

USE CASE # QAD-01: "QUICK ADD"]

Last Updated: May 6, 2013

DESCRIPTION:

This Use Case outlines the user's ability to quickly (within 30 seconds) add a wine to the inventory.

DESIRED OUTCOME:

The System will store a quick description of a wine in the User's inventory.

USER GOALS:

The User wishes to quickly add a wine for future reference.

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- The User is logged in.
- The System is at the inventory screen.

POST-CONDITIONS:

- The System will add a new wine or wines into the inventory.

TRIGGER:

- A button labeled "Quick Add" pressed by the user while on inventory screen

WORKFLOW:

1. The system shall launch the QuickWine activity to collect the name of the wine and any comments.
2. The system shall validate the entered text.
3. The system shall return the entered text as a result to the InventoryMain activity.
4. The system shall store the returned result into the local database for syncing.

ALTERNATE PATHS:

- The User may bring up the QuickWine activity multiple times in sequence by selecting the "Continue Adding" button.
- Each wine added returns it's result as normal before relaunching the activity.

OPTIONS:

The User shall be able to quick add more than one wine at a time.

USE CASE # SCH-01: “BASIC SEARCH”]

DESCRIPTION:

The user shall be able to search for a wine by entering partial information related to that wine, such as its category, grape, taste, name, origin, etc.

DESIRED OUTCOME:

The information of the searched wine is shown to the user.

USER GOALS:

The user wants to search for a wine.

DEPENDENT USE CASES:

DAT-01

PRE-CONDITIONS:

- The user knows some information about the wine.

POST-CONDITIONS:

- The user gets the information of the searched wine.

TRIGGER:

- The user taps the search bar.

WORKFLOW:

1. The system shall validate the entered text
2. The system shall submit a query to the database containing the keywords entered by the user.
3. The database shall return a list of all wines matching that criteria
4. SearchMain shall display the returned wines in list format.

ALTERNATE PATHS:

- The app fails to find any related wines.
 1. SearchMain shall display “No Results”.
 2. The sytem shall check for any potentially misspelled words.
 3. The system shall display possible spelling corrections as links for a new search if any are found.

OPTIONS:

N/A

USE CASE # SCH-02: "SEARCH WINE BY NAME"]

DESCRIPTION:

The user shall be able to search for a wine by entering its name in the search bar.

DESIRED OUTCOME:

The information of the searched wine is shown to the user.

USER GOALS:

The user wants to search for a wine.

DEPENDENT USE CASES:

DAT-01

PRE-CONDITIONS:

- The user knows the name of the wanted wine.

POST-CONDITIONS:

- The user gets the information of the searched wine.

TRIGGER:

- The user taps the search bar.

WORKFLOW:

1. The system shall validate the entered text.
2. The system shall submit a query to the database containing the name entered by the user.
3. The database shall return a list of all wines matching that name
4. SearchMain shall display the returned wines in list format.

ALTERNATE PATHS:

- The user enters a name that does not exist or is incorrect.
 1. The system shall display "No Results" and provide suggestions of wines that the user may want.

OPTIONS:

N/A

USE CASE # BAC-01: “SAFE BAC FOR DRIVING SYSTEM”

DESCRIPTION:

The user shall have the ability to check whether it is safe to drive after drinking a particular set of wine(s), or see how long it will take for them to safely drive again if they drink a planned set of wine(s).

DESIRED OUTCOME:

After inputting the appropriate information, show a recommended period of time to wait before driving.

USER GOALS:

The user wants to know if it is safe, or when it will be safe to drive again after drinking a set of wine(s).

DEPENDENT USE CASES:

USR-02

USR-03

PRE-CONDITIONS:

- The user has created an account and can log in.
- They have put in their weight and height.
- The set of wine(s) and their amounts have been input.
- If applicable, how long has it been since your last drink.

POST-CONDITIONS:

- A period of time to wait is output.

TRIGGER:

- The user selects the “BAC Estimate” option from one of the available screens.

WORKFLOW:

1. The system shall start the BACMain activity.
2. The system shall request the current user’s information from the database.
3. The system shall request information about the consumed wine from the user.
4. The user shall fill in the requested information to the best of their ability.
5. The system shall estimate the user’s BAC based on the returned information.
6. The system shall display the estimated BAC along with a legal disclaimer.

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A

USE CASE # INV-01: “ADD A WINE”]

DESCRIPTION:

This Use Case outlines the user’s ability to add a wine into the application for the system to keep track of what wines the user currently owns.

