Podo:
Software Design Document

Team Members:

Hyerin Choi (hyerin.choi.1@stonybrook.edu)
DukYoung Eom (dukyoung.eom@stonybrook.edu)
Woohyung Lee (woohyung.lee.1@stonybrook.edu)
Woohyun Park (woohyun.park@stonybrook.edu)

Table of Contents

1. Introduction

1.1 Product description

Many people buy wine for their dinner table without knowing what would fit best. *Podo* is an online wine search and recommendation service for general and beginning users who want to buy a decent wine. It aims to offer valuable resources that wine beginners can navigate themselves to select the right wines matching their tables. Users can review and rate the wines they had, share their experience and refer to professional sommeliers' guides for their dinner table. Sommeliers' reviews will offer more professional and informative features about the wine, listed on the top section of the review. *Podo* helps people have more opportunities to try out various wines that correspond to their taste and buy the right wine for their dishes by offering a carefully selected wine list. Sommeliers can easily share their recommended wine lists and knowledge to the public through *Podo*, which allows general consumers and restaurants to choose the right wine for their special dining.

1.2 Scope

People enjoy wine on special occasions, which means they do not drink wine often or on a daily basis. This app, *Podo*, will be a great reference for those who do not have professional knowledge, but who want to choose wine that fits their taste and purpose(general users). In particular, by displaying ratings and reviews of the experts (sommelier) and those of general users at the same time, general users can get more objective information and evaluations of wine by combining both professional and general reviews.

However, it will not support actual shopping or payment of the wine, but only will provide information about wine to the users. It will be available on both web browsers(focused mainly on Chrome - Internet Explorer will not be supported) and mobile devices.

1.3 Users

Target users in this project will be people aged in late 20s up to 40s who do not have professional knowledge about the wine; those who want wines that fits their tastes and special purpose - such as a present, for a picnic, or fine dining. The primary assumption is that general users are familiar with web browsing or mobile devices.

1.4 User Feedback

Our team will start to collect user feedback as soon as we release our first prototype. Target users will use our website for a couple of days and our team will interview them about the experience, satisfaction with the findings and feedback. Our first goal is to check the user experience that people who are new to wines are able to navigate themselves to find wines they want. If users feel uncomfortable or find it hard to search wines, we will have to reorganize how we guide the user from the main page. This process will be conducted at every prototype release. Moreover, we will contact wine experts or professional sommeliers to talk about how we can deliver the correct information and guidance for wine beginners.

1.5 Existing Alternatives

Reviews of wines can be easily obtained through the Internet, but they may be unreliable and inaccurately reflect what you would experience. Furthermore, some users claim that most of the existing alternatives tend to focus more on advertising specific products than delivering the correct and detailed information on all products.

There are several solutions to solve these problems. One of the existing alternatives is a web community called *The Wine Community*. In order to know the exact aroma and taste of wine, it provides a wine tasting program and even offers regular subscription services. However, it is difficult to obtain detailed information or general evaluation of people on each wine because it has only a simple description, and does not allow users to post reviews. Moreover, it provides limited services to certain countries - Australia and New Zealand.

Another solution is *Vivino*. It provides information on wine and induces purchases. One great advantage of this is that it allows wines to be searched by region, grape, and type. However, this website does not provide any features related to professional sommeliers within the service, making it less reliable. Due to this problem, this service tends to focus more on providing information on leading trends and advertisements.

From those alternatives, we adopted *The Wine Community's* wine detail description UI and *Vivino's* wine search UI because we thought those UIs were intuitive designs for our target users to search and gain knowledge. Unlike the previous two websites, we want to target users to get professional and reliable information about the wine. Therefore, Podo's ultimate goal is to help people without knowledge of wine to get professional and reliable information about wine and easily choose the product they need. Currently, only sommeliers can share wine lists, but if the users actively experience more wines and become knowledgeable and reliable, we are open to creating another class that can create wine lists and promotes them. We aim our website to be a guide for wine beginners eventually.

