Team Volkswagen

Michigan State University

Volkswagen Shopping App with Augmented Reality

Project Plan

Fall 2023

Volkswagen Contacts

Igor Efremov

Hassan Elnajjar

Eugene Pavlov

Michigan State Capstone Team

Bryce Cooperkawa

Nahom Ghebredngl

Rikito Takai

Swathi Thippireddy

Richard Zhou

Table of Contents

Executive Summary

Volkswagen Group of America (VWGoA) is the U.S. subsidiary of Volkswagen AG, one of the world's leading automobile manufacturers and the largest carmaker in Europe. VWGoA serves as the hub for U.S. operations of well-known brands like Audi, Bentley, Bugatti, Lamborghini, and Volkswagen, along with VW Credit, Inc. VWGoA also owns VWGoA Chattanooga Operations, which operates an assembly plant in Chattanooga, Tennessee. Established in 1955, VWGoA is headquartered in Herndon, VA., employs around 6,000 people in the United States, and sells its vehicles through approximately 1,000 independent dealers.

Volkswagen prioritizes "customers' needs - from the early development of the vehicles, through the configuration and purchase of a car", says Ralf Brandstätter, CEO of the Volkswagen brand. Volkswagen envisions a future where customers can explore, customize, and visualize their desired Volkswagen vehicles from the comfort of their own spaces using Augmented Reality (AR) technology.

Team Volkswagen strives to achieve this goal by empowering potential customers with an immersive and interactive car viewing experience, bringing window shopping from the dealership to the home. Through a mobile app, users can select their desired Volkswagen car model and make personalized customizations, which are all brought to life in AR. This innovative approach represents Volkswagen's commitment to embracing technology and enhancing the customer experience, ultimately redefining the way we shop for and engage with automobiles.

Functional Specifications

Car dealerships offer potential buyers the opportunity to view and test drive certain vehicles, allowing them to make informed purchasing decisions. Unfortunately, scheduling an appointment and finding time to visit a dealership during their operating hours is a challenging task for many individuals, especially when in-demand vehicles are often not in stock.

To deliver convenience, our AR application allows for placement of real-scale Volkswagen vehicles in any suitable location. The app is also implemented with color customization and accessory attachments to match the user's unique preferences, providing even more features than a typical dealership.

Up to two cars may be placed in the real-world scene, with intuitive controls for doing so. Pointing the device's camera in different directions situates a car during its placement phase, and dragging along the screen rotates the model on its surface plane. The car's size can also be manually reduced for further convenience, though clear indication is provided to ensure the user understands what the car's actual scale should be.

The inside of a vehicle is viewed in its own scene, where the user doesn't go through a placement sequence, instead experiencing the car as if they were actually sitting in any of the available seat positions. In this environment, they can examine the car's interior components and interact with certain controls, such as the horn, without worrying about their real-world position.

Design Specifications

Overview

The Volkswagen Shopping App with Augmented Reality is an iOS mobile application that allows users to select from a range of life-sized VW car models, like the ID Buzz and Atlas, from the comfort of their homes. It offers an engaging experience with customizable accessories, interactive features such as honking the horn, and capturing photos and videos. Users can access detailed vehicle specifications through the VW website, save their customized vehicle configurations, and even share their unique creations on social media.