DESIRED OUTCOME:

The System will store the user’s wine information that will be used for either reference or other features of the app.

USER GOALS:

The user wishes to add information of a wine that they currently own into their inventory.

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- The User is logged in.
- The System is at the inventory screen.

POST-CONDITIONS:

- The System will add a new wine or wines into the inventory.

TRIGGER:

- A button labeled “add wine” pressed by the user while on inventory screen

WORKFLOW:

1. The System shall start the EditWine activity as a new wine.
2. The user shall enter the information of the desired wine.
3. The System shall query the database for matching wines to provide additional details.
4. The User shall confirm the information is correct by touching the add button.
5. The system shall validate the entered text.
6. The system shall return the entered text as a result to the InventoryMain activity.
7. The system shall store the returned result into the local database for syncing.

ALTERNATE PATHS:

- The User shall, after entering data for one wine, be prompted “Continue Adding?”
- The user shall select “Yes” and be directed to workflow steps 7-9

OPTIONS:

The User shall be able to add more than one wine at a time.

USE CASE # INV-02: “SELECT A WINE”]

DESCRIPTION:

This Use case outlines the user’s ability to select a wine from the inventory.

DESIRED OUTCOME:

The system will show the details about the wine in the inventory.

USER GOALS:

The User wishes see the details about a particular wine in the inventory.

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- The User is logged in.
- The System is at the user’s inventory screen.
- The User has at least one wine data stored in their inventory.

POST-CONDITIONS:

- The System displays the details about the wine selected.

TRIGGER:

- The User selects a wine from the list.

WORKFLOW:

1. The System shall launch the WineDisplay activity.
2. The system shall query the database for the selected wine’s info
3. The system shall display the returned information for the user.

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A

USE CASE # INV-03: “DELETE A WINE”]

DESCRIPTION:

This Use case outlines the user’s ability to delete a wine from their inventory.

DESIRED OUTCOME:

The System shall remove wine data and display the user’s updated inventory (with the deletion).

USER GOALS:

The User wishes to remove a wine from their personal inventory.

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- The User is logged in.
- The System is at the user’s inventory screen.
- The User has at least one wine data stored in their inventory.

POST-CONDITIONS:

- The System deletes the desired wine data from the user inventory.

TRIGGER:

- The User selects the “Delete” button

WORKFLOW:

1. The System shall confirm that the user wants to delete the selected wine.
2. The System shall prompt the database to remove the entry for the selected wine.
3. The System shall return the user to the InventoryMain activity.

ALTERNATE PATHS:

- N/A

OPTIONS:

The User shall be able to delete more than one wine by selecting more than one check box before hitting “Delete.”

USE CASE # INV-04: “FILTER/SEARCH INVENTORY”]

DESCRIPTION:

This Use case outlines the user’s ability to look at specific types of wine within their inventory.

DESIRED OUTCOME:

The System will display a portion of the user’s wine inventory (in relation to a specific attribute of the wines).

USER GOALS:

The user wishes to find specific types of wine in their inventory

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- The User is logged in.
- The System is at the Inventory Screen.
- The User has existent wine data in their inventory.

POST-CONDITIONS:

- The System will present wine data that consistent with wine attributes selected by user.

TRIGGER:

- The User shall select a “Filter” button.

WORKFLOW:

1. The system shall overlay the Filter activity over InventoryMain.
2. The user shall select one of the stored wine attributes by which to filter.
3. The system shall query the database for the relevant tags for all wines in the inventory.
4. The system shall provide the user a series of data groups based on the returned data.
5. As each filter is added, the system shall update the currently applicable results to match the new criteria
6. The system shall query the database for a list of wines and display the results

ALTERNATE PATHS:

- The User shall, be able to add or remove filters, returning the user to step 2.

OPTIONS:

Wine attributes (dry, sweet, red, white, high tannin, low tannin, country or origin, year, ideal food pairing).

USE CASE # INV-05: “PAIR FOOD WITH WINE IN THE INVENTORY”]

DESCRIPTION:

This Use case outlines the User’s ability to pair a wine with food.

DESIRED OUTCOME:

The System will display a recommended food to pair with a wine (or display a recommended wine to pair with a food).

USER GOALS:

The User wishes to know which of wines pair well with a particular type of food that they are eating (or vice versa: which foods pair well with a particular type of wine that they are drinking).

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- The User is logged in.