1.6 Definitions

Like(s) : refers to a bottle of wine or wine list that the user marked with a heart.
Review : Star rates with related tags and short comments on each wine.
Sommelier user : user who is verified as a professional sommelier by the admin.
User : a person who can access public information on the website.
Verification : admin granting a special badge to a verified sommelier.
Wine list : wine recommendation list written by a sommelier with a strong theme.

1.7 References

The wine Community (2015) : Retrieved from https://www.thewinecommunity.com.au/pages/tasting

Vivino (2010) : Retrieved from https://www.vivino.com/US/en

2. Requirements

2.1 Functional requirements

All the users shall:

1. Be able to search for wine by
    - Name
    - Filters (tags, price range, star rates)
2. Be able to look at the posted reviews
3. Be able to look at sommelier's wine lists
4. Be able to see the FAQ

The signed-in users shall:

1. Be able to review wine
    - Favorites
    - Rate the wines (star rate, comment, tags)
2. Be able like the posted review and wine lists
3. Be able to add and edit the account information
    - Password
    - Phone number
    - Email address
    - Profile pictures
4. Be able to request for sommelier verification
5. Be able to delete their accounts
6. Be able to follow users and view their reviews and favorites
7. Be able to issue a support ticket
8. Be able to look at similar wine posts (tag, characteristics)

The Sommelier users shall:

1. Be able to post their wine list

The admin users shall:

1. Be able to manage the permission level of user
    - Signed-in Users
    - Sommelier
    - Admin
2. Be able to manage post, delete, ban users and posts

The system shall:

1. Be able to extract wine information from the wind database

2.2 Use Cases

2.2.1 Use Case : Search wine with filters

Primary Actor	All users
Priority	essential
Scenario	1. User clicks a search button at the right of the top navigation bar. 2. User enters a keyword one wants to search for. 3. After entering a keyword, the system will show results, and provide ‘filter’ and ‘sort’ buttons. 4. User clicks the 'filter’ button which will then show a full page modal that shows a list of tags available for search, price ranges, and star rates. 5. User clicks ‘find’ or presses enter key after selecting all related tags to one's needs. 6. The filtered wines will appear in order of high star rates by default.
Extension	3.c Empty string input will show an error message to the user. 5.a User does not necessarily have to select a tag. 6.a When the user clicks the ‘sort’ button, the system will show a full page modal with options of - highest rating, most liked, price low to high, price high to low.

2.2.2 Use Case : View tickets

Primary Actor	Signed-in Users
Priority	Nice to have
Scenario	1. Signed-in user clicks the menu icon at the left of the top navigation bar. 2. User will see the side menu from the left side, which has a ‘View tickets’ button at the bottom. 3. Signed-in user clicks the 'View tickets’ button. 4. User will be able to see the tickets and the status that the signed-in user has submitted since one’s registration to the admin in a list format. 5. Signed-in user clicks one of the tickets to check out the details.
Extension	4.a The tickets will by default be listed in the order of newest to oldest. 4.b The view will show an empty page with the message ‘no ticket created!’.

2.2.3 Use Case : Select related tag for each wine on review page

Primary Actor	Signed-in Users
Priority	Nice to have
Scenario	1. Extension from 2.2.1 2. A section for review will be shown on the screen to users when a signed-in user clicks a specific wine page by default. 3. Signed-in user writes a review on a wine and adds some tags. 4. Signed-in user clicks the ‘post a review’ button to save a revised review. 5. Signed-in user will be able to see one’s own review on the detailed wine page.
Extension	3.a The review written by a verified sommelier will have a special badge next to the username and will have a highlighted outline with a special color(burgundy). 3.b Signed-in user’s own review will be shown at the top of all reviews. 3.c Signed-in users can edit one’s review by clicking the ‘pen’ icon. The pen icon will be filled after clicking once, and then outlined after clicking again. 4.a If the signed-in user clicks the 'pen’ icon again, the revised review will not be saved.

2.2.4 Use Case : View a detailed wine page

Primary Actor	All users
Priority	Essential
Scenario	1. Extension from 2.2.1 2. User clicks the wine from the search results. 3. The information about the wine and related tags will appear on the screen. 4. User sorts the reviews by rating and latest.
Extension	1.a User can also access the wines by clicking the ‘wine’ button in the side menu. 3.a User can click the tag to search for other wines with the same tag. 3.b User can click the ‘like’ button to add the wine to personal favorites.