Login/Create Account

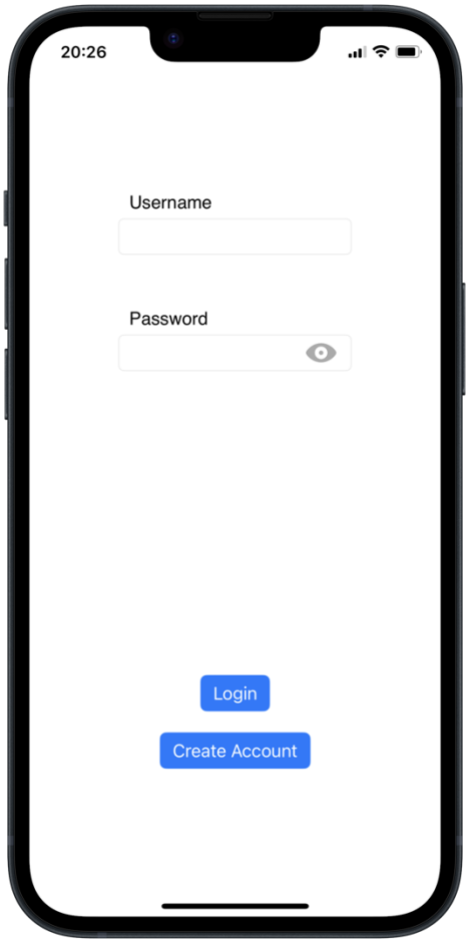

Figure 1 – Login

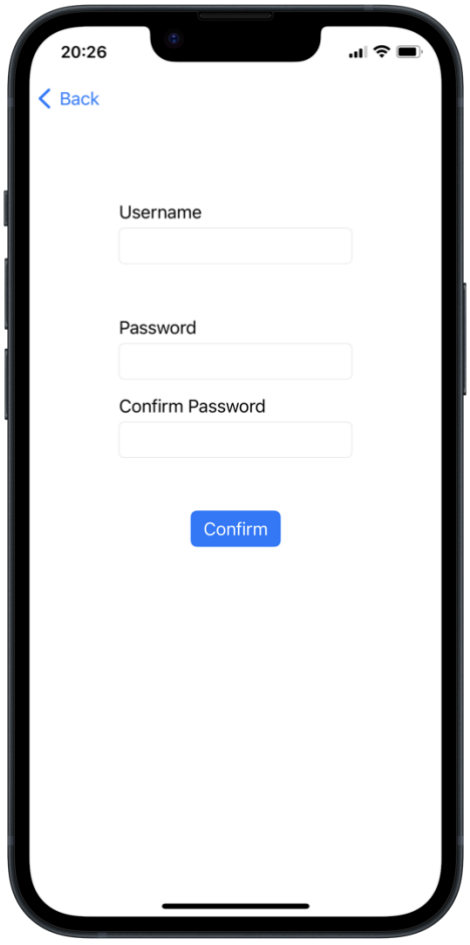

Figure 2 – Create Account

The login screen shown in Figure 1 presents a straightforward interface with some text fields for entering a username and password. First-time users may navigate to the “Create Account” screen by tapping the associated button. On the account creation interface (Figure 2), users input a unique username and password, re-enter the password, and upon pressing "Confirm," they are redirected back to the login page for immediate access with the new credentials.

Selection View

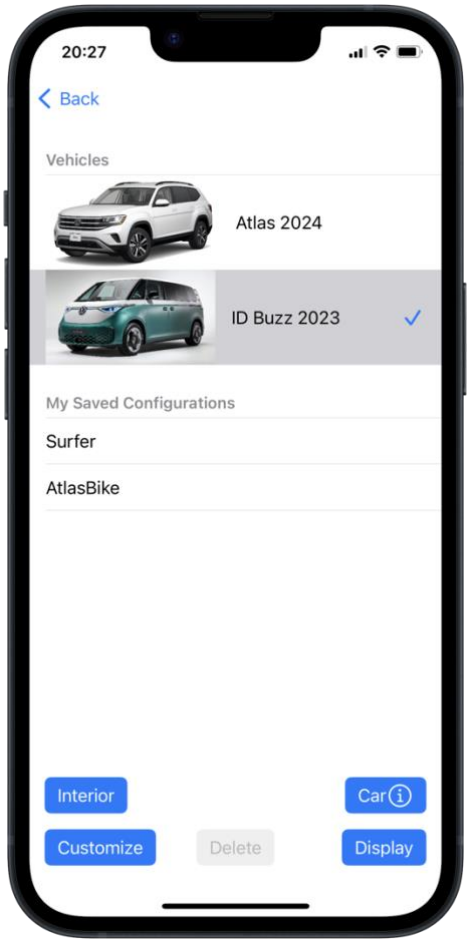

Figure 3 – Model Selection

The Selection view displays available Volkswagen vehicles. If the user has saved a customized vehicle before, it will be shown with its given name. For users interested in customizing a Volkswagen vehicle, they can select a model and tap the "Customize" button to transition to the Customization View. To delete saved configurations, users

can select one and tap the delete button. To explore the interior of the chosen vehicle, users can tap the "Interior" button to switch to an augmented view of the interior. For additional details like price or car specifications, users can press the "Car info" button to visit the Volkswagen website for the chosen model. Lastly, to view the vehicle in Augmented Reality, tapping the "Display" button transitions to the Car Display View.

Interior View

Figure 4 – Interior

The Interior View showcases the interior of the selected vehicle in augmented reality as shown in Figure 4. Users can explore the interior by rotating their mobile device. Tapping the "next seat" button allows users to switch between all available seats in the vehicle. If the user is in the driver's seat, tapping on the steering wheel will simulate the sound of the horn.

