# ASSIGNMENT 1:

# iOS Development

Name: Truong Hong Van (s3957034)

Instructor: Mr. Tom Huynh

COSC2659|COSC2813 iOS Development

RMIT University Vietnam, Semester 2024B

# Table of Contents

# Introduction

## I. Project background

The word “cocktail” is rich in history and tradition since it takes a long time to distill and gather the bases of a cocktail, such as whiskey, tequila, and vodka. From classic cocktails to modern espresso martini, cocktails have become an integral part of social gatherings and events. Based on this significance, several cocktail recipe apps have emerged to suit the needs of those passionate about this type of mixology. However, these apps usually are not well-organized and lack visually appealing themes. With motivation in mind, I decided to develop *Le Tonneau* – an application that drives users to the world of cocktail making.

## II. Project overview

### 1. Goals and objectives

The goal of this project is to offer a comprehensive and engaging platform for users to explore different types of cocktails. The objectives include:

- Provide users with a reasonable database of cocktails and mocktails, ranging from classic to modern cocktail types. Each cocktail includes detailed ingredients and possible locations to buy them.
- Offer the visualization of cocktail-making procedures and drink decorations for users.
- Design an intuitive user interface with a vintage aesthetic design to create an appealing and unique user experience.

### 2. Target audience

The target users for Le Tonneau are limitless, meaning that any users who are passionate about cocktails can join the app and try a recipe. The documentation can be read by any developers and designers who are interested in understanding the implementation details and design patterns of Le Tonneau. This report can also serve as a user manual to guide users through the functionality of the app.

### 3. App name and logo explanation

The app name *Le Tonneau*, meaning The Barrel in English, is inspired by the fact that barrels have long been used for the wine aging process throughout history. Wine is the main ingredient of a cocktail, and this is also the spirit of this kind of drink. Besides, the app logo centers a shadow of the Old-fashioned cocktail, an infamous drink ever made in the earliest days of cocktails (NIO n.d.). Through Le Tonneau's logo design, the app aims to bring the tradition, craftsmanship, and history of cocktails to those interested.

# Implementation details

## I. In-scope

*Le Tonneau* consists of three main views:

- Welcome view: The welcome view introduces the application logo, title, and slogan to the users, with the author's information provided.
- Navigation list view: This view is the primary interface with which users can interact. In particular, the user can choose from a list of categories to view all related cocktail drinks. There is also a search bar and filter options so that the users can retrieve the recipes that meet their needs.
- Detailed recipe view: The Details view provides essential information about a cocktail, including history, level, wine base, ingredients, and instructions. Besides, a collection of images introducing different ways to decorate the drink and locations to buy the ingredients are also provided.

Besides, users can change the theme of the app based on their preferences. The app's theme follows the system mode by default but can be changed to light or dark mode by the user.

The detailed description and implementation explanation of each feature is outlined below.

### 1. Welcome view

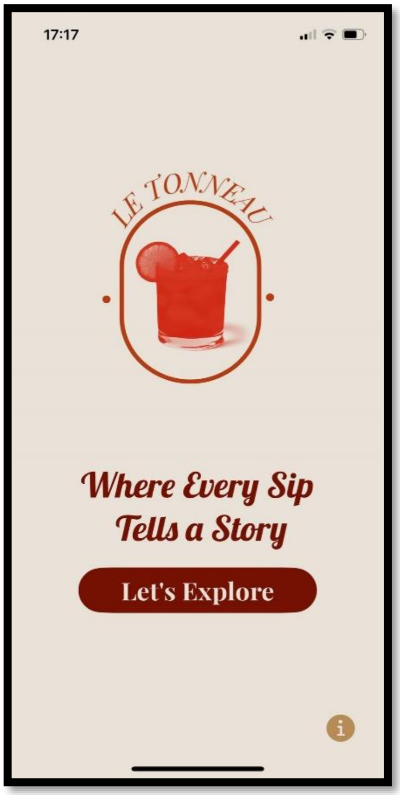
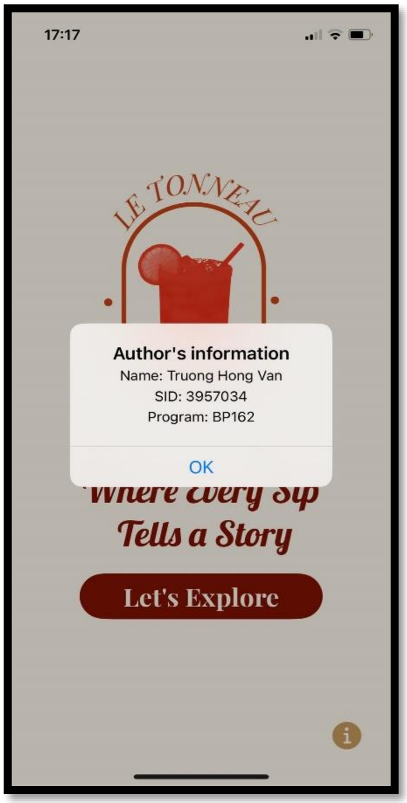


*Figure 1. Welcome view*



Author's information
Name: Truong Hong Van
SID: 3957034
Program: BP162

*Figure 2. Welcome view – Alert dialog*

The Welcome view consists of three main components: application's overview, navigation button and pop-up information box (Figure 1 and 2). These elements are organized in a main VStack view.

| Functionality | Description |
|---|---|
| Application overview | The overview includes the application name, Le Tonneau, the logo image and the slogan. |
| Navigation button (Figure 1) | The custom button is wrapped by NavigationLink, which controls the navigation presentation (Apple Developer 2024). This view enables directing to the Item List View destination when clicked on. The default NavigationLink behavior has a "back" button for the users to navigate to the Welcome View. However, the "back" button would be set hidden in this case. |
| Pop-up information box (Figure 2) | The information box is mainly controlled by the state behavior and Alert view (Apple Developer 2024), allowing the box to be open/closed. When the user clicks on the information icon, the showAlert state is toggled, and the alert dialog is displayed. The content of the author's information is covered in the dialog. Finally, the relative position of the information icon is modified using the *offset()* modifier. |

*Figure 3. Welcome view – Feature description*

## 2. Navigation list view

The navigation list view is divided into five main components, organized in a VStack view. Additionally, ScrollView view is used to wrap around the main VStack view, so that the users scroll down to see a wide range of cocktails with ease.

