



ASSIGNMENT 1

AspireVogue Application

Course Name: iOS Development

Course code: COSC2659

Lecturer's name	Mr. Tom Huynh
Submitted by:	Duong Vu Thanh Ngoc – S3924496
Due Date	August 7 th , 2023
Submission Date:	April 28 th , 2023

I declare that in submitting all work for this assessment I have read, understood, and agree to the content and expectations of the Assessment declaration.

TABLE OF CONTENTS

I.	Introduction.....	2
1.	Application Introduction.....	2
2.	Motivation and Inspiration	2
II.	Project Description	3
1.	App Overview and User Guide	3
2.	Lessons Learned.....	4
III.	Implementation Details	4
1.	Technical Design and Implementation Details	4
2.	Main features and Advance features.....	6
3.	Application Demonstratrion	12
4.	Known Bugs/Problems	12
IV.	Conclusion	12
V.	Reference	13

I. Introduction

1. Application Introduction

Fashion holds a crucial role in boosting women's self-confidence, and attire is often used as a tool for expressing one's individuality, personality, and style (Keogan 2013). It is precisely for this reason that the AspireVogue application came into existence. As implied by its name, "AspireVogue" conveys the essence of a fashion platform with the purpose of inspiring users to seek and pursue a chic and stylish appearance. This application combines various fashion stores, offering an extensive variety of styles, ranging from casual ensembles to exquisite party attires. AspireVogue provides women with a source of inspiration, empowering them to make choices from a variety of apparel options, thereby enhancing confidence and elevating their self-esteem. Furthermore, the application includes convenient search and save functionalities for favored boutiques, allowing users to easily find and access their preferred store in the future. Comprehensive information about each store, including prices, addresses, and maps, becomes readily available with a simple click, making it easier for users to visit and try on clothes. The incorporation of social media channels, such as Facebook and Instagram, enhances the shopping process and enables users to seek advice from vendors directly. In summation, this application passionately encourages users to seize and integrate the latest fashion trends into their personal style, inviting them to explore and enhance their fashion style through the application's content and features.

2. Motivation and Inspiration

In addition to the desire to instill confidence in women, similar to the slogan "Elevate Your Style, Radiate Your Confidence," AspireVogue aims to be an outfit guide that provides a constant source of fashion inspiration by providing a wide range of outfit ideas for different occasions, seasons, and personal styles. Consequently, women can explore various looks and discover new ways to express themselves through fashion. Furthermore, this app is specifically designed for women, considering that societal norms often place more emphasis on women's appearance and clothing choices. Societal pressures to meet beauty standards can lead to body dissatisfaction and negative body image among women (Braizaz 2018). Therefore, women may face more pressure to dress appropriately for different occasions. A fashion guide application can assist them in meeting these expectations and navigating the complexities of dress codes in a variety of contexts. Another important reason that AspireVogue seeks to address is its support for local Vietnamese brands. AspireVogue aims to showcase the diversity of Vietnamese products, ranging from designs to quality. Each shop chosen by AspireVogue reflects its unique personality and keeps up with the latest fashion trends. Moreover, many products have gained popularity among both domestic and international celebrities, yet remain remarkably affordable. In addition to that, AspireVogue desires to promote the spirit of supporting Vietnamese products and receive enthusiastic responses from young people, thereby aiding the development of local Vietnamese brands, not only recognized domestically but also globally.

II. Project Description

1. App Overview and User Guide

The purpose of our application is to facilitate users in exploring contemporary fashion trends, acquiring inspiration for outfits, and feeling confident in their style selections. One prominent benefit of this application is its capacity to provide users with a diverse range of outfit suggestions and the opportunity to explore different fashion trends by means of recommended stores. The application features a user-friendly interface that promotes effortless navigation through various apparel categories, enables store search functionality, and provides the option to save preferred stores. By utilizing real-time trend alerts, individuals are able to proactively anticipate and adapt to the ever-changing fashion landscape. Since then, it has served as a source of fashion inspiration, enabling individuals to explore modern attire options suitable for a diverse range of events, spanning from informal gatherings to elegant parties.

When a user launches an application by choosing its icon on their mobile device, a welcome screen will appear, presenting a brief overview of the application, covering its name, logo, and slogan. The application also includes an information button that displays the author's information, as well as a "Get Started" button that directs users to the app's navigation list.

Upon accessing the list, users will be presented with a full listing of all the stores presently accessible on AspireVogue. The list includes important details such as the store's name, address, pricing information, the shop's clothing style image, and a little heart icon to indicate whether or not the store is one of their favorites. The application logo will be positioned at the uppermost section of the page. This area will also include various functionalities, such as the ability to display preferred stores by clicking on a heart-shaped icon, enabling a dark mode theme by selecting a moon-shaped icon, and providing a search bar to facilitate the user's search for specific shop names. Below that is a section with different options to filter the shop based on its styles, including streetwear, casual, punk, romantic, and glamorous. When users click on a specific style, it will display a list of shops that sell clothes in that particular style. To display users' preferred fashion shops in a specific style section, users can tap on the heart icon and choose the style they desire. Alternatively, they can select the style first and then click on the heart icon.