2.2.5 Use Case : Follow other users

Primary Actor	All users
Priority	Moderate
Scenario	1. User opens the side menu icon located top left. 2. User clicks their username or profile image to go to the profile page. 3. User clicks the search button located on the top right. 4. User types a username in the search bar and press enter or click the search icon. 5. The search bar shows the list of users depending on the input string and user clicks one of the users to go to other user's profile page. 6. User clicks the 'follow' button located right below the profile picture.
Extension	5.a If user A is already following user B, the button will be transformed into 'unfollow' instead of 'follow'.

2.2.6 Use Case : Create Wine List

Primary Actor	Sommelier users
Priority	Essential
Scenario	1. Sommelier user clicks the side menu icon located top left. 2. Press the 'Create Wine List' Button. 3. Type the title of the list. 4. Sommelier user types wine names in the search bar to find the wines. 5. Search results are shown on the screen and the sommelier user clicks the 'Add' button to add the wine to the list. 6. Write a brief description of the selected wine. 7. Sommelier user upload photos of the wines he listed 8. Write the description about the whole list and click the 'Submit' button.
Extension	6.a Sommelier user can repeat steps 4-6 to add other wines. 8.a Sommelier user clicks the 'X' icon on the top right to go back to the main page.

2.2.7 Use Case : Verifying Sommelier

Primary Actor	Admin users
Priority	essential
Scenario	1. User opens a side menu by clicking the menu icon located at the top left navigation bar. 2. Admin clicks the 'verify sommeliers' button at the bottom of the side menu. 3. Admin clicks one of the verification requests to check out the certificate from a sommelier applicant. 4. Admin will see the user's paper verification of the sommelier with a comment section and icons for 'approve' and 'reject'. 5. After the admin fills out the comment, the admin clicks either one of the acceptance or rejection buttons based on the content of the picture.
Extension	

2.2.8 Use Case : Request for verification

Primary Actor	General users
Priority	essential
Scenario	1. User opens a side menu by clicking the menu icon located at the top left navigation bar. 2. User clicks the 'become sommelier' button at the bottom of the side menu bar. 3. User clicks the 'choose a photo of your sommelier certificate to be verified'. 4. User chooses a photo. 5. When the upload is complete, user clicks the 'submit' button. 6. User will see the message "submission complete the result will be notified in 2-3 business days".
Extension	3.a User clicks the 'X' icon on the top right to cancel the submission. 4.a User clicks the photo to reupload. 4.b User clicks the 'view request history' button to view verification history.

2.3 User Interfaces

Link to Figma

2.4 Non-functional Requirements

1. Users will be able to complete all the use cases for the website on either a mobile or desktop web browser(except Internet Explorer, focused on Chrome).
2. All the users' data will require authentication to access and only be accessible by the user themselves.
3. All web communication should be secured using SSL.
4. The website will be able to load the pages in under 500 ms on average.
5. The website will have all the text localized and available in English.

