

Wine Manager

5th December 2019

OVERVIEW

Wine Manager is an organizational solution for wine collectors. It allows the user to visualize and search their wine collection(s) without having to physically go look. It makes being a collector easier to enjoy.

GOALS

1. Visualize wine storage in an appealing manner, indicating full and empty spaces
2. Display wine location based off of search results, or indicate no wine is available
3. Display wine details on selection of a slot in a wine storage, includes origin, varietal, taste profile, *isUserFavorite
4. Add / Remove user's wine from storage
5. Display list of favorite wines and current quantity
6. Create / Delete user's wine storage locations
7. Be mobile friendly: responsive design

FUTURE GOALS

1. Web scraper to seed Wine database
2. Enable grid layout within multiple shapes - hexagon, circle, rectangle, etc.
 - a. Grid recognition from image
 - b. Enable user click and drag within shape to create grid underlay
3. Integrate LED light hardware with app
 - a. Lights illuminate based on user search results, physically identifying where bottles are in storage location
4. Integrate image scan with app
 - a. Allow user to scan barcode to search for wine in database
 - b. Allow user to scan image to search for wine in database

PHASES & TARGETS

COMPLETE

Define

By Sun 8-Dec

Identify the following solution paths:

Open

- Hosting database - Firebase (FirebaseDatabase.net), Azure, etc
- Front-end technology - Razor, React, JQuery, etc
- Finalize project concept

Plan

By Sun 8-Dec

Create outline of schedule and specific objective milestones

Open

Design

By Tues
10-Dec

Wireframe concept views: consider computer, tablet, iPhone viewports

Design data models & data flow

Open

Develop

By Tue 17-Dec

Develop:

Open

- Front-end views of wine location (1)
- Back-end logic for wine data storage
- Authentication
- Hosting

Test

By Tue 17-Dec

Complete:

Open

- Unit testing of major logic functions
- User testing of front-end

Deploy

By Thur 19-Dec

Host database and website online

Open

STRUCTURE

The purpose of this section is to provide general concepts for the views and functions of the Wine Manager app. The (*) indicates future considerations.

1. Login page:
 - a. WineManager Logo
 - b. Text fields for user name and password
 - c. Login & Registration buttons
 - d. About link?
 - e. FUNCTION: route user to homepage on successful login, display descriptive error message on failed login
2. Home page:
 - a. Display username & WineManager logo (header?)
 - b. Link to account details page
 - c. List of links for existing wine locations
 - d. List of favorite wines w/ storage location name ⇒ goes to wine detail display or storage location
3. Account page:
 - a. Password reset
 - b. * Settings (useLights y/n)
4. Wine Location page:
 - a. Display location name
 - b. Display representation of storage using grid
 - i. * Allow for user to create custom storage locations by creating shape outline and moving grid pattern underlying the shape
 - c. Identify empty and full slots in storage
 - d. FUNCTION:
 - i. Add wine (click on empty slot)
 - ii. Display existing wine (click on full slot)
5. Wine Detail page:
 - a. ⇒ Display changes based on viewport size:
 - i. Computer or tablet: display as sidebar
 - ii. Mobile: display as entire screen with added link to highlight in wine location view
 - b. Display wine name
 - c. Display wine details: varietal, origin, tasting profile, etc
 - d. Display current storage location(s) and total quantity

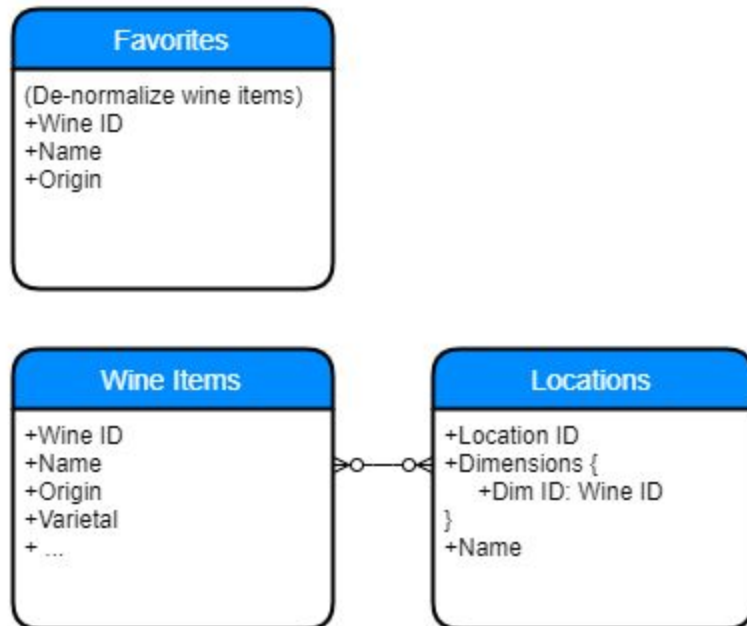
-
- e. Button to remove wine from storage
 - f. Button to add wine to Favorites list
6. Add Wine page:
- a. ⇒ Three views: Options view, Search for wine view, Create new wine view
 - i. Options: display links to search for existing wine and create new wine;
FUNCTION: add to slot after successful search or create
 - ii. Search: search form with submit button; FUNCTION: display paginated wine summaries based on search, link to wine display pages
 - iii. Create: create new wine & add to database, user must confirm values before database push; FUNCTION: create wine item & push to database on confirm
7. * Online / Offline capability?
- a. Offline: view only functions
 - b. Online: add / remove wine, create / delete wine locations
8. * Wine “periodic” table page ⇒ need copyright approval?
- a. Display wine periodic table
 - b. FUNCTION: on click of box in table, display all applicable wines in user’s collection with corresponding wine location
9. BACKEND DATABASE:
- a. Wine data table
 - i. Wine ID, unique
 - ii. Name, varietal, origin, tasting profile, etc
 - b. Wine storage table
 - i. Wine storage ID, unique
 - ii. Name, dimensions
 - c. Collection table (join Wine data & Wine storage)
 - i. Collection ID, unique
 - ii. Wine ID
 - iii. Wine storage ID
 - iv. Wine storage location ID (based on dimensions, e.g 1-2 or A2)

Backend Model

Favorites will be a filtered list of Wine Items that the user indicates they like and want to keep track of *even if not in a location*.

Wine Items are individual objects with properties containing wine details (e.g. id, name, origin, etc). They do not have to be in a storage location, but can be referenced from many locations.

Locations are objects containing details about the storage location, including a sub-object using dimension id keys to wine id values. Locations do not have to have wine items, but can contain many.



<https://www.draw.io/?libs=general;uml;er#G1RzSv-UB57Nz1HAmpJry5h6dstJkdZNWo>

NOTES

6-Dec-2019

- Learnings from research: Firebase uses NoSQL databases (not relational), but can be used with C# projects (I especially like the FirebaseDatabase.net library)
 - Can't find reliable, *free* VPS SQL server hosting - all are limited time or trial based.
 - Moving forward with Firebase option ⇒ Need to restructure expected backend data model