Upon selecting a particular store from the list, important details regarding the selected store will be displayed. This information includes a carousel showcasing clothing items in alignment with the store's distinctive style, along with the price range, store location, and a map for guidance. Users have the ability to designate a specific store as a favorite on this page by clicking on the heart icon, then the store will be promptly saved and shown on the navigation list page. In addition to providing essential information, AspireVogue facilitates the presence of social network icons, such as Facebook and Instagram. This feature enables users to conveniently place orders and establish direct communication with shop owners via these platforms. Presented here is a concise overview of the AspireVogue application and its potential applications for users' convenience.

2. Lessons Learned

I was able to get a fundamental grasp of SwiftUI, which is Apple's declarative user interface framework, by executing this application. This involves having an understanding of how to design and arrange views, use SwiftUI modifiers to style and layout user interface elements, and manage user interactions. In addition, developing a fashion guide app requires designing an attractive and intuitive user experience. By working on this project, I was able to gain experience in the principles of user interface design, color schemes, typography, and visual hierarchy in order to offer an app that is aesthetically pleasing. Figma is a tool that I have grasped, and as a result, I am able to make accurate prototypes and applications that are appropriate before proceeding with it. In addition, the fashion inspiration app requires data in order to display store information, address, price range, and social media networks. As a result, I learned how to handle data, construct data models, and interface the application with other data sources. Mainly due to the fact that navigating between various sections and displays is a highly crucial part of any program. I have learned how to improve the user experience by implementing navigation features such as modal views and navigation views. Moreover, it is common for fashion apps to entail a carousel of photos, which necessitates the use of effective image loading and caching strategies to ensure a smooth user experience and optimal performance. In addition to that, I discovered how to implement data persistence to preserve favorite stores in such a way that they are accessible even after the app has been closed and reopened. And finally, during my development, I encountered and solved bugs and issues that come in various forms. Some include the elements does not appear as intended due to attributes from different elements. Some are more related to the logic in the code, specifically in the carousel feature, where the image was unable to click on due to an error in the logic behind that piece of code. There was an instance where my app would crash suddenly with just an accident deletion of one line of code and the debug message was unable to detect the error. Fortunately, I was able to revert that change and make the app return to its initial state. I believe that through this experience, my capabilities in testing and troubleshooting SwiftUI code will grow in the future.

III. Implementation Details

1. Technical Design and Implementation Details

The development of a fashion guide application necessitates the creation of an aesthetically pleasing and user-friendly interface. On the welcome view page, a fashionista image was designed featuring the app name displayed prominently in bold white capital letters. The effect was achieved by using three layers of ZStack, a white text image, the model image, and the white outline image, stacking on top of each other. The letters are arranged in a manner that corresponds to each layout and is set in front of a gradient background that transitions from black to white. This design aims to capture users' attention and emphasize the model and name of the AspireVogue application, effectively conveying its identity as a fashion app. Additionally, the logo of the AspireVogue application is composed of the letters A and V, symbolizing the abbreviated form of the app's name. It has been meticulously crafted with a gentle typographical style, featuring prominent bold strokes, thereby evoking a distinct aura of fashion for the user. The logo is positioned in the upper left corner of the page, aligned with a small information button that indicates the author's information. Upon clicking the button, it will show a pop-up view similar to a modal design that shows the information of the author. The button will change the state of the pop-up view which

allow the view to appear or be hidden. The app aspires to enhance users' personal style and boost their self-assurance, as indicated by the slogan "Elevate your style, radiate your confidence" positioned at the bottom of the page. Located beneath the slogan, a button labeled "Get Started" will be present, serving as a hyperlink that redirects the user to the page containing the list of available stores. To provide the user with a more immersive feel, the button will also change its color when the button is held, this is achieved using the button style configuration attribute.

The subsequent view relates to the navigation store listing page, which encompasses a multitude of features. Consequently, the page has been designed with a recognized header to enhance user visibility and facilitate ease of use. The application logo is placed in the upper left corner of the header. In the right corner, there will be three distinct functionalities. These include the option to showcase preferred stores by activating the heart icon, the ability to switch to the dark mode theme by selecting the moon icon, and a search bar to facilitate users in the process of finding specific store names. There is also a section below it that offers various filtering options for the shop's styles, covering streetwear, casual, punk, romantic, and glamorous aesthetics, made by utilizing a segmented picker. Upon selecting a particular style, a full list of stores specializing in clothing of that specific style will be presented. By that, users can easily find shops that sell clothes in their desired style. Presented below is a full listing of all the stores that are presently accessible on the AspireVogue application. The listing comprises each store's essential information, including the name, addresses, pricing information, the store's clothing style image, and a symbolic heart icon denoting its favorite status among other stores. By utilizing the provided information, users have the ability to obtain comprehensive details about the shop, enabling them to determine its compatibility with their desired style and fashion preferences. There is also an up-to-top button to help easily navigate back to the top. Upon choosing a particular store from the provided list, all relevant details related to the selected store will be presented.

