T

*Last Updated: April 21, 2013*

[**USe CASE # INV-0: “INventory”]**

|  |
| --- |
| Description:  The User shall be able to modify their wine inventory and their wish list which includes picture of the wine, price, grouping, location in the house, rating, comment, and notes. |
| Actors:  The user |
| Desired Outcome:  The data regarding the user’s wine inventory and wish list is updated and store back into the database. |
| User Goals:  To update their wine inventory and wish list according to their current needs. |
| Dependent Use Cases:  N/A |
| Involved Requirements:   * N/A |
| Details:   * **Priority:** 1 * **Progress Status:** Implementing * **Test Phase Status:** Planned * **Frequency:** Weekly |
| Pre-conditions:   1. User is able to modify their wine inventory and their wish list 2. Personal wine inventory and wish list for the user is already stored in the application’s database |
| Post-conditions:   1. Wine inventory and wish list for the user is updated according to their needs in the present. |
| Trigger:   1. Clicking on “Inventory” on the application’s home screen |
| Workflow:   1. User is at the application’s home screen 2. User shall select “Inventory” 3. System shall display the user’s inventory information in thumbnail format with price, name, winery, rating, and year. 4. User shall update the inventory list by selecting “Add” or “Remove” 5. User select “Add” will take the user to a add page. 6. Add page shall include option to take picture, use online picture, add name of the wine, the year of the wine, the name of the winery, the price of the wine, the location of the wine in the house, and side notes. 7. User shall update the value by inputting the correct values in the corresponding categories 8. User shall select “Confirm” 9. System shall store the updated values in the database 10. User select “Remove” will prompt a message with “Yes” and “No” option. 11. If User select “Yes”, system shall remove the item and update values in the database 12. If User select “No”, system shall cancel such action. |
| Alternate Paths:   1. N/A |
| Options:  N/A |

*Last Updated: April 19, 2013*

[**USe CASE # INV-1: “Add a Wine”]**

|  |
| --- |
| Description:  This use case covers the user’s ability to add a wine to the application for the system to keep track of what wines the user currently owns. |
| Actors:  The user |
| Desired Outcome:  The application will store the user’s wine info that will be used for either reference or other features of the app |
| User Goals:  The user wishes to add information of a wine they currently own into their inventory |
| Dependent Use Cases:  TBD  Creating a profile  Logging in |
| Involved Requirements:   * TBD |
| Details:   * **Priority:** 1 * **Progress Status:** Planning * **Test Phase Status:** Planned * **Frequency:** Custom |
| Pre-conditions:   1. User has logged in onto their profile 2. User is currently at their inventory display |
| Post-conditions:   1. The app will add the user’s info on the wine into the inventory |
| Trigger:   1. A button labeled “add wine” pressed by the user while on inventory screen |
| Workflow:   1. User logs onto their profile 2. App directs user to the home screen 3. User selects “inventory” 4. App collects the data of the user’s wine inventory 5. App displays inventory of the user’s wine in an organized fashion 6. User selects “add wine” 7. App displays a pop-up allowing user to input wine info 8. User inputs info via a picture of wine or image search of said wine 9. User confirms info is correct and hits a “confirm” button 10. User is brought back to inventory screen with recently added wines at “top” of inventory. |
| Alternate Paths:   1. Upon entering data for one wine, User will be prompted if they want to continue adding wines into their inventory. 2. If user selects yes, user will go through the workflow steps 7-9 3. At any time during the user input step, the user can cancel an add by hitting “cancel” |
| Options:  User may add more than one wine at a time. |

*Last Updated: April 19, 2013*

[**USe CASE # INV-2: “delete a wine”]**

|  |
| --- |
| Description:  This use case covers the user’s ability to delete a wine from their inventory in the event that the wine in question is no longer under their ownership (e.g. the wine is empty). |
| Actors:  The user |
| Desired Outcome:  The application will remove the data of the wine in question and will display the user’s inventory back to them to allow the user to confirm (if they please) that the info has been deleted. |
| User Goals:  The user wishes to remove data on a wine they don’t own/have anymore. |
| Dependent Use Cases:  TBD  Creating a profile  Logging on |
| Involved Requirements:   * TBD |
| Details:   * **Priority:** 1 * **Progress Status:** Planning * **Test Phase Status:** Planned * **Frequency:** Custom |
| Pre-conditions:   1. The user is logged onto their profile 2. The user is currently on the inventory screen 3. The user has at least one wine data stored in their inventory |
| Post-conditions:   1. The app will delete the desired wine data from the user’s inventory |
| Trigger:   1. A “delete” button |
| Workflow:   1. User logs onto their profile 2. App directs user to the main screen 3. User selects “inventory” 4. App collects the data of the user’s wine inventory 5. App displays inventory of the user’s wine in an organized fashion 6. User selects a checkbox next to the wine they wish to delete 7. User selects “delete” 8. App prompts user if they are sure they want this particular wine deleted 9. User hits confirm (or something similar). 10. App deletes the wine data in question 11. App returns user back to the inventory screen with the wine data in question deleted |
| Alternate Paths:   1. At any time during step 6, the user can cancel the deletion process by hitting “cancel” |
| Options:  User may delete more than one wine at a time by tapping more than one check box prior to hitting “delete.” |