POST-CONDITIONS:

- The System displays a list of wines that pair well with the food (or vice versa).

TRIGGER:

- The User selects the “Pair” button.

WORKFLOW:

1. The System shall launch the PairingMain activity.
2. The system shall process the intent with which the activity was started.
3. The System shall query the database for the most applicable pairings based on the data processed.
4. The System shall display a list recommended foods and wines.

ALTERNATE PATHS:

OPTIONS:

Different types of foods and wines.

USE CASE # INV-06: “EDITING WINE DATA”]

DESCRIPTION:

This Use case outlines the user’s ability to edit their personal wine inventory.

DESIRED OUTCOME:

The Systems updates the data of an existent wine of the user’s inventory.

USER GOALS:

The User wishes to change information on a wine in their inventory.

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- The User is logged in.
- The System is at the inventory screen.
- The User has existent wine data in their inventory.

POST-CONDITIONS:

- The System shall display an updated inventory.
- The User shall confirm the changes to their inventory

TRIGGER:

- The User selects the “Edit” button.

WORKFLOW:

1. The System shall start the EditWine activity.
2. The system shall populate the text fields with stored data.
3. The User shall select “Done” when done editing wine data.
4. The System shall send the updated information to the database.
5. The System shall return to InventoryMain.

ALTERNATE PATHS:

- N/A

OPTIONS:

The User can make changes to more than one wine data before going back to the main inventory screen.

USE CASE # REC-01: “RECOMMENDATION SYSTEM”]

DESCRIPTION:

This case outlines the capacity of the system to provide wine recommendations using user entered parameters.

DESIRED OUTCOME:

The user shall be able to receive a recommendation based upon their selected criteria and their previous wine ratings.

USER GOALS:

The user wants a recommendation from the system about what wine to drink.

DEPENDENT USE CASES:

DAT-01

USR-02

RAT-01

PRE-CONDITIONS:

- The user has created an account and can log in.

POST-CONDITIONS:

- The user is displayed the page of the wine they were recommended for the purpose of rating.

TRIGGER:

- The user selects the “Recommend A Wine” option from one of the available screens.

WORKFLOW:

1. The System shall start the RecommendationMain activity.
2. The User shall select a type of recommendation.
3. The System shall query the server for a recommendation based on the current profile.
4. The System shall return the result via RecommendationDisplay.
5. The User shall select a wine to try.
6. The System shall launch DisplayWine with data based on selection.

ALTERNATE PATHS:

- N/A

OPTIONS:

The screen that the user enters the recommendation system from shall pre-define certain features of the recommendation algorithm. For example, if the user enters the recommendation screen from a wine pairing, the recommended wine type will be pre-entered as a filter for the algorithms search.

USE CASE # SMI-01: “FRIEND RECOMMENDATION”]

DESCRIPTION:

This use case outlines the feature of having a friend recommend you a wine. If there is a huge list of recommendations, then the user will see an abbreviated list.

DESIRED OUTCOME:

The user will be shown recommendations from friends.

USER GOALS:

The user wants to know about wine recommendations from friends.

DEPENDENT USE CASES:

DAT-01

USR-04

PRE-CONDITIONS:

- The user is logged on.
- The user is in the correct screen for viewing.
- The user has a friend recommendation.

POST-CONDITIONS:

- A window showing friend recommendation is shown.

TRIGGER:

- From the social page of the wine app, the user clicks on the “Recommendations” button.

WORKFLOW:

1. The System shall start the RecommendationFriend activity.
2. The System shall query the server for friend recommendations.
3. The System shall launch the RecommendationDisplay activity with the returned results.

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A

USE CASE # SMI-02: “RATINGS AND COMMENTS”]

DESCRIPTION:

This use case outlines the ability for the user to: (1) see recent comments and (2) ratings of wine from the user’s friends from a social-networking apparatus.

DESIRED OUTCOME:

A list of comments and ratings from friends will be displayed.

USER GOALS:

The user will be informed of recent comments/ratings from friends.

DEPENDENT USE CASES:

DAT-01

USR-04

RAT-01

PRE-CONDITIONS:

- The user is logged on to social-networking apparatus.
- The user has at least one friend.
- The user has at least one comment/rating from a friend.

POST-CONDITIONS:

- The user can see recent comments and ratings from friends.

TRIGGER:

- The home page will alert the user with a recent items indicator.
- The social page will have a “recent comments” button.

WORKFLOW:

1. The System shall launch SocialMain.
2. The System shall query the server for friend data.
3. The System shall populate the activity with the returned results.

