2013

Wine Interface Necessary for Oenophiles, W.I.N.O.

Compilation Date:
June 7, 2013
Compiled by:
Steven Buell

Authors:

Asad Rana – Project Manager
Derek Bever – Senior System Analyst
Scott Mo – Software Development Lead
Jon Agustin – Subject Matter Expert I
Michael Yao – Software Architect / SME II
Steven Buell – Algorithm Specialist
Anh Tran – Database Specialist
Daniel Levine – Quality Assurance Lead I
Andrew Farin – Quality Assurance Lead II
Ayesha Mazumdar – User Interface Specialist

[DESIGN USE CASES]

Use Case Description:

As requested on March 28, 2013 by the customer, this Use Case Model will act as a reference tool to guide and depict the various aspects pertaining to the wine application. This guide will serve to represent an android phone application aiming to offer a feature set that is beneficial to oenophiles and novices a like. The main customer will be the professor of a particular class but also the CEO of CSE 110 Software Engineering, INC. This application will however not be designed to fit only our customer's needs, but other elements and uses contributing to the extensibility of this program. As a result, many other users will be able to find utility in this tool.

Within the wine application, known as "WINO," all the features and elements of the application will be accessible by any actor. With any user able to use the program, there is no restrictions or privileges that one user may have while another may not. The only restricted access is accessing a user's account without the correct user name and corresponding password.

Our app is meant not to dumb down the material and features such that Oenophiles are annoyed, but the app is also not built to only cater to individuals that have a similar level of knowledge of wine as an oenophile does. With this app, we are determined to provide tools that oenophiles will find to be invaluable, and Novices will have the resources necessary to enter the world of wine (towards becoming an oenophile themselves).

WINO will serve as a feature rich application that solves many problems consumers of wine have. Users will have the ability to: test for safe driving BAC, manage their "Vineyard" (inventory), search for wine across multiple media platforms, be surprised with a random recommendation, watch informational/educational tutorials on everything wine, wine and meal pairing, and a wine wish list feature. Social media integration through Facebook is going to be held off till the next release. Features that are planned to come with the Facebook integration will only increase the usability of WINO.

Table of Contents:

Blood Alcohol Content (BAC)

My Vineyard (INV) <our inventory>

Log in and Registration (LNR)

Quick Add (QAD)

Social Media Integration (SMI)

Search Tool Feature (STF)

Surprise (SUR)

Tutorials (TUT)

Wine Meal Pairing (WMP)

Wish List (WSL)

Use Case Variable Status Key:

Frequency:	
On Demand	The user can access this feature whenever they choose to.

Priority:	
High	These items must be implemented for the most basic of functions to be accessible. In most cases this will include the ability to create, login, retrieve password/account, view and modify wine inventory, and search the inventory for a wine.
Medium	These items represent secondary functions, which include BMI, advanced search features, social integration, and user tutorials.
Low	These items represent additions to existing functions that make them more convenient and user-friendly. In most cases this will include the ability to have more interactive tutorials, special search features, and track wine activities of other users, such as wine recommendations, comments on wines, and wine hosting.
Complete	The feature's implementation has been finished to the specification of the use case.

Progress Status:	
Planning	Use Case is currently being considered and planned
Planned	Design has been determined and implementation is ready to begin
Implementing	Use Case is currently being implemented
Implemented	Use Case is fully implemented into application and ready to be tested by QA specialist.
Deferred	Development of use case is currently on hold.

Test Phase Status:	
Planned	Testing will commence when development is completed
In Progress	Currently testing with team
Complete	Test Phase Complete, Use case currently functional

[USE CASE # BAC-01: "SAFE BAC FOR DRIVING SYSTEM"]

Last Updated: June 7, 2013

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).

Actors:

The user

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:

LNR-01

Involved Requirements:

- ACC-01
- ACC-02

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

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

Post-conditions:

1. A period of time to wait is output.

Trigger:

1. The user selects the "BAC" option from one of the available screens.

Workflow:

1. The System shall start BACActivity.
2. The System shall request the current user's weight.
3. The System shall request the current user's gender.
4. The System shall request the number of drinks.

