Wine Information Network

Test Cases

Authors

Shannon Harris – Project Manager

Eliah Overbey – Senior Systems Analyst

Michael Nguyen - Software Development Lead

Andrew Hwang - Database Specialist

Brian La - Quality Assurance Lead

David Smith - Software Architect

Rex West - Subject Matter Expert

Derek Chan – Algorithm Specialist

Kyle Gullicksen - Algorithm Specialist

Edward Wang – User Interface Specialist

Compiled by: Brian La
Compilation Date: 5/19/2013
Last Updated: 6/8/2013

Table of Contents

Test Case Overview

	Pre-Built Entries
User:	• John Donut ○ Email Address: wintester@gmail.com ○ Password: cs110s ○ Location: UCSD CSE Basement ○ Birthday: 1980-05-31
Inventory Locations:	• Kitchen • Master Bedroom • Living Room • Garage
Wines in Inventory:	• Chardonnay (Cloudlift Cellars / 2010) – (Located in 'Kitchen' with quantity of '5' and is in the wishlist) • Syrah (JM Cellars / 2010) – (Located in 'Kitchen' with quantity of '1') • Moscato (Columbia Crest / 2011) – (Located in 'Master Bedroom' with quantity of '2') • Cabernet Sauvignon (Owen Roe / 2006) – (Located in 'Master Bedroom' with quantity of '4')
Shelves (and their seeded wines):	• Red Wines ○ Bel Canto Red Wine Cara Mia Vineyard / Cadence Winery / 2006 ○ Taptiel Vineyard Red Wine / Cadence Winery / 2006 ○ Ciel du Cheval Vineyard Red Wine / Cadence Winery / 2006 ○ Red Wine / Corliss Estate / 2003 • White Wines ○ Viognier / K Vintners / 2007 ○ Chardonnay Stillwater Creek / Saviah Cellars / 2006 ○ Chardonnay / Sapolil Cellars / 2007 ○ Roussanne / Doyenne / 2007 ○ Metier Blanc / Doyenne / 2007 • Dessert Wines ○ Moscato / Columbia Crest / 2011

- **The Test Cases listed here are from the Use Cases that are currently implemented (does not include ones that are listed as 'Planned' or 'Deferred', as they are saved for Ver. 2)**
- **The Test Cases and their Expected Results are assumed to be independent of each other.**
- For the best user experience use Chrome to browse the website.
- Labels are represented within single quotation marks (' ')
- User inputs are represented within double quotation marks (" ")
- On the 'Inventory' page, 'List' mode is preferred when verifying the correct wines are present.
- For 'Delete an Account' Test Case, use the following account:

  E-Mail: wintester1@gmail.com   Password: cs110s

Test Case Information Tables

	Test Case Groupings
TIS	Test case for Inventory System
TRS	Test case for Recommender System
TIRS	Test case for both Inventory and Recommender Systems
TFP	Test case is for the Front Pages, seen by the user before they choose to access their inventory or recommendations
TIWP	Test case for Specific Wine Pages that can be accessed from both the inventory system and the recommender system

	Priority
1	Highest Priority – Vital to Application
2	Medium Priority – Desirable, but not essential
3	Low Priority – Implement time permitting

	Progress Status
Planning	Use case is being considered for project.
Designing	Details of use case are being determined.
Developing	Use case is being programmed.
Testing	Use case is being tested to ensure it is working as designed.
Completed	Use case is done. There is no more work to do.
Pending	Development of use case is on hold.

TIS-1: Add a Wine

Description:	This Test Case outlines the process of a user adding a bottle of wine to their inventory.
Desired Outcome:	The bottle of wine is added to the user’s inventory.
User Goals:	The user wants to be able to add a specific bottle of wine to their inventory, with the ability to add the following information about the bottle of wine: Wine Name, Inventory Location, Winery, Region, Varietal, Vintage, Quantity, Rating, Wine Image, Qualities, and Description.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	S/R-1: Store Wine Bottle S/R-2: Store Wine Attributes S/R-4: Store User Ratings S/R-5: Store User Reviews/Descriptions
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the ‘Inventory’ page. 3. The user has at least one inventory location.
Post-conditions:	1. The user’s inventory shall reflect the addition of the bottle of wine.
Trigger:	1. The user, John Donut, clicks on the large ‘+’ button (if in Grid mode) or the ‘Add Wine’ button (if in List mode) under the ‘Search’ bar.
Workflow:	1. A modal window appears with the ‘Add Wine’ label above a set of forms and the text ‘Auto-complete Candidates:’ is to the left of the set of forms. 2. John Donut fills out the following forms: Wine Name: “Sauvignon Blanc”, Inventory Location: “Kitchen”, Winery: “Bernardo Winery”, Region: “San Diego - CA” Varietal: “Sauvignon Blanc” Vintage: “2009”, Quantity: “3”, Rating: “5”, Qualities: “Fruity”, Description: “5/5 Would drink.” 3. John Donut clicks the ‘Add Wine’ button. 4. The ‘Inventory’ page is reloaded and the wine is added to the inventory display.