3. System Architecture

3.1 Overview

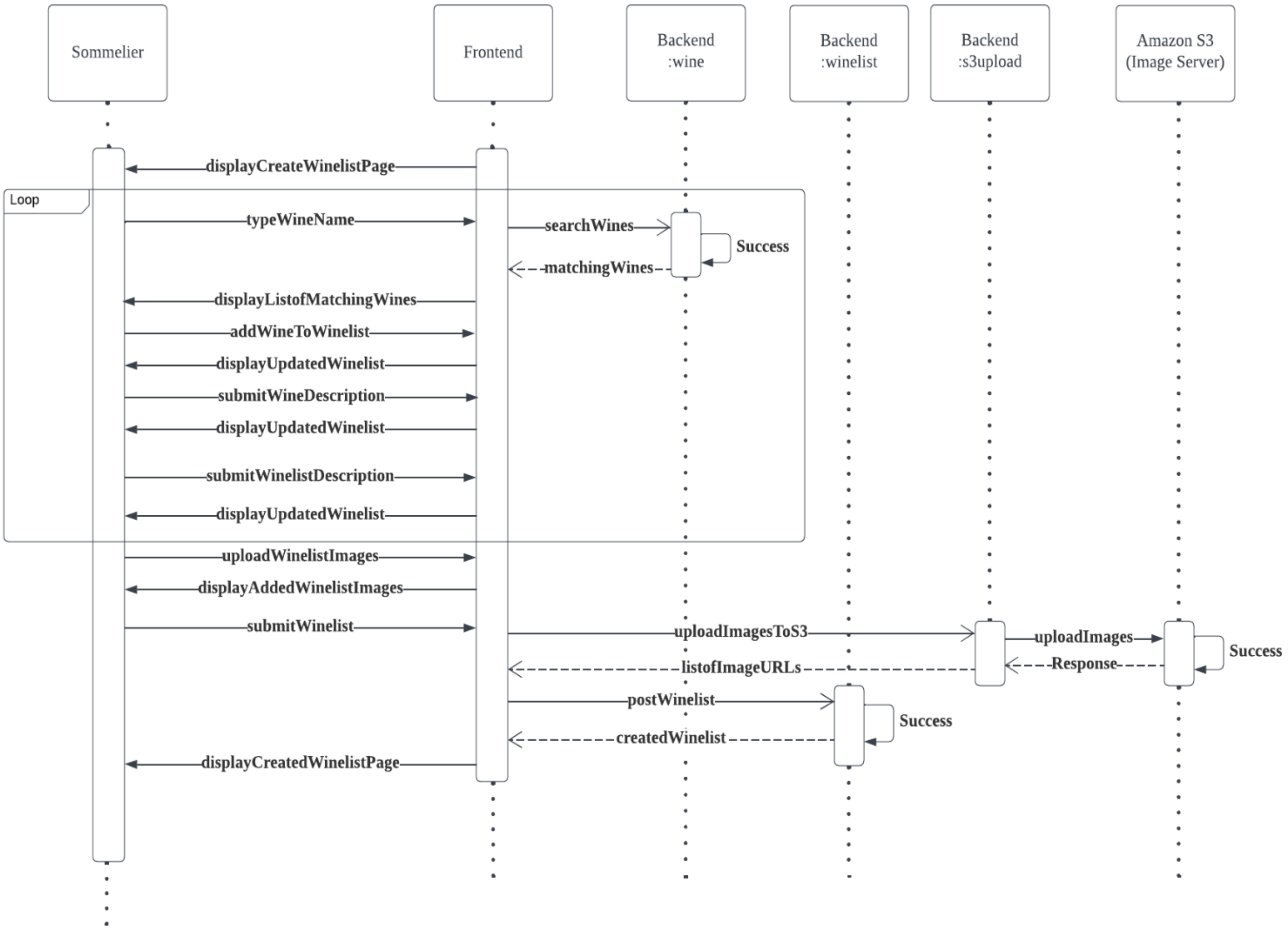
Part	Tech stack	Reasoning
Frontend	HTML	A web browser renders documents created in this format. Creates structured documents by denoting structural semantics for text.
	CSS	Design for better user interactions and layouts of pages.
	React.js (17.0.2)	Easy to reuse components that appear multiple times on the website. There are many libraries that can be used in React.
	JSX	React uses JSX which allows HTML-like code in React.
	Material-UI	CSS framework used within React.js to provide a professional design.
Backend	Python 3.8	Fairly latest version of the python (even if 3.10 is available, it is not stabilized for all the libraries), also supports the latest versions of all libraries
	MongoDB	Easy to modify with flexible schemas, simple installation, relatively easy to use, good documentation
	FastAPI	Modern, and fast. Helps validate data types, fast development and supports SwaggerUI which makes API test run much easier. Also Netflix adopted FastAPI (reliable)
	Docker	Increases overall portability. Ensures an isolated environment. Greater parity between development and production environment. Easy to transfer to other platforms (ex. Heroku -> AWS). Many IT companies adopt Docker.
	Amazon S3	Used for image hosting. Offers a set of APIs for controlling files, also the hosting server is reliable at the same time. Easy to upload and access images.