5. The System shall request the duration since the last drink.
6. The User shall fill in the requested information to the best of their ability.
7. The System shall estimate the user's BAC by calling processBAC from UserFunctions.
8. The System shall display the estimated BAC along with a legal disclaimer.

ALTERNATE PATHS:

1. N/A

OPTIONS:

N/A

[USE CASE # INV-01: "ADD A WINE"]

Last Updated: June 7, 2013

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.

ACTORS:

The User

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 in their Vineyard.

DEPENDENT USE CASES:

LNR-01
LNR-02

INVOLVED REQUIREMENTS:

- ACC-01
- ACC-02
- INV-02a-f

DETAILS:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

PRE-CONDITIONS:

1. The User is logged in.
2. The System is at the "My Vineyard" screen.

POST-CONDITIONS:

1. The System will add a new wine or wines into the Vineyard.

TRIGGER:

1. A button labeled "add wine" pressed by the user while on "My Vineyard" screen

WORKFLOW:

1. The System shall start the InventoryEditWine activity as a new wine.
2. The User shall enter the information of the desired wine.
3. The User shall confirm the information is correct by touching the add button.
4. The System shall validate the entered text.
5. The System shall call addWine from UserFunctions to store the result to the local SQLite

database.

6. The System shall return to the InventoryActivity.

Alternate Paths:

1. The User shall, after entering data for one wine, be prompted " Continue Adding?"
2. 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"]

Last Updated: June 7, 2013

DESCRIPTION:

This Use case outlines the user's ability to select a wine from the Vineyard.

ACTORS:

The User

DESIRED OUTCOME:

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

USER GOALS:

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

DEPENDENT USE CASES:

INV-01

INVOLVED REQUIREMENTS:

- N/A

DETAILS:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

PRE-CONDITIONS:

1. The User is logged in.
2. The System is at the user's "My Vineyard" screen.
3. The User has at least one wine data stored in their Vineyard.

POST-CONDITIONS:

1. The System displays the details about the wine selected.

TRIGGER:

1. The User selects a wine from the list.

WORKFLOW:

1. The System shall launch the InventoryViewInfo activity.
2. The System shall query the database for the selected wine's info by calling DBHandler's getWine().
3. The system shall display the returned information for the user.

ALTERNATE PATHS:

1. N/A

OPTIONS:

N/A

USE CASE # INV-03: “DELETE A WINE”

Last Updated: June 7, 2013

Description:

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

Actors:

The User

Desired Outcome:

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

User Goals:

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

Dependent Use Cases:

INV-01

Involved Requirements:

- INV-03a-b

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The User is logged in.
2. The System is at the user’s “My Vineyard” screen.
3. The User has at least one wine data stored in their Vineyard.

Post-conditions:

1. The System deletes the desired wine data from the user Vineyard.

Trigger:

1. The User selects the “Delete” button

Workflow:

1. The System shall listen for an extended touch event on the wine list.
2. The System shall query the User: “Do you want to delete this wine?”
3. The User shall touch “Ok”.
4. The System shall call WineDBHandler’s deleteWine().
5. The System shall return the update the list adapter to refresh the wine list.

Alternate Paths:

1. The User shall, at any time during step 2, be able to cancel the deletion process by hitting “Cancel”.

OPTIONS:

The User shall be able to delete more than one wine by selecting more than one check box before hitting “Delete.” (Not Implemented)

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

Last Updated: June 7, 2013

Description:

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

Actors:

The user

Desired Outcome:

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

User Goals:

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

Dependent Use Cases:

INV-01

Involved Requirements:

- SQ-01a

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The User is logged in.
2. The System is at the “My Vineyard” Screen.
3. The User has existent wine data in their Vineyard.

