BSCS-S24-010

03-134211-021 Muhammad Abdullah Subhani

03-134202-025 Hasan Umair

AR Visualizer

Redefining Design Visualization with Realistic Augmented Reality Experiences

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: JUNAID NASIR

Department of Computer Sciences

Bahria University, Lahore Campus

December 2024

C e r t i f i c a t e

We accept the work contained in the report titled

“AR Visualizer”

written by

Muhammad Abdullah Subhani

Hassan Umair

as a confirmation to the required standard for the partial fulfilment of the degree of

Bachelor of Science in Computer Science.

Approved by:

Supervisor: **Junaid Nasir**

_____ (Signature)

December 05, 2024

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134211-021	**Muhammad Abdullah Subhani**	
03-134202-025	**Hassan Umair**	

Date : December 05, 2024

Specially dedicated to
my Supervisor, Parents and my friends
(**Muhammad Abdullah Subhani**)
my Supervisor, Parents and my friends
(**Hassan Umair**)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to my project supervisor, **Mr. Junaid Nasir** for his invaluable advice, guidance and his enormous patience throughout the development of the project.

In addition, we would also like to express my gratitude to our loving parents and friends who had helped and given me encouragement.

Muhammad Abdullah Subhani
Hassan Umair

AR Visualizer

Redefining Design Visualization with Realistic Augmented Reality Experiences

ABSTRACT

Augmented Reality will change the way users approach interior design, immerse them in it interactively, and help bridge the gap between imagination with implementation. The AR Visualizer project is a cross-platform mobile application built with Flutter, Firebase, and ARCore/ARKit. This is an application that lets its users visualize furniture and other decorations in their real space by providing key features such as object placement, dynamically resizing, and integrating custom objects. These functionalities enable the user to fiddle around with layout and designs, change dimensions of items in space, or upload their favourite furniture images and convert them into 2-D models and visualize them in their space.

This AR visualizer is built by using Agile methodologies iteratively, in order continuously to improve according to users' feedback and testing. System architecture is such that it integrates a responsible frontend, a Firebase Backend, and most advanced AR frameworks just to make things work seamlessly as real-time updated applications or platforms. Rigorous testing on both Android and iOS devices showed the usability, scalability, and performance of the application.

The project achieved all its objectives, but challenges faced during the development were related to device compatibility, accuracy in detecting the surface, and optimization of performance. Given the associated challenges, this AR Visualizer has great potential to revolutionize the current design process for ordinary people and

interior designers by making the designing process easier, more engaging, and accessible.

Future enhancements could be suggested to extend the capacity of this application: advanced augmented reality, AI recommendation, offline mode, and compatibility with various wearable devices. With more innovation, the AR Visualizer will become a well-rounded tool for homeowners, designers, and architects.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

AR	Augmented Reality
2D	2 dimensional
3D	3 Dimensional
UI	User Interface
UX	User Experience
JPG	Joint Photographic Experts Group
PNG	Portable Network Graphics
USDZ	Universal Scene Description Zip
glTF	Graphics Language Transmission Format
API	Application programming interface
OBJ	File format for 3D models
FBX	Filmbox

CHAPTER 1

INTRODUCTION

1.1 Background

Interior design significantly impacts the functionality and aesthetic appeal of living and commercial spaces. However, traditional methods of designing rooms such as manual furniture placement or experimenting with decor arrangements can be time-consuming and costly. For instance, physically moving furniture to explore layouts is laborious and impractical, especially for larger or more complex spaces. Additionally, homeowners and designers often struggle to accurately visualize how various elements will look together in a specific room and their vision is limited but with the app they can easily experiment with different layouts and furniture placements

While current digital tools exist, they often fall short in terms of customization, realism, and user-friendliness, as they have the built-in furniture products available only. This limits their effectiveness for both professional designers and homeowners seeking practical, accessible solutions.

Advancements in Augmented Reality (AR) offer a transformative opportunity to address these challenges. AR technology allows users to overlay virtual objects onto their real-world environments through smartphones or tablets, enabling an interactive experience. Building on this potential, the AR Visualizer project will focus on simplifying the design process by developing a mobile application that focuses on pre-available furniture and decor items. This allows users to place the objects virtually in their space and visualize them in real time. The application will also be able to

transform photographs of real-world objects into 2-D representations, which a user can use to add custom objects into their design. This approach encourages creativity, and the users will intuitively and precisely be able to predict the result of their choice in design without necessarily trying this physically.

1.2 Problem Statement

Traditional ways of designing and personalizing interior spaces are cumbersome and full of inefficiencies. The laying out of furniture and decoration is highly a process of manual labor with many logistical problems, hence very impracticable, especially for big or complicated interior spaces. Moreover, users often can't accurately perceive how various layouts of furniture or décor elements will fit into their space.

The existing digital tools try to solve these issues, but most of them fail in the name of usability, realism, and flexibility. Many applications lack intuitive interfaces or the possibility of adjusting to individual users' needs, often not allowing the user beyond the boundaries of predefined object libraries, leaving no room for custom items. Such an attitude will make the users frustrated and uncertain about the result of the choices made.

These challenges call for a solution that will simplify furniture and decor arrangement, immersing users in an accurate visualization experience. This includes placing pre-available virtual objects in their environment in real time and transforming photographs of physical objects into 2D representations for integration into designs. The given solution uses various advanced technologies, including Augmented Reality, that make the interior design process much easier to cope with; thus, it stimulates creativity and develops confidence in the design decisions made.

1.3 Aims and Objectives

The AR Visualizer project focuses on developing a mobile application to transform the interior design experience using augmented reality (AR) technology. The main goals and objectives of the project are:

i) **Real-Time Visualization:** Allow the users to view furniture and decoration layouts using AR technology in their real spaces.

ii) **Virtual Furniture Placement:** The users are allowed to choose from a library of available furniture and decor or upload images of their own. These images would then be turned into 2D objects for integration into design plans

iii) **Cross-Platform Accessibility:** This app needs to be operable both on Android and iOS so that the maximum number of audiences can access it.

iv) **User-Friendly Interface:** Provide a friendly, engaging interface that ensures this application can be well accessed and utilized by everyone, whether they are homeowners and professionals in interior design and architecture

v) **Collaborative Development:** Engage with industry experts and other stakeholders in the process of fine-tuning the features for pragmatic needs of both professionals and casual users.

By achieving these objectives, the AR Visualizer is expected to provide an innovative, efficient, and user-oriented solution to the challenges encountered in the interior design process.

1.4 Scope of Project

The **AR Visualizer** project provides an all-inclusive package to fill the gaps throughout the whole process of house interior design for homeowners and interior designers/architects. This is a mobile application that lets users virtually experience room setups and place decor without the hassle involved in physical rearrangements or costly experiments. The platform empowers users to visualize real-time placement and size-up virtual furniture and decor elements in space with precise and intuitive accuracy by using augmented reality technology.

The app shall have an in-built library of pre-available furniture and decor objects, so a lot of different layouts can be tried. Users would also upload images of their furniture or decorations, turned into a 2D match for their project. To make users' designs appear in a close-to-life way, we add yet another feature: lighting simulation, which allows a design to be previewed under different lighting conditions.

The AR Visualizer project will consider cross-platform use and will therefore be available on both Android and iOS to reach a large audience without an expensive AR device. The interface will be very friendly; therefore, it could be called versatile, since a non-expert designer can easily use the app. The main targets which the functionality of the application must cover are practicality and AR: advanced AI features such as creative content generation, predictive design recommendations, or automatic design completion will not be included. AR Visualizer is intended to bring simplicity, access, and creativity in making interior design easy and enjoyable for everybody.

CHAPTER 2

LITERATURE REVIEW

The integration of AR in processes is revolutionizing the ways of homeowners, designers, and architects in visualizing the use of space. By overlaying digital information on real-world settings, augmented reality enables views and testing of design decisions without making changes to physical space. In doing so, the solution mitigates some pressing issues such as inefficiencies within furniture arrangement and unreal display of spatial visualization tools.