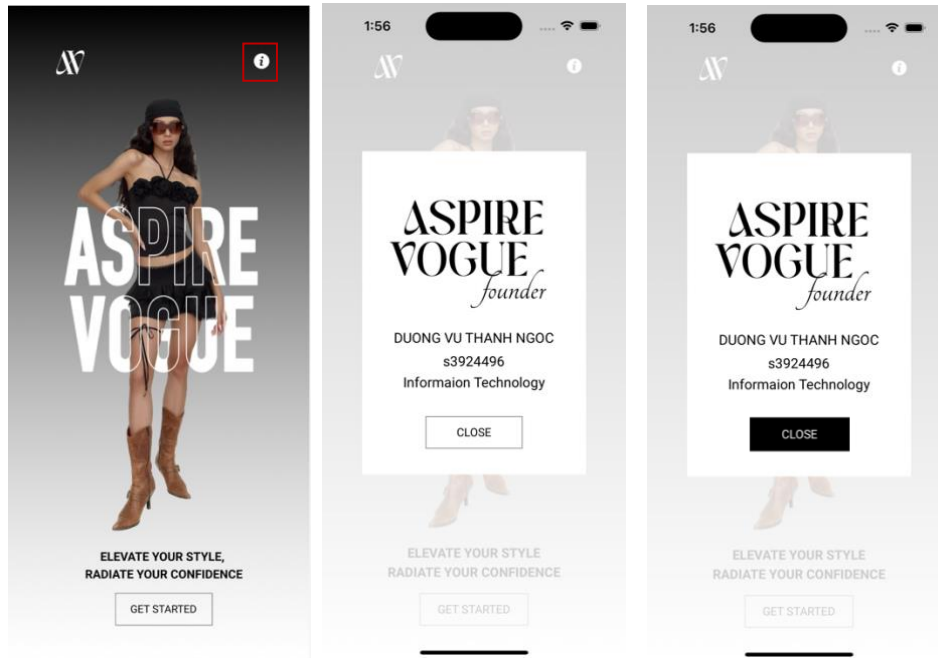
The final view is the detailed display of store information. In the right corner of the header section, a back button will be placed allowing the user to navigate to the navigation list page. The detail view will have presentationMode, and upon clicking on the back button, it will dismiss this presentationMode wrapper value, returning to the previous view, the list view. On the header right corner, there will be a heart icon that allows the user to designate the store as a favorite. Subsequently, the store will be promptly saved and presented on the navigation list page. Adjacent to it is the moon icon which allows for the transition to a dark theme mode. The following section will display a carousel of images showcasing the fashion style of the store. The section below the carousel will provide comprehensive store information, covering details such as the store's name, price range, address, and store introduction. In addition, alongside the store's name, there are also social network icons, including Facebook and Instagram icons. This functionality enables users to conveniently carry out orders and establish direct communication with the store via these platforms. At the bottom of the page, users will find a map displaying the precise location of the store, denoted by a pinned marker. This design feature aims to facilitate users in identifying the store's location, enabling them to conveniently visit the store by selecting and trying on the shop clothing.

2. Main features and Advance features

- **Welcome screen view:**

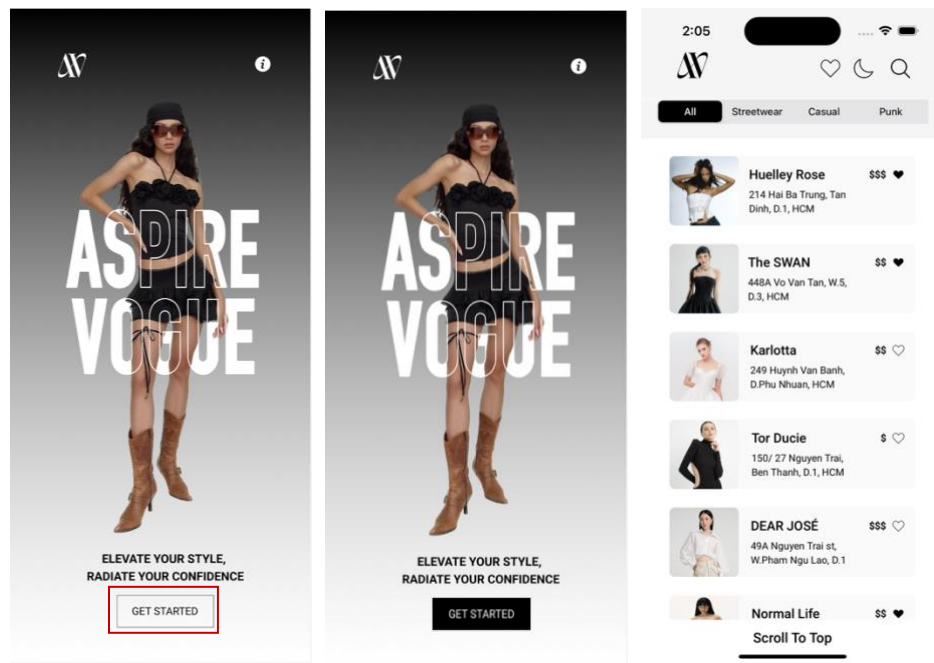
- **Information button that shows the author's information**