Post-conditions:

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

Trigger:

1. The User shall select a “Filter” button.

Workflow:

1. The System shall listen for a text change on the filter field.
2. The User shall type in all or part of a wine name (Starting at the first character).
3. The System apply the filter to the list adapter and update the wine list by calling adapter.filter().

Alternate Paths:

1. N/A

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 VINEYARD”]

Last Updated: June 7, 2013

Description:

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

Actors:

The User

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:

WMP-01

Involved Requirements:

- PAR-03

Details:

- **Priority:** Complete
- **Progress Status:** implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The User is logged in.

Post-conditions:

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

Trigger:

1. The User selects the “Pair” button.

Workflow:

1. The System shall launch Main_WMPActivity.
2. The System shall prompt the user to select a pairing origin.
3. The System shall direct user to the wine/meal pairing screen by starting Wine_WMPActivity and Meal_WMPActivity respectively.
4. The User shall select a wine or food (or multiple foods) for pairing.
5. The System shall call matchWine() or findWine().
6. The System shall display a list recommended foods or wines.

Alternate Paths:

1. N/A

OPTIONS:

Different types of foods and wines.

[USE CASE # INV-06: "EDITING WINE DATA"]

Last Updated: June 7, 2013

Description:

This Use case outlines the user's ability to edit their personal wine Vineyard.

Actors:

The User

Desired Outcome:

The Systems updates the data of an existent wine of the user's Vineyard.

User Goals:

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

Dependent Use Cases:

INV-01

Involved Requirements:

- INV-02a-f

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The User is logged in.
2. The System is at the "My Vineyard" screen.
3. The User has existent wine data in their Vineyard.

Post-conditions:

1. The System shall display an updated Vineyard.
2. The User shall confirm the changes to their Vineyard

Trigger:

1. The User selects the "Edit" button.

Workflow:

1. The System shall launch InventoryEditWine in "edit" mode.
2. The User shall modify the data in any of the now editable fields.
3. The User shall select "Save" when done editing wine data.
4. The System shall return to InventoryActivity and call updateList().

Alternate Paths:

1. N/A

Options:

The User can make changes to more than one wine data before going back to the main "My

Vineyard" screen. (Not yet Implemented)

[USE CASE #LNR-01: “USER LOG-IN”]

Last Updated: June 7,, 2013

Description:

This Use Case details the User’s ability to log into the System with a valid email and password combination stored in the database.

Actors:

The Application User

Desired Outcome:

The User shall be logged in and directed to main page of the application.

User Goals:

The user wants to log into the application.

Dependent Use Cases:

LNR-02

Involved Requirements:

- ACC-02

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The User knows their valid email and password combination
2. The User has registered with the application.

Post-conditions:

1. The User is logged into application.

Trigger:

1. The User types in email and password and presses log-in button.

Workflow:

1. The User shall enter email and password into the text fields.
2. The User shall press log-in button.
3. The System shall call validateLogin()
4. The System shall call UserFunction’s loginUser().
5. The System shall return an OK to the MainActivity and close the LoginActivity.

Alternate Paths:

1. The User shall enter an invalid email and password combination.
2. The User shall press log-in button.
3. The System shall display "Invalid Login".

Options:

N/A

[USE CASE #LNR-02: "USER REGISTRATION"]

Last Updated: June 7, 2013

Description:

This Use Case details the User's ability to register with the application and create a custom log in by entering First Name, Last Name, Email, Username, Password, DOB, Gender, Weight

Actors:

The Application User

Desired Outcome:

The System shall store User's information in the database.

User Goals:

The User shall be able to log in with their custom email and password combination.

Dependent Use Cases:

LNR-01

Involved Requirements:

- ACC-01

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The Username and Email have not already been used.

Post-conditions:

1. The User shall be logged into the application.
2. The System shall store the User's information.

Trigger:

1. The User shall press Register on the Login page.

Workflow:

1. The System shall launch RegisterActivity.
2. The User shall enter First Name, Last Name, Email, Username, Password, DOB, Gender, and Weight.
3. The User shall press Submit.
4. The System shall validate the input.
5. The System shall call registerUser().
6. The Server shall verify unique user data.
7. The System shall store the user data to the local SQLite database and return to MainActivity.