ALTERNATE PATHS:

- The System shall verifies if there are any new comments
- The System shall indicates that there are new comments on the home page

OPTIONS:

N/A

USE CASE #SMI-03: "RECENT ACTIVITY"]

DESCRIPTION:

This use case outlines the user's interaction related to friend's wine activities. The user shall be informed of places, recommendations, tagged friends, pictures, and messages.

DESIRED OUTCOME:

The user will be shown an abbreviated list of recent wine related activities. The user shall be able to click on any one of the activities and have a more elaborate description of the activity.

USER GOALS:

The user wants to know about wine related activities from friends.

DEPENDENT USE CASES:

USR-04

PRE-CONDITIONS:

- The user is logged on
- The user has friend activity

POST-CONDITIONS:

- The System shall show the user's friends activity within the last week.

TRIGGER:

- The system syncs with the Social server.

WORKFLOW:

1. The System shall query the server for updated activity.
2. The System shall modify the local database to reflect the existence of new activity.

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A

USE CASE #SMI-04: "GIFT RECOMMENDATION"]

DESCRIPTION:

The user shall be able to see their friend's favorite wine list, which can be used for gift ideas.

DESIRED OUTCOME:

The user shall be able to see your friend's favorite wine list.

USER GOALS:

Find a wine to buy their friend or just to see what type of wine their friends like.

DEPENDENT USE CASES:

USR-04

PRE-CONDITIONS:

- The user shall have an account.
- The user shall have logged in.
- The user shall have a friend.
- The user's friend shall have wines in his favorite list.

POST-CONDITIONS:

- A list of wines that the user's friend favored.

TRIGGER:

- The user shall choose the "Suggest Gift" option in the social feature.

WORKFLOW:

1. The System shall launch SocialWishlist.
2. The User shall select a friend from a dropdown box.
3. The System shall launch WishlistDisplay with the selected friend's ID.

ALTERNATE PATHS:

- The user entered the screen from a friend's page; the system instead directly loads WishlistDisplay.

OPTIONS:

N/A

USE CASE # SMI-05: “INVITE FRIENDS”]

DESCRIPTION:

This use case outlines the user’s ability to invite their friends to use WINO.

DESIRED OUTCOME:

The user shall be able to post on their friend’s wall, urging them to join WINO or the user shall be able to email invitations.

USER GOALS:

To invite their friends to use the app.

DEPENDENT USE CASES:

USR-04

PRE-CONDITIONS:

- The user shall have an account.
- The user shall have logged in.
- The user shall have a friend.

POST-CONDITIONS:

- The user’s current friend list is displayed.

TRIGGER:

- The user shall select Invite a Friend from their friend list.

WORKFLOW:

1. The System shall query the server for current friends not registered to the app.
2. The System shall push a notification through the Facebook API

ALTERNATE PATHS:

- The User enters an e-mail address into the new friend field.
- The System shall send a packet to the server requesting that an e-mail be sent.

OPTIONS:

N/A

USE CASE # SMI-06: “CREATE AN EVENT”]

DESCRIPTION:

The user shall have the ability to create an event.

DESIRED OUTCOME:

The user shall be able to create an event that is viewable to all of the user’s friends.

USER GOALS:

To create an wine event viewable to their friends.

DEPENDENT USE CASES:

USR-04

PRE-CONDITIONS:

- The user shall have an account.
- The user shall have logged in.

POST-CONDITIONS:

- An event created by the event.

TRIGGER:

- The user shall choose the “create an event” option

WORKFLOW:

1. The System shall launch SocialEvent.
2. The System shall validate entered information.
3. The System shall push notifications through the Facebook API

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A

USE CASE # TUT-01: “NAVIGATE TO THE TUTORIAL MENU”]

DESCRIPTION:

The user shall be able to navigate to the tutorial menu from the main menu.

DESIRED OUTCOME:

The system navigates from the main menu to the tutorial menu.

USER GOALS:

To view the tutorial menu.

DEPENDENT USE CASES:

USR-02

PRE-CONDITIONS:

- User is viewing the main menu.

POST-CONDITIONS:

- The user is now viewing the tutorial menu.

TRIGGER:

- User selects the “Tutorials” option from the main menu.

WORKFLOW:

1. The system launches the TutorialMain activity.

ALTERNATE PATHS:

- The user is viewing a screen which can be accessed from the tutorial menu.
- The user selects the “Back” option until they reach the tutorial menu.
- The system shall navigate back to the tutorial menu.