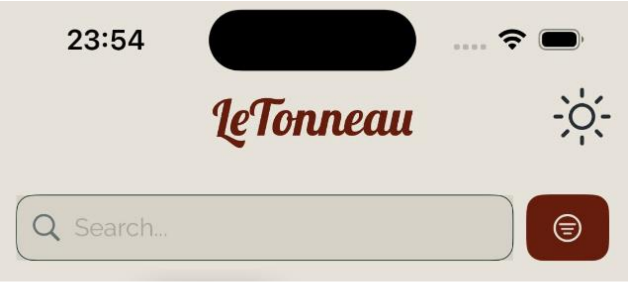


*Figure 4. Navigation – Search bar*

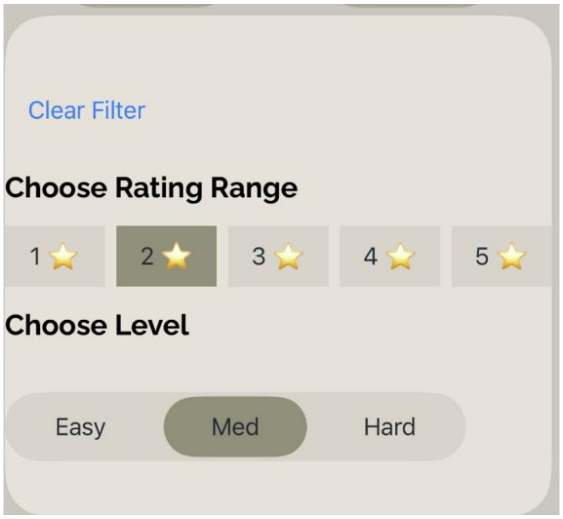


*Figure 5. Filtering options*

| Functionality | Description |
|---|---|
| Navigation bar (Figure 4) | The navigation bar contains the app name and the theme changer button, which displays in a fixed position at the top of the screen, even when the user scrolls down the view. This is achieved by implementing the NavigationView view with its modifiers. In particular, the display mode of the bar title is set to display in an inline style, and the button is added to the bar using the *toolbar()* modifie (SoftwareReader 2024). The ToolbarItem component is then used to style and position the navigation buttons. |
| Search bar (Figure 4) | Regarding the style, the search bar is a TextField view that holds the user input, allowing users to find a cocktail recipe by name. It retrieves the user input from a property wrapper Binding, which enables tracking the data changes through different views (Apple Developer 2024). The searching process of a recipe is performed simultaneously with other filter options. |
| Filtering (Figure 5) | The filtering function is implemented to filter out items based on some criteria, such as category, rating, level, and search query. Each criterion is managed by a state and is passed to the child view using binding. To explain, if any of the values in the criteria list are provided, the system will filter out the items, ignoring case differences for String. The filtering criteria retrieve values in three ways:<br>- The search query is retrieved from user input in the search bar.<br>- The selected category is decided when the user clicks on a category in the category list.<br>- The rating and level values can be chosen from the Filter modal in a sheet view. The sheet view is open over the existing view when clicking on the filter button, which is implemented using the combination of *the sheet()* modifier and a state to display the sheet (Hudson 2022). Furthermore, the user can easily drag down to dismiss the sheet, creating a clean and user-friendly layout. |
| Category List (Figure 7) | The category list displays six categories, which are scrollable horizontally. If a category is chosen, its icon appears larger with a shadow to distinguish itself from the others. Additionally, the category name of the selected category is displayed accordingly, along with the recipes of that category. |

| | |
|---|---|
| Drink List (Figure 7) | The drink list is arranged as grids in a LazyVGrid view, allowing it to grow vertically as the number of items increases (Apple Developer 2024). The 2-column GridItem also offers flexibility in controlling the size of each drink card. Furthermore, each item card is a NavigationLink that directs the user to the related recipe information. |
| Dark/Light mode | The theme changer button allows users to switch between different themes in the app (dark, light, system default). The theme is managed by the AppStorage property wrapper, which remembers the theme value across app launches (Apple Developer 2024). The theme picker is displayed in a sheet view. When the user chooses a theme, all the views in the app immediately change the color palettes to fit the theme. Regarding the theme changer button, it has the sun symbol for the light theme, and the moon symbol for the dark theme. |