3.2 Data Design

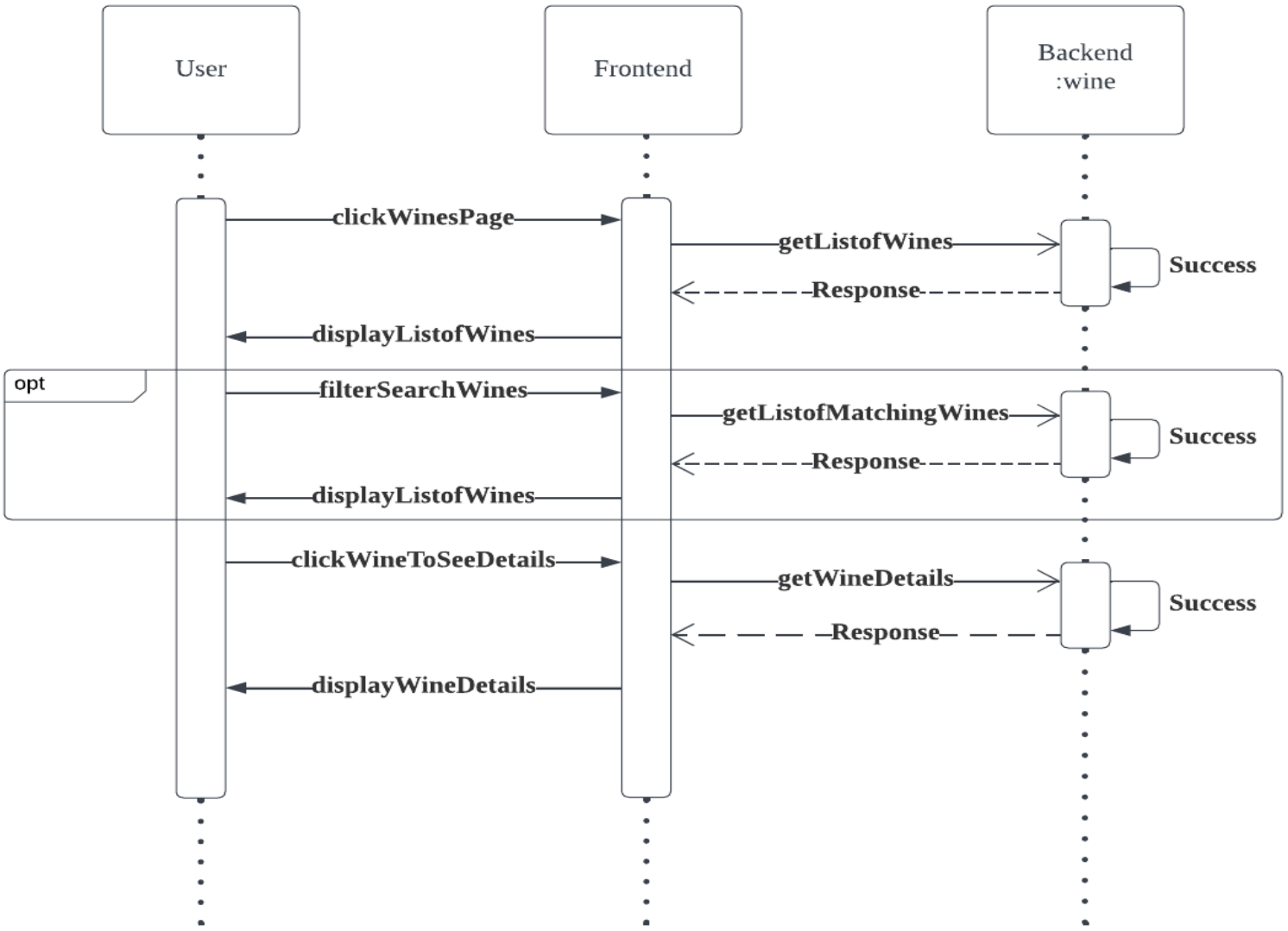
Link to Data Design

3.3 UML Sequence Diagrams

3.3.1 Use Case : Create Wine List



3.3.2 Use Case : Go to Wine details page



3.4 API Design

Link to API Design

3.5 Deployment

Front-end deployment would be on Firebase, since it is easy to manage authentication of the login process. Back-end deployment would be on Heroku. It allows people to use it for free for up to 1,000 hours.

