RMIT UNIVERSITY

COSC2659: IOS DEVELOPMENT

ASSIGNMENT 1

TRAN HOANG VU – S3915185

Table of Contents

Table of Figures:

I. INTRODUCTION

The app I am introducing here is about giving information about luxury cars and their brands. It gives us a better understanding of luxury cars that are famous for their global brands; they are old my favorites thing to talk about, at least I have a material goal to look forward to afford one of these by myself. We can have a look at their overview, contact information, history, additional information, and including their models like a car notebook.

The purpose of developing an iOS application for a mobile phone about cars is quite vague, because of the purpose of this is not to write my own car notebook that is on mobile phone, but is to remember that this simple notebook will be implemented everywhere and can be developed on any operating system, not just on iOS. You would ask me "why luxury cars and why these brands?". Basically, I got information shared by coworkers currently working in automobile vehicles, and told me that the company are implementing not just a simple model software, but a operating system, which means that everything that a mobile phone can do, a future car would be able to do so. Then I was inspired by that, not just because these cars are eye-catching, but also they would be initial brands that would release any operating systems on their vehicles, that we can call video, chat, view map, … basically, like a normal mobile phone ON a car. One day we can code a program that can view on any of these operating systems.

My motivation for building this app is not only to develop the program on iOS, but also to enhance my thinking skills to solve problems better when upcoming challenges ahead are waiting for me, so that in every state, I would be able to learn quickly and adapt information wisely.

II. PROJECT DESCRIPTION

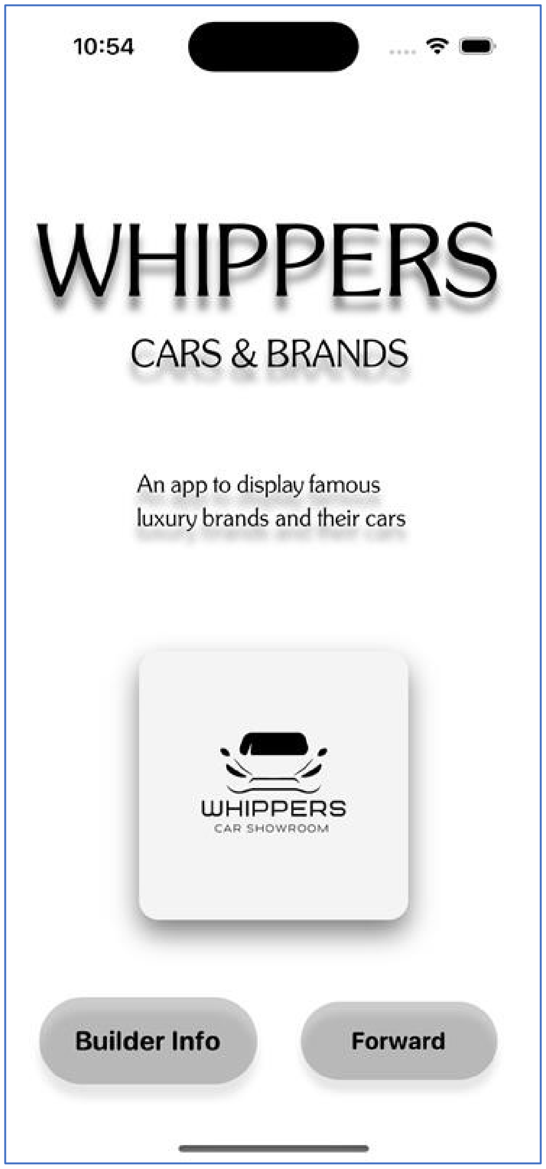
The app I've developed is about giving users a brief understanding of each luxury car and brand. It contains basically cars that are usually famous for their brands and that are unique enough to catch people attention.

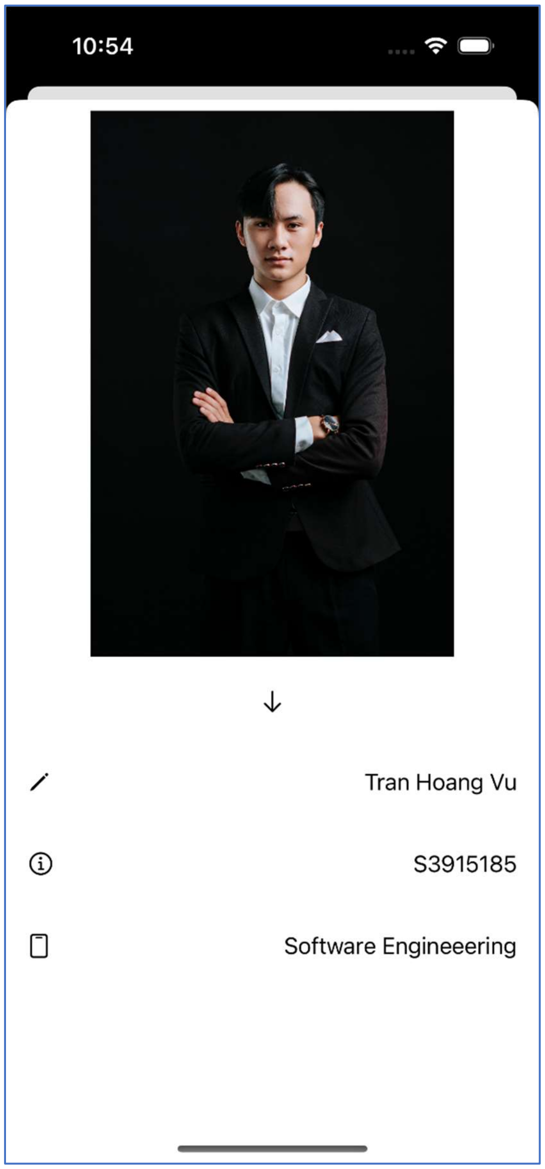
By first running the program, user can have a look at a welcome view, and they can either look at my information, or get started to the notebook. In my information, the user can view my information including my image, name, sID, and my learning career. In the notebook, users can get to pick a brand that is available, including search and sorting information, and also able to select dark and light mode. After selecting a button that represents the brand, the app will direct you to that brand card. In there, you can look at: overview, location of their headquarters on a map, their available contact information, a link to their website, additional information, and their car catalog including transparent pictures of the car from back view. Users can also search for car, and sort them in ascending or descending of letters. If users click to a car in a catalog, the app will direct to the card for the car information, then, it will display information of the car's logo (if it exists), the name, the overview, and the image of the car.

I have learned a lot coding work for developing an app on iOS devices, and was able to get huge amounts of knowledge and ideas that I can provide for my software development career in the future. I got to work with views, animations, binding variables with states of them; I've learned about how should we develop the program so that it can have great UX/UI, and able to implement MapView, or even environment and decoding json files.

III. IMPLEMENTATION DETAILS

1. Welcome Screen and My Information





Tran Hoang Vu

S3915185

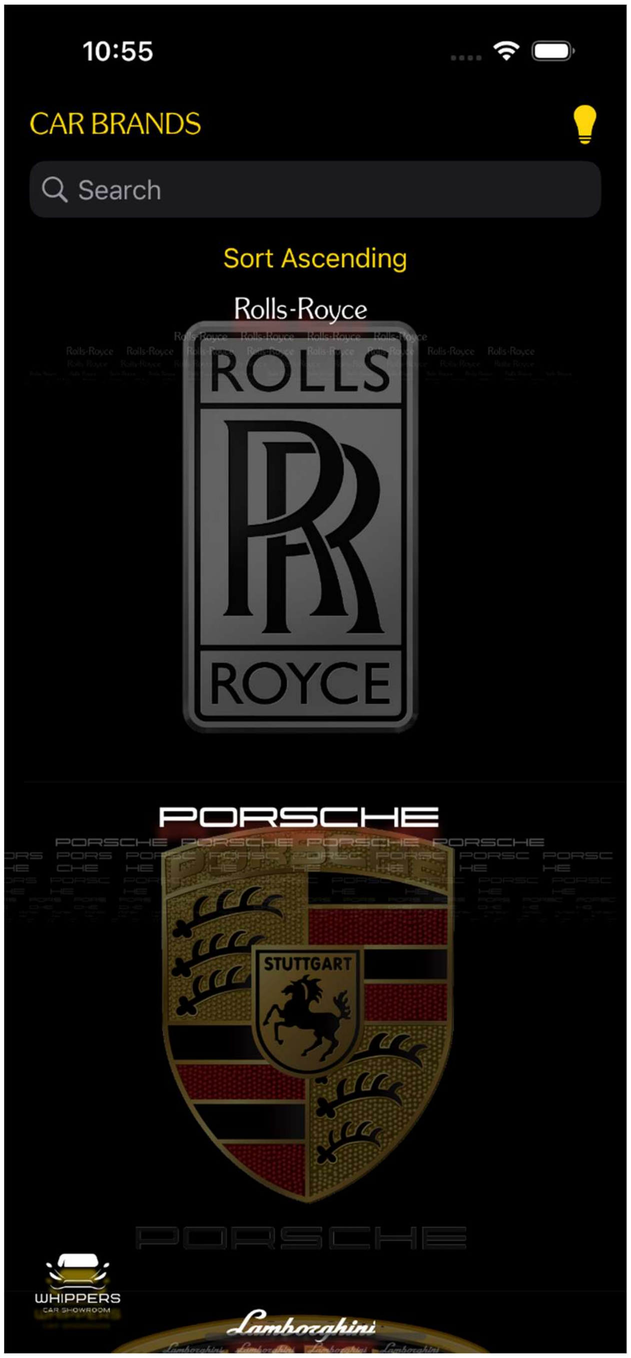
Software Engineeering

Figure 1: Welcome View and Builder Info

By first clicking on the app, the welcome page will be first directed to users to view. This includes the name of the app (including the type of information that the app is talking about, is "Car & Brand"). Also there is a image of the app icon, including with the buttons that link to my information and to the main app. The welcome screen also has some animations of texts start to appear from left to right, with the logo of the app ease-in-out with repetitiveForever() that creates an interesting first look at the program. Here is the code that setState() for the animating part in Appendix 1. By changing the states of these values with the TimeInterval, implementing the function below gives us the amazing texts animation in Appendix 2. For displaying my information on a sheetView that pops up, I set states and would be toggled when button clicked, then it would pop up the screen on the Figure … above. Here is how the code works, and it's just a state that the sheet would presented if the state is true in Appendix 3. Here is the short video of the feature's above, and I don't know why the resolution is bad like this:

Figure 2: Welcome View Animation GIF

2. Main Page – List of Brands



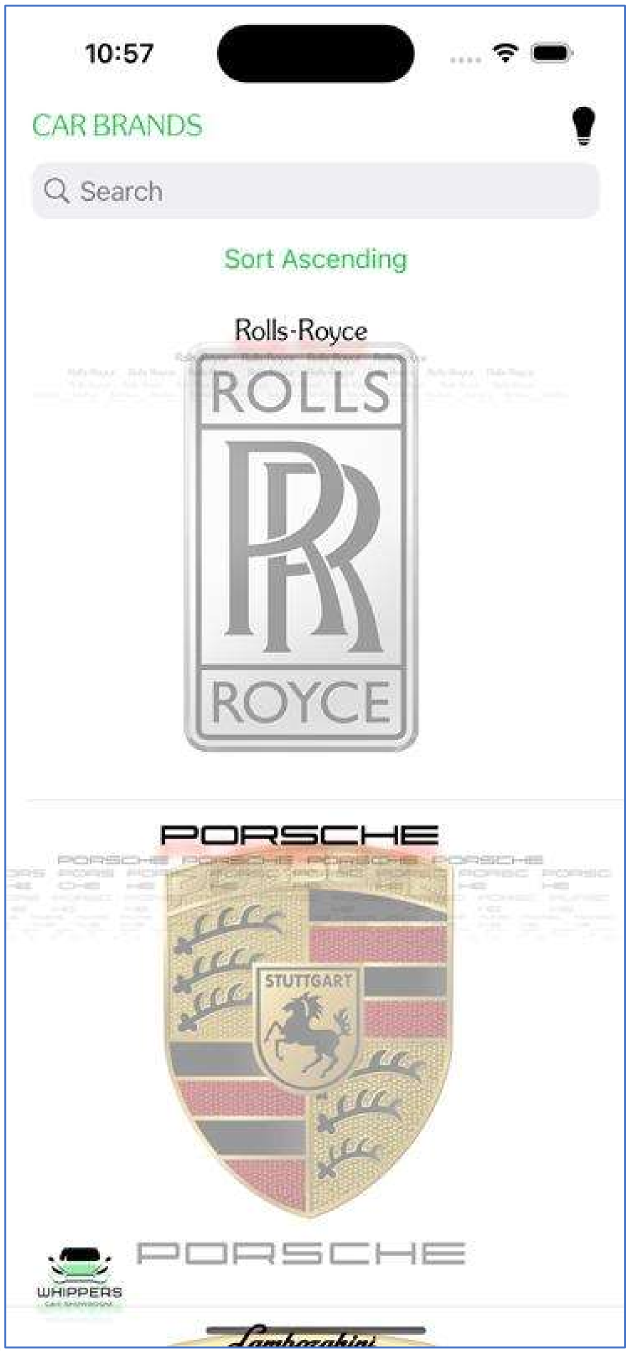


Figure 3: Main Brand List Dark & Light Mode

These illustrations are for dark mode and light mode of the main page whenever the user is directed to the main page. Every brand that is available will exist on the NavigationView in the List(), and once you click it, it will direct you to the brand card. The navigationLink is decorated with the BrandRow.swift that has generated multiple texts that is using the Brand Logo's font for better visualization, including with their transparent logo at the back. The text are also change with the environment of the app, including the logo at the bottom left, which is also changing its shadow based on the mode. Colors are picked based on the understanding of choosing contract colors between black and white for visual appealing. There is also a search bar and sort ascending or descending whenever users want to search for a brand. Here is a gif that demonstrate the information above:

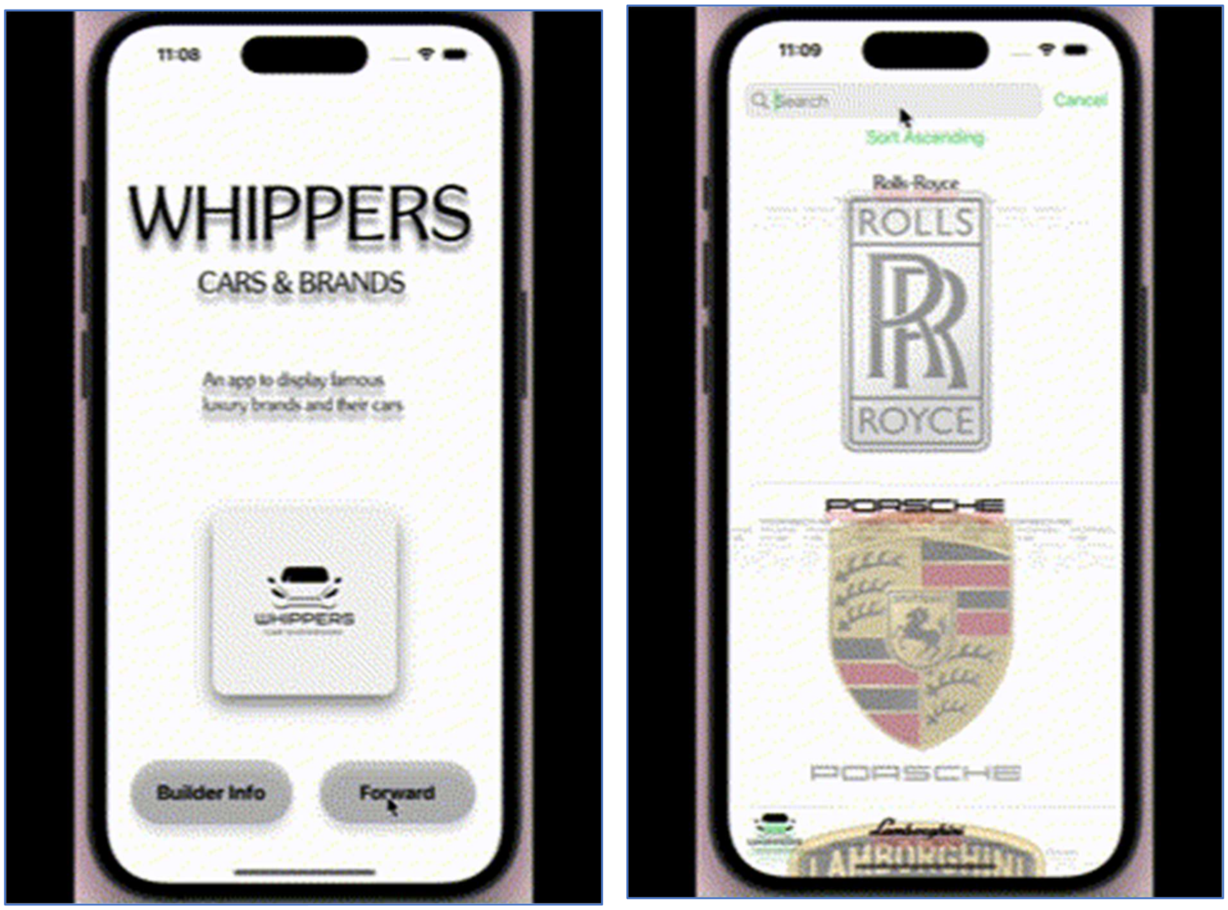


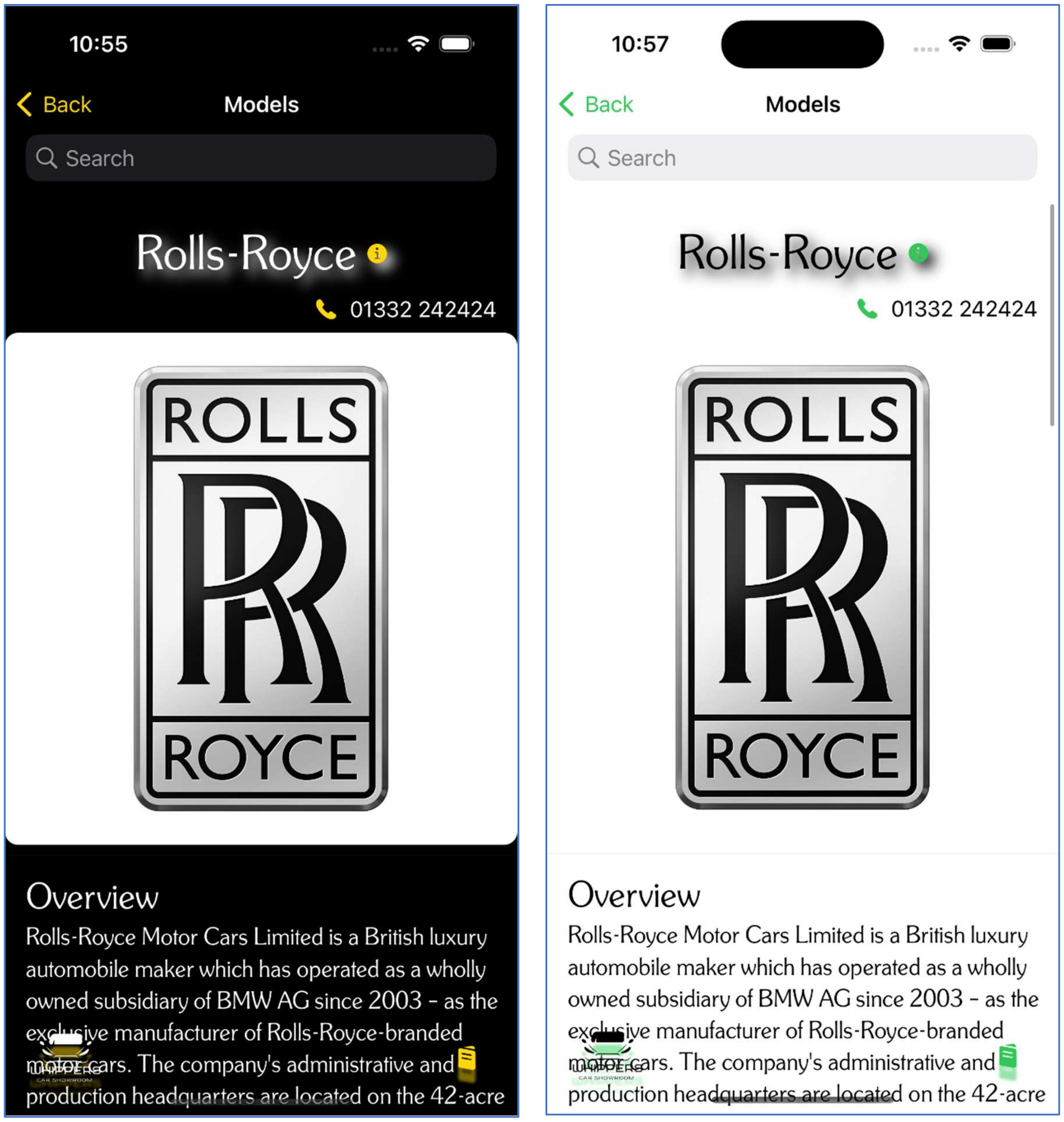
Figure 4: Main Brand List from Welcome View Animate GIF

This is the short version code that helps implementing such visualization and features above in Appendix 4. Because the search bar is already built in, I just had to add .searchable(text: $searchText) with setting states in the beginning to make this work, and whenever the search bar's value changes, the brands list will be modified by using filters($0.localizedCaseInsensitiveContains()), and sorted() with lhs < rhs for ascending and lhs > rhs for descending in Appendix 5.