Expected Results:	John Donut is able to see the following wines in their overall inventory, with the new wine having a Sauvignon Blanc wine bottle as its icon (in List mode): 1. Chardonnay (Cloudlift Cellars / Chardonnay / `10) 2. Syrah (JM Cellars / Syrah / `10) 3. Moscato (Columbia Crest / Moscato / `11) 4. Cabernet Sauvignon (Owen Roe / Cabernet Sauvignon / `06) 5. Sauvignon Blanc (Bernardo Valley / Sauvignon Blanc / `09) Clicking the 'Kitchen' location filter, John Donut is able to see the following wines (in List mode): 1. Chardonnay (Cloudlift Cellars / Chardonnay / `10) 2. Syrah (JM Cellars / Syrah / `10) 3. Sauvignon Blanc (Bernardo Valley / Sauvignon Blanc / `09) Clicking the image of the 'Sauvignon Blanc' entry will open a modal window of the wine with the following wine information: 'Sauvignon Blanc' 'Bernardo Winery' 'Sauvignon Blanc' 'San Diego – CA' '2009' '5/5 Would drink.'
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the 'Homepage' to login. This path occurs following step 3. 2. The system determines that the form content is not valid (did not fill out 'Wine Name' and 'Varietal' or invalid 'Rating') and sends John Donut back to the 'Inventory' page. This path occurs following step 3. 3. The system determines that the inventory location does not exist and sends John Donut back to the 'Inventory' page. This path occurs following step 3. 4. The system determines that the wine has already been added to the inventory and sends John Donut back to the modal. This path occurs following step 3.

TIS-2: Add an Inventory Location

Description:	This Test Case outlines the process of a user adding an inventory location.
Desired Outcome:	The inventory location is added to the user's inventory.
User Goals:	The user wants to be able to add a new inventory location in their inventory with the following information: Location Name, Location Image.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	S/R-3: Store Storage Location S/R-7: Store Storage Location Attributes
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Inventory' page.
Post-conditions:	1. The user shall have a new inventory location with which he/she can add wines in their inventory under that location.
Trigger:	1. The user, John Donut, clicks on the large '+' button under the 'Locations' label within the 'Inventory Controls' area on the left side of the page.
Workflow:	1. A modal window appears with the 'Add Location' label above a set of forms that includes 'Location Name', 'Location Image', and an 'Add Location' button below. 2. John Donut fills out the following forms: Location Name: "Basement" 3. John Donut clicks the 'Add Location' button. 4. The 'Inventory' page is reloaded and the location is added to the inventory display under the 'Locations' label.
Expected Results:	John Donut is able to view the following inventory locations under the 'Locations' section: 1. Kitchen 2. Master Bedroom 3. Living Room 4. Garage 5. Basement
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the 'Homepage' to login. This path occurs following step 3. 2. The system determines that the form content is not valid and sends John Donut back the 'Inventory' page. This path occurs following step 3. 3. The system determines that the location name already exists as an inventory location. John Donut is returned to the 'Inventory' page. This path occurs following step 3.

TIS-3: Edit an Inventory Location

Description:	This Test Case outlines the process of a user editing an inventory location in their inventory.
Desired Outcome:	The user’s inventory location has been edited.
User Goals:	The user wants to be able to change the following details of an inventory location: Location Name, Location Image.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	Misc-1: Modify Storage Location
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the ‘Inventory’ page. 3. The user has at least one inventory location.
Post-conditions:	1. The user’s inventory shall reflect the changes to the inventory location.
Trigger:	1. The user, John Donut, hovers over the ‘Living Room’ icon/filter, under the ‘Locations’ label within the ‘Inventory Controls’ area on the left side of the page, and clicks on the ‘?’ icon that appears on the top-right of the ‘Living Room’ image.
Workflow:	1. A modal window appears with the ‘Edit Location Living Room’ label above a set of forms that includes ‘Location Name’, ‘Location Image’, and an ‘Edit Location’ button below. “Living Room” is filled in the ‘Location Name’ form. 2. John Donut changes “Living Room” to “Bathroom” in the ‘Location Name’ form. 3. John Donut clicks the ‘Edit Location’ button. 4. The ‘Inventory’ page is reloaded and the location name is modified, seen under the ‘Locations’ label.
Expected Results:	John Donut is able to view the following inventory locations under the 'Locations’ section: 1. Kitchen 2. Master Bedroom 3. Bathroom 4. Garage
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the ‘Homepage’ to login. This path occurs following step 3. 2. The system determines that the form content is not valid and sends John Donut back the ‘Inventory’ page. This path occurs following step 3. 3. The system determines that the inventory location does not exist and sends John Donut back to the ‘Inventory’ page. This path occurs following step 3.

	4. The system determines that the location name already exists as an inventory location. John Donut is returned to the ‘Inventory’ page. This path occurs following step 3.

TIS-4: Editing a Wine Entry

Description:	This Test Case outlines the step process of the user editing a wine entry in their inventory.
Desired Outcome:	Edited parts of the wine entry will be saved to the database, while unedited parts remain unchanged.
User Goals:	The user wants to be able to edit information of a specific bottle of wine to their inventory, with the ability to edit the following information: Wine Name, Winery, Vintage, Inventory Location, Quantity, Rating, Qualities, Description, and Wine Image.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2 Detailed Wine Information – TIWP-1
Requirements:	Misc-2: Modify Wine Attributes
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Inventory' page. 3. The user has at least one inventory location. 4. The user has at least one wine.
Post-conditions:	1. The only parts of the wine entry that are modified are those specified by the user.
Trigger:	1. The user, John Donut, clicks the image of the 'Syrah' wine entry.
Workflow:	1. A modal window appears with the image of the image of the wine appearing on the top-left. To the right, there is a big text, 'Syrah', followed by the wine information underneath it. There are two buttons under the wine image, 'Edit Wine' and 'Delete Wine'. 2. John Donut clicks on the 'Edit Wine' button. 3. A modal window appears with the 'Edit Wine' label above a set of forms, which are already filled out with its information. 4. John Donut changes the following forms displayed: Inventory Location: "Kitchen" changed to "Master Bedroom" Vintage: "2010" changed to "2009" 5. John Donut clicks the 'Edit Wine' button. 6. The 'Inventory' page is reloaded and the wine entry has been modified with the fields above.