*Figure 6. Navigation list view – Feature description*

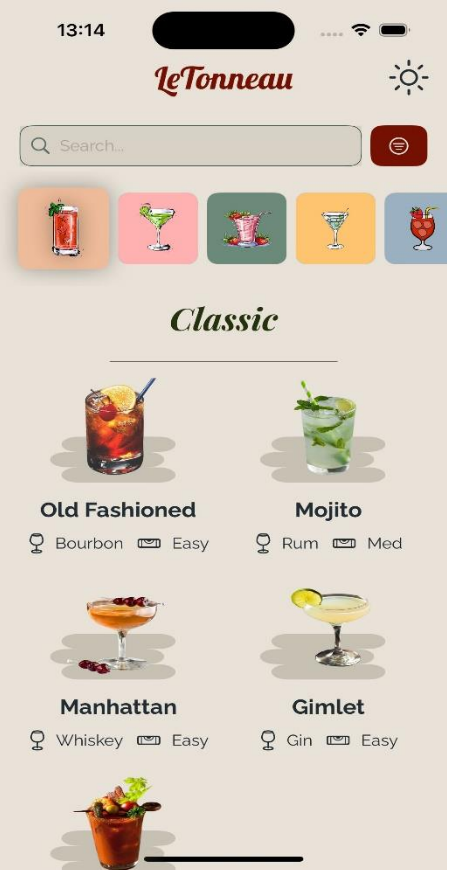
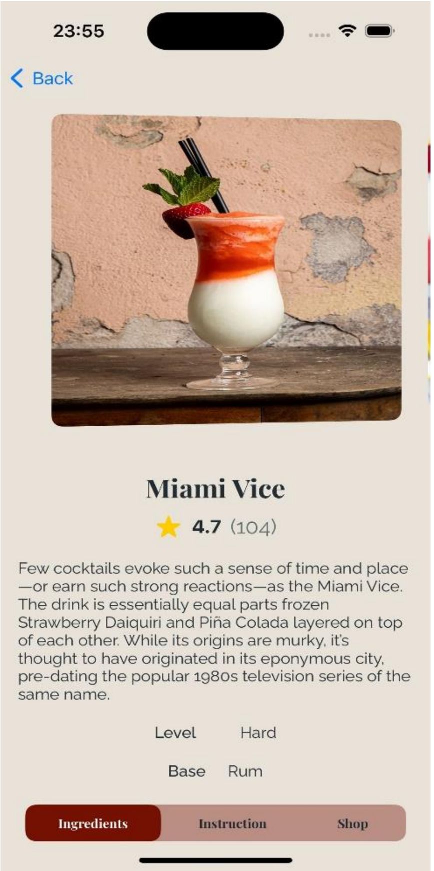


*Figure 7. Navigation list view*



Few cocktails evoke such a sense of time and place—or earn such strong reactions—as the Miami Vice. The drink is essentially equal parts frozen Strawberry Daiquiri and Piña Colada layered on top of each other. While its origins are murky, it's thought to have originated in its eponymous city, pre-dating the popular 1980s television series of the same name.

Level Hard

Base Rum

*Figure 8. Details view*

### 3. Details view

The Details view presents all the vital information about a cocktail recipe. In this view, we can see the story behind the cocktail, along with its rating, level, and wine base. There are seven main components in this view: carousel of images, cocktail descriptions, tab views, ingredients table, instruction steps with video tutorials, clickable social icons, and map view.

For visualization, please visit the Appendix.

| **Functionality** | **Description** |
|---|---|
| Carousel of images | The carousel displays a collection of images showing different cocktail decorations. It is implemented mainly by HStack and ScrollView, allowing the users to swipe the collection horizontally. Regarding the 3D rotation effect, we need the GeometryReader to access the position and size of each image view (Apple Developer 2024). Then, the 3D rotation effect is applied using *the rotation3DEffect()* modifier. |
| Cocktail description | The description displays static information about a cocktail, such as name, description, level, and wine base. The information is arranged in a VStack view. |
| Tab view | This view contains three tabs: Ingredients, Instruction, and Shop. It is implemented by building a horizontal stack to arrange the tab titles side by side. When a tab is selected, the state that manages the selection is toggled, and the selected tab's background color changes accordingly to make it outstanding.<br>The tab titles are placed in an array and are iterated by the ForEach function. The tab content is rendered based on the index of the tab title. For example, if the Ingredients title is at the first position (0), the Ingredient content appears. |
| Ingredients table | The 2-column table organizes a list of ingredients needed to make a cocktail. The table offers a sorting function, enabling the sorting of ingredients by name or quantity. By that, the cases for sorting are defined in an *enum,* namely nameAscending, nameDescending, quantityAscending, and quantityDescending. The sorting function is implemented by comparing the name or quantity of each ingredient. To enable the sorting, the users can click on arrows beside the name or quantity title. Additionally, the direction of the arrow changes according to the direction of the sorting. |
| Instruction steps with video tutorials | In terms of instruction steps, an array of instructions is retrieved from the Item model. It is iterated for the display, with the index set as the step number. |

| | |
|---|---|
| | Regarding the video tutorials, the YouTube videos are embedded with the functions provided by the WebKit module (Ko 2022). Firstly, the VideoView structure is created. It must conform to the UIViewRepresentable protocol so that WKWebView – an UIKit view class (Apple Developer 2024), could be implemented. Then, we create a URL, with the video ID retrieved from an Item object. Finally, the video is loaded into the view for display. |
| Clickable social icons | The social icons, including Facebook and Shopee, act as buttons to open other platforms. This action is achieved by using UIAppliation.shared.open(URL), which opens a given URL and directs users to social media applications (Apple Developer 2024). |
| Map view | The Map View indicates a store location to buy the cocktail ingredients. Firstly, we define the initial region of the map with the location retrieved from the Item model. Then, a pin is added at the location's coordinates. By tapping on the pin, an information dialog showing the name and detailed address of the store is displayed. This dialog is also controlled by a "popup" state variable. |

*Figure 9. Details view – Feature description*

### II. Out of scope

Although Le Tonneau is developed devotedly and carefully, there are some features that the app currently does not support and also known problems.

**1. Limitations**

- Insufficient data: The app currently only supports 5 categories of cocktails and mocktails, with 15 cocktail drinks. The store location for map view is also limited.

**2. Unsupported features**

- Font size customization: The users cannot modify the font size to their preferences to aid readability.
- Add to Favorite: The users cannot save their favorite recipes.
- Recipe portion: The users cannot view the changes of the recipes based on the portions they intend to serve.