This is logic for the transparent icon at the bottom left of the app, and it would change shadow based on the environment is in which mode in Appendix 6.

This is how the generated text that are fading and getting smaller in the BrandRow.swift. This would work better if I can generate an image for it, so the visualization can be much smoother in Appendix 7.

3. Brand Card – Brand Information



Overview
Rolls-Royce Motor Cars Limited is a British luxury automobile maker which has operated as a wholly owned subsidiary of BMW AG since 2003 – as the exclusive manufacturer of Rolls-Royce-branded motor cars. The company's administrative and production headquarters are located on the 42-acre

10:57
Back
Models
Search
Rolls-Royce
01332 242424
Overview
Rolls-Royce Motor Cars Limited is a British luxury automobile maker which has operated as a wholly owned subsidiary of BMW AG since 2003 – as the exclusive manufacturer of Rolls-Royce-branded motor cars. The company's administrative and production headquarters are located on the 42-acre

Figure 5: Brand Cart in Dark & Light Mode

After clicking to a brand Row that links to a Brand Card, users will then get directed to this place. Users will get access to the information of the brand's overview, real website (at the small i button next to the brand's name) in Appendix 8, the location, their additional information like history, contact information, or something else, and last but not least, their famous car catalog. Under this is the interaction when users click to the small i button in Appendix 9:

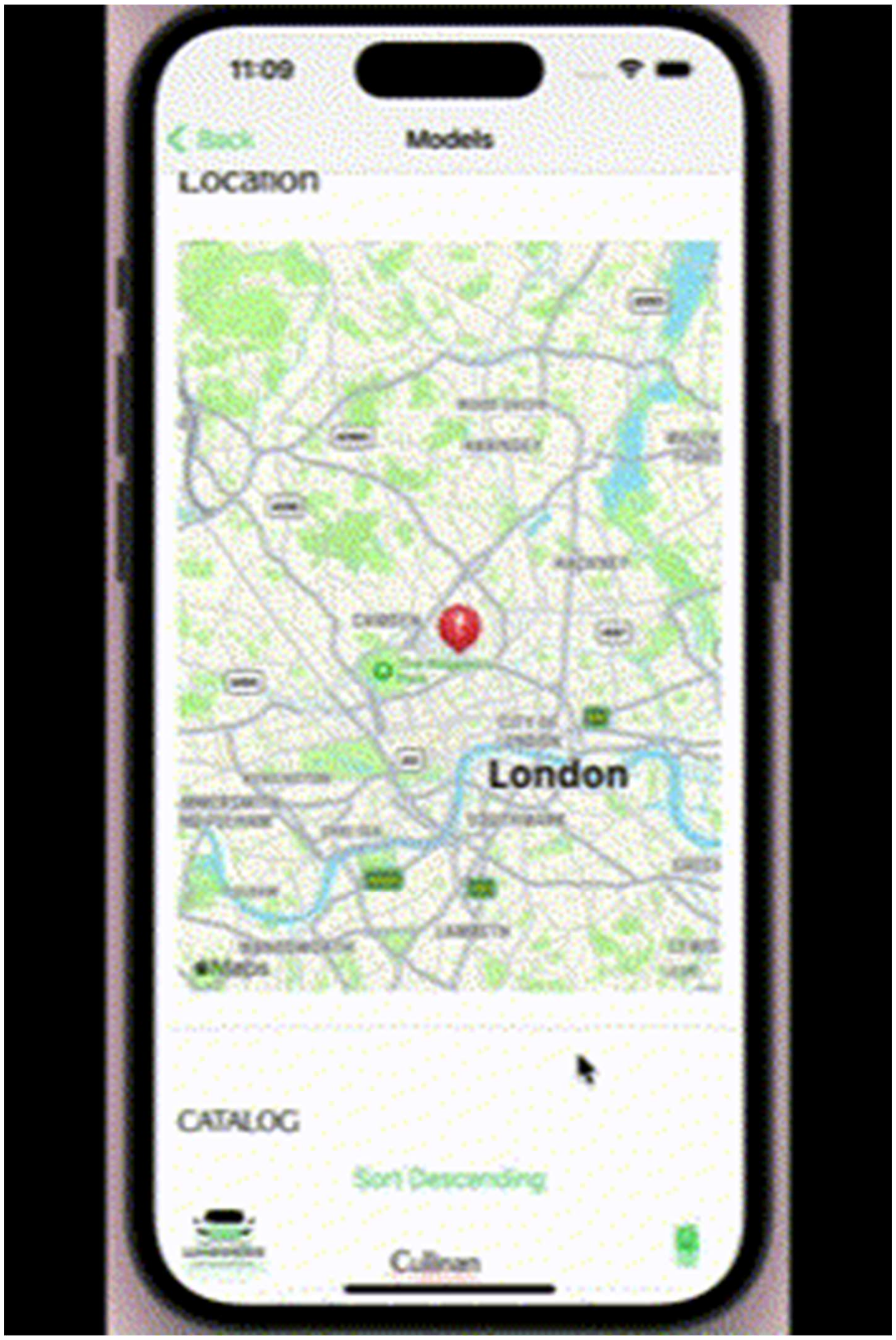
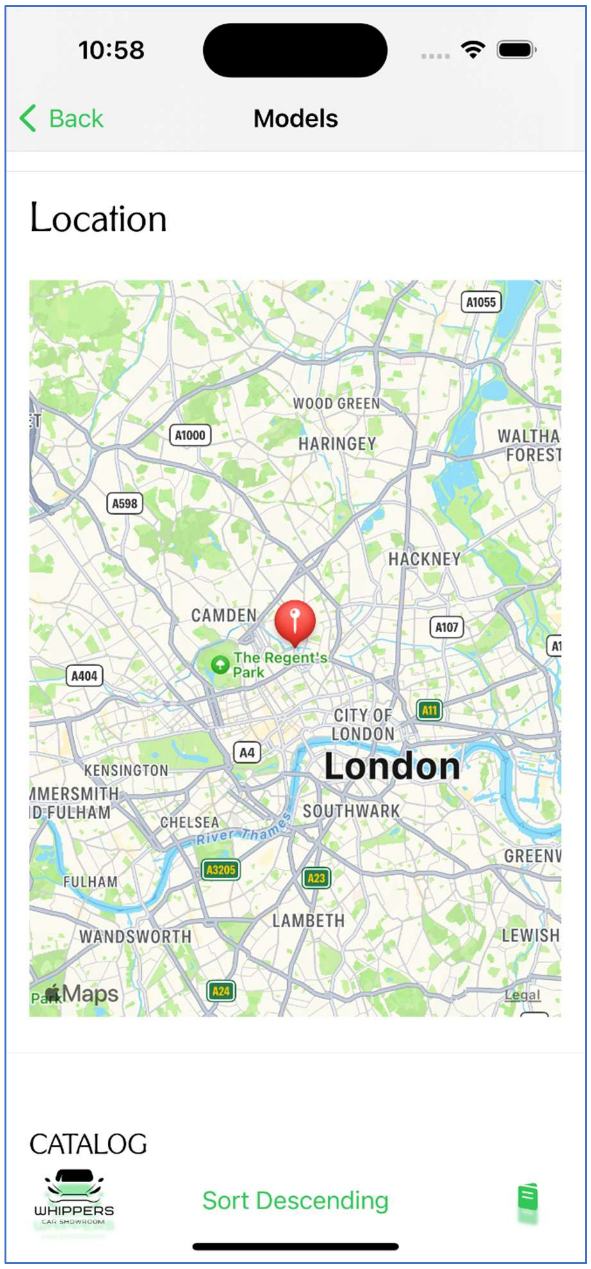


Figure 6: Click Logo links to Website Animate GIF

This is the theme for light mode:



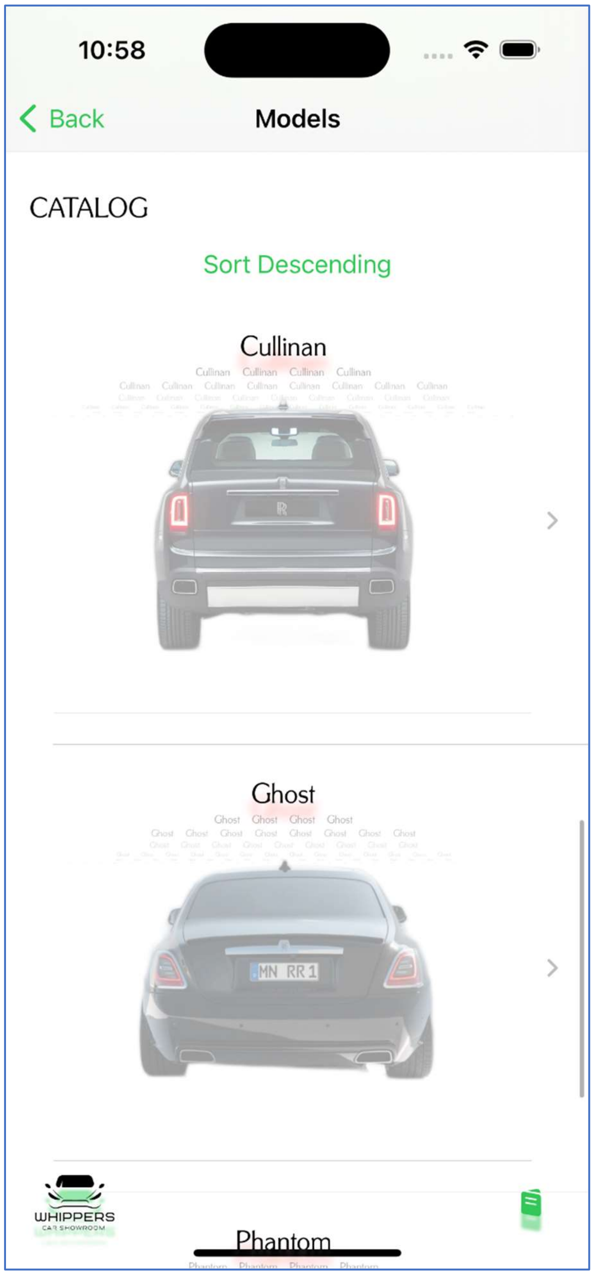
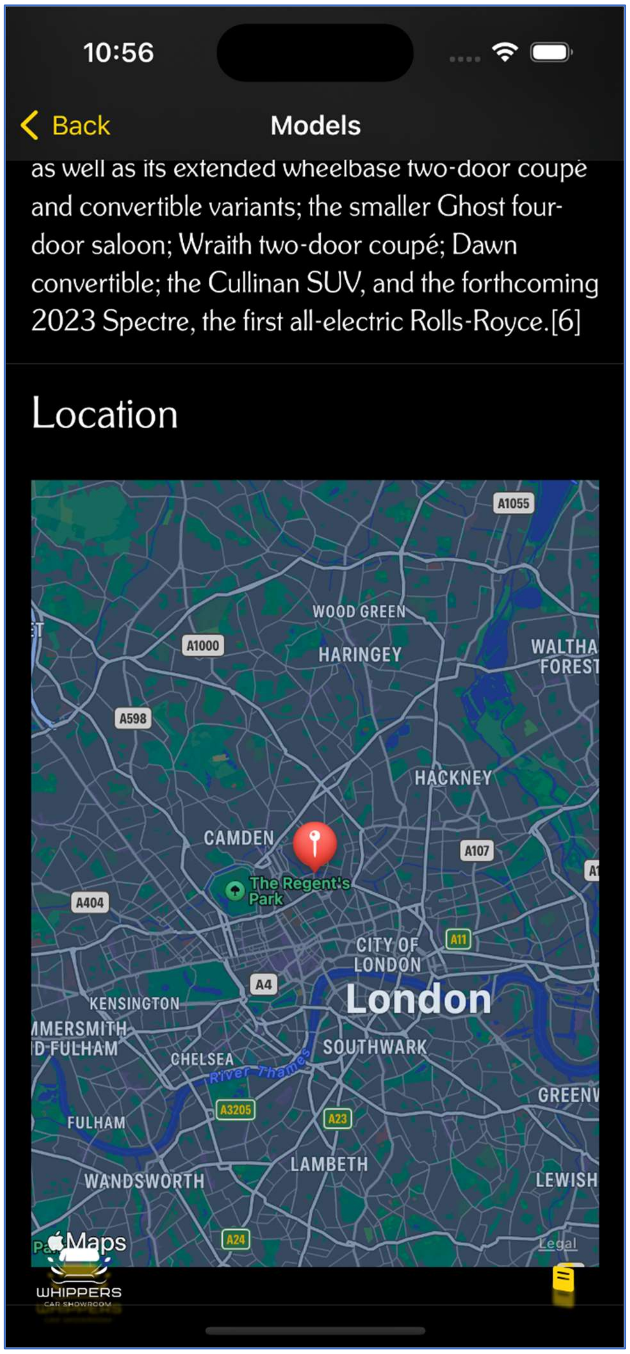


Figure 7: Map and Car Catalog in Light Mode

Basically the user can interact with the map with location pinned on the map, and also scroll the car catalog to pick the cars users would like to see. There is also a search bar at the top and the sort button for cars in the Brand Cart. The structure of the brandCard is a normal ScrollView() that has multiple small Vstack() that is embed in a big VStack(), this helps user can scroll every information without getting trouble to not able to view some. I also enhance your mapView() to be able to pin the location on the map in Appendix 10, make some changes that would make the card look fit from head to tail. Also, the accentColor would change according to the color theme of the app. The CarRow are used the same logic to implement such visualization.

Under this is the theme for dark mode:

Figure 8: Location and Car Catalog in Dark Mode

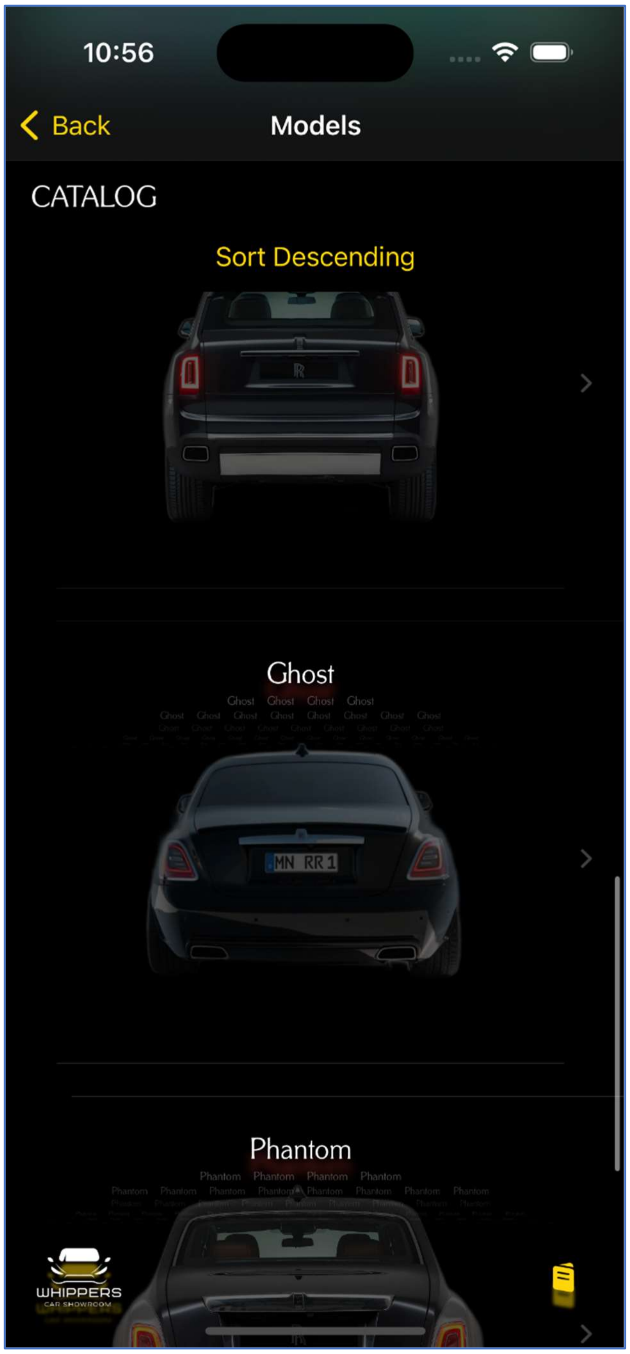


10:56

Back Models

as well as its extended wheelbase two-door coupé and convertible variants; the smaller Ghost four-door saloon; Wraith two-door coupé; Dawn convertible; the Cullinan SUV, and the forthcoming 2023 Spectre, the first all-electric Rolls-Royce.[6]

Location



10:56

Back Models

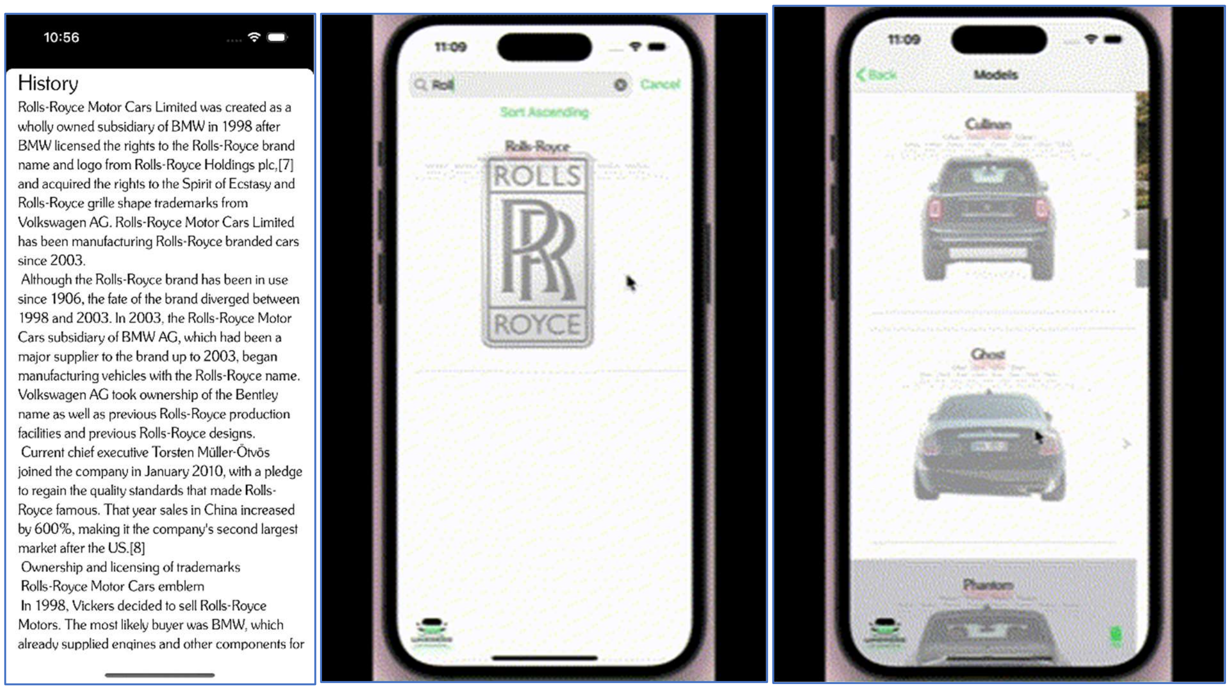
CATALOG

Sort Descending

Ghost

Phantom

At the bottom right corner, there is the notebook icon, and when users click it, a sheet page will appear for users to read additional information like history and some others:



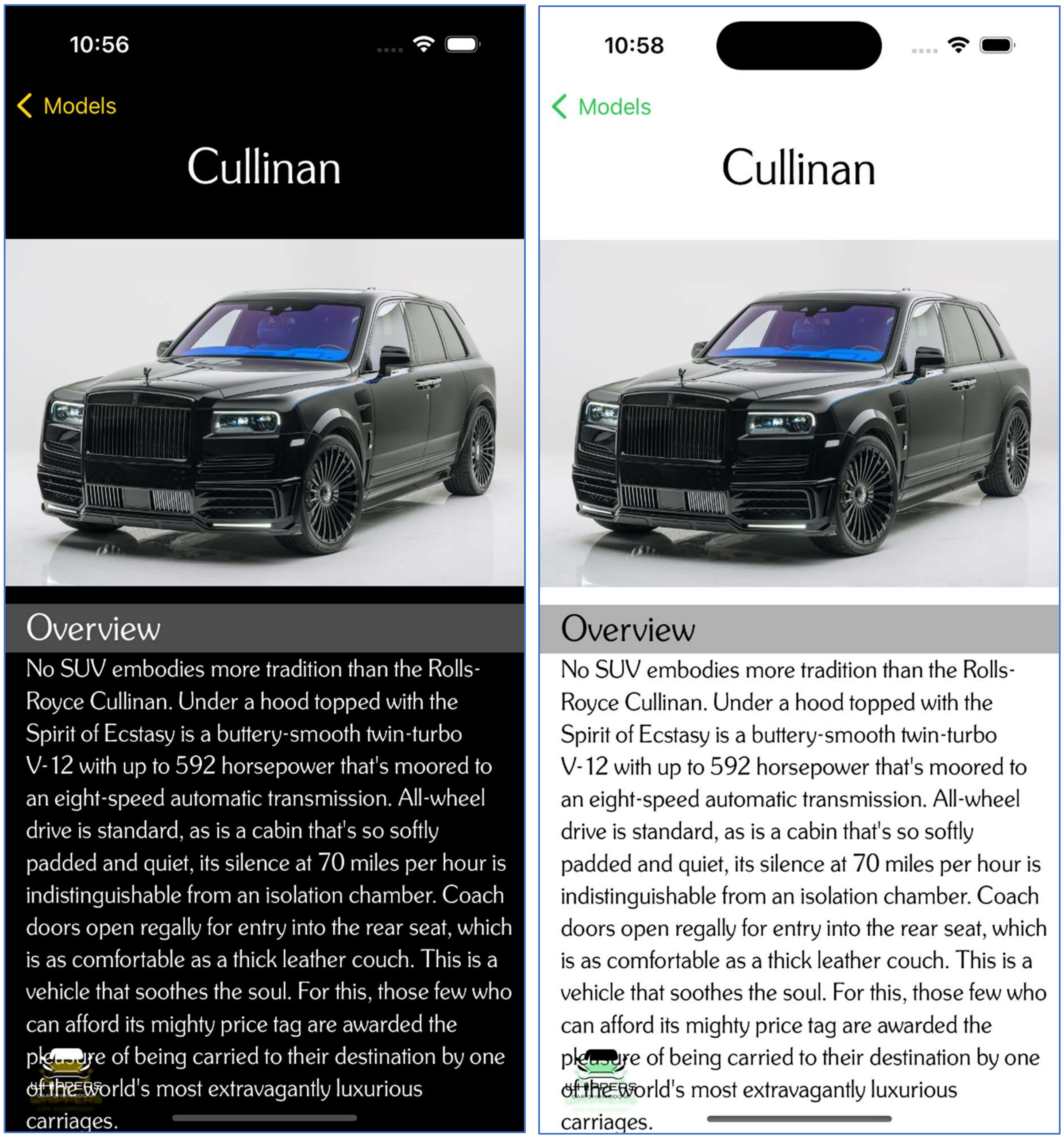
History

Rolls-Royce Motor Cars Limited was created as a wholly owned subsidiary of BMW in 1998 after BMW licensed the rights to the Rolls-Royce brand name and logo from Rolls-Royce Holdings plc,[7] and acquired the rights to the Spirit of Ecstasy and Rolls-Royce grille shape trademarks from Volkswagen AG. Rolls-Royce Motor Cars Limited has been manufacturing Rolls-Royce branded cars since 2003.
Although the Rolls-Royce brand has been in use since 1906, the fate of the brand diverged between 1998 and 2003. In 2003, the Rolls-Royce Motor Cars subsidiary of BMW AG, which had been a major supplier to the brand up to 2003, began manufacturing vehicles with the Rolls-Royce name. Volkswagen AG took ownership of the Bentley name as well as previous Rolls-Royce production facilities and previous Rolls-Royce designs.
Current chief executive Torsten Müller-Ötvös joined the company in January 2010, with a pledge to regain the quality standards that made Rolls-Royce famous. That year sales in China increased by 600%, making it the company's second largest market after the US.[8]
Ownership and licensing of trademarks
Rolls-Royce Motor Cars emblem
In 1998, Vickers decided to sell Rolls-Royce Motors. The most likely buyer was BMW, which already supplied engines and other components for

Figure 9: Additional Info SheetView with Animate GIF

The additionalInfo button works the same as the sheetview at the welcome view, and here is the logic code is still changing states of the sheet.isPresented() to display such information in Appendix 11.

4. Car Card – Car Information:


10:56

Models

Cullinan

Overview

No SUV embodies more tradition than the Rolls-Royce Cullinan. Under a hood topped with the Spirit of Ecstasy is a buttery-smooth twin-turbo V-12 with up to 592 horsepower that's moored to an eight-speed automatic transmission. All-wheel drive is standard, as is a cabin that's so softly padded and quiet, its silence at 70 miles per hour is indistinguishable from an isolation chamber. Coach doors open regally for entry into the rear seat, which is as comfortable as a thick leather couch. This is a vehicle that soothes the soul. For this, those few who can afford its mighty price tag are awarded the pleasure of being carried to their destination by one of the world's most extravagantly luxurious carriages.

10:58

Models

Cullinan

Overview

No SUV embodies more tradition than the Rolls-Royce Cullinan. Under a hood topped with the Spirit of Ecstasy is a buttery-smooth twin-turbo V-12 with up to 592 horsepower that's moored to an eight-speed automatic transmission. All-wheel drive is standard, as is a cabin that's so softly padded and quiet, its silence at 70 miles per hour is indistinguishable from an isolation chamber. Coach doors open regally for entry into the rear seat, which is as comfortable as a thick leather couch. This is a vehicle that soothes the soul. For this, those few who can afford its mighty price tag are awarded the pleasure of being carried to their destination by one of the world's most extravagantly luxurious carriages.

Figure 10: Car Card - Car Information in Dark & Light Mode

After selecting a car, users will then get directed to the CarCard.swift, and users can see the car's overview, including with its image. This is builted with a bigVstack() that is embed in a scrollView(), and there is functions to check whether to display the car name or the car logo, if the model does not have a logo, then display a normal text like images above. The code can be found in Appendix 11.

There are bugs that I've not solved:

1. The light & dark mode can't set individually in any pages, so I removed and only leave it in main.
2. The bugs that the lightbulb keep generating many forms kind of stuck, and I can't still figure it out yet. For example, after we went back from clicking the link to the brand's website and go back, there is one more light bulb that exists.
3. Bugs of the scrollView for car catalog, I was not able to let it full display, but only a portion and you can only see it when you scroll it.

IV. CONCLUSION

There are a lot of criteria for me to considered, and when the app can be further developed, I would:

1. Implement fetch data algorithm to better categorize many more information faster and better for a notebook, so that the notebook would work as a handy notebook.
2. Adding which dealers at what locations are available for that car to be bought or test-drive, so that users can get great contact information for their needs.
3. Adding theme for each brand, as because on their own website, they all developed their website in a very good-looking way to attract visitors, so that I would try to do them also. Like the videos displaying or sound of the cars in N mode.
4. Adding more images for the car card, but because I made all of the data by hand, and there are a lot of cars, I would not be able to collect all of the images, so I would convert them all to links.
5. Adding more functionality to set dark mode, login, log out for each user to record their notes about this; they also all can comment on the cars and other users can look at it.

V. REFERENCES

[1] "Automobili Lamborghini - Official Website," Lamborghini.com, https://www.lamborghini.com/en-en (accessed Aug. 7, 2023).

[2] TomHuynhSG, "TOMHUYNHSG/SSET-Contact-List-IOS: RMIT SSET Contact list IOS app!," GitHub, https://github.com/TomHuynhSG/SSET-Contact-List-iOS (accessed Aug. 7, 2023).

[3] "Home," Porsche Asia Pacific - Dr. Ing. h.c. F. Porsche AG, https://www.porsche.com/pap/_vietnam_/ (accessed Aug. 7, 2023).

[4] "Bentley Motors website Home Page," Official Bentley Motors website, https://www.bentleymotors.com/en.html (accessed Aug. 7, 2023).

[5] "Bugatti Automobiles," Bugatti Homepage, https://www.bugatti.com/ (accessed Aug. 7, 2023).

[6] “Home,” Aston Martin | Iconic Luxury British Sports Cars | Aston Martin, https://www.astonmartin.com/en/ (accessed Aug. 7, 2023).

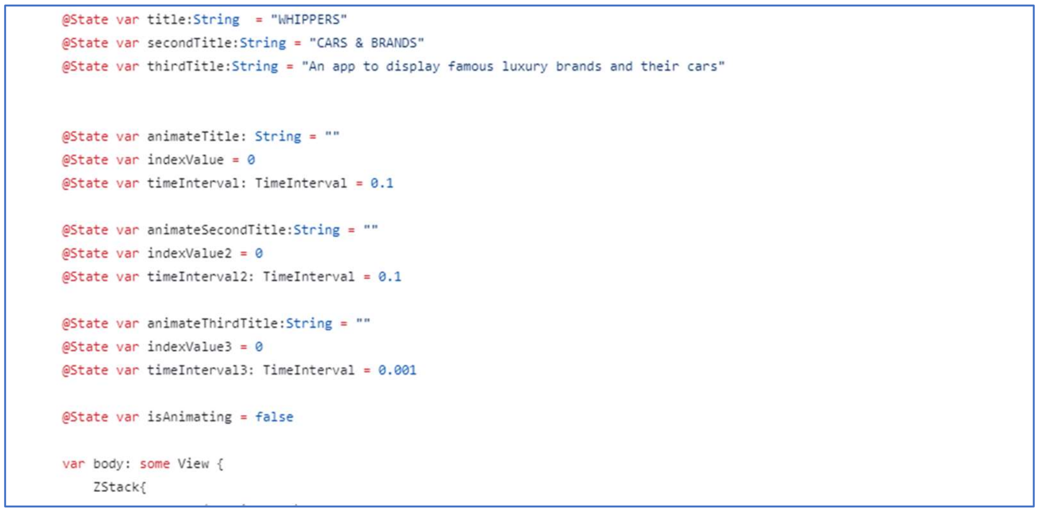
[7] “Jaguar sedans, suvs and sports cars - official site: Jaguar USA,” landrover, https://www.jaguarusa.com/index.html (accessed Aug. 7, 2023).