2.1 Applications of Augmented Reality in Interior Design

With Augmented Reality, a person could sit in the real world with a virtual couch or chair along with other kinds of physical home furniture. IKEA Place is an example of how AR can help cut black box decision-making by letting consumers view real-world possible configurations without the need for a traditional prototype [1-2]. In this case, the time and testing resources required from conventional homes decors would also be reduced.

Research also suggests that AR serves to boost engagement and trust among users. Findings by Trivedi Hritik support that AR-based design platforms give significant spatial insights, enabling a user to make much better and precise decision making [3]. Overall, AR is considered an essential tool for the modernization and optimization of interior design practice as it bridges the gap between conceptual design and tangible outcomes.

Most existing tools don't let one import from the factory but only from the built-in furniture and decoration library, thereby severely restricting what can be done with them. The AR Visualizer comes here, bringing the option of adding personalized objects to virtual layouts and, therefore, providing a more personal and flexible design experience.

2.2 Furniture Arrangement and Visualization

Furniture installation is a central point of interior designing because it basically refers to visual aesthetics and functional use of the place. The most traditional furniture arrangement often requires heavy physical labor, which is very harrowing, at least in a larger or more complicated layout. To solve that problem, different kinds of digital tools like AutoCAD, SketchUp, and Room Planner have been specifically created to allow users to view layouts inside a virtual environment. Advanced functionalities make this kind of tool complicated and usually fit for professional designers.
Augmented reality (AR) solutions such as Planner 5D and Magic Plan have even introduced an even more user-friendly interface to lure non-specialist users [5]. However, most of those 'user-friendly' tools do not provide precision and customization that an individualized design could provide. For example, most of them do not allow importing specific furniture or decor owned by the user.

The AR Visualizer seeks to solve all these problems by providing export functionality to the user as well, wherein the user can click photographs of real-world, and the object will be transformed into 2D virtual assets. This functionality tremendously increases personalization because users can include their actual furniture and decorative items in the design process. The application's interactive control bar for resizing objects also ensures that very precise adjustments are made, enabling items to fit perfectly into specific spaces. Thus, combining ease-of-use with complex customization, the AR Visualizer forms a practical and user-friendly approach to furniture installation and interior design.

2.3 User Interface Design for AR Applications

Quality in augmented reality applications need proper hands-on experience with expertise in the interface and user experience to make it effective. Studies show that UI that is simplified and intuitive, becomes good for the usability and adoption of such inputs by nonprofessionals [7]. In contrast, complex and poorly designed interfaces usually scare off users when they are supposed to entice them to technology. Studies also show clear and minimized such design elements while allowing the participant to be free to discover those complicated and creative components involved in designing [8].

The AR Visualizer demonstrates all this, with a specific interface adapted easily for both professional designers and household casual users. The critical aspect of this theme is the Object Resizing, which cannot be achieved through an intuitive slider that allows for accurate continuous and seamless adjustments. In general, this leads to a non-hassling and pleasant design experience thereby being convenient and practical for just about anyone.

2.4 Existing Gaps and Opportunities:

Many of the current tools are based on pre-created libraries of objects limited to some personalized profiles and users might find it difficult to create their own customized interiors as per the requirements of the users [4]. More Importantly, if not all, of the highly significant functionalities like resizing the object are missing, making it impossible to make accurate, functional layouts [6]. Accessibility is yet another main issue, where all high-class professional tools such as Revit and Sketchup are designed with expert users in mind, leaving behind casual users [5]. Some interfaces prove to be really complicated to keep away from users, especially for those without training in design [8].

The identified problems lead us to think that really such an AR application should be customizable, relatively easy to access and certainly make some use of to be usable by itself. The AR Visualizer now fills these gaps, offering such features as real-time object spacing, integration of custom objects, and object resizing. Above all it provides an

end-to-end solution toward interior designing as much as it is intuitive and inclusive for addressing a range of users.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Agile Development:

The AR Visualizer project is implemented using an Agile development methodology. Agile is chosen because it is flexible and iterative so that the developed features can be delivered to the customer in an incremental basis. The Agile structure of the project made it easier to bring inclusion of feedback, allow for flexible alteration of goals changes or problems faced during construction. Each period of development referred to as a sprint was two weeks and focused on developing individual features, such as object resizing or AR integration. The results from the sprint are tested with review sessions to ensure following if they meet the user requirements.

3.2 Requirements Gathering

The phase of requirements gathering proved instrumental in defining the application. The survey and interviewed homeowners, the hired interior designers and architects, and the team of developers pointed out the following issues with existing AR tools. A general complaint was regarding the non-customization features of many applications, that users were not allowed to upload their own furniture or rearrangement of dimensions dynamically. In addition, they highlighted that the interface must be very simple-intuitive to make it accessible to the masses through AR-based design tools.
Major findings from this phase comprised:

- A need for resize feature to be able to resize furniture dynamically to fit spaces well.
- Support for including a custom furniture item via user-uploaded images.
- A strong need for cross-platform compatibility to serve both Android and iOS users.

This feedback laid down the basis for defining the scope of the project and the priority of features for development.

3.3 System Design

System design of the AR Visualizer was created to achieve modularity and scalability. The system architecture is set up in three mainly layered modules: frontend, backend, and AR framework. Such system architecture allowed seamless interaction of those components while providing potential future scalability options on demand.

3.3.1 Frontend Design

The front end is built using a cross-platform framework, Flutter, to give the same functionality and look on both Android and iOS devices. Flexibility of the widget system in Flutter allowed the team to design a user interface that was both intuitive and visually engaging. Key front-end functionalities included object selection, resizing controls, and a custom image uploading interface.

3.3.2 Backend Design

Firebase acted as the backend that could offer real-time database functionality and cloud storage facilities to handle pre-existing as well as uploaded personalized object creations. Firebase completely took care of the secure storage and instant availability of user data and images that were generated through the app, thus leading to fast and seamless interactions.

3.3.3 AR Framework

The AR features in the app are developed on ARCore for Android and ARKit for iOS. This allowed the application to truly show the virtual objects in the direct real world by real-time object placing, interaction, and resizing. The application reached highly precision and responsive level exploitation of these frameworks.

