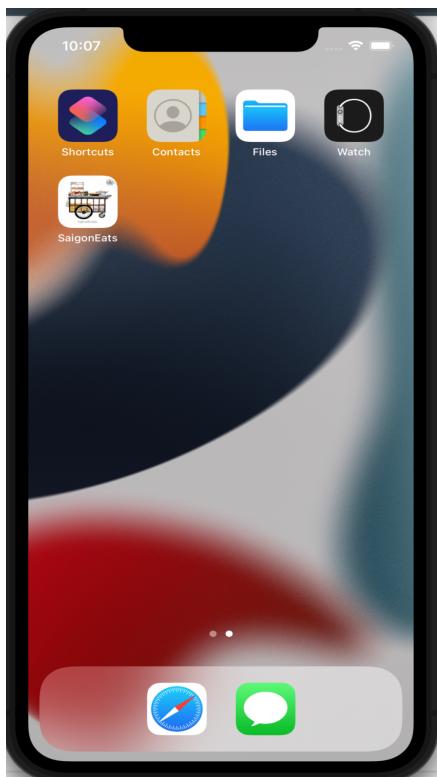


My app is called SaigonEats. It is an app that contains a list of eating spots that I think are the best Saigon can offer for anyone. I am a foodie so I have been to a good amount of restaurants. Therefore, I created an app that contains a list of my favorite eating spot in Saigon. The main feature of this application is only a navigation list that shows the users my favorite eating spots. They can scroll through the list in the navigation view and choose the spots that they would like to see my review. Then, they will be taken to a view have the basic information, such as the address, phone number, and virtual map location, of the place. If the users wish to return to the list, they are provided a back button, “The Holy Grails” on the top left to return to the list, where they can continue to navigate through the list and choose other spots. Here is how you are going to use the app:

Here is my icon for the app:



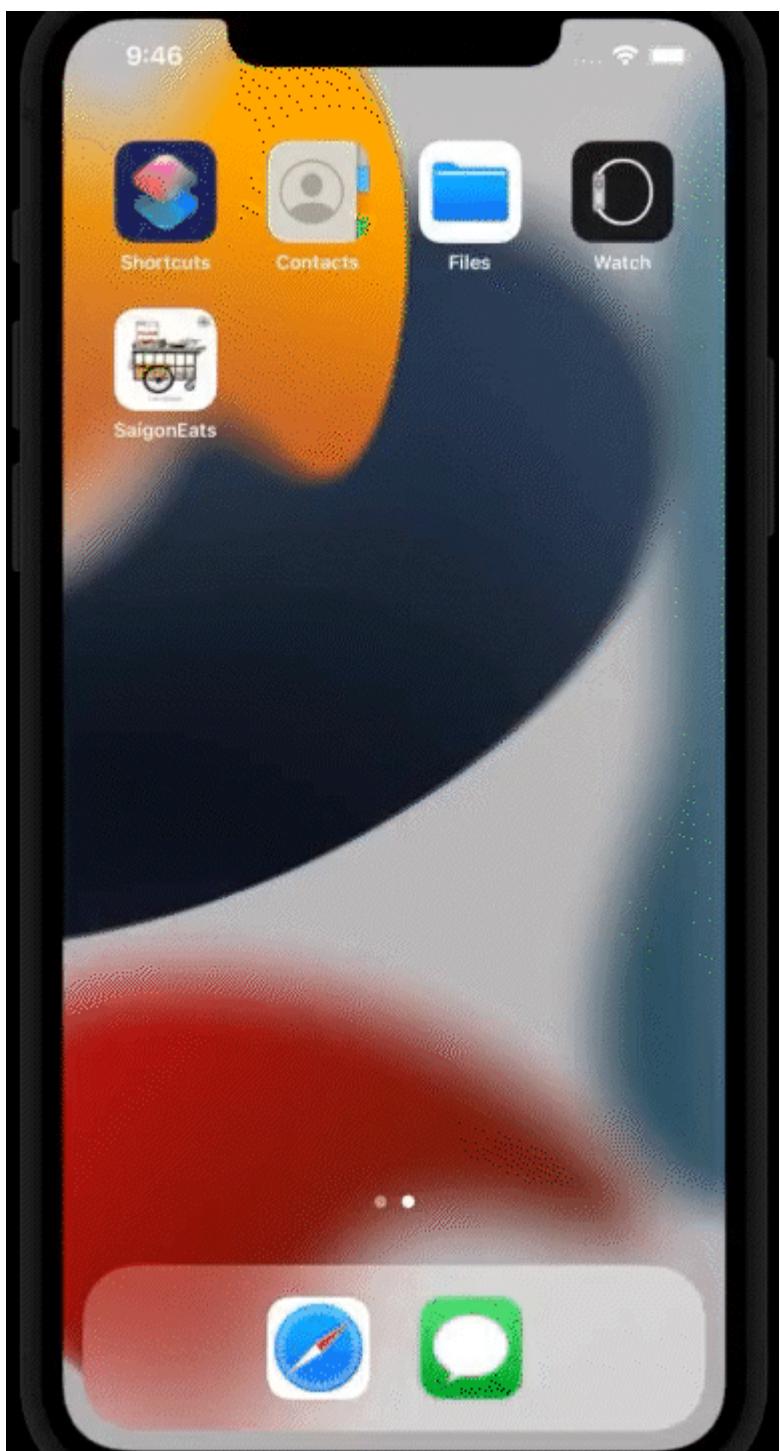
How the app icon look on an iPhone:



Here are the features of the app:

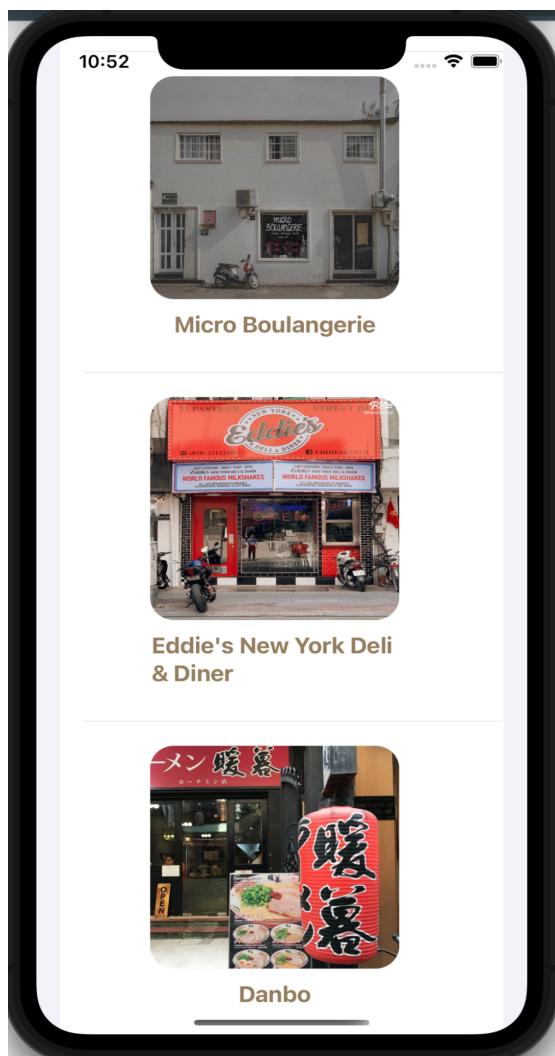
1. Greeting View





When first accessing the app, the user would be taken to the greeting view/ homepage, which briefly introduces the purpose of the app- showing my favorite food spots in Saigon. If the users are interested, they can continue to tap the “Let’s Dive In” button to access the home page/ the spot list. There are 3 views that are created to be able to create this sequence of actions. They are the OpeningView, HomePage, and the ContentView. The ContentView is to activate the OpeningView. In the OpeningView, I declare @State with the variable isOpeningActive and set it as true. Then, in the ZStack, I set the condition that the home page would be shown as default, with the isOpeningActive variable bound to an “active” element. Then, in the HomePage, I set the button “Let’s Dive In” as the catalyst, when tapped, the active state would become false and the spot list would appear.