[8] Official Ferrari website, https://www.ferrari.com/en-EN (accessed Aug. 7, 2023).

[9] “Royce Motor Cars: Inspiring greatness,” Rolls, https://www.rolls-roycemotorcars.com/en_GB/home.html (accessed Aug. 7, 2023).

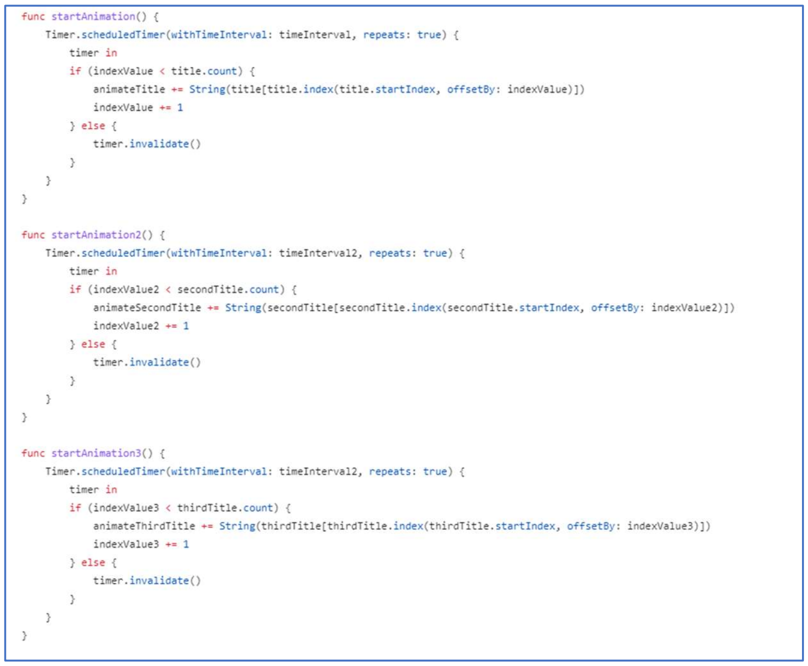
VI. APPENDIX

1.



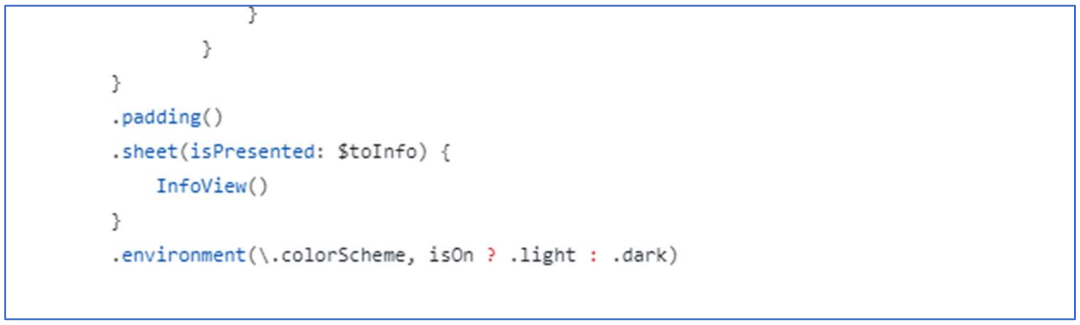
```
    @State var title:String  = "WHIPPERS"
    @State var secondTitle:String = "CARS & BRANDS"
    @State var thirdTitle:String = "An app to display famous luxury brands and their cars"


    @State var animateTitle: String = ""
    @State var indexValue = 0
    @State var timeInterval: TimeInterval = 0.1

    @State var animateSecondTitle:String = ""
    @State var indexValue2 = 0
    @State var timeInterval2: TimeInterval = 0.1

    @State var animateThirdTitle:String = ""
    @State var indexValue3 = 0
    @State var timeInterval3: TimeInterval = 0.001

    @State var isAnimating = false

    var body: some View {
        ZStack{
```

Figure 11: Coding for States() for animation

2. 

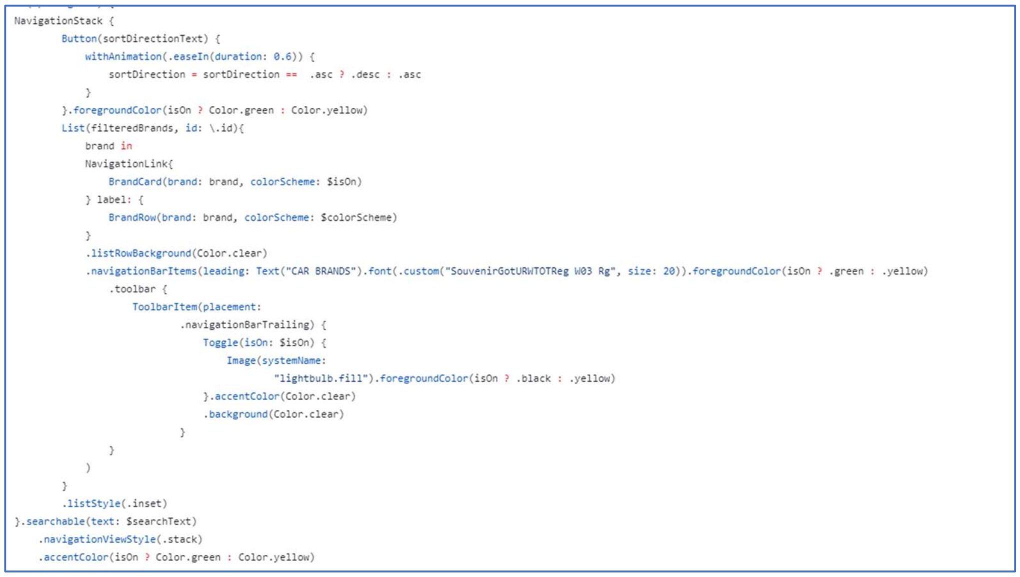
```
func startAnimation() {
    Timer.scheduledTimer(withTimeInterval: timeInterval, repeats: true) {
        timer in
        if (indexValue < title.count) {
            animateTitle += String(title[title.index(title.startIndex, offsetBy: indexValue)])
            indexValue += 1
        } else {
            timer.invalidate()
        }
    }
}

func startAnimation2() {
    Timer.scheduledTimer(withTimeInterval: timeInterval2, repeats: true) {
        timer in
        if (indexValue2 < secondTitle.count) {
            animateSecondTitle += String(secondTitle[secondTitle.index(secondTitle.startIndex, offsetBy: indexValue2)])
            indexValue2 += 1
        } else {
            timer.invalidate()
        }
    }
}

func startAnimation3() {
    Timer.scheduledTimer(withTimeInterval: timeInterval2, repeats: true) {
        timer in
        if (indexValue3 < thirdTitle.count) {
            animateThirdTitle += String(thirdTitle[thirdTitle.index(thirdTitle.startIndex, offsetBy: indexValue3)])
            indexValue3 += 1
        } else {
            timer.invalidate()
        }
    }
}
```

Figure 12: Functions to call for animation in WelcomView

3. 

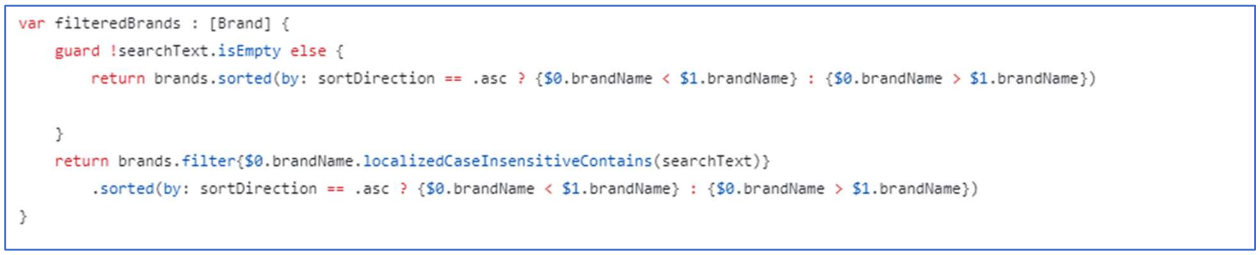
```
                    }
                }
            }
            .padding()
            .sheet(isPresented: $toInfo) {
                InfoView()
            }
            .environment(\.colorScheme, isOn ? .light : .dark)
```

Figure 13: SheetView.isPresented() for WelcomeView to Display Builder Info

4. 

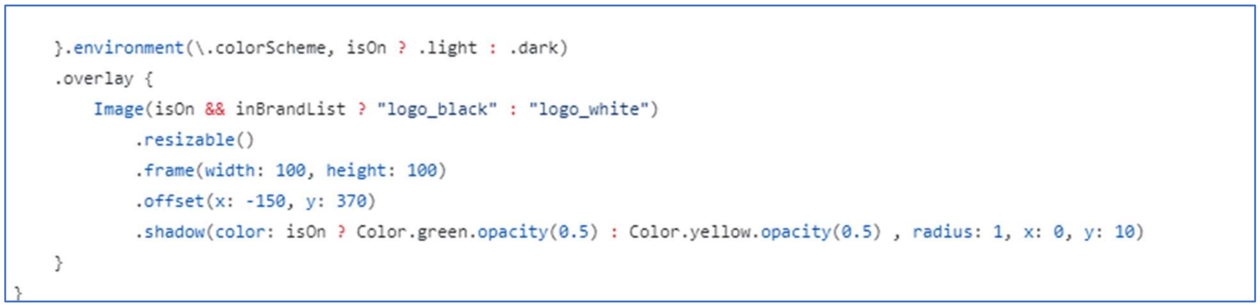
```
NavigationStack {
        Button(sortDirectionText) {
            withAnimation(.easeIn(duration: 0.6)) {
                sortDirection = sortDirection ==  .asc ? .desc : .asc
            }
        }.foregroundColor(isOn ? Color.green : Color.yellow)
        List(filteredBrands, id: \.id){
            brand in
            NavigationLink{
                BrandCard(brand: brand, colorScheme: $isOn)
            } label: {
                BrandRow(brand: brand, colorScheme: $colorScheme)
            }
            .listRowBackground(Color.clear)
            .navigationBarItems(leading: Text("CAR BRANDS").font(.custom("SouvenirGotURWTOTReg W03 Rg", size: 20)).foregroundColor(isOn ? .green : .yellow)
                .toolbar {
                    ToolbarItem(placement:
                            .navigationBarTrailing) {
                                Toggle(isOn: $isOn) {
                                    Image(systemName:
                                            "lightbulb.fill").foregroundColor(isOn ? .black : .yellow)
                                }.accentColor(Color.clear)
                                .background(Color.clear)
                            }
                }
            )
        }
        .listStyle(.inset)
}.searchable(text: $searchText)
    .navigationViewStyle(.stack)
    .accentColor(isOn ? Color.green : Color.yellow)
```

Figure 14: NavigationLink for BrandCard

5. 