Customization View

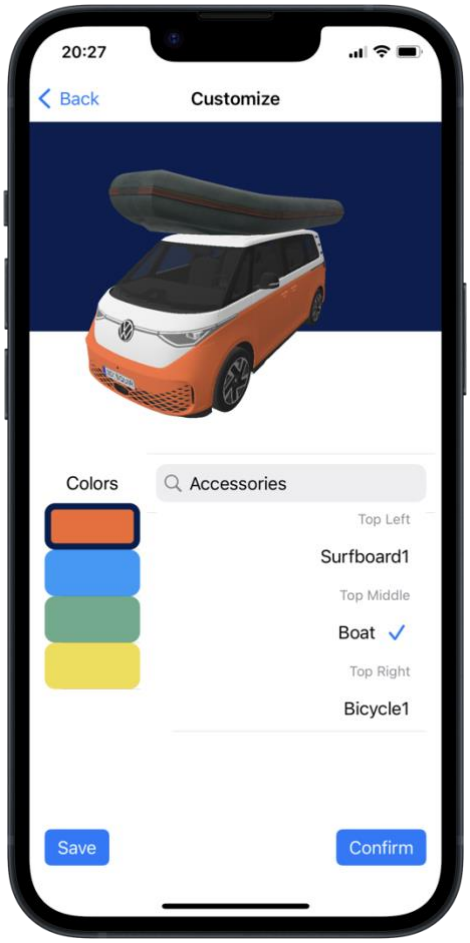

Figure 5 – Buzz Customization

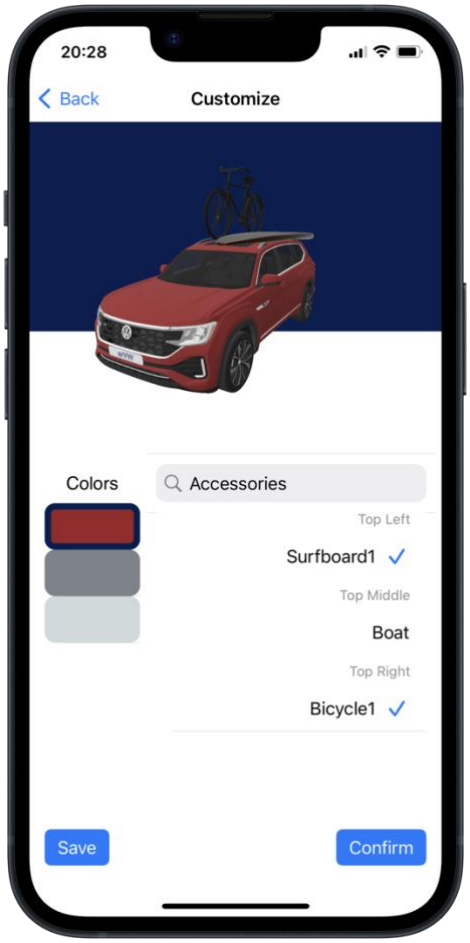

Figure 6 – Atlas Customization

In this view, users can customize a Volkswagen vehicle, as depicted in Figures 5 and 6. The vehicle's color can be changed using the color boxes on the left, featuring options specific to the chosen model. Additionally, users can personalize the car by adding accessories such as a surfboard and a bicycle to assess their fit and enhance the overall appearance when displayed. Once satisfied with the customization, users can either tap the "Save" button to store the configuration for later use or tap the "Confirm" button to transition to the Car Display View.