3.6 Code Conventions

Part	Convention	Description
Front-end	Prettier	A code formatter plugin to provide formatting consistency. It automatically formats the codes in a predefined way.
	Airbnb React/JSX Guide	Used to provide the standard that is currently prevalent in React/JSX
	BEM methodology	Used to enhance readability and clarity for CSS naming
Back-end	CamelCase	For the python code, namings will be in CamelCase to provide consistency and readability
	REST API	API URIs will follow the REST API convention, which provides readability and great flexibility

4. Schedule

<table>
<tr><th colspan="4">Milestone 1</th></tr>
<tr><td>Hyerin Choi</td><td>Woohyung Lee</td><td>Woohyun Park</td><td>Dukyoung Eom</td></tr>
<tr><td>● Wine component
● Tag component
● FAQ component</td><td>● Sidebar Component
● Footer Component
● Review component
● Button component</td><td>● Desktop View (Figma)
● Header Component
● Winelist Component</td><td>● Scraping Vivino
● Docker setting
● Populating DB with dummy data</td></tr>
<tr><th colspan="4">Milestone 2</th></tr>
<tr><td>Hyerin Choi</td><td>Woohyung Lee</td><td>Woohyun Park</td><td>Dukyoung Eom</td></tr>
<tr><td>● Ticket Modal
● FAQ page
● Wine page
● Wine-comment page</td><td>● Filter Modal
● Sort Modal
● Verification Modal
● Register Page</td><td>● Search Modal
● Wine List Page
● Wine Theme Page
● Wine List Them Page
● Search Result Page</td><td>● CRUD Implementation for front pages (it is flexible depending on other teammates work progress)</td></tr>
<tr><th colspan="4">Milestone 3</th></tr>
<tr><td>Hyerin Choi</td><td>Woohyung Lee</td><td>Woohyun Park</td><td>Dukyoung Eom</td></tr>
<tr><td>● Login page
● Verify sommelier
● Manage tickets</td><td>● Profile Main Page
● Follower/Follow List Page
● Edit Profile Page 2</td><td>● Search Result Page
● Create Winelist Page
● Manage Reviews (Admin)</td><td>● Image upload to S3 and image optimization
● Work on rest of the APIs.
● Bug fixes</td></tr>
<tr><th colspan="4">Milestone 4</th></tr>
<tr><td>Hyerin Choi</td><td>Woohyung Lee</td><td>Woohyun Park</td><td>Dukyoung Eom</td></tr>
<tr><td colspan="3">● Beta release, bug fixes
● Work on optional features
a. Comment on comments
b. View option buttons
c. Ask experts to examine Podo</td><td>● Recommendation algorithm
● Bug fixes</td></tr>
<tr><th colspan="4">Milestone 5</th></tr>
<tr><td>Hyerin Choi</td><td>Woohyung Lee</td><td>Woohyun Park</td><td>Dukyoung Eom</td></tr>
<tr><td colspan="4">Refactor(tentative)</td></tr>
<tr><th colspan="4">Milestone 6</th></tr>
<tr><td>Hyerin Choi</td><td>Woohyung Lee</td><td>Woohyun Park</td><td>Dukyoung Eom</td></tr>
<tr><td colspan="4">Testing (tentative)</td></tr>
</table>

5. Contributions

Name	Contributions
DukYoung Eom	● Modified Database Schema Design ● API Design ● UML Sequence Diagrams ● Participated in distributing schedules to milestones
Hyerin Choi	● Participated in distributing schedules to milestones ● Contributed equally to creating the Software Design Document
Woohyun Park	
Woohyung Lee	