Expected Results:	John Donut is able to see the following wines in their overall inventory (in List mode): 1. Chardonnay (Cloudlift Cellars / Chardonnay / `10) 2. Syrah (JM Cellars / Syrah / `09) 3. Moscato (Columbia Crest / Moscato / `11) 4. Cabernet Sauvignon (Owen Roe / Cabernet Sauvignon / `06) Clicking the 'Master Bedroom' location filter, John Donut is able to see the following wines (in List mode): 1. Syrah (JM Cellars / Syrah / `09) 2. Moscato (Columbia Crest / Moscato / `11) 3. Cabernet Sauvignon (Owen Roe / Cabernet Sauvignon / `06) Clicking the image of the 'Syrah' entry will open a modal window of the wine with the following wine information: 'Syrah' 'JM Cellars' 'Syrah' 'Columbia Valley – WA' '2009' 'A brooding, mineral laden wine with peppery spice, smoked meat, blueberry, and floral notes. The palate has bold fruit flavors with firm, grainy tannins and a warm finish. 100% Syrah. Klipsun and Stillwater Creek (80%) vineyards. Aged 18 months in Russian and French oak (50% new). 15.0% alcohol. 225 cases produced. Sample provided by winery.'
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the 'Homepage' to login. This path occurs following step 5. 2. The system determines that the form content is not valid (did not fill out 'Wine Name' and 'Varietal') and sends John Donut back to the modal. This path occurs following step 5. 3. The system determines that the inventory location does not exist and sends John Donut back to the 'Inventory' page. This path occurs following step 5. 4. The system determines that the wine has already been added to the inventory and sends John Donut back to the modal. This path occurs following step 5.

TIS-6: Removing a Wine Entry

Description:	This Test Case outlines the process of a user removing a particular wine in the inventory.
Desired Outcome:	The wine specified by the user is removed from the user's inventory.
User Goals:	To remove wines they no longer have in their inventory.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2 Detailed Wine Information – TIWP-1
Requirements:	S/R-9:Delete Wines
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Inventory' page. 3. The user has at least one wine entry in his/her inventory.
Post-conditions:	1. The specified wine entry is removed from the inventory. 2. Other items remain unchanged in the inventory. 3. The specified entries are not removed from the general database.
Trigger:	1. The user, John Donut, clicks the image of the 'Cabernet Sauvignon' wine entry.
Workflow:	1. A modal window appears with the image of the image of the wine appearing on the top-left. To the right, there is a big text, 'Cabernet Sauvignon', followed by the wine information underneath it. There are two buttons under the wine image, 'Edit Wine' and 'Delete Wine'. 2. John Donut clicks on the 'Delete Wine' button. 3. A modal window appears with the message, 'Are you sure you want to delete your wine?', and the button, 'Yes, I'm sure.' 4. John Donut clicks the 'Yes, I'm sure.' button. 5. The 'Inventory' page is reloaded and the wine entry has been deleted from the inventory.
Expected Results:	John Donut is able to see the following wines in their overall inventory (in List mode): 1. Chardonnay (Cloudlift Cellars / Chardonnay / `10) 2. Syrah (JM Cellars / Syrah / `10) 3. Moscato (Columbia Crest / Moscato / `11) Clicking the 'Master Bedroom' location filter, John Donut is able to see that there is only one wine under this filter, which is 'Moscato'.
Alternate Paths:	1. The system determines that the entry has already been removed from the inventory and returns John Donut to the 'Inventory' page. This path occurs following step 4.

	TIS-8: Filter Inventory
Description:	This Test Case outlines the process of the ability for the user to filter their wine inventory by selecting from inventory location, filters, and/or tags of wine.
Desired Outcome:	The user shall be able to filter their wines in their inventory location, filters, and/or tags.
User Goals:	The user wants to filter their wine inventory by selected inventory location, filters, and/or tags.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	S/S-1: Filter Wines S/S-2: Sort Wines by Attribute
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the ‘Inventory’ page. 3. The user has at least one wine entry in his/her inventory.
Post-conditions:	1. The system shall filter the wines in the filters requested.
Trigger:	1. The user, John Donut, selects the ‘Chardonnay’ filter button under the ‘Filters’ label within the ‘Inventory Controls’ area.
Workflow:	1. The wines in the inventory are filtered under the word filter, ‘Chardonnay’, and are shown in the inventory.
Expected Results:	John Donut is able to see the wines listed in the inventory (in List mode): 1. Chardonnay (Cloudlift Cellars / Chardonnay / `10)
Alternate Paths:	None

TIS-10: Search Inventory

Description:	This Test Case outlines the process of the ability for the user to search from their wine inventory for wines by name, inventory location, winery, region, varietal, vintage, quantity, rating, qualities, and description.
Desired Outcome:	The user shall be able to search for wines in their inventory by name or filters.
User Goals:	The user wants to find the wines they are looking for.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	S/S-6: Search Database for Wine – Name S/S-7: Search Database for Wine – Attribute S/S-8: Search Inventory for Wine – Name S/S-9: Search Inventory for Wine – Name
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is at the ‘Inventory’ page. 2. The user has at least one wine entry in his/her inventory.
Post-conditions:	1. The system shall search the wines in the user inventory. 2. The system shall display the wine(s) specified.
Trigger:	1. The user, John Donut, fills out the search form with “Moscato”.
Workflow:	1. The wines in the inventory are filtered so that only the entries that contain the word, “Moscato”, and are shown in the inventory.
Expected Results	John Donut is able to see the following wine on their inventory page (in List mode): Moscato (Columbia Crest / Moscato / `11)
Alternate Paths:	None