Car Display View

Figure 7 – Place Vehicle

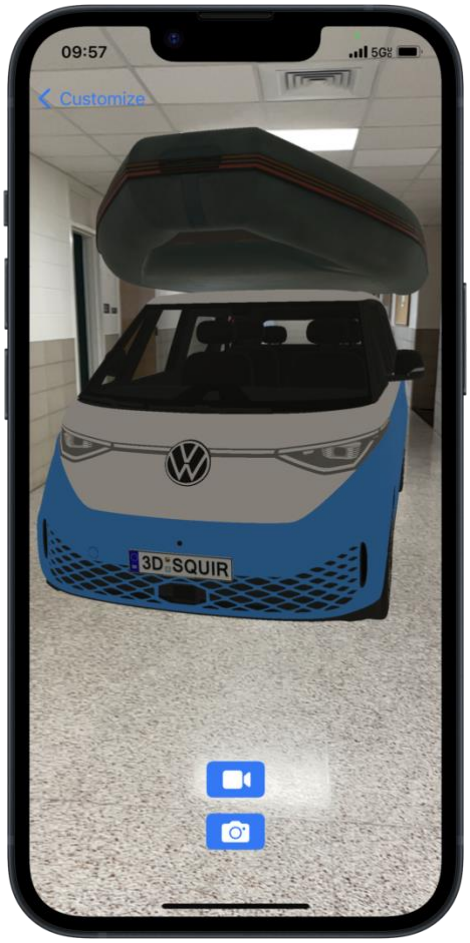

Figure 8 – Capture Video/Photo

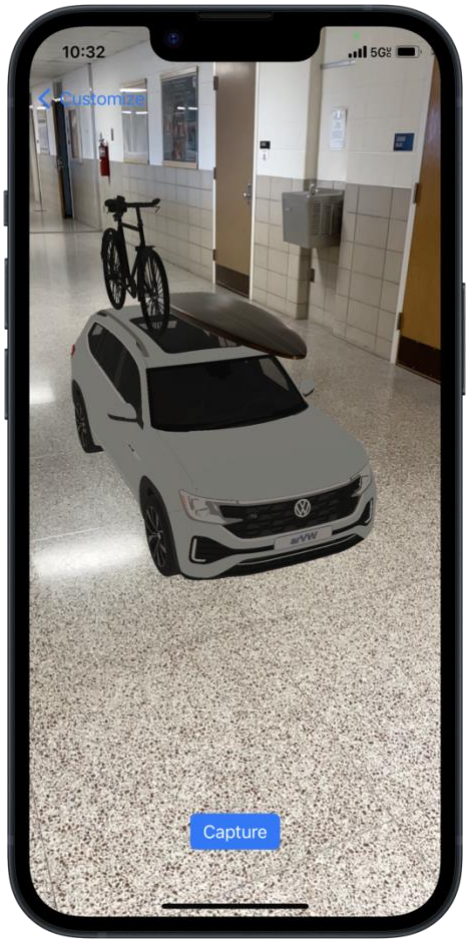

Figure 9 – Medium Scaled Vehicle

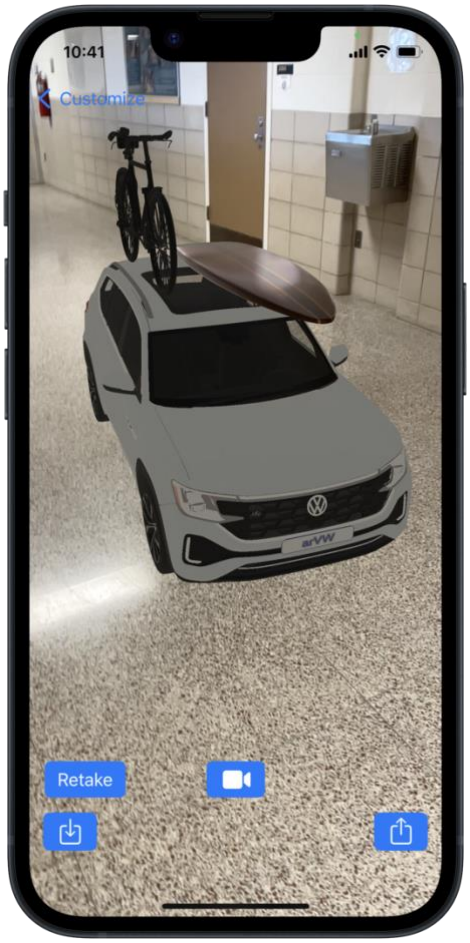

Figure 10 – Save/Share Image

The Car Display View offers an interactive experience with dynamic UI changes as the user progresses. In this view, users need to find an open space to place the real-sized model of the vehicle as shown in Figure 7. Once a suitable space is found and the vehicle is displayed, users can choose between the actual scaled, medium, or small-sized model as depicted in Figure 9. Upon selecting the size, users can place the model by pressing the "Add" button. After placement, users have the option to capture a video by pressing the video recorder icon or take a photo by pressing the camera icon as depicted in Figure 8. After capturing a photo, users can choose to retake it by pressing the “Retake” button as shown in Figure 9. If satisfied with the photo, they can press the bottom left button to save the image or tap the bottom right button to share the image with friends and family.