Alternate Paths:

N/A

OPTIONS:

N/A

[USE CASE # LNR-03: "LOG OUT OF APPLICATION"]

Last Updated: June 7, 2013

Description:

This use case outlines the ability of the user to log out of the application.

Actors:

The Application user

Desired Outcome:

The user is logged out of the application and no user data is present.

User Goals:

The user wants to log out of his or her application account.

Dependent Use Cases:

N/A

Involved Requirements:

- ACC-01
- ACC-02
- ACC-03

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The User is logged into the application

Post-conditions:

1. The User is logged out of the application

Trigger:

1. The User shall select the "Logout" option in the menu.

Workflow:

1. The System shall clear the current user's data from the SQLite database.
2. The System shall reload the MainActivity.
3. The System shall redirect the user to the LoginActivity.

Alternate Paths:

1. N/A

Options:

N/A

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

Last Updated: June 7, 2013

DESCRIPTION:

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

ACTORS:

The User

DESIRED OUTCOME:

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

USER GOALS:

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

DEPENDENT USE CASES:

LNR-01

INVOLVED REQUIREMENTS:

- N/A

DETAILS:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

PRE-CONDITIONS:

1. The User is logged in.
2. The System is at the "My Vineyard" screen.

POST-CONDITIONS:

1. The System will add a new wine or wines into the Vineyard.

TRIGGER:

1. A button labeled "Quick Add" pressed by the user while in MainActivity

WORKFLOW:

1. The System shall start the InventoryEditWine activity as a new wine.
2. The User shall enter the information of the desired wine.
3. The User shall confirm the information is correct by touching the add button.
4. The System shall validate the entered text.
5. The System shall call addWine from UserFunctions to store the result to the local SQLite database.
6. The System shall return to the MainActivity. (Currently loads InventoryActivity instead)

ALTERNATE PATHS:

1. N/A

OPTIONS:

[USE CASE # SMI-01: "FRIEND RECOMMENDATION"]

Last Updated: June 7, 2013

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.

ACTORS:

The Application user.

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:

LNR-01

SMI-07

INVOLVED REQUIREMENTS:

- N/A

DETAILS:

- **Priority:** Low
- **Progress Status:** Deferred
- **Test Phase Status:** Planned
- **Frequency:** On Demand

PRE-CONDITIONS:

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

POST-CONDITIONS:

1. A window showing friend recommendation is shown.

TRIGGER:

2. From the social page of the wine app, the user clicks on the "Recommendations" button.

WORKFLOW:

1. The System shall query the Facebook API for a list of friends registered to the app.
2. The System shall filter the Recommendation's using the returned list.
3. The System shall display the friend recommendations.

ALTERNATE PATHS:

1. N/A

OPTIONS:

N/A

[USE CASE # SMI-02: "RATINGS AND COMMENTS"]

Last Updated: June 7, 2013

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.

Actors:

The Application user.

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:

LNR-01
SMI-07

Involved Requirements:

- N/A

Details:

- **Priority:** Low
- **Progress Status:** Deferred
- **Test Phase Status:** Planned
- **Frequency:** On Demand

Pre-conditions:

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

Post-conditions:

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

Trigger:

1. The home page will alert the user with a recent items indicator.
2. The social page will have a "recent comments" button.

Workflow:

1. The System shall query the Facebook API for a list of friends registered to the app.
2. The System shall query the database for comments on the currently viewed wine.
3. The System shall filter the returned data using the friend list.
4. The System shall display the comments.

Alternate Paths:

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

Options:

N/A

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

Last Updated: June 7, 2013

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.