## Design elements and User experience

### I. Visual appeal

Since Le Tonneau is inspired by retro-theme cocktail bars, the app is expected to create a rich and unique vintage atmosphere for users who experience it. Hence, the design elements such as fonts and colors should align with the app theme.

### 1. Fonts

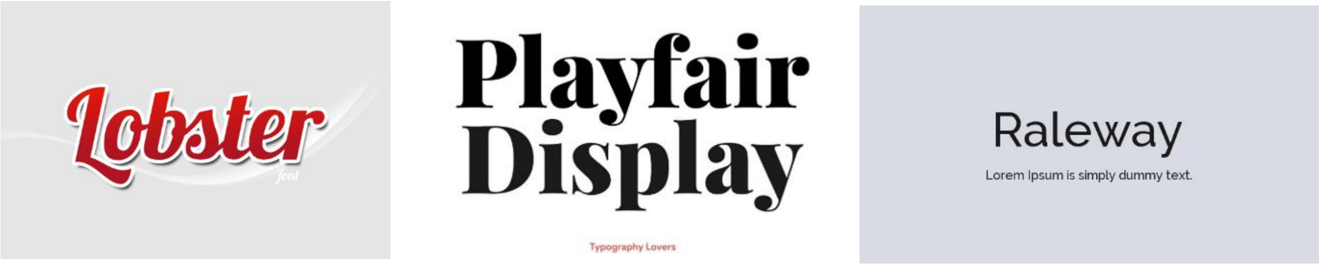


*Figure 10. Font style*

The app makes use of three main font styles to deliver the content. All font styles are customized and imported from Google Fonts.

a. **Lobster** (Google Fonts n.d.)

This cursive script font brings a sense of vintage feeling, and it is also often seen in classic cocktail bars in the mid-20th century (Hilton 2019). Thus, it is chosen to display the app name *Le Tonneau, hoping it gives a memorable and distinctive appearance to the app's identity*. However, it is not used for the content, due to the concern that its script style affects the readability during the experience.

b. **Playfair Display** (Google Fonts n.d.)

Playfair Display has long been seen on the menu of some cocktail places, bringing a sense of history and tradition. Because of that, it is selected to indicate headings and subheadings. The high contrast between the thin strokes and serifs gives it a classic and elegant representation. Because of its distinctive appearance, it can naturally draw attention to headings and subheadings, effectively guiding the user’s eye.

c. **Raleway** (Google Fonts n.d.)

Raleway is a minimalist sans-serif font that gives a neat and organized look. Therefore, it is applied to deliver most of the app’s content, including body text and descriptions. The primary aim of choosing this font is to ensure a high level of readability of important information, thereby enhancing user experience. Moreover, this font provides a balanced contrast to previous decorative fonts like Lobster and Playfair Display.

## 2. Color schemes

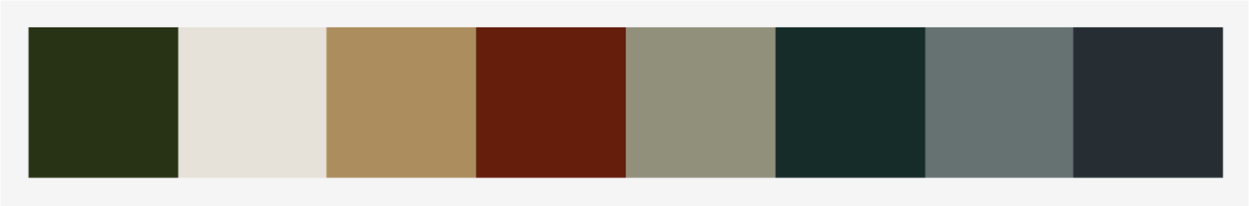


*Figure 11. Color palettes*

The main color schemes used in Le Tonneau consist of earthy and muted tones which evoke a sense of nostalgia and history, thereby aligning with the vintage theme of the app. Besides, the contrast and complementary amongst colors are also taken into account, so as to bring a balanced look and ensure that the design is visually appealing without being overwhelming for users. The color combination for light and dark theme is explained below. For visualization, see Appendix.

### a. Light theme

In the light theme, the primary color chosen is Cranberry (#741102). It is used mainly for app identity and primary buttons. Meanwhile, the Vintage Gold (#B78C56) is applied to secondary buttons and information, ensuring it still looks outstanding in the background but does not overwhelm the red color. Next, the green-shaded colors act as accent colors, often used for shadow and unimportant texts. Overall, the light theme design aims to give a fresh and clean look, making the app more open to users than the dark theme.

### b. Dark theme

In marked contrast, the dark theme makes the most use of green colors, including Midnight (#042D29) for the background. The primary color is switched to Vintage Gold, ensuring the primary interactive elements are noticeable to users. Choosing Midnight as the background, the dark theme aims to convey a more luxurious and elegant atmosphere, compared to the light theme.

## 3. Layouts

Since the colors used in Le Tonneau are usually dark, the layout of the app is expected to follow a minimalist style so that the users are not overwhelmed with complex and strong-colored designs. With that principle in mind, each section in the app is distinguishable, with headings and indicators to guide the users throughout the experience. In particular, the main layout is structured with the navigation bar at the top, followed by the search and filter buttons. Then, the list of categories and items is displayed below, making the overall design neat and organized. Because of that, the layout is believed to enhance the visual appeal in combination with colors and font styles.

## II. Consistency

### 1. Fonts

Each font style has its usage without overlapping with one another. Throughout three main views, we consistently use Lobster for branding, Playfair Display for headings, and Raleway for body text with description.

Besides, the text follows a structured hierarchy. In particular, the brand name is always outstanding at the top of the screen with primary colors. Besides, the 1st heading (the name of each category) is displayed in italics with a larger size than the sub-headings. Finally, the body text has the smallest font size compared to the others, with sufficient spacing to enhance readability.