TIS-11: Wine Wishlist

Description:	This Test Case outlines the process of the ability for the user to view the wines in their wishlist.
Desired Outcome:	The user shall be able to access their wishlist quickly and easily modify this list as desired.
User Goals:	The user wants to keep track of wines that they would like to have in the future, but currently are not able to acquire.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	S/R-10: Add Item to Wishlist S/R-11: Delete Item from Wishlist Di-11: Display Wishlist
Details:	Priority Level: 2 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Inventory' page. 3. The user has at least one wine entry in his/her wishlist.
Post-conditions:	1. The system shall display the wines that are in the wishlist.
Trigger:	1. The user, John Donut, selects the 'Wish List' filter button under the 'Filters' label within the 'Inventory Controls' area.
Workflow:	1. The wines in the inventory are filtered under the 'Wish List' filter and are shown in the inventory.
Expected Results:	John Donut is able to see the wines listed in the inventory that are in their wishlist (in List mode): 1. Chardonnay (Cloudlift Cellars / Chardonnay / `10)
Alternate Paths:	None

TIS-14: Remove an Inventory Location

Description:	This Test Case outlines the process of a user removing an inventory location from their inventory.
Desired Outcome:	The user’s inventory location has been removed.
User Goals:	The user wants to be able to remove an inventory location from their inventory.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	S/R-12: Delete an Inventory Location
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the ‘Inventory’ page. 3. The user has at least one inventory location.
Post-conditions:	1. The user’s inventory shall remove the inventory location.
Trigger:	1. The user, John Donut, hovers over the ‘Garage’ icon/filter, under the ‘Locations’ label within the ‘Inventory Controls’ area on the left side of the page, and clicks on the ‘X’ icon that appears on the top-right of the ‘Garage’ image.
Workflow:	1. A modal window appears with the message, ‘Are you sure you want to delete Location Garage?’ with a ‘Yes, I’m sure’ button below. 2. John Donut clicks the button, ‘Yes, I’m sure’. 3. The ‘Inventory’ page is reloaded and the location is removed.
Expected Results:	John Donut is able to view the following inventory locations under the 'Locations’ section (with ‘Garage’ being removed): 1. Kitchen 2. Master Bedroom 3. Bathroom
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the ‘Homepage’ to login. This path occurs following step 2. 2. The system determines that the inventory location does not exist and sends John Donut back to the ‘Inventory’ page. This path occurs following step 2.

TRS-1: View Recommendations

Description:	This Test Case outlines the process of a recommendation system for wines the user may want to try, using what the User has seeded shelves with.
Desired Outcome:	The user shall be able to view recommended wines alongside wines they are looking at and as a dedicated service.
User Goals:	To discover new wines or decide what wine he or she should try.
Primary Actor:	The System
Dependency Test Cases:	Login Account – TFP-2
Requirements:	HP-5: View Random Wine HP-6: View Random Wine Info RS-11: Shuffle Shelf RS-14: Generate Recommendations
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Recommender' page. 3. The user has at least one shelf.
Post-conditions:	1. The system displays a list of at least three wines as recommendations
Trigger:	1. The user, John Donut, selects the 'White Wines' button under the 'Tasting Shelves' label.
Workflow:	1. The system displays a few recommended wines that are related to 'White Wines'.
Expected Results	John Donut is able to view a list of several related wines, which are all white wines: 1. Viognier (K Vintners 2007) 2. Chardonnay Stillwater Creek (Saviah Cellars 2006) 3. Chardonnay (Sapolil Cellars 2007) 4. Roussane (Doyenne 2007) 5. Metier Blanc (Doyenne 2007)
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the 'Homepage' to login. This path occurs following step 1.

TRS-5: Create a Shelf

Description:	This Test Case outlines the process of the user's ability to create a recommendation shelf based off of one or more wines.
Desired Outcome:	A shelf is made based off of the wine(s) the user selects and added to the user's shelf list
User Goals:	The user wants to create a shelf based off of one or more wines
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2 View Recommendations – TRS-1
Requirements:	HP-2: Login to System RS-1: Create Shelf RS-3: Create Shelf – Preexisting Seed RS-9: View Current Shelf
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in 2. The user is at the 'Recommender' page.
Post-conditions:	1. The system shall create a shelf 2. The system shall add the shelf to the user's shelf list 3. The system shall display recommendations from this shelf
Trigger:	1. The user, John Donut, selects the 'Add Shelf' button under the 'Tasting Shelves' label.
Workflow:	1. A modal window appears with the header, 'Create New Shelf', and several forms to add various wines to create a shelf. 2. The user, John Donut, puts "Donut Shelf" as the 'Shelf Name'. 3. John Donut fills out the following forms: <u>Wine Name</u>: "Riesling" <u>Winery</u>: "Barrage Cellars" 4. A list of wines that include the fields above will appear below the forms. 5. John Donut clicks on the text, '2009 Barrage Cellars's Riesling'. 6. The selected wine is removed from the list and appears on the right, under the text, 'We'll recommend wines similar to these'. 7. John Donut clicks on the 'Create Shelf' button. 8. The 'Recommend' page is reloaded and the wine is added to the recommendation display.
Expected Results:	The user, John Donut, is able to see a list under the 'Tasting Shelves' label: 1. Add Wine 2. Edit Shelf 3. Red Wines 4. White Wines 5. Dessert Wines 6. Donut Shelf

	The user, John Donut, is able to see a list of recommended wines when clicking the 'Donut Shelf' button: 1. Riesling (Barrage Cellars 2009) 2. Riesling Cold Creek Vineyard (Chateau Ste. Michelle 2010) 3. Single Leaf Riesling (Buried Cane 2009)
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the 'Homepage' to login. This path occurs following step 7. 2. The system determines that the form content is not valid and sends John Donut back to the modal. This path occurs following step 7.

