ratings (*More specification regarding the view option)

Involved Requirements:

SR 08 (Give Rating)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The system shall be updated with the users rating of the wine they decided to rate. The user is given an option of 10 random wines to choose from to rate, in the future release the user shall be able to search for a specific wine they would like

Trigger:

The user wants to rate a wine they have had before, therefore they navigation to the 'Rating' page and click on 'Give a Rating'

Workflow:

1. From the home page, the user shall click on: **The grape symbol (rate)**
2. The system shall take you to the rating page and display two buttons: **Give Rating and View Ratings**

3. From the ratings page, the user shall click on: **Give Rating**
4. The system shall display to the user 10 random wines to choose from
5. The user shall click on: **The first wine in the list** (make note/remember this wine name)
Example: Duckhorn Napa Cabernet Sauvignon
6. The system shall display to the user 5 wine classes (indicating a rating of 1-5)
7. The user shall click on: **2 Wine glasses**
8. The system shall display to the user: **A thank you message for submitting, a button to rate another wine, and a button to view ratings**
9. The user shall click on: **View Ratings**
10. The system shall display to the user: **The name of the wine they just rated and the rating they gave it**

Example: Duckhorn Napa Cabernet Sauvignon 2

Expected Results:

The test case passes if the following is true:

The wine the user rated is displayed under 'View Ratings' (View ratings has its own constraints, it only displays the first 5 wines the user rates). If the user rates more than 5 wines, they cannot view it under 'View Ratings'.

Under give rating the wine list just have the name of the wine: Duckhorn Napa Cabernet Sauvignon

Under the view ratings, the reviewed wines will appear like so: Duckhorn Napa Cabernet Sauvignon 2

Alternate Paths:

None

Test Case #9: View Ratings

Description:

This test case outlines the ability for the user to view the first five wines they rate. In future releases they shall be to view all the wines they have rated.

User Goals:

The user wants to view the wine ratings they have given to different wines

Dependent Test Cases:

TC 8: Give Rating

Involved Requirements:

SR 09 (View Ratings)

Details:

Priority - 0

Progress Status - Completed

Test Phase - Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The system shall display to the user a list of the first five wines the user rated. The information displayed will be just the wine name and the corresponding rating.

Example: Duckhorn Napa Cabernet Sauvignon 2

Trigger:

The user wants to view the ratings of different wines

Workflow:

1. From the home page, the user shall click on: **The grape symbol (rate)**
2. The system shall take you to the rating page and display two buttons: **Give Rating and View Ratings**
3. From the ratings page, the user shall click on: **View Rating**
4. If no wines have been rated prior to this test case, no wines should be displayed and there should also be displayed a button indicating to rate a wine. If there is already wines that have been rated, skip to Alternate Path
5. From the view ratings page, the user shall click on: **Give Rating**
6. The user shall click on: **The third wine in the list** (make note/remember this wine name)

Example: Duckhorn Napa Cabernet Sauvignon

7. The system shall display to the user 5 wine classes (indicating a rating of 1-5)
8. The user shall click on: **5 Wine glasses**
9. The system shall display to the user: **A thank you message for submitting, a button to rate another wine, and a button to view ratings**
10. The user shall click on: **View Ratings**
11. The system shall display to the user the wine they just rated with the rating of 5 next it.

Example: Duckhorn Napa Cabernet Sauvignon 5

Expected Results:

The test case passes if the following is true:

While view ratings does not have five or more wines, any wine the user rates from the give ratings page shall display under view ratings. The name should be the same from when it was rated, however under view ratings, their rating that was given to the wine shall be displayed with it

Alternate Paths:

*If there is already five different wine ratings being displayed, the user will not be able to view any new additional wine ratings. If there is less than five, then proceed with the following steps

5. From the view ratings page, the user shall click on: **The grape symbol (rate)**
6. The system shall take you to the rating page and display two buttons: **Give Rating and View Ratings**

7. From the ratings page, the user shall click on: **Give Rating**
8. The system shall display to the user 10 random wines to choose from
9. The user shall click on: **The 5th wine in the list** (make note/remember this wine name)