### 2. Colors

The color palettes, either in light or dark themes, are applied uniformly across the app. If the app is switched to a dark theme, all the views, including sheet views, are displayed accordingly. Moreover, the use of primary colors for primary buttons, secondary colors for less important elements, and accent colors to highlight important parts are applied throughout the app.

### 3. Imagery

The application of images and icons is consistent across three main views. To elaborate, the image in the item cards needs to have a transparent background, while the collection of images displayed in the carousel should maintain similar width and height. Additionally, all the images are high-quality to maintain a professional and engaging look.

Regarding icons and buttons, they should follow similar styles and be large enough for users to notice and interact with.

## III. User-friendly interface and user-centered design

Le Tonneau is expected to be intuitively used even by novice users. That concern is addressed by analyzing two aspects, usability considerations and prioritizing user needs.

### 1. Prioritizing user needs

The design elements align with the Human Interface Guidelines (Apple Developer 2024). Additionally, the app functionality centers around the users, making sure that users can find the preferred cocktail recipes with ease.

### a. Design

**Navigation**: The navigation bar is placed at the top, followed by the search bar and category icons. Each element in the bar contains straightforward symbols. For example, the search bar has a magnifying glass, while the theme changer button has a sun symbol. This practice ensures that the users intuitively understand the usage of the elements.

**Iconography**: Icons used in the app are taken from SF Symbols (Apple Developer 2024). The icons are large enough and straightforward to understand at a glance.

**Typography**: The text follows a hierarchical order with different font sizes, ensuring the users can distinguish between body text and headings.

**Interactive elements**: Buttons and interactive elements are large enough with simple symbols to be easily tapped.

### b. Functionality

**Customization**: The users can switch between light and dark modes to choose the interface that best suits their preferences.

**Finding items**: The users can search for an item with specific names or browse the recipes by level and rating with the filtering options.

**Ingredients and Instructions**: The users can sort the ingredients by name or quantity. Besides, the video tutorials facilitate the cocktail-making process for novice users when they can observe and follow the instructions.

**Shop**: The app provides users with physical and online locations to immediately shop for their ingredients without the need to look up themselves.

### 2. Usability considerations

To evaluate the usability of Le Tonneau, we can analyze some aspects of the app following Nielsen’s Ten Usability Heuristics (NN/g 2024).

**Match between the system and the real world:** The app content is written in simple language, which is understandable for novice English users. The icons and symbols are straightforward in meaning, making sure the users intuitively interact with them.

**User control and freedom**: The users always have the option to undo or cancel an action. Regarding filtering, they can tap “Clear filter” to remove all the criteria applied. They can also exit a chosen category by tapping on it again. Additionally, the “back button" is always visible for users to exit the Details view.

**Recognition rather than recall**: The interactive elements such as buttons and maps are clearly labeled, helping the users to recognize and seamlessly navigate. For the category list, the users can easily understand which category they choose by tapping on a category and seeing the name displayed.

With these usability considerations, it can be seen that Le Tonneau is relatively user-friendly and user-centric designed.

## Conclusion

Throughout the development of Le Tonneau, I have gained valuable insights as a software developer. Firstly, I had first-hand experience building a simple iOS application with SwiftUI. The feature implementation significantly sharpened my skills regarding logical thinking and programming. Secondly, I learned how to design a user-friendly interface and pay more attention to building an application that meets user needs. Finally, I recognized the importance of scope management at the start of the project since there were some intended features that I had not successfully implemented.

In the future, there are several areas where Le Tonneau could be enhanced to offer a more comprehensive user experience. The recipe generation feature can be integrated, so the users can randomly have a drink recipe based on their current ingredients. Moreover, the users can save their favorite recipes to a list with backend integration.

## References

NIO Cocktails (n.d.) *Old-fashioned cocktail history and Ingredients guide*, NIO Cocktails website, accessed 2 August 2024. https://niococktails.co.uk/blogs/news/old-fashioned-cocktail-history-ingredients-guide

Apple Developer (2024) *Alert*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/documentation/swiftui/alert

Apple Developer (2024) *AppStorage*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/documentation/swiftui/appstorage

Apple Developer (2024) *Binding*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/documentation/swiftui/binding

Apple Developer (2024) *GeometryReader*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/documentation/swiftui/geometryreader

Apple Developer (2024) *Human Interface Guidelines*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/design/human-interface-guidelines

Apple Developer (2024) *LazyVGrid*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/documentation/swiftui/lazyvgrid

Apple Developer (2024) *NavigationLink*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/documentation/swiftui/navigationlink

Apple Developer (2024) *SF Symbols*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/sf-symbols/

Apple Developer (2024) *UIApplication*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/documentation/uikit/uiapplication/1648685-open

Apple Developer (2024) *WKWebView*, Apple Developer website, accessed 4 August 2024. https://developer.apple.com/documentation/webkit/wkwebview

Google Fonts (2024) *Lobster*¸ Google Fonts website, accessed 4 August 2024. https://fonts.google.com/specimen/Lobster

Google Fonts (2024) *Playfair Display*¸ Google Fonts website, accessed 4 August 2024. https://fonts.google.com/specimen/Playfair+Display

Google Fonts (2024) *Raleway*¸ Google Fonts website, accessed 4 August 2024. https://fonts.google.com/specimen/Raleway

Hilton N (2019) *Long live Lobster,* Medium website, access 4 August 2024. https://modus.medium.com/consider-the-lobster-75929aa299f6

Hudson P (2022) *How to present a new view using sheet,* Hacking with Swift website, accessed 4 August 2024. https://www.hackingwithswift.com/quick-start/swiftui/how-to-present-a-new-view-using-sheets