TRS-6: Remove a Shelf

Description:	This Test Case outlines the process of the user removing a shelf from their recommendation stations page.
Desired Outcome:	The selected shelf shall be removed from the user's recommendation stations page.
User Goals:	The user shall desire to remove a shelf from their recommendation stations page.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	RS-4: Remove Shelf – Database RS-5: Remove Shelf – Display RS-6: Remove Shelf – Button RS-7: Store Shelves RS-8: View Shelf List (Sidebar) RS-9: View Current Shelf
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Recommender' page. 3. There exists at least one shelf.
Post-conditions:	1. The selected shelf shall be absent from the user's recommendation stations page.
Trigger:	1. The user, John Donut, selects the 'Edit Shelf' button under the 'Tasting Shelves' label.
Workflow:	1. A modal window appears with the header, 'Edit a Shelf', and several forms to add various wines to add to the seed list. The seed list already has several wines underneath. There are two buttons on the top-right, 'Delete Shelf' and 'Save Shelf'. 2. John Donut opens the dropdown under the text, 'Pick a shelf to edit', and selects the option, 'Dessert Wines'. 3. John Donut clicks the 'Delete Shelf' button. 4. A modal window appears with the message, 'Are you sure you want to delete this shelf?' with a 'Yes, I'm sure' button below. 5. John Donut clicks the button, 'Yes, I'm sure'. 6. The 'Recommend' page is reloaded and the 'Dessert Wines' shelf is deleted from the list under the label, 'Tasting Shelves'.
Expected Results:	The user, John Donut, is able to see the following list of shelves they have under the label, 'Tasting Shelves', (with 'Dessert Wines' being deleted): 1. Add Shelf 2. Edit Shelf 3. Red Wines 4. White Wines

Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the ‘Homepage’ to login. This path occurs following step 5. 2. The system determines that the wine channel has already been removed from the channel list and returns John Donut to the ‘Recommender’ page. This path occurs following step 5.

TRS-7: Navigate Shelves

Description:	This Test Case outlines the process of the user navigating to and between their existing shelves.
Desired Outcome:	The user shall be presented with the shelf page of their choice.
User Goals:	The user shall desire to view the shelf page.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2 View Recommendations – TRS-1
Requirements:	RS-7: Store Shelves RS-8: View Shelf List (Sidebar) RS-9: View Current Shelf RS-10: Select Different Shelf
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Recommender' page. 3. There exist at least two shelves. 4. The user has the "Red Wines" shelf filter selected.
Post-conditions:	1. The user shall be presented recommended wines from the selected shelf.
Trigger:	1. Under the label, 'Tasting Shelves', the user, John Donut, selects the "White Wines" shelf button to navigate from the "Red Wines" shelf to the "White Wines" shelf.
Workflow:	1. The system removes the current recommended wines from the list. 2. The system displays a few recommended wines that are related to 'White Wines'.
Expected Results:	John Donut is able to view a list of several related wines, which are all white wines: 1. Viognier (K Vintners 2007) 2. Chardonnay Stillwater Creek (Saviah Cellars 2006) 3. Chardonnay (Sapolil Cellars 2007) 4. Roussane (Doyenne 2007) 5. Metier Blanc (Doyenne 2007)
Alternate Paths:	None

TRS-9: Edit a Shelf

Description:	This Test Case outlines the process of the user editing a shelf by modifying the shelf's seeded wines from their recommendation stations page.
Desired Outcome:	The selected shelf shall have its seeded wine list modified, which shall affect the recommendation system.
User Goals:	The user shall desire to edit a shelf by modifying the shelf's seeded wines.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	RS-15: Edit a Shelf
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Recommender' page. 3. There exists at least one shelf.
Post-conditions:	1. The selected shelf shall have its seeded wine list modified, which shall affect the recommendation system.
Trigger:	1. The user, John Donut, selects the 'Edit Shelf' button under the 'Tasting Shelves' label.
Workflow:	1. A modal window appears with the header, 'Edit a Shelf', and several forms to add various wines to add to the seed list. The seed list already has several wines underneath. There are two buttons on the top-right, 'Delete Shelf' and 'Save Shelf'. 2. John Donut opens the dropdown under the text, 'Pick a shelf to edit', and selects the option, 'Red Wines', if it is not already selected. 3. Under the text, 'Your Seed list', John Donut clicks the red button with an 'X' next to the following: '2003 Corliss Estates's Red Wine'. 4. John Donut clicks the button, 'Save Shelf'. 5. The 'Recommend' page is reloaded and the 'Red Wines' shelf has been modified so that the recommendation system is based on the updated seeded wine list.
Expected Results:	Clicking the 'Red Wines' shelf (if it is not already selected), John Donut is able to view a list of related wines, which are all red wines: 1. Bel Canto Red Wine Cara Mia Vineyard (Cadence Winery 2006) 2. Taptiel Vineyard Red Wine (Cadence Winery 2006) 3. Ciel du Cheval Vineyard Red Wine (Cadence Winery 2006)
Alternate Paths:	3. The system determines that John Donut is not logged in and sends John Donut to the 'Homepage' to login. This path occurs following step 4. 4. The system determines that the wine shelf does not exist and returns John Donut to the 'Recommender' page. This path occurs following step 4.