Example: Duckhorn Napa Cabernet Sauvignon

10. The system shall display to the user 5 wine classes (indicating a rating of 1-5)
11. The user shall click on: **3 Wine glasses**
12. The system shall display to the user: **A thank you message for submitting, a button to rate another wine, and a button to view ratings**
13. The user shall click on: **View Ratings**
14. The system shall display to the user: **The name of the wine they just rated and the rating they gave it**

Example: Duckhorn Napa Cabernet Sauvignon 3

Test Case #10: View Wine Terms

Description:

This test case outlines the ability for the user to view the list of all common wine terms

Desired Outcome:

The user shall be able to view the list of wine terms that they can select to choose a definition for

User Goals:

The user wants to view the list of wine terms to see if a word intrigues them or if they have a specific word in mind they can see if the word is there so they can view the definition for it

Dependent Test Cases:

Test Case 11: View Definition of Wine Terms

Involved Requirements:

SR 10 (View Wine Terms)

Details:

Priority - 0

Progress Status - Completed

Test Phase -Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The system shall display to the user a list of all the wine terms

Trigger:

The user wants to view the list of wine terms so they press on the 'Terms' tab

Workflow:

1. From the home page, the user shall click on: **Terms**
2. The system shall display the list of terms as a scrollable list and indicate to the user to select a term to view a specific definition
3. The user shall scroll through the list of wine terms

Expected Results:

The test case passes if the following is true:

The user is displayed wine terms in a scrollable list

Alternate Paths:

None

Test Case #11: View Wine Term Definitions

Description:

This test case outlines the ability for the user to view the definition of a term from the list of wine terms

Desired Outcome:

The user shall be able to view the definition of a wine term from the list of wine terms

User Goals:

The user wants to view the list of wine terms to see if a word intrigues them or if they have a specific word in mind they can see if the word is there so they can view the definition for it

Dependent Test Cases:

Test Case 10: View Wine Terms

Involved Requirements:

SR 11 (View Definition of Wine Term)

Details:

Priority - 0

Progress Status - Completed

Test Phase -Completed

Pre-Conditions:

TC 10 (View Terms) – The user must be on the terms page in order to view the list of terms

Post-Conditions:

The system shall display to the user the definition that goes with the terms they selected

Trigger:

The user wants to view the definition of a specific term from the list of wine terms, so they select that term from the list

Workflow:

1. From the terms page, the user scroll through the terms list until they see the word: **Blend**
2. The user shall click on: **Blend**
3. The system shall stay on the terms page and display to the user the definition of Blend which is: **"The process whereby two or more grape varieties are combined after separate fermentation; common blends include Cotes de Rhone and red and white Bordeaux."**
4. The user shall scroll through the terms list until they see the word: **Maderized**
5. The user shall click on: **Maderized**
6. The system shall stay on the terms page and display to the user the definition of Maderized which is: **"Stemming from the word Madeira, this term means oxidization in a hot environment."**

Expected Results:

The test case passes if the following is true:

The corresponding definition for the term is displayed to the user. The definition should be different for each term since no two terms should be the same word.

Alternate Path: None

Test Case #12: View Random Facts

Description:

This test case outlines the ability for the user to view random facts about wine

Desired Outcome:

The user shall be able to view random facts about wine

User Goals:

The user wants to view random facts about wine

Dependent Test Cases:

None

Involved Requirements:

SR 12 (View Random Facts)

Details:

Priority - 0

Progress Status - Completed

Test Phase -Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The system shall display to the user a random fact about wine. Since it is not a random fact, there is no way to predict which fact will be displayed.

Trigger:

The user wants to view random facts about wine so they press on the 'Facts' tab

Workflow:

1. From the home screen, the user shall click on: **Facts**
2. The system shall display to the user a speech bubble and a man with no text.
3. The user shall click on: **The man**
4. The system shall generate a random fact and display it to the user in the speech bubble
5. The user shall click on: **The man**
6. The system shall generate a different random fact and display it to the user in the speech bubble
7. The user shall click on: **The man**
8. The system shall generate a different random fact and display it to the user in the speech bubble

Expected Results:

The test case passes if the following is true:

A random fact is displayed each time the man is pressed. Here are a couple examples of a Random Facts:

"While tasting wine, leave the wine in your mouth for a second or two. Then, you can either spit it out in a spittoon or swallow it. A high quality wine not only tastes good but also has a long after-taste. On the other hand, a low quality wine would have a short after-taste."