The information about the author of the app is stored in a custom view that appears when the state in the welcome view screen changes. The button to toggle the pop-up view is the information button. I design the custom view similar to a modal design and wrap it in a ZStack, when the state changes the modal view will appear in front of the main welcome screen view. Inside the pop-up, there is a button that has the same button style as the get started button that once again toggles the pop-up state which allows the user to hide the view.



- **Get Started button that navigates users to the navigation list page.**

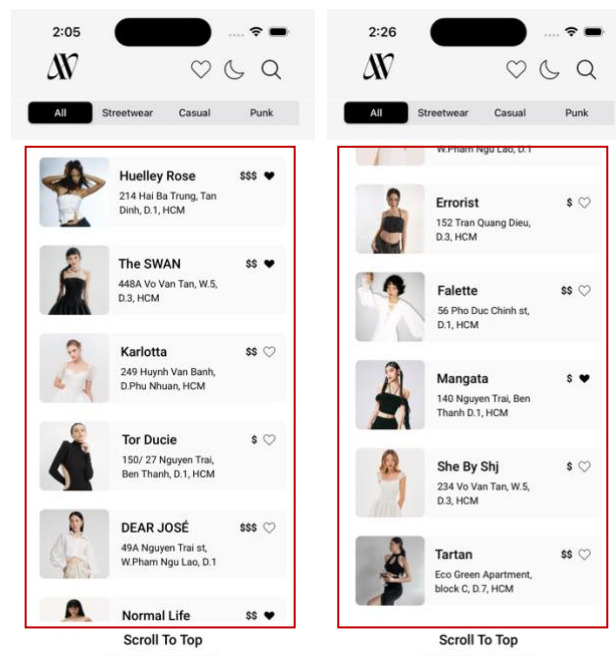
The Get Started button is a simple button that has the navigation link attribute added with the destination being the navigation list page. This is achieved by wrapping the entire view in the navigation view component. To add the fill effect and also easily reuse the styling, I also create a button style struct to be used by both the get started and the close button in the pop-up view. The fill effect is achieved by adding a logical statement inside the button style background and foreground attribute, changing the color when the button is held down.



- **Navigation list view:**

- **Show the list of all stores alongside their name, shop clothing style's image, address, price range, and heart icon showing if the shop is favorited or not.**

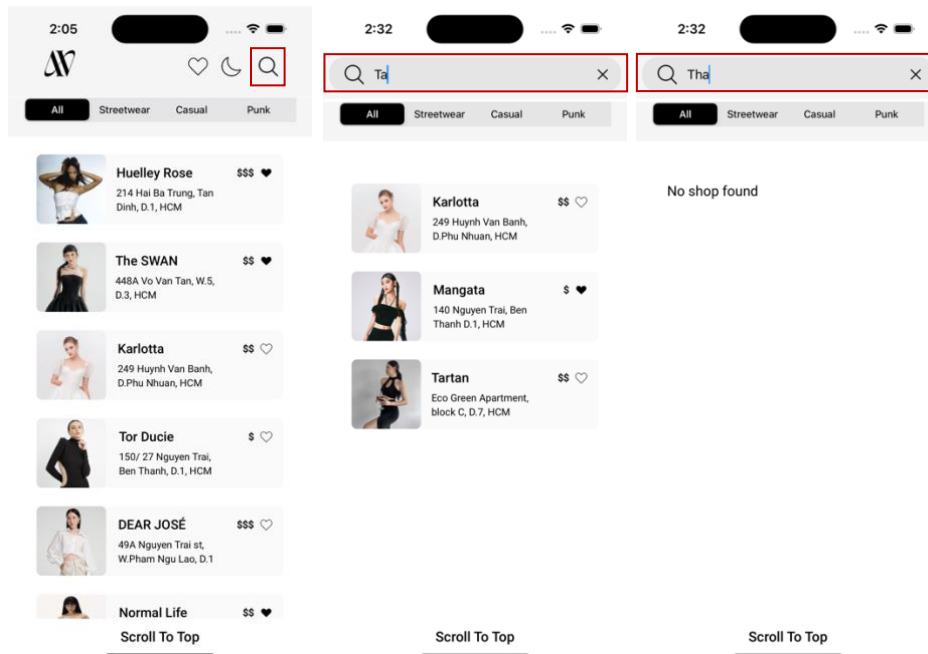
The store list is implemented by looping through the data and adding in the shop row struct with the respective data. The shop row is a struct that style each component in the row. The struct also takes in a Shop parameter which is for the information that will appear in that struct.