TIRS-1: From Recommender to Inventory

Description:	This Test Case outlines the process of the user to place wines that were recommended to them directly into their user inventories.
Desired Outcome:	The recommended wine is added to the user's inventory.
User Goals:	The user may face the situation where they have been recommended an amazing sounding wine, but already have a house full to bursting with wine. Thus, the user can retain a note of the current wine for when they have drunk enough wine to order it.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2 View Recommendations – TRS-1
Requirements:	RS-12: Recommender to Inventory RS-13: Transfer Confirmation
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2. The user is at the 'Recommender' page. 3. There exists at least one shelf. 4. There exists at least one inventory location.
Post-conditions:	1. The described wine is placed into the user's inventory.
Trigger:	1. Under the label, 'Tasting Shelves', the user, John Donut, selects the "Red Wines" shelf button, if it is not already selected.
Workflow:	1. John Donut clicks on the 'More Info' button within the wine entry, 'Taptiel Vineyard Red Wine (Cadence Winery / 2006)'. 2. A modal window appears with the image of the image of the wine appearing on the top-left. To the right, there is a big text, 'Taptiel Vineyard Red Wine', followed by the wine information underneath it. There is one button under the wine image, 'Add this to my inventory'. 3. John Donut clicks the 'Add this to my inventory' button. 4. A modal window appears with the message, 'Which location would you like to add this wine to?', and a dropdown under the label, 'Inventory Location', with the 'Add wine' button underneath. 5. John Donut selects 'Kitchen' from the 'Inventory Location' dropdown. 6. John Donut clicks the 'Add wine' button. 7. The 'Recommend' page is reloaded and the wine is added to the inventory under the 'Kitchen' location.
Expected Results:	The user, John Donut, navigates to the 'Inventory' page. The 'Taptiel Vineyard Red Wine' has been added to the inventory. Clicking the image of the 'Taptiel Vineyard Red Wine' entry will open a modal window of the wine with the correct inputted information: 'Taptiel Vineyard Red Wine' 'Cadence Winery'

	‘Bordeaux-style blend’ ‘Red Mountain – WA’ ‘2006’ ‘Light in color, this is a very well put together, smooth wine with mineral and floral notes along with Rose hips. 53% Cabernet; 32% Merlot; 15% Cab Franc. 14.4% alcohol.’
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the ‘Homepage’ to login. This path occurs following step 6. 2. The system determines that the inventory location does not exist and sends John Donut back to the ‘Recommender’ page. This path occurs following step 6. 3. The system determines that the user does not have any inventory location and sends John Donut back to the ‘Recommender’ page. This path occurs following step 6. 4. The system determines that the wine has already been added to the inventory and sends John Donut back to the modal. This path occurs following step 6.

TIRS-2: Autocomplete

Description:	This Test Case outlines the process of the user selecting wine names based off a prepopulated field of common wines when adding a wine or adding seeds to create a new shelf.
Desired Outcome:	The user shall be able to search through lists of wines using a common form element, automatically populating suggestions for wine names.
User Goals:	The user wants to quickly find what wine they're looking for.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2 Search Inventory – TIS-10 Add a Wine – TIS-1 Create a Shelf – TRS-5
Requirements:	S/S-6: Search Database for Wine – Name S/S-7: Search Database for Wine – Attribute S/S-8: Search Inventory for Wine – Name S/S-9: Search Inventory for Wine – Attribute Di-9: Autocomplete – Database Search Di-10: Autocomplete – Inventory Add
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is logged in. 2. There exist items in the search query located in the database. 3a. The user is at the 'Inventory' page. 3b. The user is at the 'Recommender' page.
Post-conditions:	1. The system shall populate a dropdown with the relevant wine names. 2. The system shall update when the form element is updated.
Trigger:	1a. The user, John Donut, clicks on the large '+' button (in Grid mode) or the 'Add Wine' button (in List mode) under the 'Search' bar. OR 1b. The user, John Donut, selects the 'Add Shelf' button under the 'Tasting Shelves' label.
Workflow:	1a. A modal window appears with the 'Add Wine' label above a set of forms and the text 'Auto-complete Candidates:' is to the left of the set of forms. 2a. John Donut fills out the following forms: Wine Name: "Riesling" Winery: "Barrage Cellars" 3a. A list of wines that include the fields above will appear below the forms. 4a. John Donut clicks the text, '2009 Barrage Cellars's Riesling'. 5a. Several forms are automatically filled with the wine information. OR 1b. A modal window appears with the header, 'Create New Shelf', and several forms to add various wines to create a shelf. 2b. John Donut fills out the following forms:

	Wine Name: “Riesling” Winery: “Barrages Cellar” 3b. A list of wines that include the fields above will appear below the forms. 4b. John Donut clicks the text, ‘2009 Barrage Cellars’s Riesling’. 5b. The selected wine is added to the list that will be used to recommend other wines.
Expected Results:	(From the ‘Add Wine’ modal window on the ‘Inventory’ page) The following fields are automatically filled: Wine Name: “Riesling” Winery: “Barrage Cellars”, Region: “Yakima Valley - WA” Varietal: “Riesling” Vintage: “2009” Description: “An aromatic wine with freshly cut golden delicious apples, oranges, and pears. On the sweeter side of off dry, the palate is full of apricot and tropical fruit flavors. Les Vignes de Marcoux (75%) and Dineen (25%) vineyards. Fermented and aged in stainless steel. 11.1% alcohol. 2% Residual Sugar. 230 cases produced.” (From the ‘Add Shelf’ modal window on the ‘Recommender’ page) Under the text, ‘We’ll recommend wines similar to these’, should have the following wine: ‘2009 Barrage Cellars’s Riesling’
Alternate Paths:	1. The system determines that there is no wine that matches the user’s entered wine name. A drop down is not shown.