OPTIONS:

N/A

USE CASE # TUT-02: “VIEWING TUTORIALS”]

DESCRIPTION:

The user shall be able to view a tutorial while at the tutorial menu.

DESIRED OUTCOME:

The system shall display to the User the desired tutorial.

USER GOALS:

To learn more about wine.

DEPENDENT USE CASES:

USR-02

PRE-CONDITIONS:

- User is viewing the tutorial menu.

POST-CONDITIONS:

- The user is shown the desired tutorial.

TRIGGER:

- User selects the desired tutorial option from the tutorial menu.

WORKFLOW:

1. The system launches TutorialDisplay with the selected tutorial stored as data.

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A

USE CASE # TUT-03: “ACCESSING WINE RESOURCES”]

DESCRIPTION:

The user shall be able to view/access resources to learn more about wine whilst at the tutorial menu.

DESIRED OUTCOME:

The system shall display to the User a list of links to resources where the User could find more information on wine.

USER GOALS:

To find resources to learn more about wine.

DEPENDENT USE CASES:

USR-02

PRE-CONDITIONS:

- User is viewing the tutorial menu.

POST-CONDITIONS:

- The user is shown a list of links to wine resources.

TRIGGER:

- User selects the wine resources option from the tutorial menu.

WORKFLOW:

1. The system launches the AdditionalLinks activity.

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A

USE CASE # TUT-04: “HELP TIPS”]

DESCRIPTION:

The user shall be able to select a help icon, and then select the item they wish to know more about.

DESIRED OUTCOME:

The system shall display to the User information describing the selected option.

USER GOALS:

To gather information about an unknown option.

DEPENDENT USE CASES:

USR-02

PRE-CONDITIONS:

- User is viewing a screen that contains the help icon.

POST-CONDITIONS:

- The User is shown a description of the selected item.

TRIGGER:

- User selects the help icon.

WORKFLOW:

1. The System shall grey out the current activity, and launch HelpMain.
2. The System shall highlight possible options for help request.
3. The System shall launch DisplayHelp with the selected object as stored data.

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A

USE CASE # WMP-01: “WINE TO MEAL PAIRING”]

DESCRIPTION:

This Use Case details the scenario when a User asks the system to suggest a meal that can be paired with a wine they already have or plan to drink

DESIRED OUTCOME:

The User will receive a meal pairing suggestion for their wine.

USER GOALS:

The User wants suggestions for what to eat with their wine.

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- N/A

POST-CONDITIONS:

- The system shall present the User with a screen that shows a few meal suggestions.

TRIGGER:

- The User presses the “Suggested Meals” option.

WORKFLOW:

1. ???

ALTERNATE PATHS:

- At step 1, the user selects “Choose Wine from Inventory”.
 1. The User shall choose a wine from the list provided.

OPTIONS:

N/A

USE CASE # WMP-02: “MEAL PAIRING RECIPE”]

DESCRIPTION:

This Use Case deals with the scenario where the User would like to see the recipe of the food pairing that resulted from the wine-meal pairing feature.

DESIRED OUTCOME:

The User will receive a recipe for the meal that was suggested with their wine.

USER GOALS:

The User wants to see the recipe for the meal that was paired with their wine.

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- The User shall have selected a wine in the “Wine to Meal” feature.
- The system shall display a list of meal that will pair well with the wine selected.

POST-CONDITIONS:

- The system shall direct the User to a website with a recipe for their meal pairing.

TRIGGER:

- The User shall click the "Find recipe" button.

WORKFLOW:

1. ???

ALTERNATE PATHS:

-

OPTIONS:

N/A

USE CASE # WMP-03: “MEAL TO WINE PAIRING”]

DESCRIPTION:

This Use Case handles the scenario when a User asks the system to suggest a wine that can be paired with the meal they already have or are planning to eat.

DESIRED OUTCOME:

The User will receive a wine pairing suggestion for their meal.

USER GOALS:

Find wine suggestions for the a meal a user attends to consume.

DEPENDENT USE CASES:

DAT-01

USR-02

PRE-CONDITIONS:

- N/A

POST-CONDITIONS:

- The system shall preset the User with a screen that shows wine suggestions.

TRIGGER:

- The User shall click the "Find a pairing" button.

WORKFLOW:

1. ???

ALTERNATE PATHS:

- N/A

OPTIONS:

N/A