3.4 UML Diagrams

3.4.1 Class Diagram

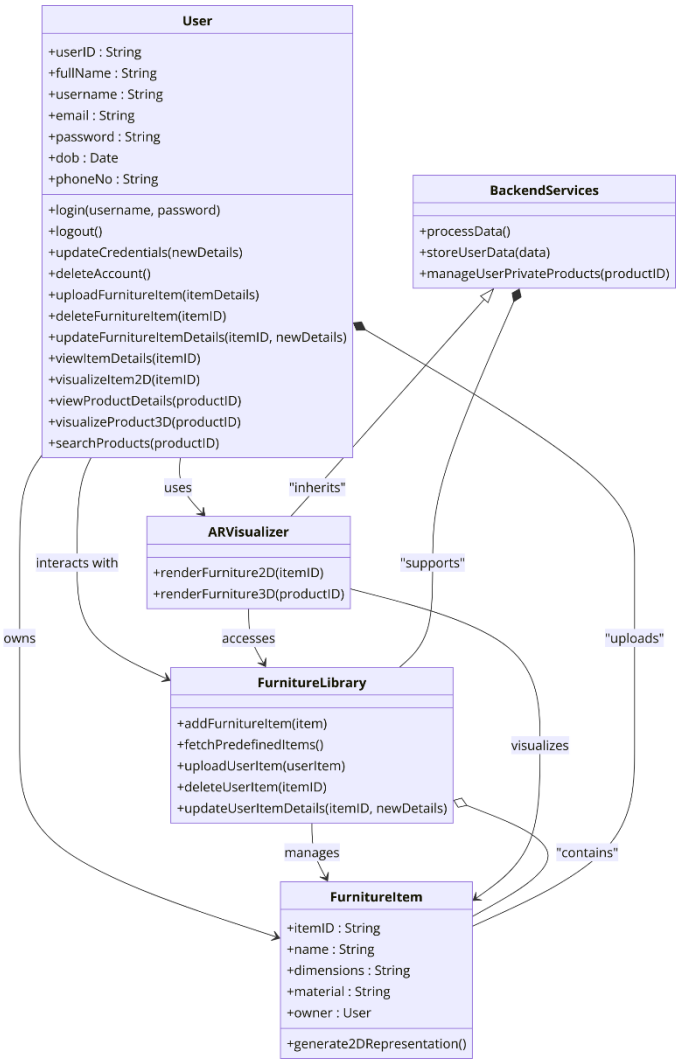

Figure 3.1: Class Diagram of AR Visualizer

3.4.2 ERD Diagram

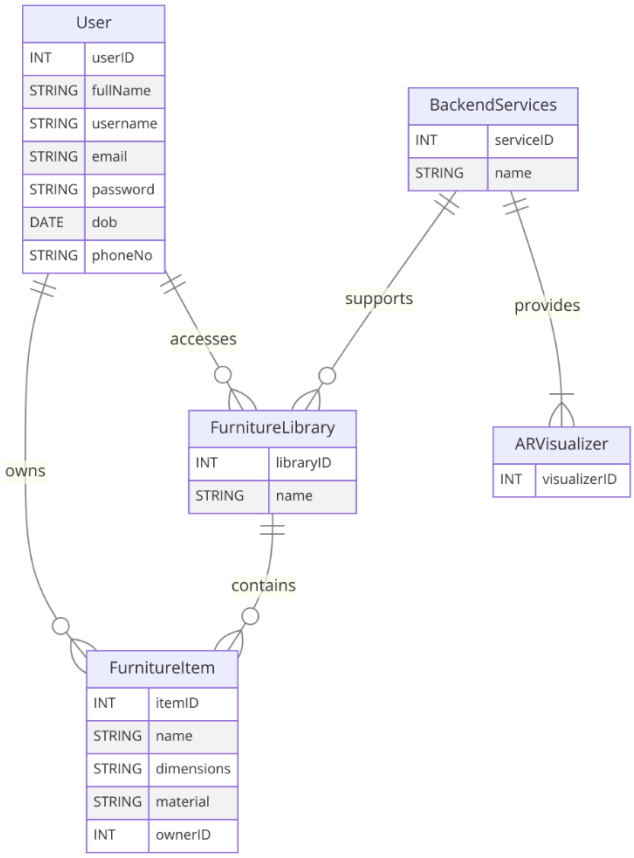

Figure 3.2: Entity Relationship Diagram of AR Visualizer

3.4.3 DataFlow Diagram

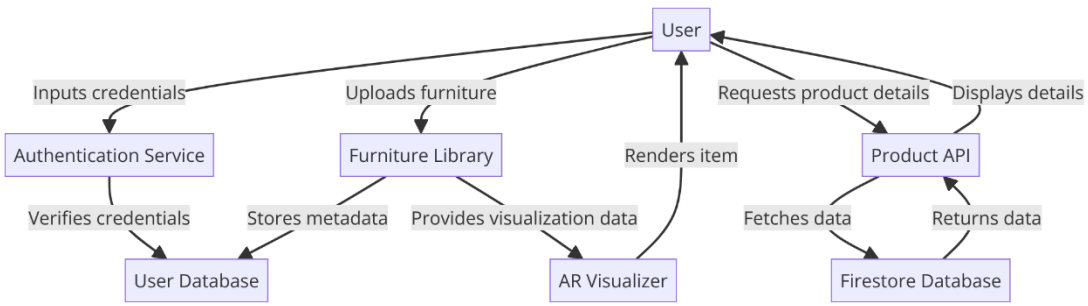

Figure 3.3: Data Flow Diagram of AR Visualizer

3.4.4 Use Case Diagram

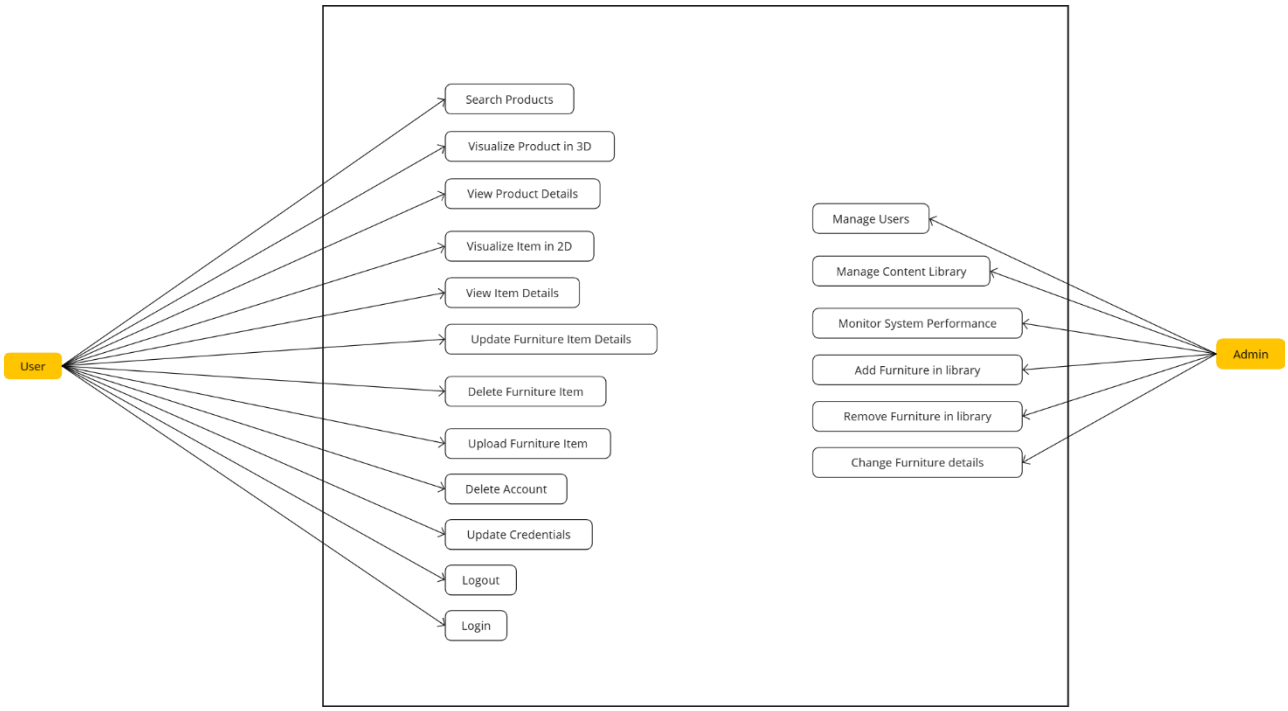

Figure 3.4: Usecase Diagram of AR Visualizer

3.4.5 System Sequence Diagram

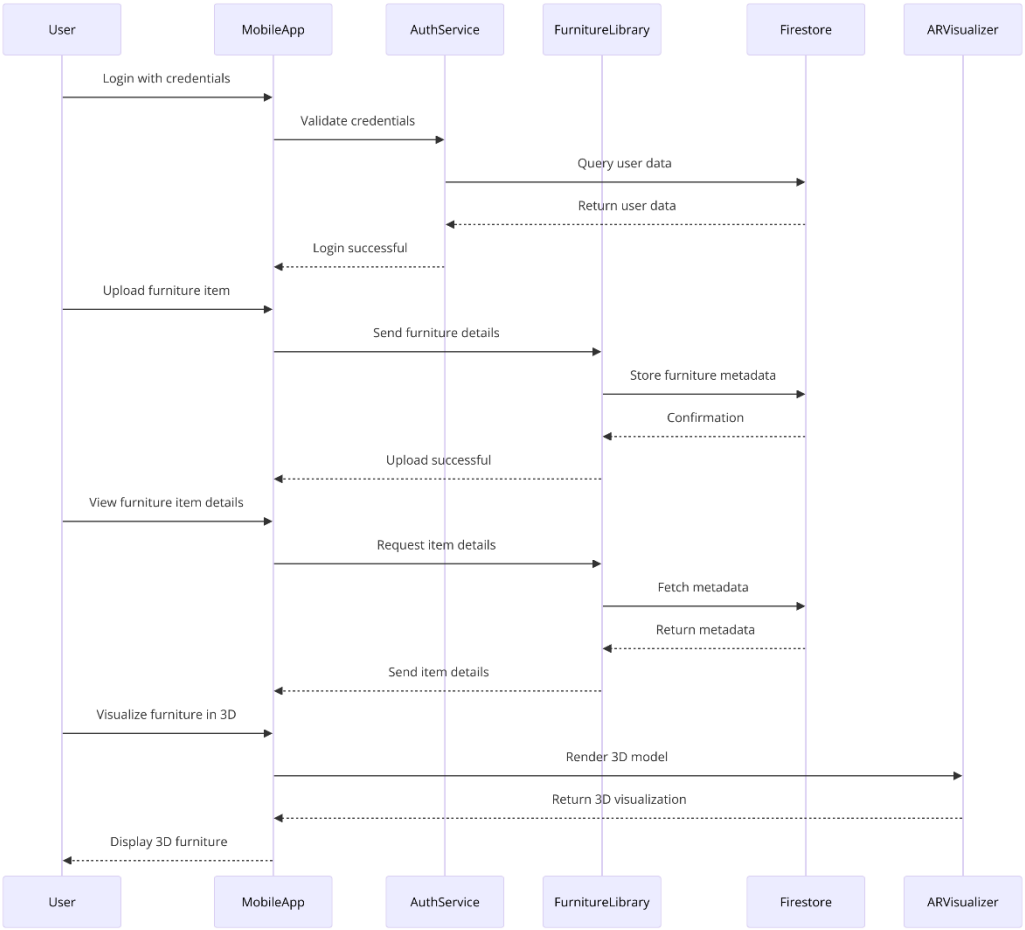

Figure 3.5: System Sequence Diagram of AR Visualizer

CHAPTER 4

IMPLMENTATION

The implementation phase of the AR Visualizer project involved translating the system's design into a fully functional application. The project integrates Flutter for building a cross-platform mobile app, Firebase for backend services, and ARCore/ARKit for providing Augmented Reality (AR) functionalities. The primary goal was to implement a seamless interior design experience that allows users to place, resize, and integrate custom objects into an AR environment. This chapter discusses the step-by-step development of the application and provides detailed insights into the key features implemented.

4.1 Flutter Mobile Application

The **Flutter mobile application** forms the foundation of the AR Visualizer, offering a single codebase for both Android and iOS devices. Flutter's widget-based framework facilitated the development of a responsive and interactive user interface.

4.1.1 User Interface Implementation

Most of the work for the UI was based on ease of use and simple navigation, which makes the application very intuitive to use. The Flutter widgets applied to build interactive components included: GridView and Stack:

- **Combined library for predefined furniture and decor items** are available scrollable grid format.
- **Dynamic slider** with which the natural resizing of object will be possible while adjusting on things in real time.

- **A file upload button** allows users to upload images of custom furniture or decorations with their description.

Animative transitions were incorporated using Flutter's AnimatedContainer and PageRouteBuilder; hence, smooth navigation was created to improve the user experience.

4.1.2 Cross-Platform Compatibility

Flutter's cross-platform capabilities ensure that this application behaved identically on both Android and iOS devices. Thus, the ar_flutter_plugin package was used to integrate Flutter with ARCore and ARKit, enabling the functionality to exist for both platforms instead of being implemented individually for each platform.

4.2 AR Functionality Integration

The heart of the AR Visualizer is its very AR capability with which users may interact with virtual objects in the real environment. This included surface detection, object placement, and rescaling by integrating ARCore and ARKit.

4.2.1 Surface Detection and Object Placement

Both the ARCore for Android and the ARKit for iOS allowed the user to understand horizontal planes like floors, tables, etc. with virtual object placing as the anchor. With the help of the ar_flutter_plugin, a placeObject() function was created to fetch the desired object on these surfaces. It ensures spatial coherence of the object by anchoring users to the same spot even as they move around.

4.2.2 Object Resizing

The object resize feature adds a whole new level to the ability for users to resize virtual objects to fit accurately into their own physical spaces. This was linked with the AR functionality through Flutter's state management using a slider component.

As the user moves the slider, the scale of the currently selected object is dynamically updated to ensure keeping proportions at play. For example:

- Where the slider changes from left to right, the object size increases in all dimensions.
- As the slider is moved, real-time feedback is given by updating the object live in the AR environment.