Admin Panel

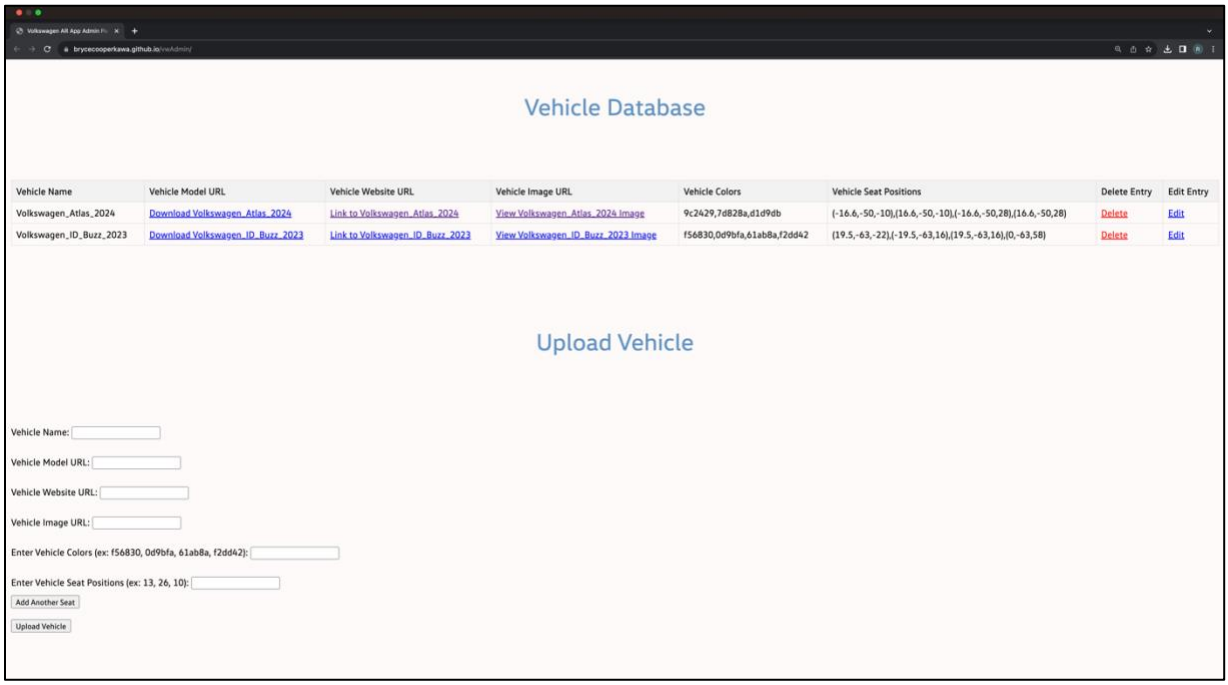

Figure 10 – Admin Panel Top

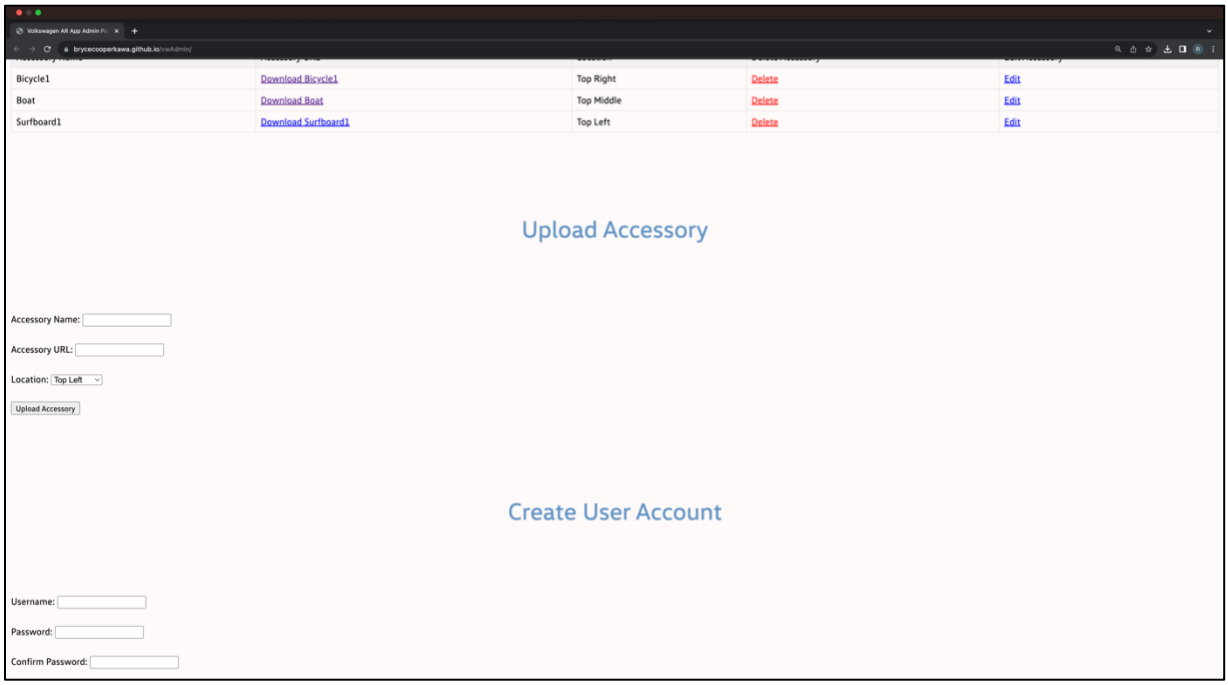

Figure 11 – Admin Panel Bottom

The applications admin can access the Admin Panel, as depicted in Figure 10. The Admin Panel serves to enable database management without the need for knowledge of database functionality. It provides the admin with the ability to add, delete, or edit model information. Adding a vehicle model requires information such as the vehicle name, S3 URL for the model, Volkswagen website link for the corresponding vehicle, S3 URL for the car image, available vehicle colors, and vehicle seat positions. Additionally, admins can add accessory information as shown in Figure 11, including the accessory name, S3 URL, and the designated location for placement. Finally, the admin has the ability to create new accounts.