Actors:

The user

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:

LNR-01

SMI-07

Involved Requirements:

- N/A

Details:

- **Priority:** Low
- **Progress Status:** Deferred
- **Test Phase Status:** Planned
- **Frequency:** On Demand

Pre-conditions:

1. The user is logged on
2. The user has friend activity

Post-conditions:

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

Trigger:

1. The User shall choose on the "Recent Activity" button.

Workflow:

1. The System shall query the Facebook API for a list of friends registered to the app.
2. The System shall query the database for user activity in the last X days.
3. The System shall filter the returned data using the friend list.
4. The System shall display the user's friend's recent activity.

Alternate Paths:

1. N/A

Options:

N/A

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

Last Updated: June 7, 2013

Description:

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

Actors:

The Application user

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:

LNR-01

SMI-07

Involved Requirements:

- N/A

Details:

- **Priority:** Medium
- **Progress Status:** Deferred
- **Test Phase Status:** Planned
- **Frequency:** Custom

Pre-conditions:

1. The user shall have an account.
2. The user shall have logged in.
3. The user shall have a friend.
4. The user's friend shall have wines in his favorite list.

Post-conditions:

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

Trigger:

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

Workflow:

1. The System shall query Facebook API for a list of friends currently registered to the app.
2. The User shall select a friend.
3. The System shall access the friend's public Wishlist and display the resulting wines in a new activity.

Alternate Paths:

1. The user shall log in.
2. The user shall choose the "Social" feature.
3. The user shall pick a friend.

4. The user shall choose the “Suggest Gift” option.

OPTIONS:

N/A

[USE CASE # SMI-05: “INVITE FRIENDS”]

Last Updated: June 7, 2013

Description:

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

Actors:

The Application user

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:

LNR-01

SMI-07

Involved Requirements:

- N/A

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The user shall have an account.
2. The user shall have logged in.
3. The user shall have a friend.

Post-conditions:

1. A list of wines that the user’s friend favored.

Trigger:

1. The user shall choose the invite a friend option in the social feature.

Workflow:

1. The System shall query Facebook API for a list of friends currently registered to the app.
2. The User shall select a friend.
3. The System shall post a linked message to the selected friend’s Facebook timeline, and send them a notification.

Alternate Paths:

1. N/A

Options:

N/A

[USE CASE # SMI-06: "CREATE AN EVENT"]

Last Updated: June 7, 2013

DESCRIPTION:

The user shall have the ability to create an event.

ACTORS:

The Application user

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:

LNR-01
SMI-07

INVOLVED REQUIREMENTS:

- N/A

DETAILS:

- **Priority:** Medium
- **Progress Status:** Deferred
- **Test Phase Status:** Planned
- **Frequency:** On Demand

PRE-CONDITIONS:

1. The user shall have an account.
2. The user shall have logged in.

POST-CONDITIONS:

1. An event created by the event.

TRIGGER:

1. The user shall choose the "create an event" option

WORKFLOW:

1. The System shall start the EventPlannerActivity.
2. The User shall input event details.
3. The System shall submit a query through Facebook's API to schedule an event using the details included.

ALTERNATE PATHS:

1. N/A

OPTIONS:

N/A

[USE CASE # SMI-07: "LOGIN TO FACEBOOK"]

Last Updated: June 7, 2013

Description:

This use case outlines the feature of the user being able to login to Facebook.

Actors:

The Application user

Desired Outcome:

The user is logged in and brought to the Social Screen.

User Goals:

The user wants to log in to Facebook in order to integrate the application with Facebook data.

Dependent Use Cases:

LNR-01

Involved Requirements:

- N/A

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. The User shall be logged into the Application
2. The User shall have a Facebook account

Post-conditions:

1. The User has successfully logged in to Facebook
2. The User is at the Social Media screen.

Trigger:

1. The User shall select the "Social" option on the Menu screen.

Workflow:

1. The System shall display the SocialActivity.
2. The User shall type his or her username.
3. The User shall type his or her password.
4. The User shall touch the Login button.
5. The System shall verify the login info with Facebook.
6. The System shall check the user's current app permissions.
7. The User shall authorize the app to interface with his or her Facebook account.
8. The System shall relaunch the SocialActivity with the User logged in.