	TFP-2: Login Account
Description:	This Test Case outlines the process of the user logging in to their account.
Desired Outcome:	The system should support user accounts and allow easy access to them for the users.
User Goals:	The user would be able to login to their account.
Primary Actor:	John Donut
Dependency Test Cases:	None
Requirements:	HP-2: Login to System HP-4: Login with Facebook PP-1: View Profile S/R-6: Store User Account Info
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user has a registered account. 2. The user is not logged in. 3. The user is at the 'Homepage'.
Post-conditions:	1. The user is logged in. 2. The user is redirected to the 'Inventory' page.
Trigger:	1. The user, John Donut, clicks on the 'Login' button.
Workflow:	1. A modal window appears with a message, 'Login to WineCellar', with two forms, 'E-Mail' and 'Password', and a 'Login' button below. 2. John Donut fills out the forms: E-Mail: "wintester@gmail.com", Password: "cs110s" 3. John Donut clicks the 'Login' button. 4. John Donut is redirected to the 'Inventory' page.
Expected Results:	The user, John Donut, is logged in and is able to see their 'Inventory' page. Clicking on the 'Profile' label on the top-bar will open a modal window with the name 'John Donut' displayed at the top, followed by several inventory statistics, 'Favorite Wines', and 'Recent Activity'.
Alternate Paths:	1. The user, John Donut, puts in incorrect credentials and clicks the 'Login' button. The system will display the message, 'Login Failed', above the 'Login' button. This path occurs following step 3.

TFP-3: Register an Account

Description:	This Test Case outlines the process of the user creating an account.
Desired Outcome:	An account is created and the ‘Inventory’ page will be shown after the account has been created.
User Goals:	The user wants to be able to create an account.
Primary Actor:	John Donut
Dependency Test Cases:	None
Requirements:	HP-3: Create Account
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is not logged in. 2. The user is at the ‘Homepage’.
Post-conditions:	1. The user is logged in. 2. The user is redirected to the ‘Inventory’ page.
Trigger:	1. The user, John Donut, clicks on the ‘Register an Account’ button.
Workflow:	1. A modal window appears with a message, ‘Create New Account’, with several forms below and a ‘Create Account’ button. 2. John Donut fills out the forms: E-Mail: “testing123@gmail.com”, Password: “cs110s” Re-Type: “cs110s” Display: “John Donut” Location: “Right here” Birthday: “1980-05-31” 3. John Donut clicks the ‘Create Account’ button. 4. John Donut has an account and is redirected to the ‘Inventory’ page.
Expected Results:	The user, John Donut, is logged in and is on the ‘Inventory’ page. Clicking on the ‘Profile’ tab on the top-bar will bring up a modal window with the display name, ‘John Donut’ shown on the top.
Alternate Paths:	1. The user, John Donut, puts in incorrect credentials and clicks the ‘Create Account’ button. The system will display the one of the following messages: ‘ This E-Mail Address is already in use.’ ‘*Bold fields are required’ ‘Passwords must match.’ ‘Password must be letters/numbers only and between 6-255 characters.’ ‘ Name must be letters/numbers/spaces and between 2 - 255 characters.’ ‘Location must be letters/numbers/spaces and between 2 - 255 characters.’ ‘ Birthday needs to be of the format YYYY-MM-DD and reasonable.’ This path occurs following step 3.

	TFP-4: Edit Account
Description:	This Test Case outlines the process of the user editing their account's information.
Desired Outcome:	An account's information is modified to the user's preference.
User Goals:	The user wants to be able to edit their account's information.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	PP-5: Edit User Account
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is logged in.
Post-conditions:	1. The user's account is modified. 2. The user is redirected to the 'Inventory' page.
Trigger:	1. The user, John Donut, clicks on the 'Profile' button on the top-bar of the page.
Workflow:	1. A modal window appears with the user's information and statistics. The text, 'John Donut', appears at the top, to the right of an image. Below that image are two buttons: 'Edit Account' and 'Delete Account'. 2. John Donut clicks the 'Edit Account' button. 3. John Donut fills out the forms: Current Password: "cs110s" Location: "UCSD" Birthday: "1981-05-31" 4. John Donut clicks the 'Edit Account' button. 5. The account has been modified and John Donut is redirected to the 'Inventory' page.
Expected Results:	The user, John Donut, is on the 'Inventory' page. Clicking on the 'Profile' tab on the top-bar will bring up a modal window with the display name, 'John Donut', shown on the top. Below the profile image are the following information: '**Location:** UCSD' '**Birthday:** 1981-05-31'

Alternate Paths:	1. The user, John Donut, puts in incorrect credentials and clicks the ‘Create an Account’ button. The system will display the one of the following messages: ‘Password is required.’ ‘Incorrect Password.’ ‘Passwords must match.’ ‘Password must be letters/numbers only and between 6-255 characters.’ ‘ Name must be letters/numbers/spaces and between 2 - 255 characters.’ ‘Location must be letters/numbers/spaces and between 2 - 255 characters.’ ‘ Birthday needs to be of the format YYYY-MM-DD and reasonable.’ This path occurs following step 4.