Ko J (13 January 2022) “Embed YouTube in SwiftUI”, *Jacob’s Dev Blog*¸ accessed 4 August 2024. https://jacobko.info/swiftui/swiftui-06/

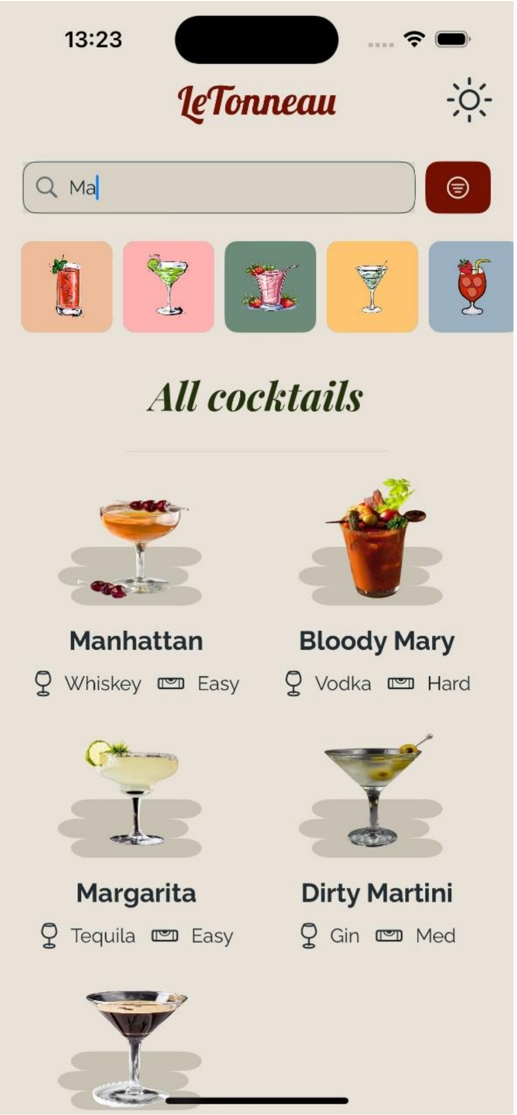
Nielsen Norman Group (NN/g) (2024) *Ten usability heuristics*, NN/g website, accessed 4 August 2024. https://www.nngroup.com/articles/ten-usability-heuristics/

Sofware Readers (2024) *Navigation bar – A complete guide*, Software Readers website, accessed 4 August 2024. https://softwareanders.com/swiftui-navigation-bar-a-complete-guide/

# Appendix

**Link to app demo: IOS-App Demo.mp4** (Please login using RMIT account)

1. **Navigation list view – Visualization**



13:23

LeTonneau

Ma

All cocktails

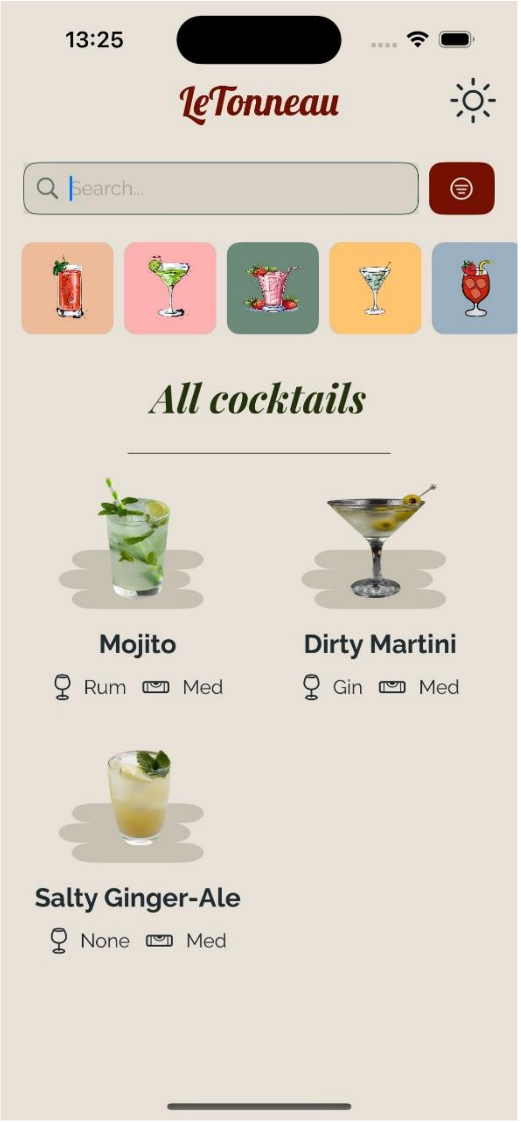
Manhattan
Whiskey Easy

Bloody Mary
Vodka Hard

Margarita
Tequila Easy

Dirty Martini
Gin Med

*Figure 12. Search*



13:25

LeTonneau

Search...