- **Search bar to find a shop by its name (Advanced feature)**

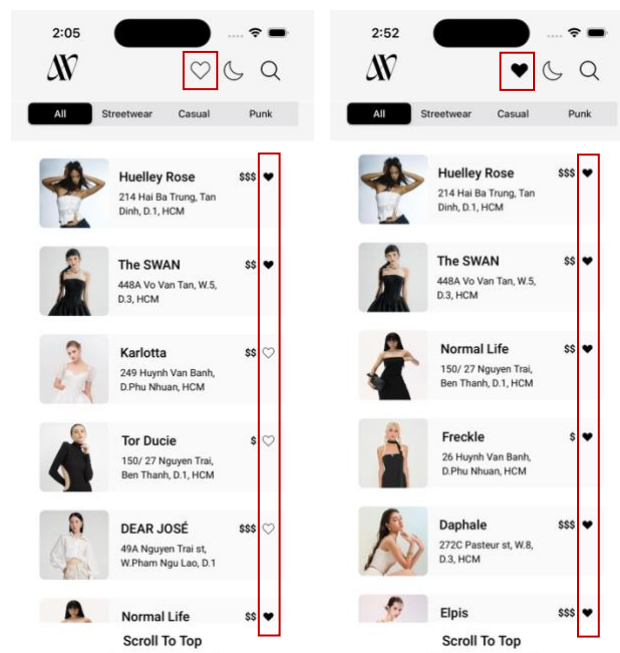
The search bar is implemented by using a text field component and using the input to filter the list. The text field visibility is controlled by the published variable search shown in the observable struct ShopDataStore. When clicking on the search icon, the variable will change and

the text field will appear. The input of the text field is stored inside a state variable called `searchTerm`. Each time this variable is changed, it will update the view to show only shops that have names containing the `searchTerm`. This is achieved by using the filter function on the shop data array, which checks for if the name data contains the `searchTerm`.



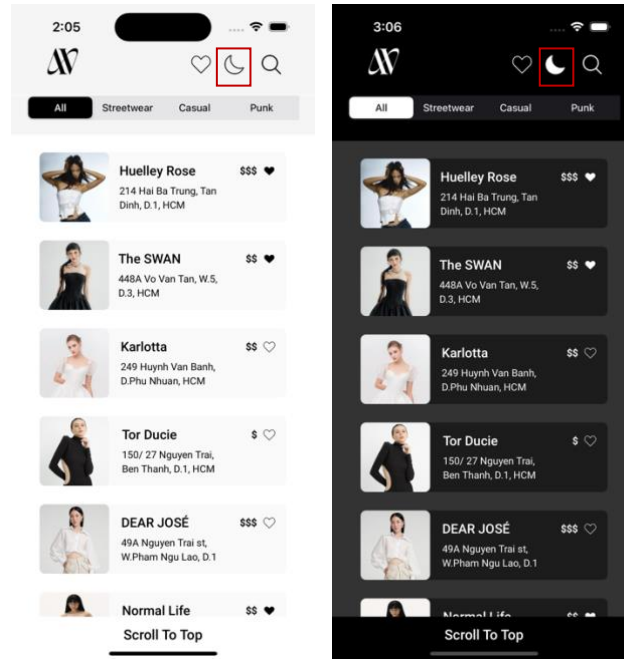
- **Show the list of favorite stores (Advanced feature)**

To have this feature, each shop object will have a `loved` boolean value. There is a published `lovedFilter` variable that is responsible for whether the list will be filtered. The heart button on the header will toggle the variable boolean value, the value will be true when the heart is filled, and false when there is only the outline. Upon the value being true, the list view will filter the shop data to include the shop which has the `loved` value being set to true.