Technical Specifications

System Architecture

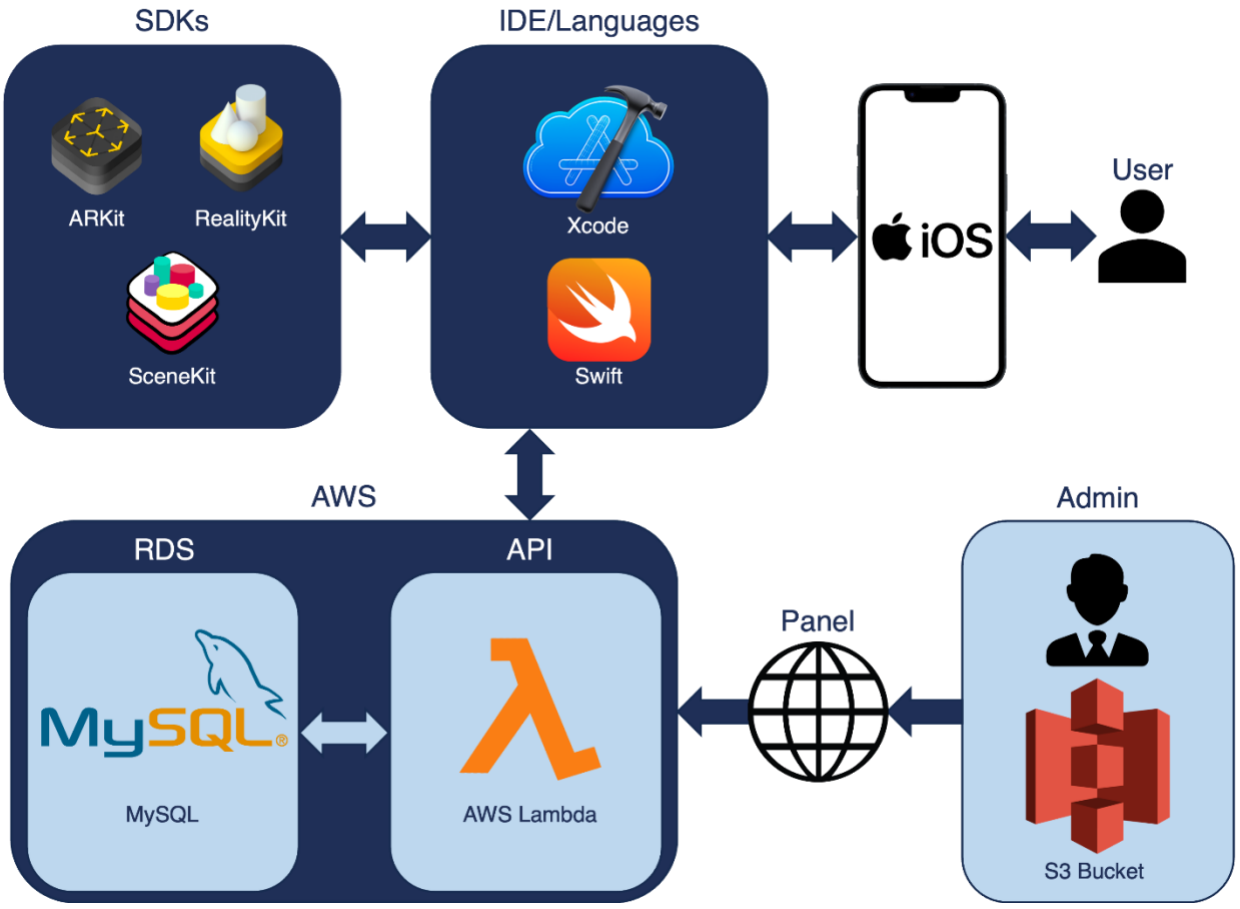

Figure 12 – System Architecture

Overview

The Volkswagen augmented reality (AR) shopping app will be deployed to iOS devices using Xcode, Apple's integrated development environment. This deployment will leverage Xcode's available AR development tools, including ARKit, RealityKit, and SceneKit.