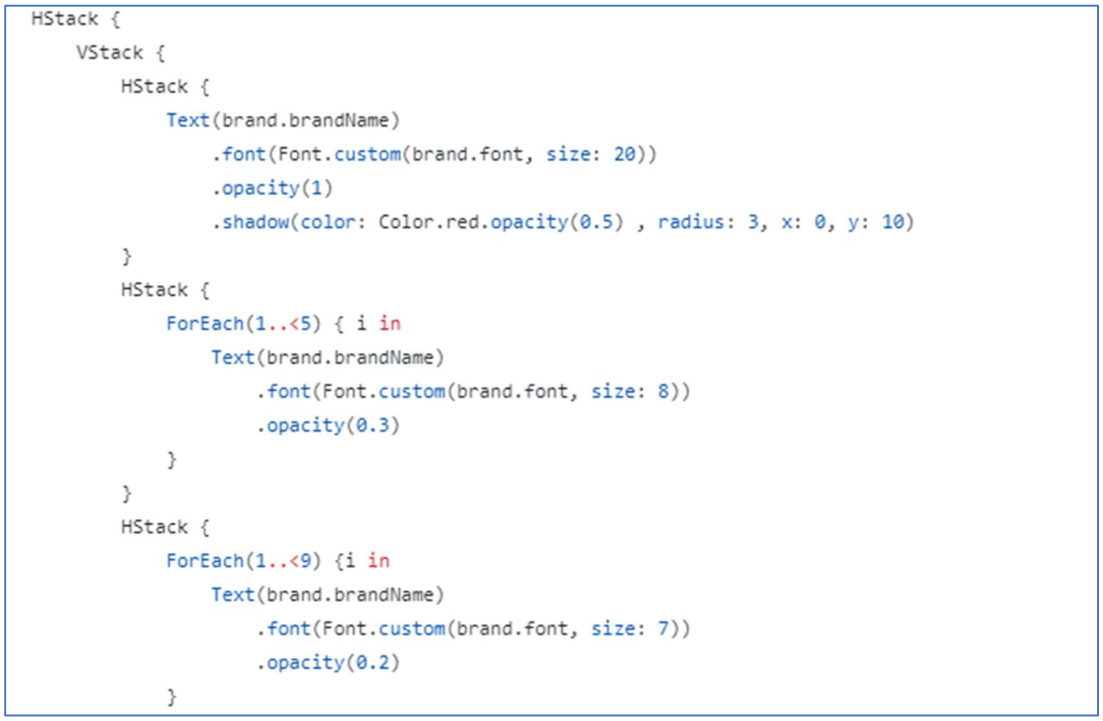
```
var filteredBrands : [Brand] {
    guard !searchText.isEmpty else {
        return brands.sorted(by: sortDirection == .asc ? {$0.brandName < $1.brandName} : {$0.brandName > $1.brandName})

    }
    return brands.filter{$0.brandName.localizedCaseInsensitiveContains(searchText)}
        .sorted(by: sortDirection == .asc ? {$0.brandName < $1.brandName} : {$0.brandName > $1.brandName})
}
```

Figure 15: Filtering Brands with sorted() and filter()

6. 

```
    }.environment(\.colorScheme, isOn ? .light : .dark)
    .overlay {
        Image(isOn && inBrandList ? "logo_black" : "logo_white")
            .resizable()
            .frame(width: 100, height: 100)
            .offset(x: -150, y: 370)
            .shadow(color: isOn ? Color.green.opacity(0.5) : Color.yellow.opacity(0.5) , radius: 1, x: 0, y: 10)
    }
}
```

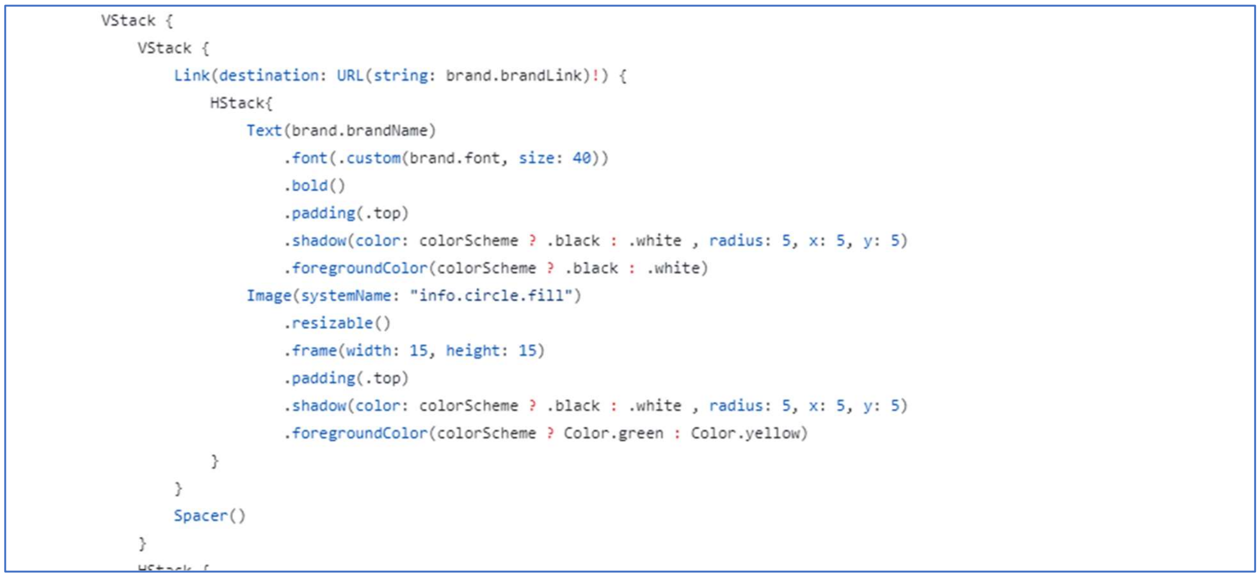
Figure 16: Overlay with the app Icon in BrandCard



7.

```
HStack {
    VStack {
        HStack {
            Text(brand.brandName)
                .font(Font.custom(brand.font, size: 20))
                .opacity(1)
                .shadow(color: Color.red.opacity(0.5) , radius: 3, x: 0, y: 10)
        }
        HStack {
            ForEach(1..<5) { i in
                Text(brand.brandName)
                    .font(Font.custom(brand.font, size: 8))
                    .opacity(0.3)
            }
        }
        HStack {
            ForEach(1..<9) {i in
                Text(brand.brandName)
                    .font(Font.custom(brand.font, size: 7))
                    .opacity(0.2)
            }
```

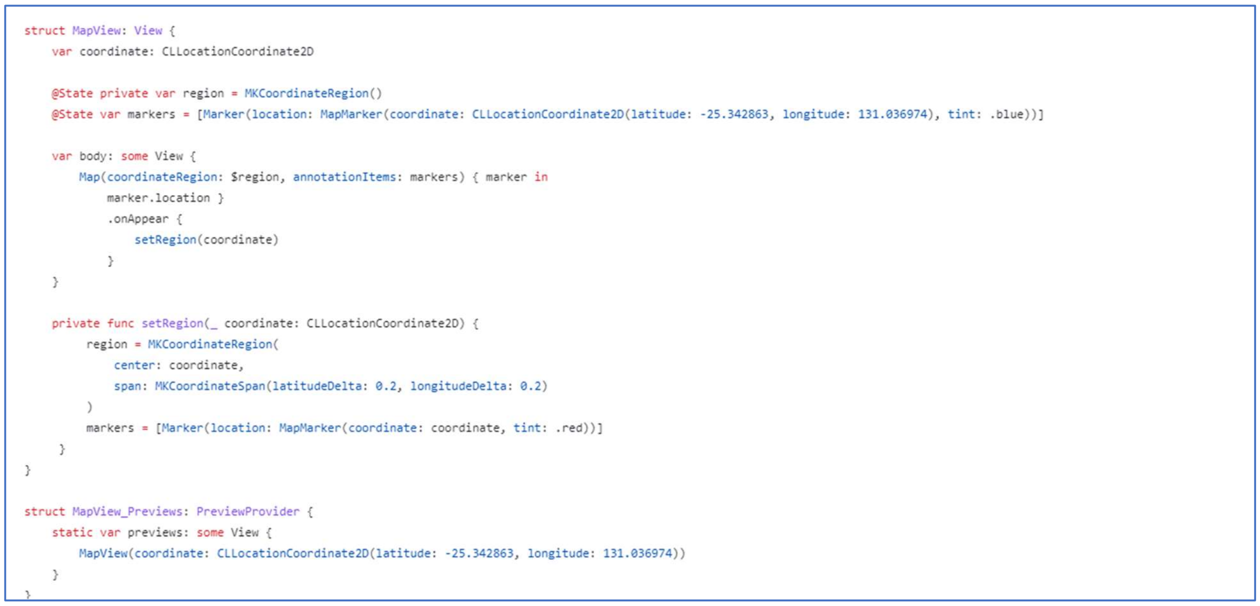
Figure 17: Build visualization for BrandRow



8.

```
        VStack {
            VStack {
                Link(destination: URL(string: brand.brandLink)!) {
                    HStack{
                        Text(brand.brandName)
                            .font(.custom(brand.font, size: 40))
                            .bold()
                            .padding(.top)
                            .shadow(color: colorScheme ? .black : .white , radius: 5, x: 5, y: 5)
                            .foregroundColor(colorScheme ? .black : .white)
                        Image(systemName: "info.circle.fill")
                            .resizable()
                            .frame(width: 15, height: 15)
                            .padding(.top)
                            .shadow(color: colorScheme ? .black : .white , radius: 5, x: 5, y: 5)
                            .foregroundColor(colorScheme ? Color.green : Color.yellow)
                    }
                }
                Spacer()
            }
```