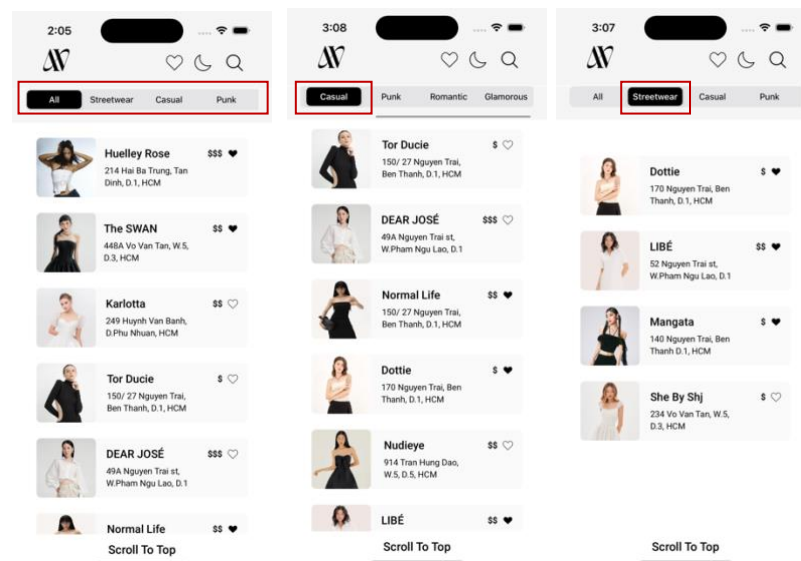
○ Dark mode theme (Advanced feature)

The dark mode is achieved by wrapping the entire view in an environment color scheme attribute, which has a ternary operator to change the scheme based on the `isDarkMode` variable. The variable `isDarkMode` is an `AppStorage` variable, which allows the variable to be saved between different launches of the app.



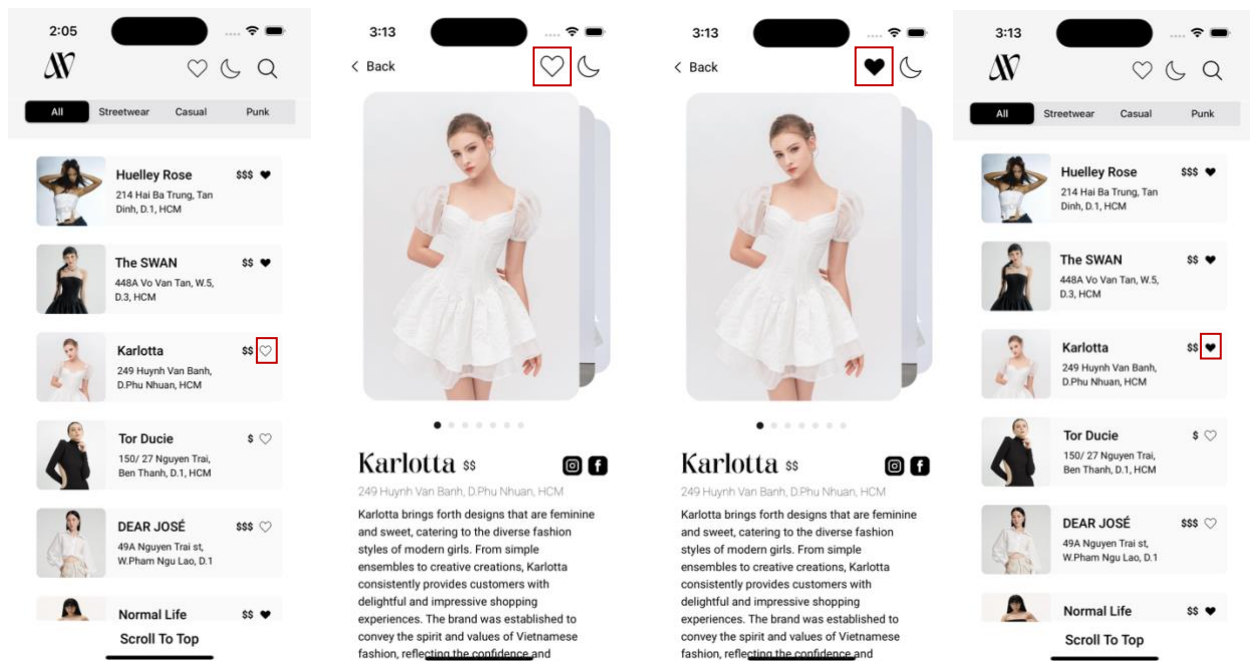
○ Filter shop by category (Advance feature)

This feature utilizes the segmented picker component in SwiftUI. The picker is customized to have its own background, and text color when selected. The picker is also put in a horizontal scroll view since the list is quite long. The picker will have two variables, `selected`, the variable to manage which category is chosen, and `shopCategory`, the category list used for the picker. Each time the user changes the selected category in the picker, the selected variable will be changed. The selected variable is used to filter the shop based on its category before the `for each` loop to display `ListRows`.