2. Spot List

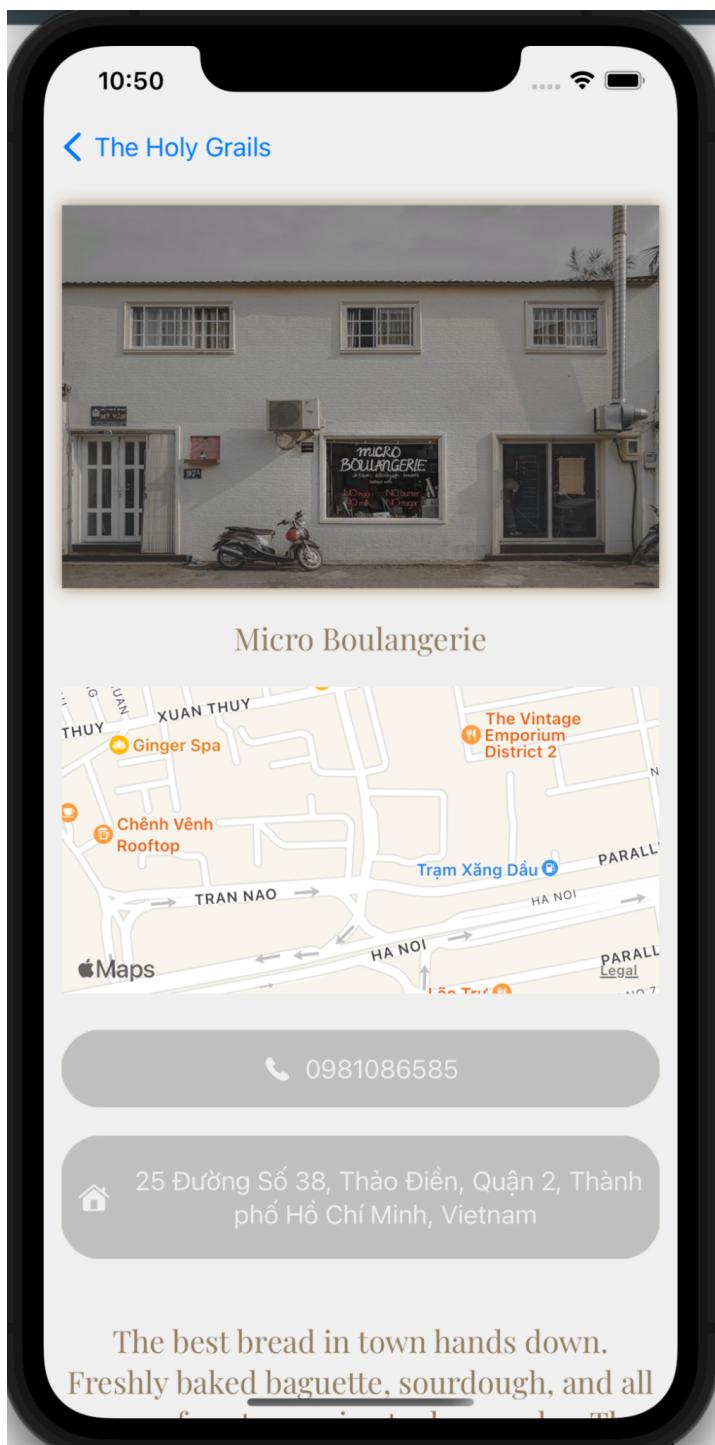


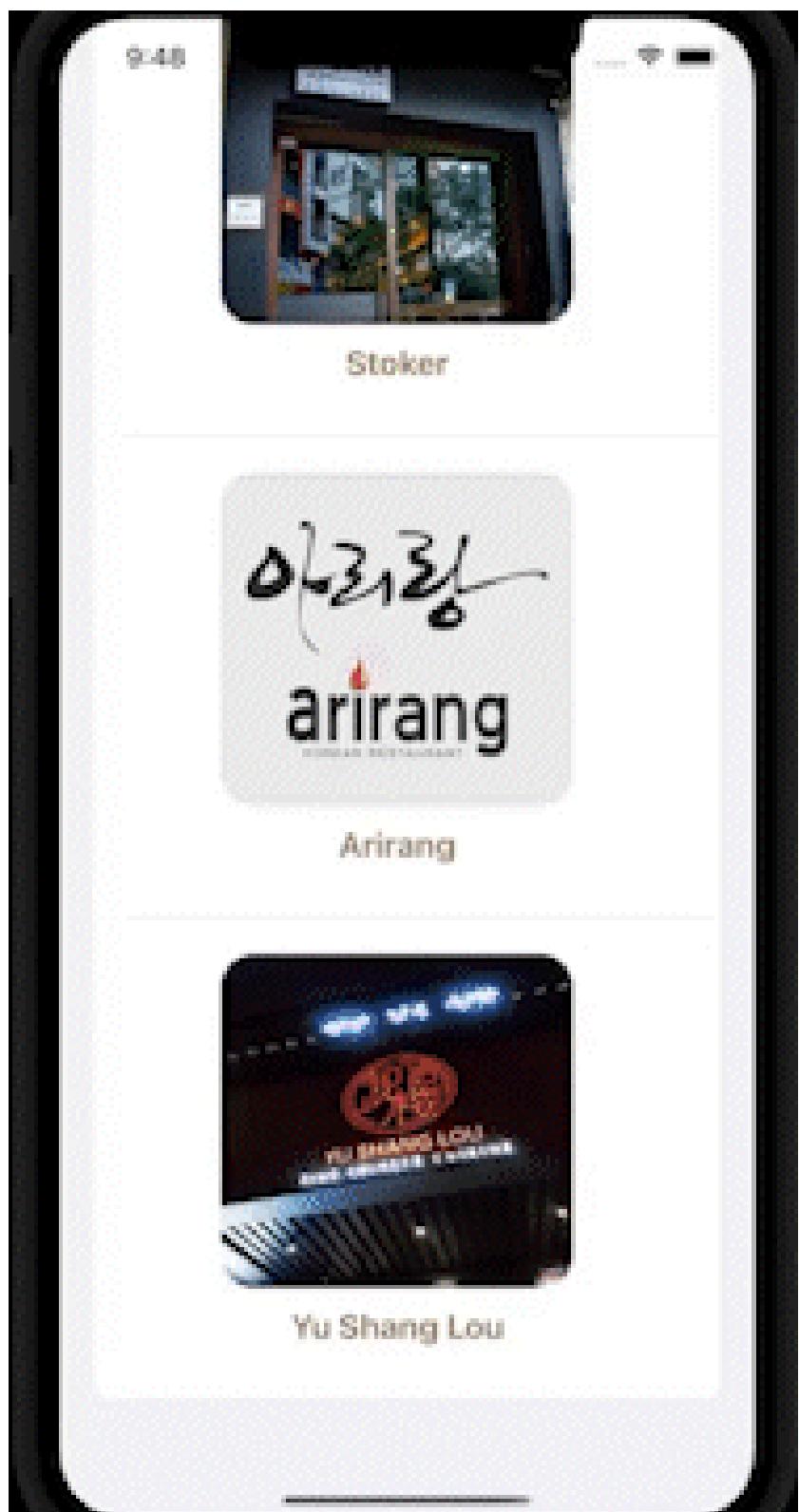


When the “Let’s Dive In” button is tapped, the users will be directed to the spot list, where all of my favorite spots are included. They can scroll up and down to navigate through the list. To achieve this, I combine integrate SpotList with SpotRow. SpotRow is the individual row and SpotList is the compounded SpotRow. The SpotRow is a VStack of “spot.image” and “spot.name”. The attributes of the spots are declared in the Spot and the specifics are stored in the spots.json file. Then in the SpotList, I create a NavigationView that contains the list of the spots, where the navigation link is the SpotCard, which contains the information about the

spots, and the label is the SpotRow. So when the users choose a spot, they will be taken to the SpotCard of each spot accordingly.

3. Spot Card





In the Spot Card view, there are the name, telephone number, address, and the map location of the place. To create this, I have to first create MapView, which contains identified

coordinates of the spots, which I have manually copied down into the spots.json file. With the help of MapKit, the system would automatically identify the coordinates and present them in the map. Then in the SpotCard view, I created a ScrollView of a VStack, which contains the information about the spots, and a brief review for each location at the bottom.

There are no creative features I was able to implement due to my lack of experience. However, later on, I would like to implement a CRUD function for myself so that I can create and update the spot list more efficiently, instead of having to manually edit the json file. As for the users, I would like to be able to have a sorting function for the spots based on the district that the spots reside. This way, the users can smoothly find restaurants that are within their desired destination. Also, the information included is static so the users can't tell the exact location of the spots by looking at the map interface provided and they can call the place directly with the telephone number given. I would like to later have that option available for the users. A forum page would be neat as well so that the users can sign in comment and leave reviews on the spots I have posted, and even have their own forum to post their own favorite spots.