All cocktails

Mojito
Rum Med

Dirty Martini
Gin Med

Salty Ginger-Ale
None Med

*Figure 13. Filter Medium level*

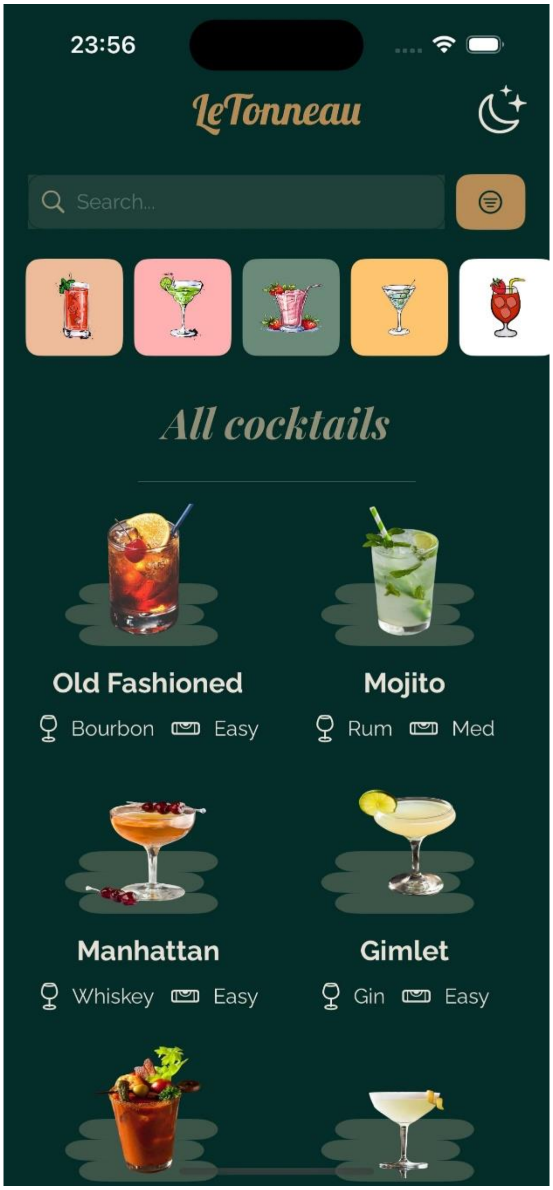


*Figure 14. Dark mode*

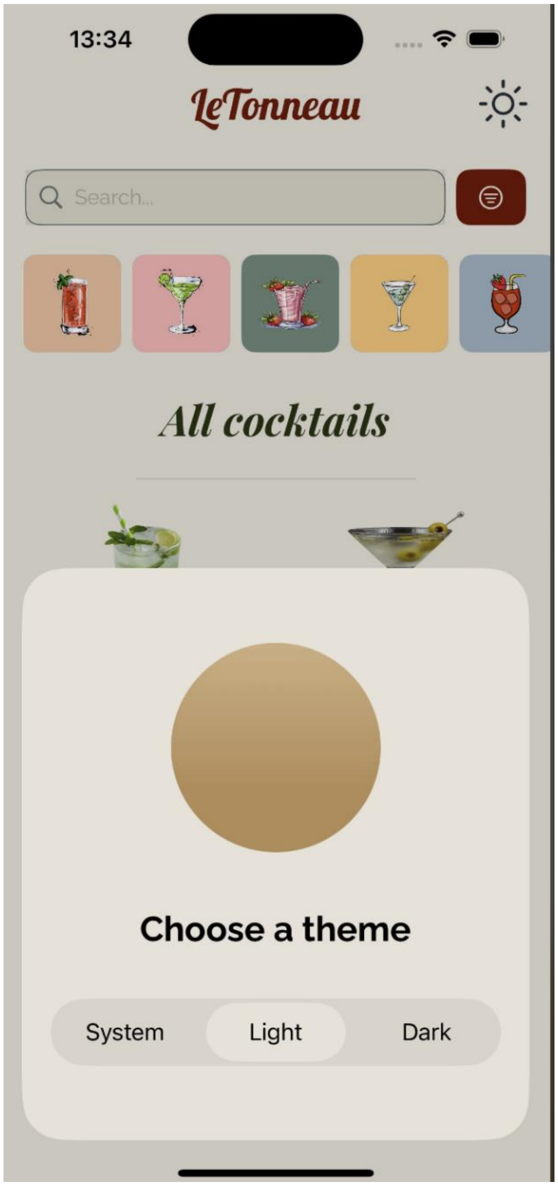
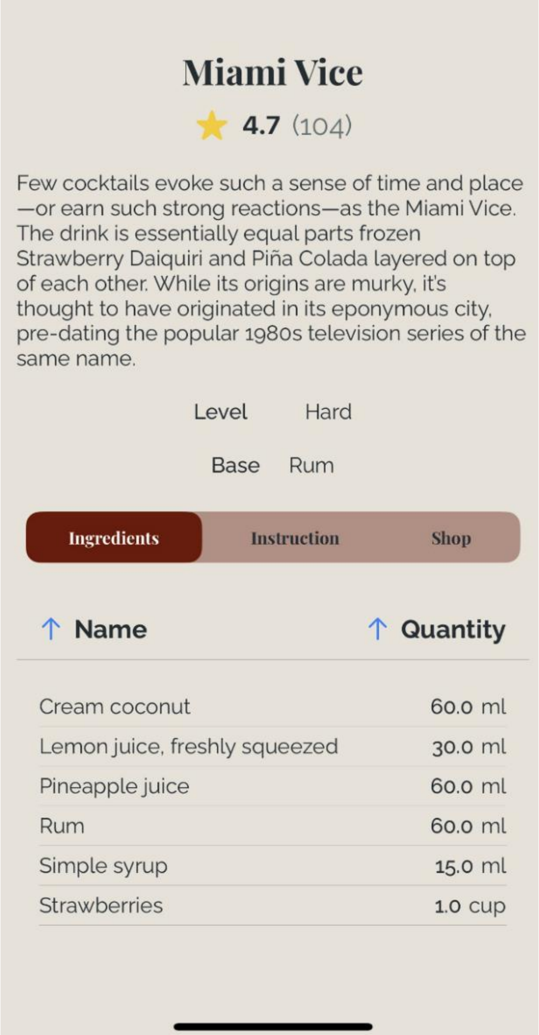


*Figure 15. Choose theme*

## 2. Details view – Visualization



## Miami Vice

★ 4.7 (104)