*Last Updated: April 19, 2013*

[**USe CASE # INV-3: “editing wine data”]**

|  |
| --- |
| Description:  This use case covers the user’s ability to edit wine data within their inventory. |
| Actors:  The user |
| Desired Outcome:  The application will be able change/update the data of any given wine already stored in a user’s inventory. |
| User Goals:  The user wishes to change information on a wine in their inventory |
| Dependent Use Cases:  TBD  Creating a Profile  Logging in  IN [1] |
| Involved Requirements:   * TBD |
| Details:   * **Priority:** 1 * **Progress Status:** Planning * **Test Phase Status:** Planned * **Frequency:** Custom |
| Pre-conditions:   1. User has logged onto their profile 2. User is at the inventory screen 3. User has data for at least one wine stored into their inventory |
| Post-conditions:   1. The app will present the user’s wine inventory after the change is executed. This allows the user to confirm that the change was successful (if they wish to do so). |
| Trigger:   1. An “edit” button |
| Workflow:   1. User logs onto their profile 2. App directs user to the main screen 3. User selects “inventory” 4. App collects data of the user’s wine inventory 5. App displays inventory of the user’s wine in an organized fashion 6. The user selects “edit wine” 7. App goes into edit mode 8. User selects the wine they wish to make changes to 9. App will display up-to-date wine data that the user can change (details of how this is presented is TBD). 10. User will hit confirm/finish when user is done editing wine data 11. App will direct user back to inventory screen in “normal mode” |
| Alternate Paths:   1. Once a user confirms a change, a prompt will pop up asking user if they want to change more wines. 2. If user selects yes, then user will be directed through workflow steps 8-10 again |
| Options:  User can make changes to more than one wine data before going back to the main inventory screen. |

*Last Updated: April 19, 2013*

[**USe CASE # INV-4: “Filter/search inventory”]**

|  |
| --- |
| Description:  This use case covers the user’s ability to look at specific types of wine within their inventory. For example, the user may wish to drink a dry wine, so the user will want to look at all the dry wines that the user has, if they have any at all. |
| Actors:  The user |
| Desired Outcome:  The application will display all of the user’s wine data stored in their inventory given a specific attribute of wines. |
| User Goals:  The user wishes to find specific types of wine in their inventory |
| Dependent Use Cases:  TBD  Creating a Profile  Logging in  IN [1] |
| Involved Requirements:   * TBD |
| Details:   * **Priority:** 1 * **Progress Status:** Planning * **Test Phase Status:** Planned * **Frequency:** Custom |
| Pre-conditions:   1. User has logged onto their profile 2. User is currently at their inventory screen 3. User has at least one (or two?) wine data stored in their inventory |
| Post-conditions:   1. The app will only present wine data that consistent with wine attributes selected by user. |
| Trigger:   1. A “filter” button |
| Workflow:   1. User logs onto their profile 2. App directs user to the home screen 3. User selects “inventory” 4. App collects the data of the user’s wine inventory 5. App displays inventory of the user’s wine in an organized fashion 6. User selects “filter” 7. App displays a pop-up menu (or something similar) giving the user options of what wines within their inventory they want to look at. 8. User selects one or more attributes of wine that they wish to look for 9. User hits search/filter 10. App removes from view, all the wines that do NOT have the attributes that the user wished to look for 11. When finished, user may hit an “un-filter” button (same place as filter button) to look at all of their wine inventory 12. Once a user is finished with the filtered list, the user can select “back” to view all of their wines again |
| Alternate Paths:   1. After a filter execution, a user may wish to start a new filter/search for a different attribute of wine. 2. User will hit “new filter” to reselect a new wine attribute or wine attributes. 3. User can also add more filters to current ones by hitting “add filter” |
| Options:  Wine attributes (dry, sweet, red, white, high tannin, low tannin, country or origin, year, ideal food pairing, etc.) |

*Last Updated: April 19, 2013*

[**USe CASE # INV-5: “Pairing food with wine in inventory”]**

|  |
| --- |
| Description:  This use case covers the app’s feature to access a user’s wine inventory when looking for wines to pair with food. |
| Actors:  The user |
| Desired Outcome:  The app will successfully retrieve wine data from a user’s inventory when looking for good food/wine pairings |
| User Goals:  The user wants to know which of their wines pair well with a particular type of food that they’re eating. |
| Dependent Use Cases:  TBD  Name of Food/Wine Pairing goes here =)  Creating a Profile  Logging on  IN [1] |
| Involved Requirements:   * TBD |
| Details:   * **Priority:** 3 * **Progress Status:** Planning * **Test Phase Status:** Planned * **Frequency:** Custom |
| Pre-conditions:   1. User has logged onto their profile 2. User has selected the wine/food pairing feature of the app 3. User selected “match food with my wine” (or something like that) |
| Post-conditions:   1. The application displays a list of wines that pair well with the food in question |
| Trigger:   1. A “pair with my wine” button/option |
| Workflow:   1. User logs onto their profile 2. The application directs user to their home screen 3. User selects the wine/food pairing feature 4. The application directs user to the wine/food pairing screen 5. User selects “food -> wine” 6. The application displays a list of food categories to the user 7. User selects a type of food they wish to pair 8. The application prompts user if they wish to pair that particular type of food with any existing wines or wines that the user currently owns. 9. User selects “wine that I own” 10. The application accesses the user’s wine database and searches the user’s wines that pair well with that type of food 11. The application constructs a list of the user’s ideal wines that will be displayed to the user |
| Alternate Paths:   1. User can also select the reverse of the match: user will select a wine from their inventory and the app will present the user with a list of types of foods that pair well with said wine. |
| Options:  Different types of foods: seafood, red meat, white meat, starches, vegetables, chocolate, etc. |