4.2.3 Custom Object Integration

Users can upload images showing how their furniture or decor would look in their homes, further enhancing the personalized experience. Firebase Storage uploads and processes those images for display as 2D representations within the AR setting. With this addition, versatility is introduced to the app, allowing users to visualize custom designs in their places.

4.3 Backend Implementation

Firebase served as the backend for the AR Visualizer, providing authentication, data storage, and real-time database functionality.

4.3.1 User Authentication

Firebase Authentication was implemented to manage user accounts securely. The login and sign-up flows were developed using Flutter's firebase_auth package. This ensures that each user's data is stored securely and accessed only by authenticated accounts. Authentication tokens generated by Firebase allowed seamless session management.

4.3.2 Data Storage

Cloud Firestore and Firebase Storage were used to handle user and object data:

- **Cloud Firestore**: Stores metadata for predefined objects, including names, dimensions, and associated 3D models. It also stores user preferences and interaction logs.

- **Firebase Storage**: Manages the uploaded custom object images. When an image is uploaded, a downloadable URL is generated and linked to the user account, enabling quick retrieval during AR rendering.

The data flow between the front end and backend was optimized to ensure minimal latency, providing users with smooth experience even during data-intensive operations.

4.4 Key Features Implemented

The AR Visualizer includes several innovative features designed to enhance the user experience. Below is a detailed explanation of the key features implemented in the project:

4.4.1 Predefined Object Placement

This feature allows users to select furniture or decor items from a library of predefined objects and place them in their real-world environment. The object library was designed as a scrollable grid, enabling users to browse items easily. Once an object is selected, it is rendered in the AR environment using ARCore/ARKit, aligned with detected surfaces.

4.4.2 Real-Time Object Resizing

Users can resize the virtual objects with dynamic scale transformations. The able use of slider on the UI allows them to scale objects to match their dimensions in real space by scaling them up or down. This feature is particularly significant in visualizing how large a piece of furniture will be placed within a given room. The resizing function keeps the aspect ratio of the object, so it is proportional in all dimensions.

4.4.3 Custom Object Integration

There is a provision that users can also upload their own images of furniture or objects to personalize the design experience. The uploaded images are processed into Firebase Storage. When processed, they will appear in the object library and can then be placed in the AR environment just like any predefined objects. This feature allows users to experiment with things that already exist within their homes, closing the gap between the virtual and the real.

4.4.4 Cross-Platform Functionality

The application is fully compatible with both Android and iOS devices, thanks to Flutter's cross-platform capabilities. The integration of ARCore for Android and ARKit for iOS ensures that the AR features work seamlessly across platforms, providing a consistent experience for all users.

4.4.5 Intuitive User Interface

The user interface was designed with simplicity in mind to ensure that even non-technical users could navigate the application easily. Key UI elements include:

- A grid-based object library for browsing furniture and decor items.
- A slider for resizing objects dynamically.
- A drag-and-drop interface for placing objects in the AR environment. This intuitive design reduces the learning curve and makes the application accessible to a broader audience.