"Women who drink excessive wine are at an increased risk of suffering from side effects of wine because the lining of the stomach needed to metabolize alcohol is less in them as compared to men."

Alternate Paths:

None

Test Case #13: Get Help

Description:

This test case outlines the ability for the user to get help on how to use Wine-Know. The user shall be able to access the help page from any sub page

Desired Outcome:

The user shall be able to get help on how to use Wine-Know

User Goals:

The user wants to get help on how to use Wine-Know

Dependent Test Cases:

None

Involved Requirements:

SR 13 (Get Help)

Details:

Priority - 0

Progress Status - Completed

Test Phase -Completed

Pre-Conditions:

*Make sure you begin from the home page. If you are not on the home page press the home button in the top left corner



Post-Conditions:

The system shall display a help sheet that describes in more detail what 'Sip', 'Terms', 'Rate', 'Facts' and 'Eat' refer to

Trigger:

The user wants help on how to use Wine-Know so they press on the help button 

Workflow:

1. From the home page, click on: **the help button** 
2. The system shall display a help page that describes in more detail the functionality of Wine-Know
3. To exit from the help page, the user shall click on: **The exit button** 
4. The system shall remove the help page from the screen and return the user to the **home page**
5. From the home page, click on: **Sip**
6. From the sip page, click on: **the help button**
7. The system shall display a help page that describes in more detail the functionality of Wine-Know

8. To exit from the help page, the user shall click on: **The exit button**
9. The system shall remove the help page from the screen and return the user to the: **home page**
10. From the home page, click on: **Terms**
11. From the terms page, click on: **the help button**
12. The system shall display a help page that describes in more detail the functionality of Wine-Know
13. To exit from the help page, the user shall click on: **The exit button**
14. The system shall remove the help page from the screen and return the user to the: **home page**
15. From the home page, click on: **Review (The grape image)**
16. From the review page, click on: **the help button**
17. The system shall display a help page that describes in more detail the functionality of Wine-Know
18. To exit from the help page, the user shall click on: **The exit button**
19. The system shall remove the help page from the screen and return the user to the: **home page**
20. From the home page, click on: **Facts**
21. From the facts page, click on: **the help button**
22. The system shall display a help page that describes in more detail the functionality of Wine-Know
23. To exit from the help page, the user shall click on: **The exit button**
24. The system shall remove the help page from the screen and return the user to the: **home page**
25. From the home page, click on: **Eat**
26. From the eat page, click on: **the help button**
27. The system shall display a help page that describes in more detail the functionality of Wine-Know
28. To exit from the help page, the user shall click on: **The exit button**
29. The system shall remove the help page from the screen and return the user to the: **home page**

Expected Results:

The test case passes if the following is true:

Each time the help button is pressed, the help image is displayed. Each time the exit button is pressed from the help screen, the user is taken to the home page.



(Help)



(Home)

Alternate Paths:

None

Test Case #14: Go Home

Description:

This test case outlines the ability for the user to return to the home page from any page within the app

Desired Outcome:

The user shall be able to return to the home page

User Goals:

The user wants to return to the home page

Dependent Test Cases:

None

Involved Requirements:

SR 14 (Go Home)

Details:

Priority - 0

Progress Status - Completed

Test Phase -Completed

Pre-Conditions:

None

Post-Conditions:

The system shall return the user to the home page

Trigger:

The user wants to return to the home page so they press on the home button



Workflow:

1. From the home page, the user shall not see the home button
2. The user shall click on: **Sip**
3. The system shall display to the user the sip page and now a home button should appear in the top left corner of the screen
4. The user shall click on: **the home button**
5. The system shall take the user back to the home screen
6. From the home page, the user shall not see the home button
7. The user shall click on: **Eat**
8. The system shall display to the user the eat page and now a home button should appear in the top left corner of the screen
9. From the eat page, the user shall click on: **Recipes**
10. The system shall display to the user the recipes page and a home button
11. The user shall click on: **the home button**

12. The system shall take the user back to the home screen

Expected Results:

The test case passes if the following is true:

When the user is on the home page, the home button does not display. When the user is on any page within the app, a home button displays in the top left corner of the page. The user shall be able to click on the home button at any point and the system shall redirect the user to the home page. The home page is this:



Alternate Paths:

None