Alternate Paths:

1. If permissions have already been granted for that user in the past, Step 7 will be

skipped.

OPTIONS:

N/A

[USE CASE # SMI-08: “RECOMMEND THE APPLICATION”]

Last Updated: June 7, 2013

DESCRIPTION:

This use case outlines the feature of the user being able to recommend the application.

ACTORS:

The Application user

DESIRED OUTCOME:

The user posts a Facebook Wall post that recommends the application to friends.

USER GOALS:

The user wants to promote the application or invite users to use the application.

DEPENDENT USE CASES:

LNR-01
SMI-07

INVOLVED REQUIREMENTS:

- N/A

DETAILS:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

PRE-CONDITIONS:

1. The User shall be logged into the Application
2. The User shall be logged into Facebook.

POST-CONDITIONS:

1. The User has a Facebook wall post that recommends the application

TRIGGER:

1. The User shall select the “Recommend WINO” option on the Social screen.

WORKFLOW:

1. The User shall be on the Social Media screen.
2. The User shall select the “Recommend WINO” option
3. The User shall be able to see Facebook wall posting that recommends the application

ALTERNATE PATHS:

1. N/A

OPTIONS:

N/A

[USE CASE #STF-01: "BASIC SEARCH"]

Last Updated: June 7, 2013

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.

Actors:

User of application

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:

LNR-01
INV-01

Involved Requirements:

- SQ-01a-b

Details:

- **Priority:** High
- **Progress Status:** Implementing
- **Test Phase Status:** Complete
- **Frequency:** On Demand

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 option from the MainActivity.

Workflow:

1. The System shall launch the SearchActivity.
2. The user shall tap the search bar.
3. The user shall enter some related information of the wine.
4. The user shall select the wanted wine shown on the list of search result.
5. The System shall load a SearchViewInfo with the details of the selected wine.

Alternate Paths:

- The user misspells words.
    1. The app shall provide suggestions of the wines that the user may want.
- The app fails to find any related wines.

1. The app shall return “No Results”.

OPTIONS:

N/A

[USE CASE #STF-02: "INTERACTIVE SEARCH"]

Last Updated: June 7, 2013

Description:

The user shall be able to interactively search for a wine by answering questions.

Actors:

User of application

Desired Outcome:

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

User Goals:

The user wants to search for a wine using the interactive search feature.

Dependent Use Cases:

LNR-01
INV-01

Involved Requirements:

- SQ-01a-b

Details:

- **Priority:** Medium
- **Progress Status:** Planned
- **Test Phase Status:** Planned
- **Frequency:** On Demand

Pre-conditions:

- N/A

Post-conditions:

- The user gets the information of the wine the app provides.

Trigger:

- The user presses the interactive search button.

Workflow:

1. The user shall press the interactive search button.
2. The user shall select one of the following options: "Grape", "Origin", "Category", "Taste".
3. The app shall move the screen to the information graph of the selected option.
4. The app shall ask the user a question and show the available options.
5. The user shall select an option for a more specific result or select "Show Result"
6. If the app reaches the end of the graph, the app shall show the results; else go to step 4.

Alternate Paths:

- N/A

Options:

N/A

[USE CASE #STF-03: "SEARCH WINE BY NAME"]

Last Updated: June 7, 2013

Description:

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

Actors:

User of application

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:

LNR-01
INV-01

Involved Requirements:

- SQ-01a-b

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

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 launch the SearchActivity.
2. The user shall tap the search bar.
3. The user shall enter some related information of the wine.
4. The user shall select the wanted wine shown on the list of search result.
5. The System shall load a SearchViewInfo with the details of the selected wine.

Alternate Paths:

- The user enters a name that does not exist or is incorrect.
    1. The app shall provide suggestions of the wines that the user may want.
- The app fails to find any related wines.
    2. The app shall return "No Results".