4.5 Testing and Validation

Rigorous testing was conducted to ensure that the application met functional and performance requirements. The testing process included:

- **Unit Testing:** Individual components, such as the resizing slider and object placement functions, were tested in isolation to verify their accuracy.
- **Integration Testing:** The communication between the frontend, backend, and AR frameworks was tested to ensure seamless functionality.
- **User Testing:** Feedback was collected from a group of target users, including homeowners and designers, to refine the app's usability and features.
- **Performance Testing:** The app's responsiveness and rendering speed were evaluated under various conditions, such as placing multiple objects simultaneously.

4.6 Tools and Technologies

The development of the AR Visualizer application relied on a variety of tools, technologies, and packages to ensure seamless implementation and delivery of the desired functionalities. Below is a detailed explanation of the primary tools and technologies used:

4.6.1 Flutter

The main framework for building the AR Visualizer application has been Flutter, which can function across Android and iOS devices. The widget-driven nature of Flutter made it possible to develop a responsive and interactive user interface. For components like the object library, resizing slider, and file upload button, such features as state management and customizable widgets were widely incorporated by Flutter developers.

4.6.2 Firebase

Firebase was integrated into the project to provide backend services, ensuring reliable data storage and user management. The following Firebase services were utilized:

- **Firebase Authentication**: To manage user login and sign-up functionalities, ensuring that each user's data was securely stored and accessed.

- **Cloud Firestore**: A NoSQL database used to store metadata for predefined objects, user preferences, and interaction logs. It allowed for real-time data synchronization, ensuring responsive user experience.
- **Firebase Storage**: Used for managing user-uploaded custom object images. The storage service provided secure and scalable cloud storage, enabling users to upload images and retrieve them during AR rendering.

4.6.3 augmented_reality_plugin

The AR features of the application were implemented by using the augmented_reality_plugin package. This plugin was used in array display of live camera preview, sources to place objects onto the surface and customize its size, position, and orientation. Some of the key features of this plugin were defined as follows:

- Live camera feed allowed the users to see their objects in their physical spaces.
- Positioning of 2D objects (PNG or JPG images) on the display in a specific place.
- Resize, drag, rotate, zoom in, and zoom out objects could be done dynamically by the user.

This plugin formed the basis for basic AR interaction in the application.

4.6.4 model_viewer_plus

This package **model_viewer_plus** supported rendering and interaction with 3D models converted in glTF-and GLB formats. It therefore allowed the application to perform the following:

- Provide rendering interactive 3D models, even animated ones.
- Allow automatic rotation or launching into a native AR viewer through ARCore or Scene Viewer on Android and through USDZ on iOS.

- Provide fallback functionality to 3D mode if ARCore services were not available, so as to guarantee graceful user experience.
- Support for custom configurations like background color and auto-rotation delay for 3D models.

This is how you couple this suite with augmented_reality_plugin, thus giving this application a push towards emerging AR features for 2D and 3D assets making it versatile and robust in terms of usability.

4.6.5 ARCore and ARKit

Advanced tracking capabilities were achieved using ARCore and ARKit enabling the application to detect surfaces, place objects, and maintain spatial consistency through Flutter plugins. These frameworks were integrated into the application to allow rendering objects precisely as they would appear within their intended environments.

4.6.6 Cloud Firestore

The primary database that was used in the application was Cloud Firestore, and it stored the following:

- Predefined object data including names, dimensions, and any associated images or 3D models.
- User preference and customization information.
- Metadata regarding uploaded custom objects. Its capacity thus for real-time sync made for fluid and responsive interaction between the front end and back end.

4.6.7 Firebase Storage

Firebase Storage was employed to manage custom object images uploaded by users. Uploaded files were securely stored in the cloud, with generated URLs allowing for

easy retrieval during AR interactions. Firebase Storage's scalability ensured that the system could handle multiple user uploads efficiently.

4.6.8 Figma

Figma was used for designing the application's user interface and creating interactive prototypes. The designs ensured that the UI was both functional and visually appealing, with clear layouts and smooth navigation for features like object selection, resizing, and file uploading.

4.6.9 Dart

Dart, the programming language used with Flutter, played a critical role in implementing the application's logic and state management. It was used to:

- Control UI components, such as the resizing slider and file upload button.
- Manage communication between the frontend and backend using Firebase APIs.
- Implement dynamic interactions, such as real-time resizing and object placement.