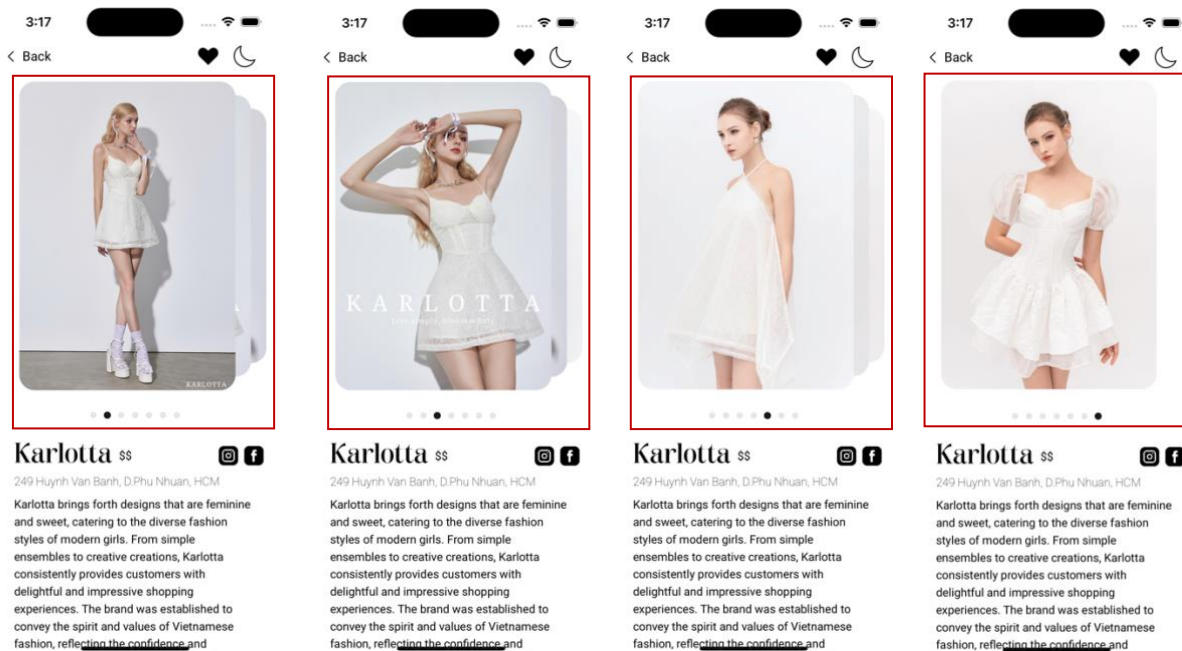
- **Detailed information about each store view:**
 - **Modify the favorite store list (Advanced feature)**

Since the shop data is a published variable, any change made to it will be reflected instantly to the app, this is the basis of this feature. The heart button will show the current loved value of that specific shop in the shop data variable, when clicked on, the loved value of the shop will change, and the data in the shop data will also change. To indicate the change, the heart button will change its image depending on the value, and all of the components that use this value will also change, such as the heart icon in the shop row in the list view. Moreover, after each time the heart button is clicked, the change will be saved to the JSON data file by the `saveDataToFile` in the `ShopDataStore` class. This save function will basically write the data in the shop variable into the original JSON file.



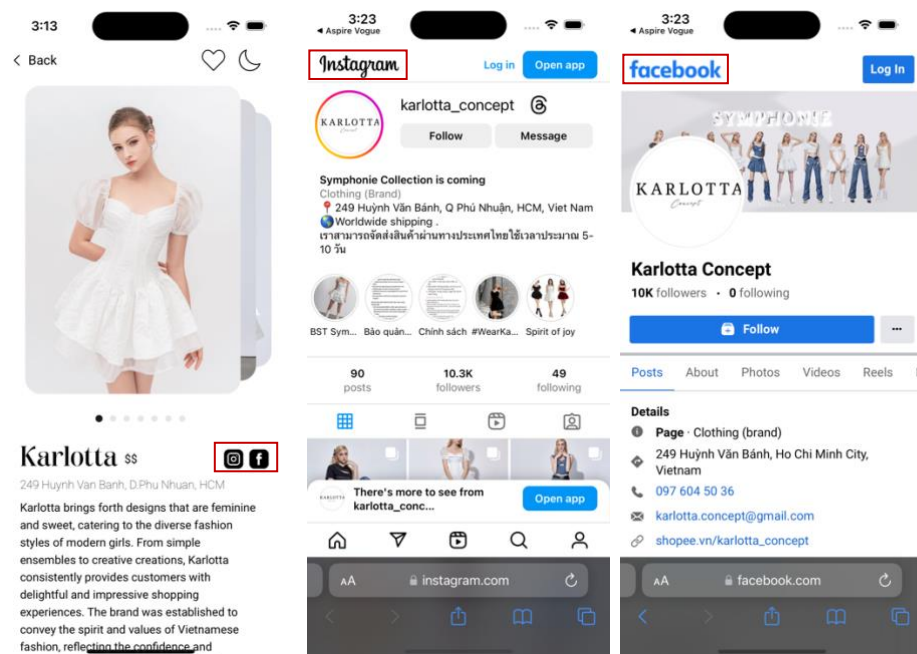
- **Carousel images of each shop (Advanced feature)**