TFP-5: Delete an Account

Description:	This Test Case outlines the process of the user deleting their account.
Desired Outcome:	The user’s account is deleted.
User Goals:	The user wants to be able to delete their account.
Primary Actor:	John Donut Jr
Dependency Test Cases:	Login Account – TFP-2
Requirements:	PP-6: Delete User Account
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is logged in.
Post-conditions:	1. The user is logged in. 2. The user is redirected to the ‘Inventory’ page.
Trigger:	1. The user, John Donut Jr, clicks on the ‘Profile’ button on the top-bar of the page. (Make sure to be logged in as the following account: E-Mail: “wintester1@gmail.com” Password: “cs110s”)
Workflow:	1. A modal window appears with the user’s information and statistics. The text, ‘John Donut Jr’, appears at the top, to the right of an image. Below that image are two buttons: ‘Edit Account’ and ‘Delete Account’. 2. John Donut Jr clicks the ‘Delete Account’ button. 3. A modal window appears with the message, ‘Are you sure you want to delete your account?’ with a ‘Yes, I’m sure’ button below. 4. John Donut Jr clicks the button, ‘Yes, I’m sure’. 5. The account has been deleted and John Donut Jr is not logged in and is redirected to the ‘Homepage’.
Expected Results:	The user, John Donut Jr, is logged out and is able to see the ‘Homepage’, which includes the ‘Register an Account’ and ‘Login’ buttons. Clicking on the ‘Login’ button and filling the following forms, E-Mail: “wintester1@gmail.com” Password: “cs110s” would have an error message, ‘Login Failed’.
Alternate Paths:	None

	TFP-6: Logout
Description:	This Test Case outlines the process of the user logging out of their account.
Desired Outcome:	The system should allow the user to logout from their account.
User Goals:	The user would be able to logout from their account.
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	Misc-6: Logout User
Details:	Priority Level: 1 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is logged in.
Post-conditions:	1. The user is logged out. 2. The user is redirected to the 'Homepage'.
Trigger:	1. The user, John Donut, clicks on the 'Logout' button on the top-bar of the page.
Workflow:	1. John Donut is logged out and is redirected to the 'Homepage' page.
Expected Results:	The user, John Donut, is logged out and is able to see the 'Homepage', which includes the 'Register an Account' and 'Login' buttons.
Alternate Paths:	1. The system determines that John Donut is not logged in and sends John Donut to the 'Homepage' to login. This path occurs following step 1 from the Trigger.

TIWP-1: Detailed Wine Information

Description:	This Test Case outlines the process of the user being given information about a wine on that wine’s individual page
Desired Outcome:	The user is more knowledgeable about the wine they are viewing, influencing their decision to try that wine.
User Goals:	The user will obtain the following information about the wine: • Wine Name • Winery • Varietal • Region • Vintage • Description
Primary Actor:	John Donut
Dependency Test Cases:	Login Account – TFP-2
Requirements:	Di-1: Display Wine Attributes
Details:	Priority Level: 2 Status: Completed
Test Phase Status:	Completed
Pre-conditions:	1. The user is registered and logged in. 2a. The user is at the ‘Inventory’ page 3a. There exists at least one wine entry. OR 2b. The user is at the ‘Recommender’ page. 3b. There exists at least one channel.
Post-conditions:	1. The system shall display a modal window that has the information about that particular wine.
Trigger:	1a. The user, John Donut, clicks on the “Chardonnay” wine icon from the Inventory page. OR 1b. Under the label, ‘Tasting Shelves’, the user, John Donut, selects the “Red Wines” shelf button, if it is not already selected.
Workflow:	1a. A modal window appears with the image of the image of the wine appearing on the top-left. To the right, there is a big text, ‘Chardonnay’, followed by the wine information underneath it. There are two buttons under the wine image, ‘Edit Wine’ and ‘Delete Wine’. OR 1b. John Donut clicks on the ‘More Info’ button within the wine entry, ‘Taptiel Vineyard Red Wine (Cadence Winery / 2006)’. 2b. A modal window appears with the image of the image of the wine appearing on the top-left. To the right, there is a big text, ‘Taptiel Vineyard Red Wine’, followed by the wine information underneath it. There is one button under the wine image, ‘Add this to my inventory’.

Expected Results:	1a. The user, John Donut, is able to see an image of a 'Chardonnay' wine bottle on the top-left, and the wine information displayed on the right of the image: 'Chardonnay' 'Cloudlift Cellars' 'Chardonnay' 'Columbia Valley – WA' '2010' 'Pale lemon yellow. A moderately aromatic wine full of smoky toast, spice, and grass. The palate is tart with juicy, full fruit flavors and grapefruit-like acidity. Bacchus Vineyard. 13.8% alcohol. 60 cases produced. Sample provided by winery.' OR 1b. The user, John Donut, is able to see an image of a wine bottle on the top-left, and the wine information displayed on the right of the image: 'Taptiel Vineyard Red Wine' 'Cadence Winery' 'Bordeaux-style blend' 'Red Mountain – WA' '2006' 'Light in color, this is a very well put together, smooth wine with mineral and floral notes along with Rose hips. 53% Cabernet; 32% Merlot; 15% Cab Franc. 14.4% alcohol.'
Alternate Paths:	None