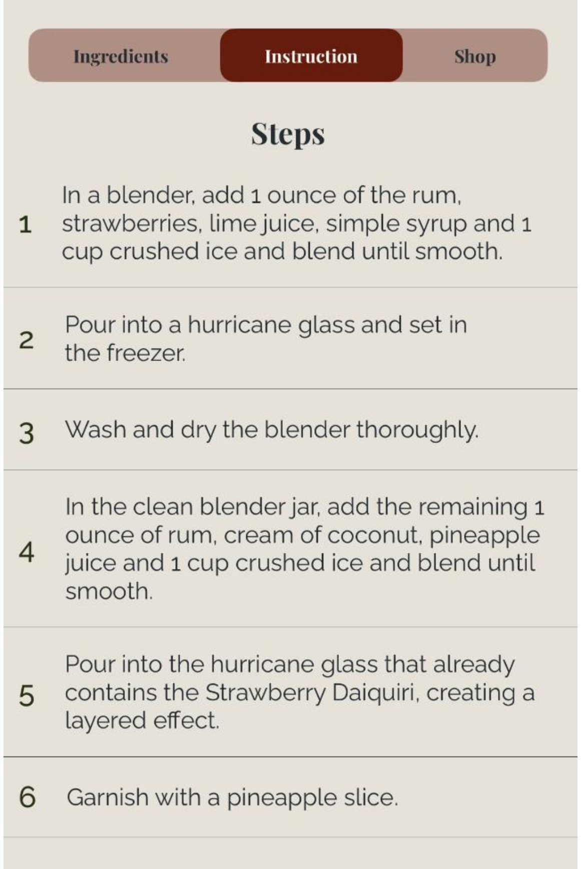
Few cocktails evoke such a sense of time and place—or earn such strong reactions—as the Miami Vice. The drink is essentially equal parts frozen Strawberry Daiquiri and Piña Colada layered on top of each other. While its origins are murky, it's thought to have originated in its eponymous city, pre-dating the popular 1980s television series of the same name.

Level Hard

Base Rum

| ↑ Name | ↑ Quantity |
|---|---|
| Cream coconut | 60.0 ml |
| Lemon juice, freshly squeezed | 30.0 ml |
| Pineapple juice | 60.0 ml |
| Rum | 60.0 ml |
| Simple syrup | 15.0 ml |
| Strawberries | 1.0 cup |

*Figure 16. Details view – Ingredients*



## Steps

1 In a blender, add 1 ounce of the rum, strawberries, lime juice, simple syrup and 1 cup crushed ice and blend until smooth.

2 Pour into a hurricane glass and set in the freezer.

3 Wash and dry the blender thoroughly.

4 In the clean blender jar, add the remaining 1 ounce of rum, cream of coconut, pineapple juice and 1 cup crushed ice and blend until smooth.

5 Pour into the hurricane glass that already contains the Strawberry Daiquiri, creating a layered effect.

6 Garnish with a pineapple slice.

*Figure 17. Details view – Steps*

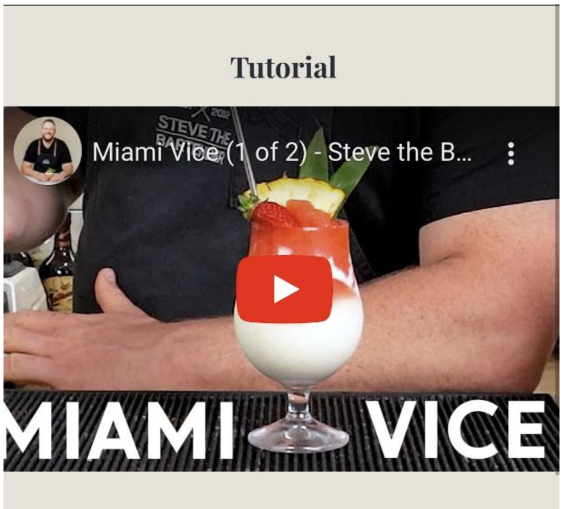
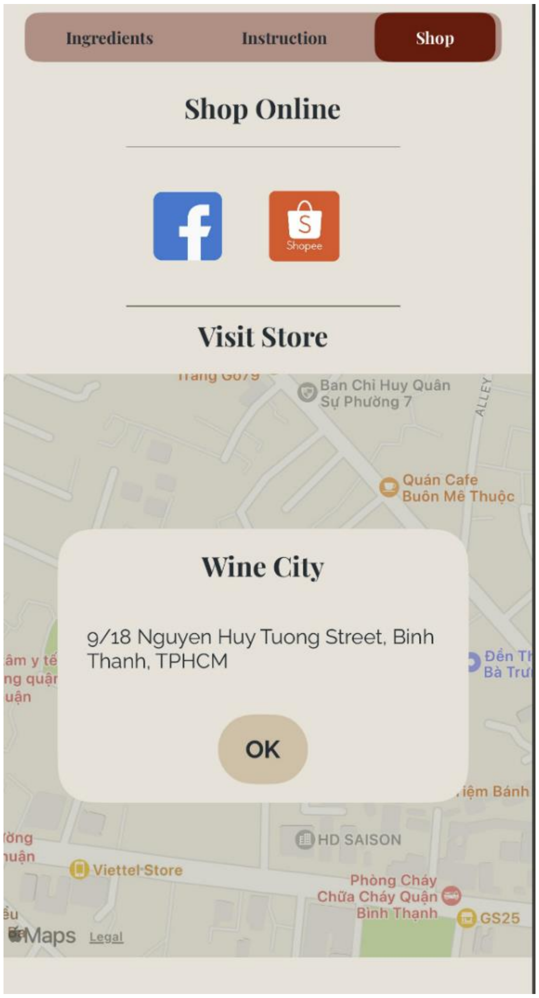


*Figure 18. Video tutorials*



Wine City

9/18 Nguyen Huy Tuong Street, Binh Thanh, TPHCM

*Figure 19. Map view and social icons*