OPTIONS:

N/A

[USE CASE #SUR-01: “SURPRISE ME”]

Last Updated: June 7, 2013

Description:

This Use Case details the scenario when a User asks the system to provide a surprise suggestion for a wine from the user’s vineyard.

Actors:

The Application User

Desired Outcome:

The User will receive a surprise suggestion from their vineyard.

User Goals:

The User wants a random suggestion to drink from their vineyard.

Dependent Use Cases:

INV-01

Involved Requirements:

- SUR-01

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. There are wines in the User’s Vineyard.

Post-conditions:

1. The System shall present the User with a screen that shows the details and picture of a wine from their vineyard.

Trigger:

1. The User shall choose the “Surprise Me” option from the menu.

Workflow:

1. The System shall query the SQLite database for a list of wines.
2. The System shall randomly select a wine from the returned list.
3. The System shall launch the InventoryViewInfo activity to display the selected wine.

Alternate Paths:

1. N/A

Options:

N/A

[USE CASE # TUT-01: "NAVIGATE TO THE TUTORIAL MENU"]

Last Updated: June 7, 2013

DESCRIPTION:

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

ACTORS:

The user

DESIRED OUTCOME:

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

USER GOALS:

To view the tutorial menu.

DEPENDENT USE CASES:

LNR-01

INVOLVED REQUIREMENTS:

- TT-02

DETAILS:

- **Priority:** Complete
- **Progress Status:** Implementing
- **Test Phase Status:** Complete
- **Frequency:** On Demand

PRE-CONDITIONS:

1. User is viewing the main menu.

POST-CONDITIONS:

1. The user is now viewing the tutorial menu.

TRIGGER:

1. User selects the "Tutorials" option from the main menu.

WORKFLOW:

1. The System shall launch the TutorialActivity.

ALTERNATE PATHS:

1. The user is viewing a screen which can be accessed from the tutorial menu.
2. The user selects the "Back" option until they reach the tutorial menu.
3. The system shall navigate back to the tutorial menu.

OPTIONS:

N/A

[USE CASE # TUT-03: "ACCESSING WINE RESOURCES"]

Last Updated: June 7, 2013

Description:

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

Actors:

The user

Desired Outcome:

The system shall display to the User the desired tutorial.

User Goals:

To learn more about wine.

Dependent Use Cases:

LNR-01

TUT-01

Involved Requirements:

- TT-01

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. User is viewing the tutorial menu.

Post-conditions:

1. The user is shown the desired tutorial.

Trigger:

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

Workflow:

1. The System shall launch a web view of the selected resource's URL.

Alternate Paths:

1. N/A

Options:

N/A

Description:

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

Actors:

The user

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:

LNR-01

TUT-01

Involved Requirements:

- N/A

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. User is viewing the tutorial menu.

Post-conditions:

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

Trigger:

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

Workflow:

1. The System shall launch TutorialResourceActivity.

Alternate Paths:

1. N/A

Options:

N/A

[USE CASE # TUT-04: "HELP TIPS"]

Last Updated: June 7, 2013

Description:

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

Actors:

The user

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:

LNR-01
TUT-01

Involved Requirements:

- N/A

Details:

- **Priority:** Medium
- **Progress Status:** Planned
- **Test Phase Status:** Planned
- **Frequency:** On Demand

Pre-conditions:

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

Post-conditions:

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

Trigger:

1. User selects the help icon.

Workflow:

1. The System shall overlay a semi-transparent activity over the current activity (Highlighting selectable options).
2. The User selects an item other than the help icon on the current screen.
3. The System shall call getHelp() using the selected item and the current page as parameters.