CHAPTER 5

USER MANUAL

5.1 User Manual

5.1.1 Splash Screen

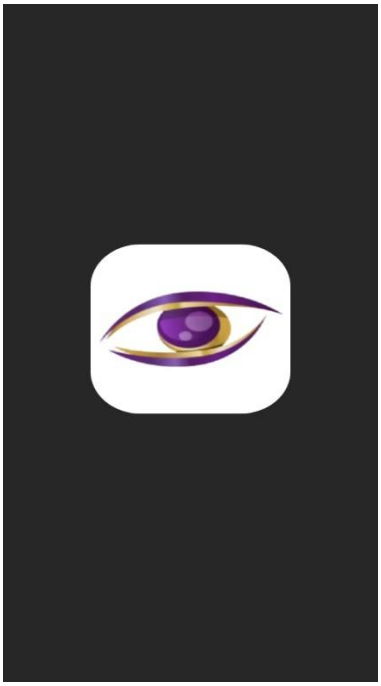

Figure 5.1: Splash Screen showing logo of AR Visualizer

Splash Screen showcase the logo of the AR Visualizer

5.1.2 Onboarding Screens

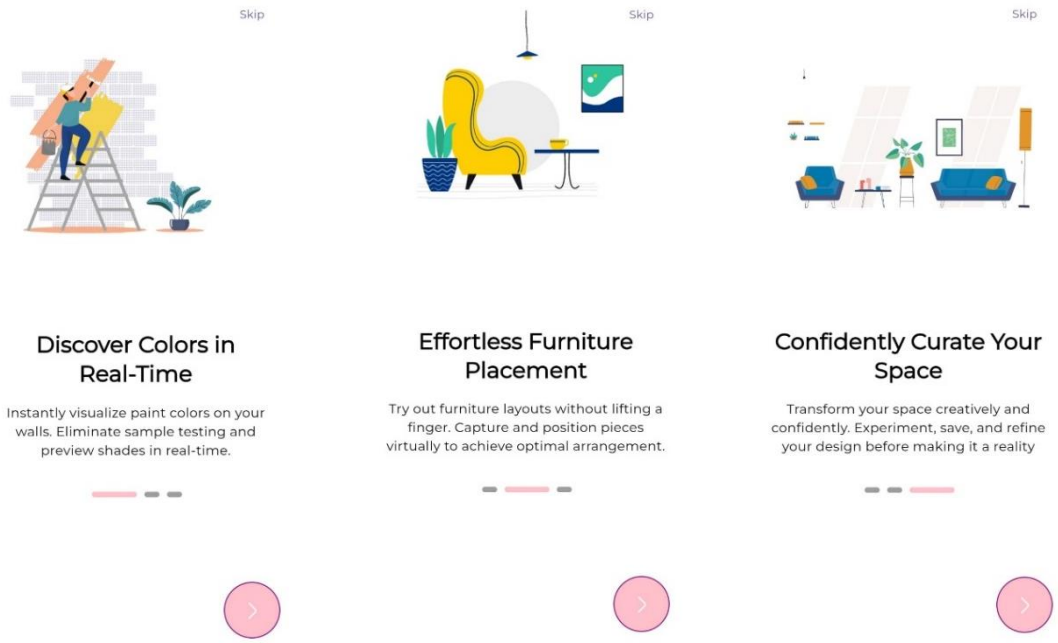

Figure 5.2: Onboarding Screens

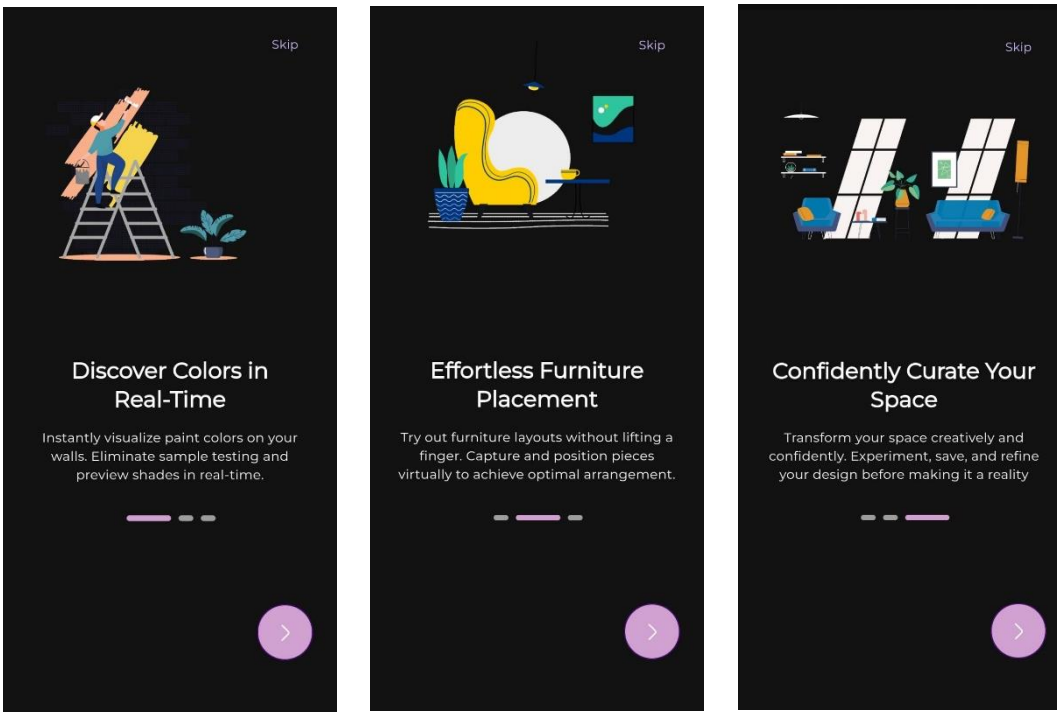

Figure 5.3: Onboarding Screens (Dark Mode)

Onboarding Screens tends to give the familiarity with the app and let the user know what to expect with the application. It also explains what you can do or not with the app.

5.1.3 Welcome Screens

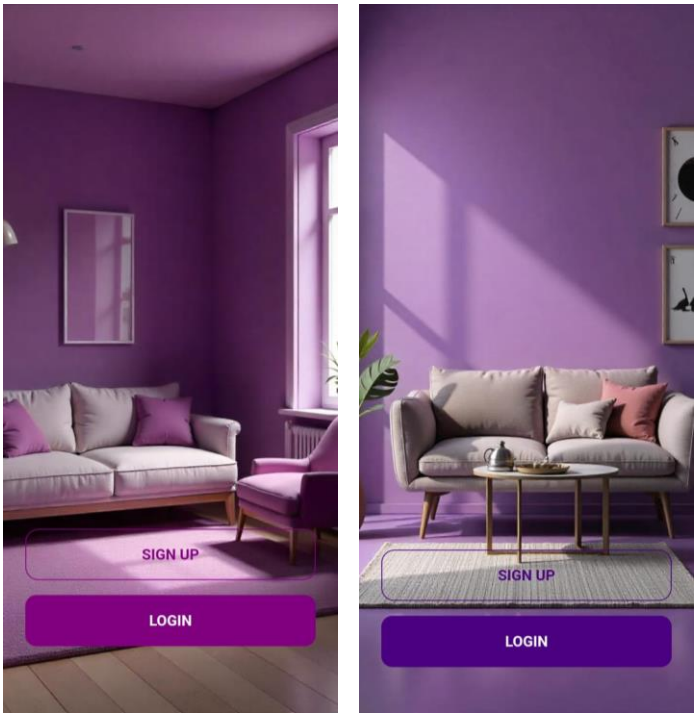

Figure 5.4: Welcome Screen in Light Mode (Left) & Dark Mode (Right)

Welcome Screen welcomes the users to the application and show them two buttons to either LogIn or SignUp using the app.

5.1.4 Log In Screen:

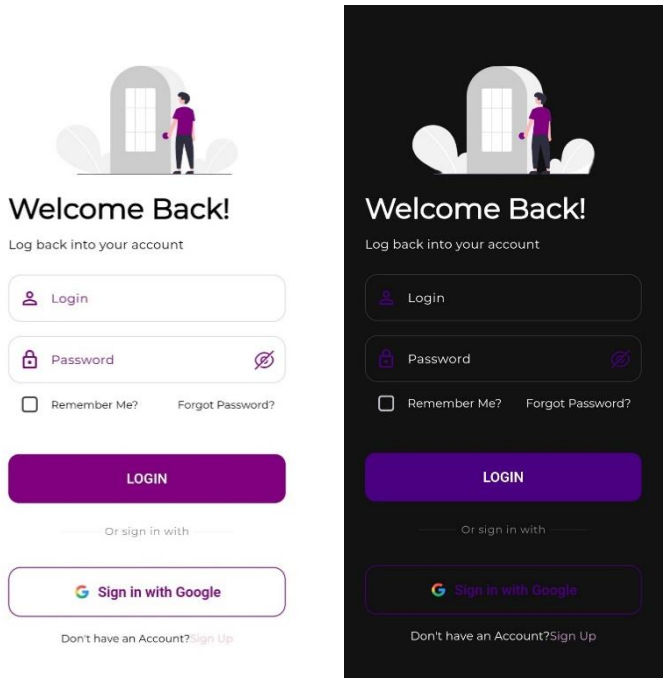

Figure 5.5: Sign-in Screens in Light Mode (Left) & Dark Mode (Right)

Sign In let the users to sign into the app using the credentials such as user names and password, even the users can login through their google account.

5.1.5 Sign Up Screen:

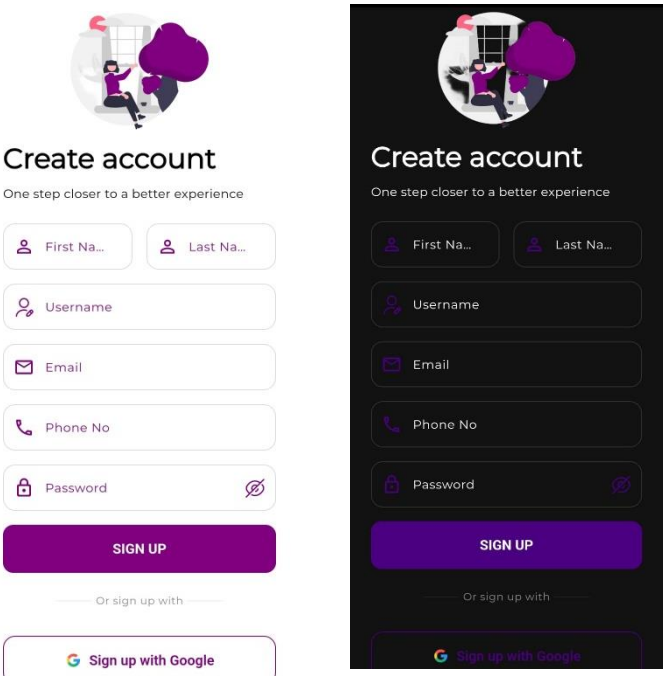

Figure 5.6: Sign-in Screens in Light Mode (Left) & Dark Mode (Right)

Sign Up let the users create an account on the app using the credentials such as name, username, email and password etc, even the users can sign up through their google account.