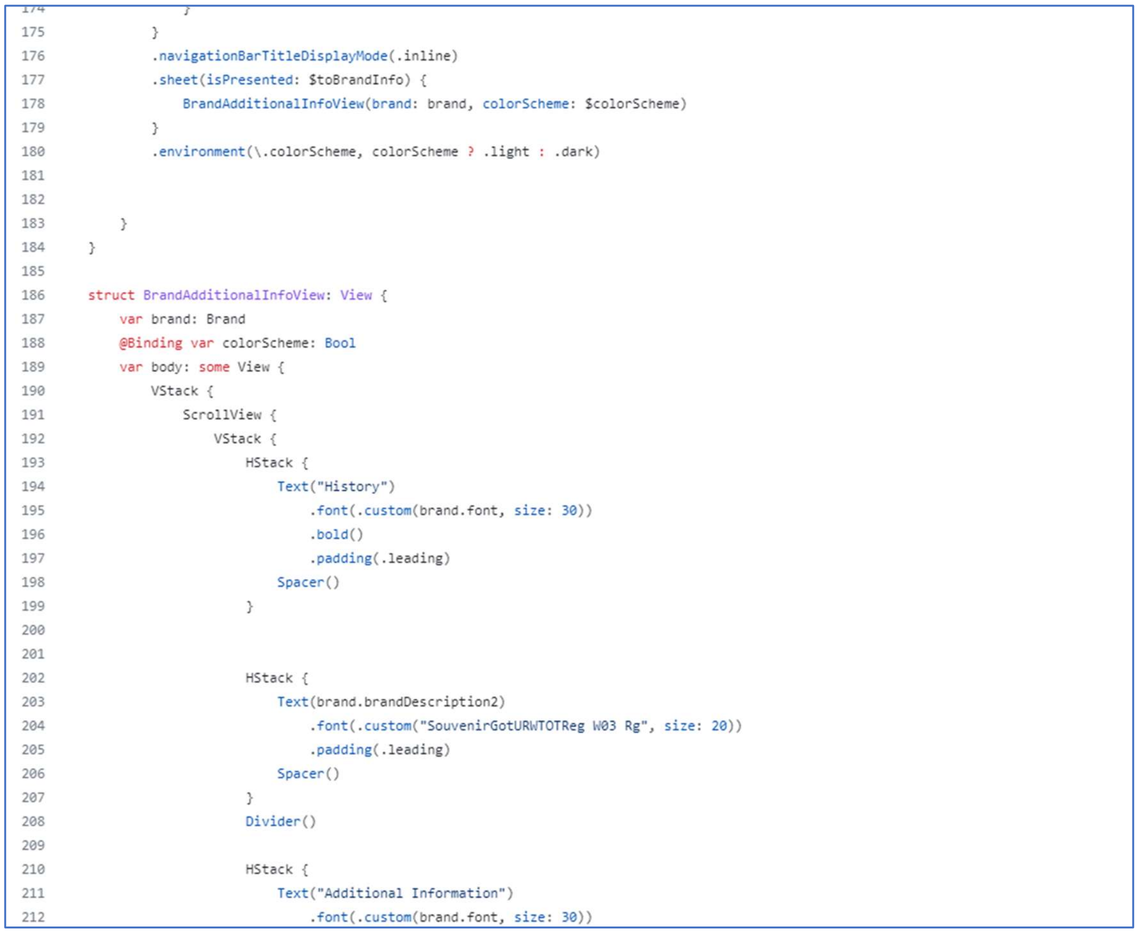
Figure 18: Buttons that link to the brand's website



9.

```
struct MapView: View {
    var coordinate: CLLocationCoordinate2D

    @State private var region = MKCoordinateRegion()
    @State var markers = [Marker(location: MapMarker(coordinate: CLLocationCoordinate2D(latitude: -25.342863, longitude: 131.036974), tint: .blue))]

    var body: some View {
        Map(coordinateRegion: $region, annotationItems: markers) { marker in
            marker.location }
            .onAppear {
                setRegion(coordinate)
            }
    }

    private func setRegion(_ coordinate: CLLocationCoordinate2D) {
         region = MKCoordinateRegion(
             center: coordinate,
             span: MKCoordinateSpan(latitudeDelta: 0.2, longitudeDelta: 0.2)
         )
         markers = [Marker(location: MapMarker(coordinate: coordinate, tint: .red))]
     }
}

struct MapView_Previews: PreviewProvider {
    static var previews: some View {
        MapView(coordinate: CLLocationCoordinate2D(latitude: -25.342863, longitude: 131.036974))
    }
```

Figure 19: MapView() modified

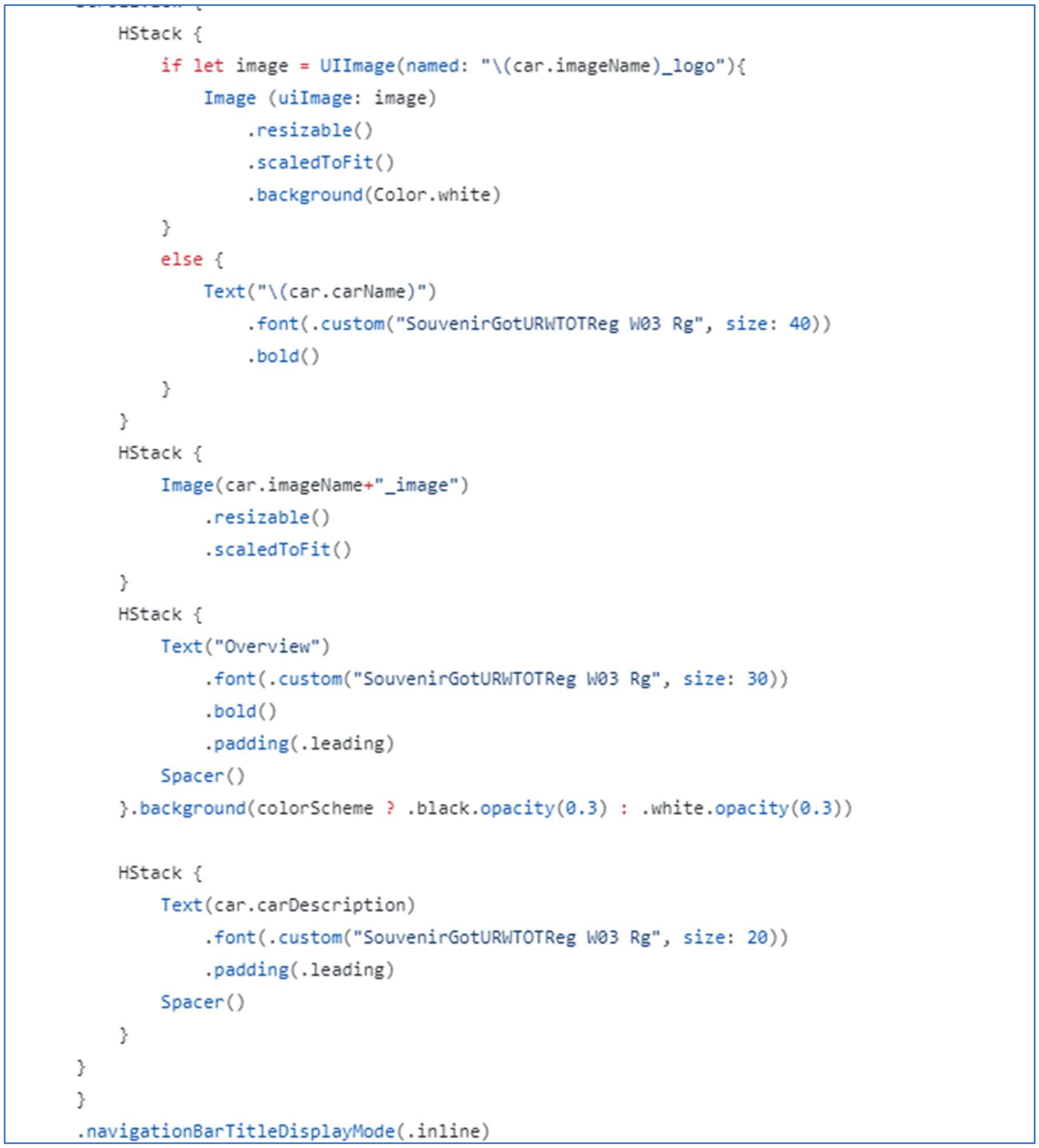


10.

```
            }
            .navigationBarTitleDisplayMode(.inline)
            .sheet(isPresented: $toBrandInfo) {
                BrandAdditionalInfoView(brand: brand, colorScheme: $colorScheme)
            }
            .environment(\.colorScheme, colorScheme ? .light : .dark)


        }
    }

    struct BrandAdditionalInfoView: View {
        var brand: Brand
        @Binding var colorScheme: Bool
        var body: some View {
            VStack {
                ScrollView {
                    VStack {
                        HStack {
                            Text("History")
                                .font(.custom(brand.font, size: 30))
                                .bold()
                                .padding(.leading)
                            Spacer()
                        }


                        HStack {
                            Text(brand.brandDescription2)
                                .font(.custom("SouvenirGotURWTOTReg W03 Rg", size: 20))
                                .padding(.leading)
                            Spacer()
                        }
                        Divider()

                        HStack {
                            Text("Additional Information")
                                .font(.custom(brand.font, size: 30))
```

Figure 20: BrandInfoView() to be presented as SheetView

11.



```
        HStack {
            if let image = UIImage(named: "\(car.imageName)_logo"){
                Image (uiImage: image)
                    .resizable()
                    .scaledToFit()
                    .background(Color.white)
            }
            else {
                Text("\(car.carName)")
                    .font(.custom("SouvenirGotURWTOTReg W03 Rg", size: 40))
                    .bold()
            }
        }
        HStack {
            Image(car.imageName+"_image")
                .resizable()
                .scaledToFit()
        }
        HStack {
            Text("Overview")
                .font(.custom("SouvenirGotURWTOTReg W03 Rg", size: 30))
                .bold()
                .padding(.leading)
            Spacer()
        }.background(colorScheme ? .black.opacity(0.3) : .white.opacity(0.3))

        HStack {
            Text(car.carDescription)
                .font(.custom("SouvenirGotURWTOTReg W03 Rg", size: 20))
                .padding(.leading)
            Spacer()
        }
    }
    }
    .navigationBarTitleDisplayMode(.inline)
```

Figure 21: Code for Car Information