Alternate Paths:

1. N/A

Options:

N/A

[USE CASE #WMP-01: "WINE TO MEAL PAIRING"]

Last Updated: June 7, 2013

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

ACTORS:

The Application User

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:

LNR-01

INVOLVED REQUIREMENTS:

- PAR-02

DETAILS:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

PRE-CONDITIONS:

1. N/A

POST-CONDITIONS:

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

TRIGGER:

1. The User shall choose the "Wine to Meal" option.

WORKFLOW:

1. The System shall launch Main_WMPActivity.
2. The System shall prompt the user to select the Wine to Meal pairing origin.
3. The System shall direct user to the start Wine_WMPActivity
4. The User shall select a wine for pairing.
5. The System shall call matchWine().
6. The System shall display a list recommended foods.

ALTERNATE PATHS:

1. The User shall choose the pairing feature.
2. The User shall choose the "Wine to Meal" option.
3. The User shall choose the "Choose Wine from Vineyard".

4. The User shall choose a wine from the list provided.

OPTIONS:

N/A

[USE CASE #WMP-02: "MEAL PAIRING RECIPE"]

Last Updated: June 7, 2013

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.

Actors:

The Application User

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:

WMP-01

Involved Requirements:

- PAR-02

Details:

- **Priority:** Medium
- **Progress Status:** Planned
- **Test Phase Status:** Planned
- **Frequency:** On Demand

Pre-conditions:

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

Post-conditions:

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

Trigger:

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

Workflow:

1. The System shall launch a webview or external application using the selected food as a search parameter.

Alternate Paths:

1. N/A

Options:

N/A

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

Last Updated: June 7, 2013

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.

Actors:

The Application User

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:

LNR-01

Involved Requirements:

- PAR-01

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. N/A

Post-conditions:

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

Trigger:

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

Workflow:

1. The System shall launch Main_WMPActivity.
2. The System shall prompt the user to select the Meal to Wine pairing origin.
3. The System shall direct user to the meal pairing screen by starting Meal_WMPActivity.
4. The User shall select a food (or multiple foods) for pairing.
5. The System shall call findWine().
6. The System shall display a list recommended wines.

Alternate Paths:

1. N/A

Options:

N/A

[USE CASE #WSL-01: “ADD TO WISH LIST”]

Last Updated: June 7, 2013

DESCRIPTION:

This Use Case deals with the scenario where the User would like to add a wine to their wish list

ACTORS:

The Application User

DESIRED OUTCOME:

The User will be able to add a wine on to their wish list in the app

USER GOALS:

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

DEPENDENT USE CASES:

LNR-01

INVOLVED REQUIREMENTS:

- WL-01

DETAILS:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

PRE-CONDITIONS:

1. Logged in
2. On the Wish list screen

POST-CONDITIONS:

1. Wine is added to the user’s wish list

TRIGGER:

1. Add to wish list has been selected on the wish list screen.

WORKFLOW:

1. The System shall start the WishlistEditWine activity as a new wine.
2. The User shall enter the information of the desired wine.
3. The User shall confirm the information is correct by touching the add button.
4. The System shall validate the entered text.
5. The System shall call addWine from UserFunctions to store the result.
6. The System shall return to the WishlistActivity.

ALTERNATE PATHS:

1. TBD

OPTIONS:

N/A

[USE CASE #WSL-02: "EDIT WINE"]

Last Updated: June 7, 2013

Description:

A user will be able to edit the attributes of wine in their wish list

Actors:

The Application User

Desired Outcome:

The user will have the ability to edit wine in their wish list.

User Goals:

Edit wine in their wish list.

Dependent Use Cases:

INV-01
WSL-01

Involved Requirements:

- N/A

Details:

- **Priority:** Complete
- **Progress Status:** Implemented
- **Test Phase Status:** Complete
- **Frequency:** On Demand

Pre-conditions:

1. Logged in to account
2. Wish list feature is selected

Post-conditions:

1. Wine in the wish list has been edited

Trigger:

1. Wish list feature is selected on the home screen.

Workflow:

1. The System shall launch WishlistEditWine in "edit" mode.
2. The User shall modify the data in any of the now editable fields.
3. The User shall select "Save" when done editing wine data.
4. The System shall return to WishlistActivity and call updateList().

Alternate Paths:

1. TBD

Options:

N/A