5.1.6 Home Screen:

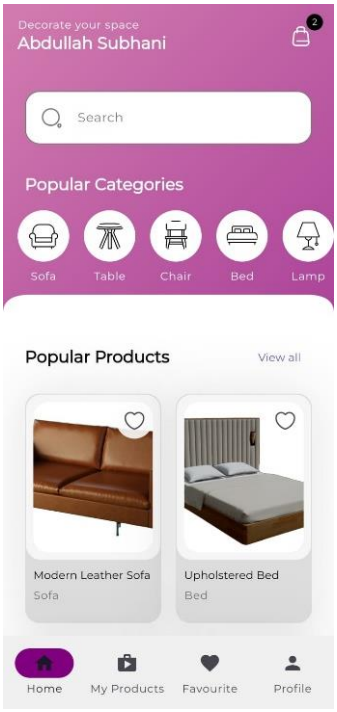

Figure 5.7: Home Screen of AR Visualizer

The Home Screen of AR Visualizer is simple, it shows the popular categories from which you can select your favourite furniture. However, it also shows the popular products that are liked by the users. There is a navigation bar at the end of the page that allows users to navigate to other pages like My Products, Favourite and Profile. The user can also search for any specific item using the search bar.

5.1.7 My Products:

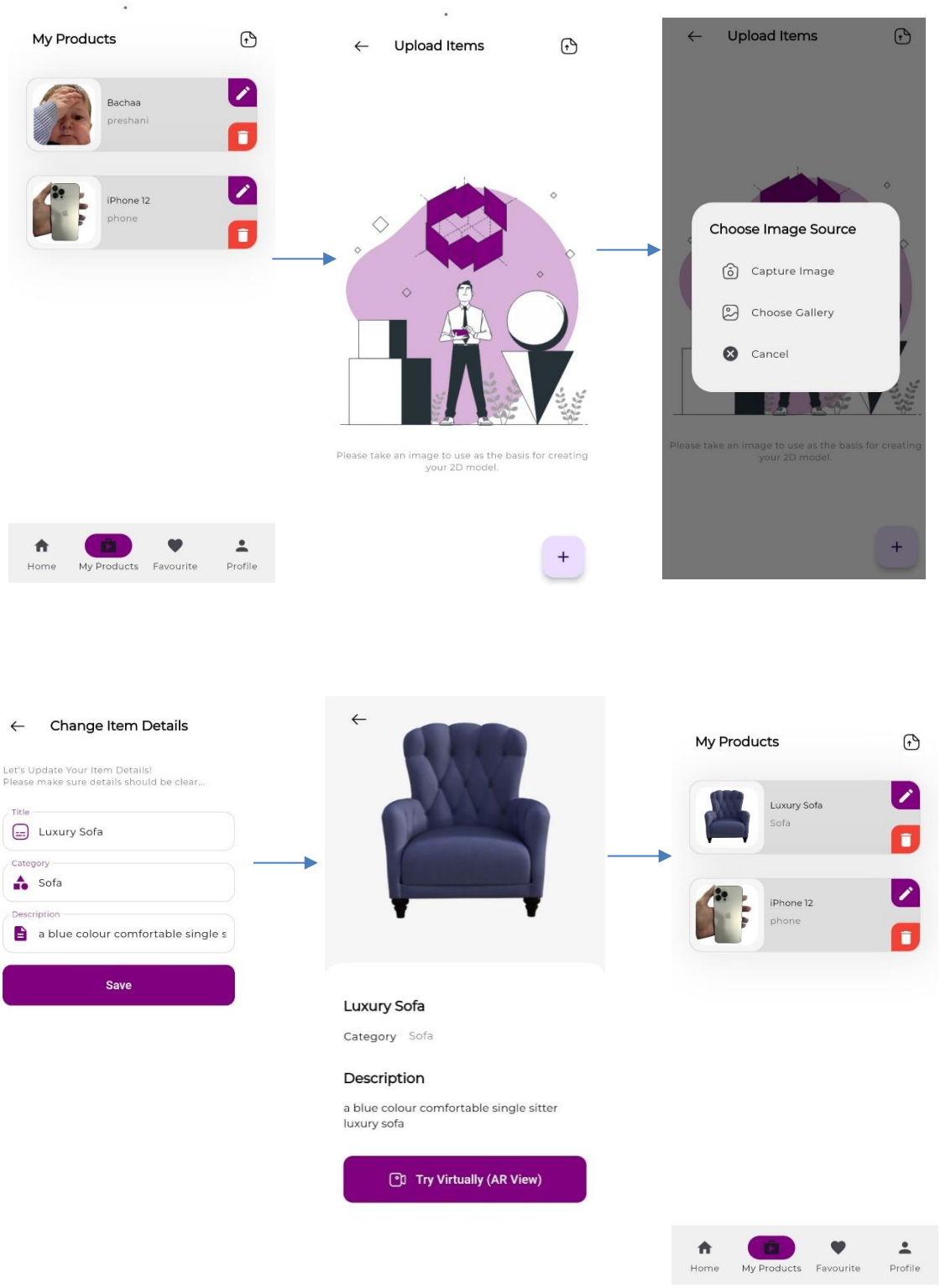

Figure 5.8: My Product Pages

My Products page contains the private products of the user, that the user upload via their camera or from their device gallery. By clicking on the top right icon on my product page, it navigates the user to another page where at the left bottom corner of screen 2, there is a plus icon. By clicking on that icon user can upload their picture. Then the picture is processed and is converted into a 2D object which the user can later visualize. After the product is added to the My Products, the user can visualize it any time soon or can also delete from their products. These personalized products are not visible to any other users.

5.1.8 Favourites:

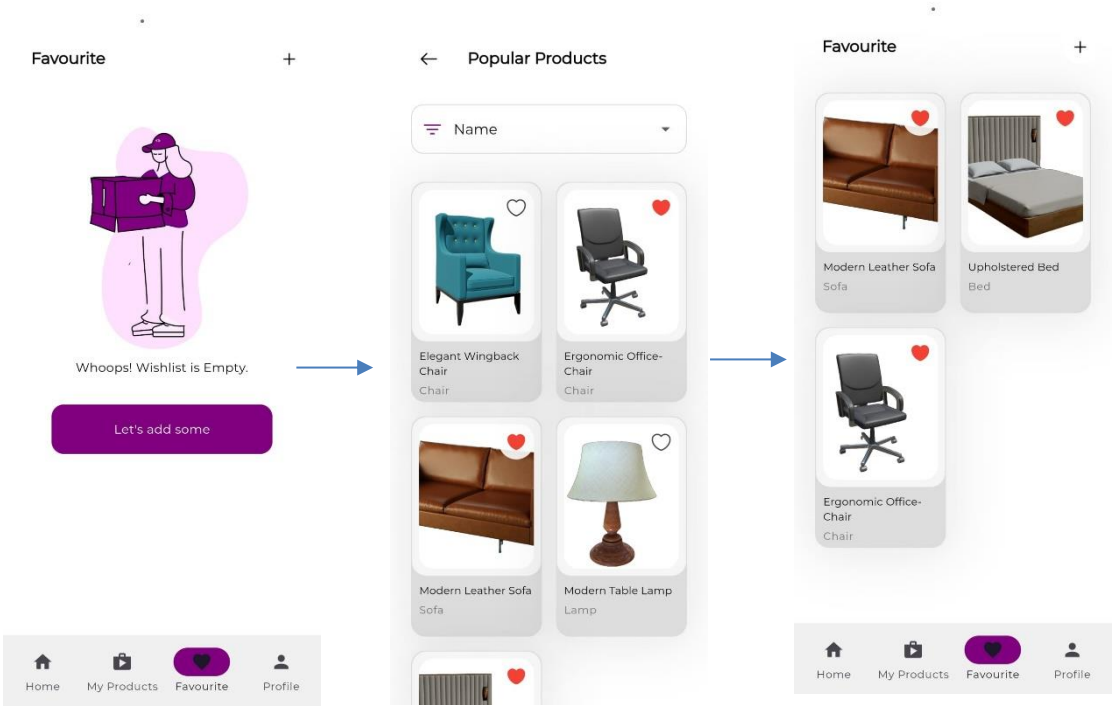

Figure 5.9: Favourite Pages

Favourite page contains the liked products by the user. By clicking on the top right heart icon on the product icon, it adds the product to the Favourite Page of the user. Each user would have different favourite products that they liked. Again, clicking on the filled heart icon, the product can be removed from the favourites of the users.