There is a scrolled state variable that is needed to implement this feature. The carousel image is all the shop images inside a `ZStack`, the later image will have an offset to the right and a smaller size compared to the first image. The offset and size will have a ternary operator to change these values. When the image is swiped away, the next image will have the same size and offset as the previous image. The swiping aspect is implemented using the `SwiftUI DragGesture` attribute. Upon swiping, if the user swipes to the left, the image is moved to the left, if the swiping distance after the release of the image exceeds a threshold, the image will be moved outside the view by changing its offset to be outside the screen view, the scrolled value will be added by one which moves the next image to the previous image position. There is also a restriction to prevent the user to swipe left on the last image by taking note of the image ID since all shops have exactly seven images. When the user swipes to the right, the opposite will occur. If the swiping distance exceeds a threshold, the previous image will come back into view and the scrolled value gets reduced by one. There is a restriction to prevent users from swiping to the right at the first image.



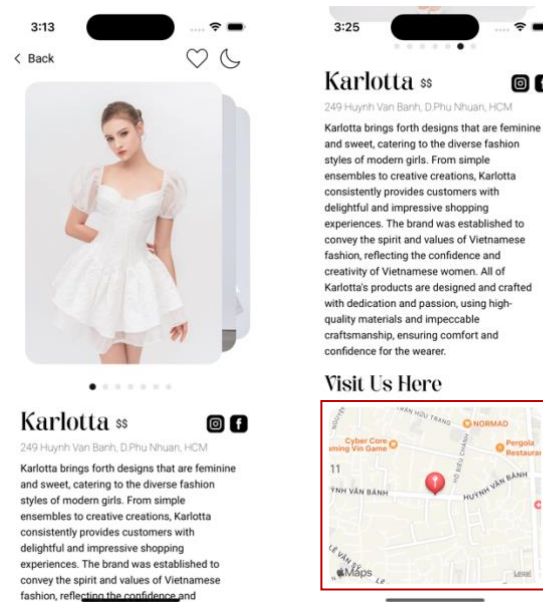
○ **The social icons that link to the social page of the shop (Advanced feature)**

Each social icon inside the detail view is a button that will open the designated URL provided by the shop data using SwiftUI UIApplication.shared.open function.



○ **Map with a pin location of the shop (Advanced feature)**

The map is created with a MapView struct, taking the parameters of the shop coordinate. There is an additional component MapMarker that also takes the shop coordinate to show the exact location marker of that shop.



3. Application Demonstratrion

The Video Link: <https://youtu.be/n7H8c7A6ecU>

4. Known Bugs/Problems

- The first problem lies in the carousel feature. Due to some technical problem, each time the image is swiped, the next image will move to the bottom by a little bit. To tackle this problem, I have to put a dynamic condition inside the padding of the below component, which is the circle indicator, that will scale the top padding based on the scrolled variable.
- The second problem is the segmented picker. Initially, the segmented picker had a built-in swipe gesture feature, so to have the ability to be able to put the segmented picker inside a horizontal scroll view and swipe it I have to have an additional bottom padding to allow space for the user to swipe. However, this issue could confuse the user since some might not understand how to swipe the scroll view since if you swipe inside the picker area, it will not work.
- The last issue is responsiveness. During the development of the code, most of the design is based on the iPhone 14 Pro, so when using the app on different phones, the layout of the app will be displayed differently.

IV. Conclusion

If given the opportunity to further enhance Aspire Vogue, I want to broaden its range of functionalities. One area that requires improvement is the app's responsiveness, specifically in terms of effectively rendering the initial design across different devices. One potential improvement for the application could involve implementing view-based padding instead of relying on fixed padding, as currently used in the initial version of the app. Furthermore, I am interested in augmenting the existing database for the application. At present, the majority of the places featured in the application are limited to the region of Ho Chi Minh City. To improve the broadness of the listings, it is desirable to broaden the scope of included establishments to encompass a broader range of locations across the nation. Finally, in light of the app's objective to enhance women's empowerment through their selection of attire, I aim to incorporate social functionalities into the application. One potential example could be a platform in which users can share their approaches to styling the items showcased within the application. In conclusion, I hope that Aspire Vogue could act as a place where women could find the resources to empower themselves through clothing.

V. Reference

- Keogan K (2013) 'The Relationship Between Clothing Preference, Self-Concepts and Self-Esteem', DBS Institutional Repository, accessed 4 August 2023. https://esource.dbs.ie/bitstream/handle/10788/1620/ba_keogan_k_2013.pdf?sequence=1.
- Braizaz M (2018), 'Femininity and Fashion: How Women Experience Gender Role Through their Dressing Practices', CERLIS, doi:[10.4000/cadernosaa.2001](https://doi.org/10.4000/cadernosaa.2001)