ARKit will give the application the power to make use of Apple's augmented reality features. Its ability to detect and track the real world through the device's camera will enable users to seamlessly place a virtual car, allowing them to gain a realistic perspective of how it fits into their surroundings.

RealityKit provides the AR scene with details that immerse the user in the experience. The AR-first 3D framework stacks on top of ARKit to provide useful tools that allow developers to seamlessly integrate virtual objects with the real world.

SceneKit will be leveraged to empower the user to customize their vehicle through its 3D model creation and editing capabilities. By allowing the manipulation of 3D assets this powerful API will be used to combine models with accessories and will allow for sound to be embedded in the scene. While other AR tools are used display models, SceneKit builds, customizes, renders, and animates them.

The tools described above will be combined through Xcode, the default IDE for developing iOS applications. Through this powerful IDE, developers will be able to write comprehensive and functional code that can be distributed to capable iOS devices.

MySQL Workbench hosted on AWS RDS will be used to host database tables. These tables contain user information, customization specifications, vehicle models, and accessory models. Serving as the foundation for the backend of the application, MySQL provides a repository for dynamically accessed and stored information.

AWS Lambda API serves as the intermediary between the application, database, and admin panel through its API functions, streamlining the flow of information. By leveraging API endpoints, it ensures enabling data to be pushed and pulled between these components. This integration links the backend of the application to the frontend.

The Amazon S3 Bucket serves as a cloud-based storage for model files and images. By leveraging the S3 Bucket, this integration enables the application to pull files from the cloud, rather than from local storage.

Each individual component plays a pivotal role in the system. Xcode will host the application and make much of the framework accessible through views. For views that require AR capabilities, ARKit will provide a base for the functionality and RealityKit will allow the developer access to that functionality through the creation of ARViews. SceneKit is instrumental in crafting highly interactive and customized augmented reality scenes. MySQL Workbench acts as the cornerstone of the backend, storing essential information within easily navigable tables. AWS Lambda API serves as the bridge

connecting the backend to the frontend and vice versa. The Amazon S3 Bucket stores model files and images in the cloud. Collectively, Xcode, ARKit, RealityKit, SceneKit, MySQL Workbench, AWS Lambda API, and the Amazon S3 Bucket collaborate to form a comprehensive framework for developing the Volkswagen Augmented Reality Shopping App. Each component playing a crucial role in achieving functionality and realism.

System Components

Xcode:

- Xcode is an integrated development environment (IDE) created by Apple Inc. that provides a comprehensive set of tools for the creation, testing, and deployment of applications for Apple devices.
- Primarily used for the development of software for Apple's macOS, iOS, watchOS, and tvOS platforms.
- Xcode also boasts an extensive collection of resources for developing Augmented Reality applications for iOS.

ARKit:

- ARKit is the foundation of AR (Augmented Reality) development on iOS. It provides the core functionality for detecting and tracking the real-world environment through the device's camera.
- ARKit is responsible for tasks such as motion tracking, world tracking (processing the 3D space and surfaces in the real world) and detecting features like points and objects.

RealityKit:

- RealityKit is a relatively new framework from Apple that is tightly integrated with ARKit and designed explicitly for AR and VR (Virtual Reality) development.
- It provides higher-level abstractions and tools for creating AR experiences, making it easier for developers to build AR apps without deep knowledge of 3D graphics or physics.
- RealityKit includes features like entity-component system architecture, physics simulation, and advanced rendering techniques optimized for AR.
- It is tightly integrated with ARKit, so developers can easily combine AR tracking capabilities with 3D rendering and interactivity provided by RealityKit.

SceneKit:

- SceneKit is a 3D graphics framework provided by Apple. It is designed for building 3D interactive scenes and can be used for making highly realistic 3D content in AR applications.
- Developers can use SceneKit to create 3D models, behaviors, and scenes that can be placed into the AR environment detected by ARKit.

- While SceneKit can be used with ARKit to render 3D content in augmented reality, it is not specifically tailored for AR development. Developers must handle AR-specific tasks (e.g., tracking and interaction) separately using ARKit.

MySQL:

- MySQL is an open-source database management system used for organizing and managing structured data efficiently.
- Offers robust capabilities for creating and managing databases, tables, and relationships between data entities.
- Developers leverage MySQL to execute queries, retrieve, insert, update, and delete data, ensuring seamless data manipulation within applications.

AWS Lambda:

- AWS Lambda is a serverless computing service provided by Amazon Web Services that allows developers to run code without provisioning or managing servers.
- Developers upload their code to Lambda, which automatically handles the execution and scaling based on incoming requests and triggers.