5.1.9 Profile:

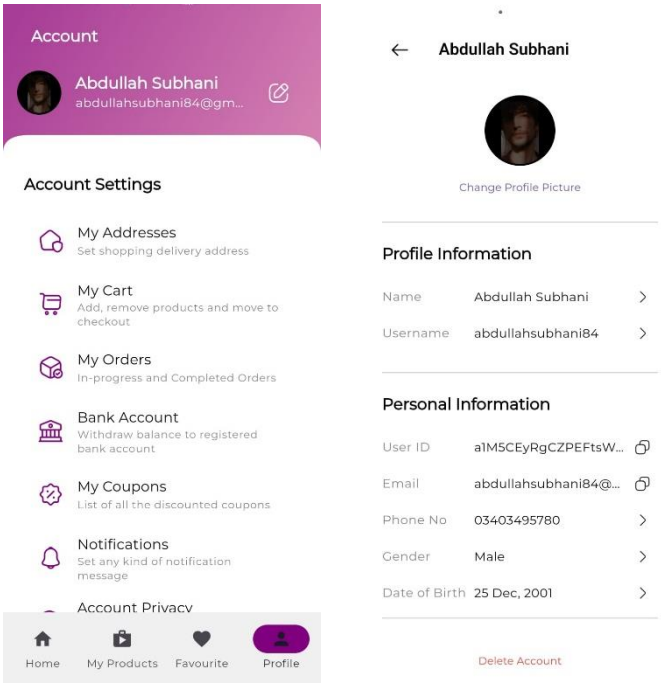

Figure 5.10: Profile Page

The AR Visualizer App also allows the users to users to manage their profiles by the profile page from which the user can edit their information by clicking on the top right icon next to the username and email address. On clicking, the icon will navigate the user to the second screen where the user can then update any of their information.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Project Achievements

The AR Visualizer has passed the development of several milestones, emerging as a very innovative and possibly practical solution to interior design problems. One of the most significant results, was the successful adoption of Flutter, a framework that permitted the creation of a unified cross-platform codebase; thus, the application runs seamlessly on Android and iOS without creating different developments specific to either system or broadens its accessibility.

The establishment of ARCore and ARKit frameworks is equally significant as advanced capabilities in augmented reality for the application. With these technologies, users can do real-time activities such as surface detection, placing virtual objects, and scaling them, with high precision. Such collected features allow users to visualize digital furniture and in-room decor elements and give them realistic experience of bringing the digitally created design into their actual space. The project has incorporated advanced tools, which demonstrate its ability to solve some of the major problems of interior design with the use of advanced technologies.

A major advantage of the AR Visualizer is that it features the possibility of integrating custom objects. For this, users uploaded pictures of their furniture or any special items that then were used to make 2D or 3D objects to be used in the AR space. Such self-inclusion of personal belongings into the design process was a very high level of personalization in the application making it meaningful in the digital and physical realms.

Keeping with the project requirements, the last part would be that the interface should be as accessible and easy to use as possible; thus, features like an interactive resizing slider, drag-and-drop object placement, and an intuitive object library were dedicatedly created and meant to ensure smooth user experience. With all these combined, this application can be accessed even by completely non-technical people.

These combined achievements show that this AR Visualizer has met all its objectives and illustrates how such applications of augmented realities enrich the field of interior design. By marrying innovation with user-oriented design principles, such a project is likely to become a strong statement on how AR technologies can radically change the existing approaches to space planning and decoration.

6.2 Implementation Issues and Challenges

6.2.1 Integration of ARCore and ARKit

- Bridging Flutter with platform-specific AR frameworks like ARCore and ARKit require the use of plugins (e.g., ar_flutter_plugin).
- Compatibility issues arose due to differences in device hardware and software environments, requiring extensive debugging.

6.2.2 Surface Detection Accuracy

- Surface detection by ARCore and ARKit was occasionally inaccurate, especially in low-light conditions or on older devices with limited AR hardware capabilities.
- This affected object placement and reduced the user experience on certain devices.

6.2.3 Custom Object Upload and Processing

- Uploading custom furniture images require backend optimization to handle large image files efficiently.

- Processing these images for AR rendering often introduced latency, particularly on lower-end devices or when network speeds were slow.

6.2.4 Device Compatibility Issues

- Variations in AR performance across Android and iOS devices created inconsistencies in features like surface tracking and rendering.
- Additional testing and adjustments were needed to ensure uniform functionality across devices.

6.2.5 Performance Optimization

- Rendering multiple objects simultaneously in AR environments caused performance issues, particularly on devices with limited hardware resources.
- Optimizations, such as reducing texture sizes and simplifying object models, were necessary to maintain smooth performance.

6.3 Future Work

6.3.1 Expanded 3D Model Support

- Support additional 3D file formats, such as OBJ and FBX, to increase compatibility.
- Provide a built-in editor for modifying 3D models directly within the application (e.g., adjusting colors, textures, or dimensions).

6.3.2 AI-Powered Design Tools

- Integrate AI to recommend furniture arrangements or decor based on room dimensions and user preferences.
- Implement automatic scaling algorithms to fit objects proportionally within the user’s available space.

6.3.3 Offline Functionality

- Add offline capabilities to allow users to access predefined object libraries and previously downloaded custom objects without an internet connection.
- Ensure smooth usability in environments with limited connectivity.

6.3.4 Collaboration Features

- Enable real-time collaboration, allowing multiple users to work on the same design project simultaneously.
- Add functionality for sharing designs directly with clients or collaborators for feedback.

6.3.5 AR Wearable Support

- Expand the application’s compatibility to support wearable AR devices such as HoloLens or Magic Leap.
- Enable users to interact with objects more naturally through gestures and immersive displays.

6.3.6 Improved Performance Optimization

- Further optimize rendering processes to ensure smooth performance, even on devices with limited hardware capabilities.
- Reduce memory and processing requirements to make the application accessible to a broader range of users.

REFERENCES

[1] PlugXR, “Augmented Reality in Interior Design - Uses, Benefits & Examples.” [Online]. Available: https://www.plugxr.com/augmented-reality/augmented-reality-interior-design/.

[2] H. Trivedi, “A Survey on Augmented Reality and its Application in the Field of Interior Design,” Journal of X University, vol. 14, p. 4321, 2020, doi: 10.37896/jxu14.5/474.

[3] R. Azuma et al., “A Survey of Augmented Reality Applications in Interior Design,” AR Journal, vol. 15, no. 4, pp. 12–22, 2021.

[4] Iflexion, “The Rise of Augmented Reality in Interior Design and Furniture Retail.” [Online]. Available: https://www.iflexion.com/blog/augmented-reality-interior-design/.

[5] SketchUp, “Features and Benefits.” [Online]. Available: https://www.sketchup.com/features/.

[6] MagicPlan, “MagicPlan AR Features.” [Online]. Available: https://magicplan.app/.

[7] A. Labrie and J. Cheng, “Adapting Nielsen's Usability Heuristics to the Context of Mobile Augmented Reality,” arXiv preprint, 2020. [Online]. Available: https://arxiv.org/abs/2008.03174.

[8] Y. Park et al., “User Experience Design for Mobile Augmented Reality Applications,” UX Journal, vol. 20, no. 3, pp. 45–55, 2020.