Amazon S3:

- Amazon S3 is an object storage service provided by Amazon Web Services, designed for secure storage of data, files and objects within the cloud.
- Enables seamless integration with other AWS services and external applications, facilitating data transfer, content delivery, and backup processes.

Risks Analysis

Working with 3D Models and adjusting them to accommodate customization features

Difficulty: Medium

Description: Changing the colors of individual parts of the vehicle and attaching accessories such as bikes or surfboards might be difficult depending on the type of 3D car model we use. Making accessories attach to this model and all future car models in a believable way could also pose some overhead challenges.

Mitigation: USDZ files can be modified in Xcode. Models can be further adjusted in other 3D modeling software.

Selecting and loading customization options

Difficulty: Easy

Description: Accessories being distinct models that must “attach” to the vehicle might reduce speed when adding more cars and accessories. Changing accessories and colors in real time (milliseconds) may not even be feasible.

Mitigation: Frontloading models and preparing them for AR on app login. SceneKit is able to interact with the individual components of 3D objects programmatically.

Object projection in different environments and conditions

Difficulty: Hard

Description: When working with Augmented Reality, it can sometimes be hard to determine the placement of an object. Inadequate space, slanted inclines, and even lighting conditions can affect how a car will appear in the scene.

Mitigation: Apple’s AR tools have intelligent solutions to placing objects in AR.

Adjusting car audio based on vehicle specifications and location

Difficulty: Easy

Description: Starting the car’s engine and honking the horn should play audio from an accurate point in space, and electric cars must have different noises than conventional cars. Vehicle distance and scale should also have an impact on noise.

Mitigation: RealityKit has a solution for location-based noise via Spatial Audio. We can scale the sound intensity based on distance using an inverse square equation based on distance.

Schedule

Week 1: 8/28 - 9/3

- Initial team meeting
- Reach out to client contact
- Set up Macs with necessary software

Week 2: 9/4 - 9/10

- Initial client meeting (schedule, office hours)
- Initial triage meeting with TM Luke Sperling
- Begin research on necessary technologies

Week 3: 9/11 - 9/17

- **Team Status Report Presentation**
- Design screen mockups and system architecture
- Work on Project Plan Document and Presentation
- Create prototypes to understand technologies

Week 4: 9/18 - 9/24

- **Project Plan Presentation**
- Project and visualize ray from camera
- Basic View structure (each planned view is initialized, and the user can navigate between them)
- Developed table view for customization
- Testing model swap for accessories

Week 5: 9/25 - 10/1

- Preview with Focus Entity
- Swap between accessories quickly and without losing anchor (Car View page alpha ready)
- Capture alpha ready (take pictures)

Week 6: 10/2 - 10/8

- Share is Alpha ready
- Interior is Alpha ready
- Car makes noise on button presses
- Customization View is Alpha ready
- Login working with database and S3 bucket
- Present alpha deliverable to client and ensure they’re happy with the direction of the project.
- Focus Entity for Preview makes it Alpha Ready

Week 7: 10/9 - 10/15

- Presentation prep
- Demo Selection page
- Refactoring to account for new database
- Front loading models to speed transitions
- **Alpha Presentations**
- Major schedule rehaul

Week 8: 10/16 - 10/22

- Capture/Share view(s) are feature ready
- Transition to Scene Kit version of Car Display (to pull changes over from Customize)

- Adding car to display only possible on plane detection

Week 9: 10/23 - 10/29

- Clean memory
- Admin Panel can push models to database
- Refactor for more robust software structure
- Work on API & database interactions
- Implemented smooth swapping between accessories in Customize

Week 10: 10/30 - 11/5

- Dynamically updating Admin Panel tables
- Delete models in Admin Panel
- Testing/Bug Fixing
- Video Capture start

Week 11: 11/6 - 11/12

- Video Capture finalized (stretch goal)
- Database & Admin Panel updates
- Info View links to Volkswagen website
- Initial filter bar for accessories
- Finished Video Capture
- Preview Finalized
- **Status Report Presentation**
- Testing/Bug-fixing

Week 12: 11/13 - 11/19

- Localized Interior sound to steering wheel
- Car scaling
- Users' saved customizations finalized
- Multiple cars feature finalized and integrated
- **Beta Presentation**

Week 13: 11/20 - 11/26

- Incorporate Beta Presentation Feedback
- Bug-fixing
- UI cleanup
- Rotation work
- Working on project video
- Prepare presentation for client

Week 14: 11/27 - 12/3

- **Status Report Presentation**
- **Project Video due**
- **Final client presentations**
- Prepare for design day

Week 15: 12/4 - 12/10

- **All deliverables due**
- **Design Day**

Week 16: 12/11 - 12/13

- **Capstone Wrap up**