ALTERIOR-AR Based Interior Design

A Minor Project Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

Hardik Baser(EN22CS301380)
Subh Labh Shrivastava(EN22CS304062)
Mahima Malviya(EN22CS304041)

Under the Guidance of
Prof.Shruti Sharma

Department of Computer Science & Engineering
Faculty of Engineering
MEDICAPS UNIVERSITY, INDORE- 453331

APRIL-2025

ALTERIOR-AR Based Interior Design

A Minor Project Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

Hardik Baser(EN22CS301380)
Subh Labh Shrivastava(EN22CS304062)
Mahima Malviya(EN22CS304041)

Under the Guidance of
Prof.Shruti Sharma

Department of Computer Science & Engineering
Faculty of Engineering
MEDICAPS UNIVERSITY, INDORE- 453331

APRIL-2025

Table of Contents

		Page No.
	Report Approval	4
	Declaration	5
	Certificate	6
	Acknowledgement	7
	Abstract	8-9
	Table of Contents	9
	List of figures	10
	Abbreviations, Notations & Symbols	11
Chapter 1	Introduction (Whichever is applicable)	12
	1.1 Introduction	13
	1.2 Literature Review	14
	1.3 Objectives	15
	1.4 Significance	16
	1.5 Research Design	17
	1.6 Source of Data	18
Chapter 2	REQUIREMENTS SPECIFICATION	19
	2.1 User Characteristics	19
	2.2 Functional Requirements	19
	2.3 Dependencies	20
	2.4 Performance Requirements	20
	2.5 Hardware Requirements	21
	2.6 Constraints & Assumptions	21
Chapter 3	DESIGN (Whichever is applicable)	22
	3.3.1 Activity Diagram	22
	3.3.2 Flow Chart	23
	3.3.3 Sequence diagram	24
Chapter 4	Implementation, Testing, and Maintenance	25
	4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation	25
	4.2 Testing Techniques and Test Plans (According to project)	26
	4.3 Installation Instructions	27
	4.4 End User Instructions	28

Chapter 5	Results and Discussions	29
	5.1 User Interface Representation (of Respective Project)	30
	5.2 Brief Description of Various Modules of the system	30
	5.3 Snapshots of system with brief detail of each	30
Chapter 6	Summary and Conclusions	32-33
Chapter 7	Future scope	34
	Appendix	37
	Bibliography	38

Report Approval

The project work **"ALTERIOR-AR Based Interior Design"** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approve any statement made, opinion expressed, or conclusion drawn there in; but approve the "Project Report" only for the purpose for which it has been submitted.

Internal Examiner	External Examiner
Name:	Name:
Designation	Designation
Affiliation	Affiliation

<u>Declaration</u>

I/We hereby declare that the project entitled **"ALTERIOR-AR Based Interior Design"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications in 'Computer Science Engineering' completed under the supervision of **'Shruti Sharma, Assistant Professer, Cse',** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Signature and name of the student(s) with date

Certificate

I/We, **Prof. Shruti Sharma** certify that the project entitled **"ALTERIOR-AR Based Interior Design"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications by **"Hardik Baser, Shubh Labh Shrivastava, Mahima Malviya"** is the record carried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Prof.Shruti Sharma	Dr. Ratnesh Litoriya
Computer Science Engineering	Head of the Department
Medi-Caps University, Indore	Computer Science & Engineering
	Medi-Caps University, Indore

Acknowledgements

I would like to express my deepest gratitude to the Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) D. K. Patnaik,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) Pramod S. Nair,** Dean, Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Ratnesh Litoriya** for his continuous encouragement for the betterment of the project.

It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Hardik Baser, Shubh Labh Shrivastava, Mahima Malviya

B.Tech. III Year (A)
Department of Computer Science & Engineering
Faculty of Engineering
Medi-Caps University, Indore

Abstract

The rapid advancement of Augmented Reality (AR) technology has revolutionized various industries, including interior design. This project presents an AR-based Interior Design Application that allows users to visualize and customize furniture, décor, and layouts in real-time within their physical space. The application leverages ARCore/ARKit for environment tracking and 3D rendering, enabling users to place virtual objects with high accuracy. Key features include real-time preview, drag-and-drop functionality, material customization, and AI-driven design recommendations.

The system follows a client-server architecture, where a mobile app interacts with a cloud-based database storing 3D models and user preferences. Usability testing indicates that the app enhances user engagement, reduces design uncertainty, and improves decision-making in home décor. Challenges such as occlusion handling, lighting adaptation, and performance optimization are addressed through iterative development.

This project demonstrates the potential of AR in democratizing interior design, offering an immersive, user-friendly, and cost-effective solution for homeowners and professionals alike. Future enhancements may include multi-user collaboration, VR integration, and advanced AI styling assistants.

Keywords:

- **Augmented reality SDK-** Software development kits for building AR applications.

- **ARCore (Android) interior design-** Google's platform for developing AR interior design apps on Android.

- **Unity AR Foundation-** A Unity framework for cross-platform AR development (supports ARKit & ARCore).

- **Blender 3D modelling-** Free, open-source 3D modeling software for creating AR assets.

- **Flutter AR plugins-** Flutter packages enabling AR functionality in cross-platform mobile apps.

- **Mobile AR Development-** Creating AR applications for smartphones and tablets.

- **Surface Detection-** AR feature that identifies and interacts with real-world surfaces.

- **Plane Tracking-** AR technology that detects and tracks flat surfaces like floors or tables.

List of Figures

Fig. No.	**Title**
Fig 1.1	**AR Workflow Architecture** (User → Flutter UI → Unity Engine → ARCore → Scene)
Fig 1.2	**System Architecture Diagram** (Client-App, Firebase, 3D Model DB, Unity Backend)
Fig 2.1	**Use Case Diagram** (Homeowner, Designer, Retailer interactions with system)
Fig 3.1	Data Flow Diagram – Level 0 (Context Level)
Fig 3.2	Data Flow Diagram – Level 1 (Furniture Placement)
Fig 3.3	Activity Diagram – AR Session Start and Object Placement
Fig 3.4	Flow Chart – Furniture Selection and Placement Logic
Fig 3.7	Sequence Diagram – Tap to Place Workflow
\Fig 5.1	Snapshots of Furniture Placement in Real Environment

Abbreviations

- **AR** – Augmented Reality
- **SDK** – Software Development Kit
- **UI** – User Interface
- **ARKit** – Apple's AR framework for iOS
- **ARCore** – Google's AR framework for Android
- **ARF** – AR Foundation (Unity)
- **UV Mapping** – Texture mapping technique (used in Blender)
- **Flutter** – Google's UI toolkit for cross-platform apps (iOS/Android)
- **Dart** – Programming language used in Flutter
- **Widget** – UI component in Flutter
- **Unity** – Cross-platform game engine (used with AR Foundation)

Notations & Symbols

- **SLAM** – Simultaneous Localization and Mapping (tracking algorithm)
- **6DoF** – Six Degrees of Freedom (AR object movement)
- **PoI** – Points of Interest (in surface detection)
- **ARAnchor** – AR object anchor point (ARKit/ARCore)
- **Raycast** – Method for detecting surfaces (e.g., *Raycast from screen touch*)
- **GameObject** – Unity's base object class (e.g., *3D model as GameObject*)
- **Transform** – Position/Rotation/Scale in Unity (e.g., *Transform.Position.X*)
- **Shader** – Graphics material effect (e.g., *Shader: Standard Specular*)
- **Blender Units** – Default scale in Blender (1 unit = 1 meter in Unity)

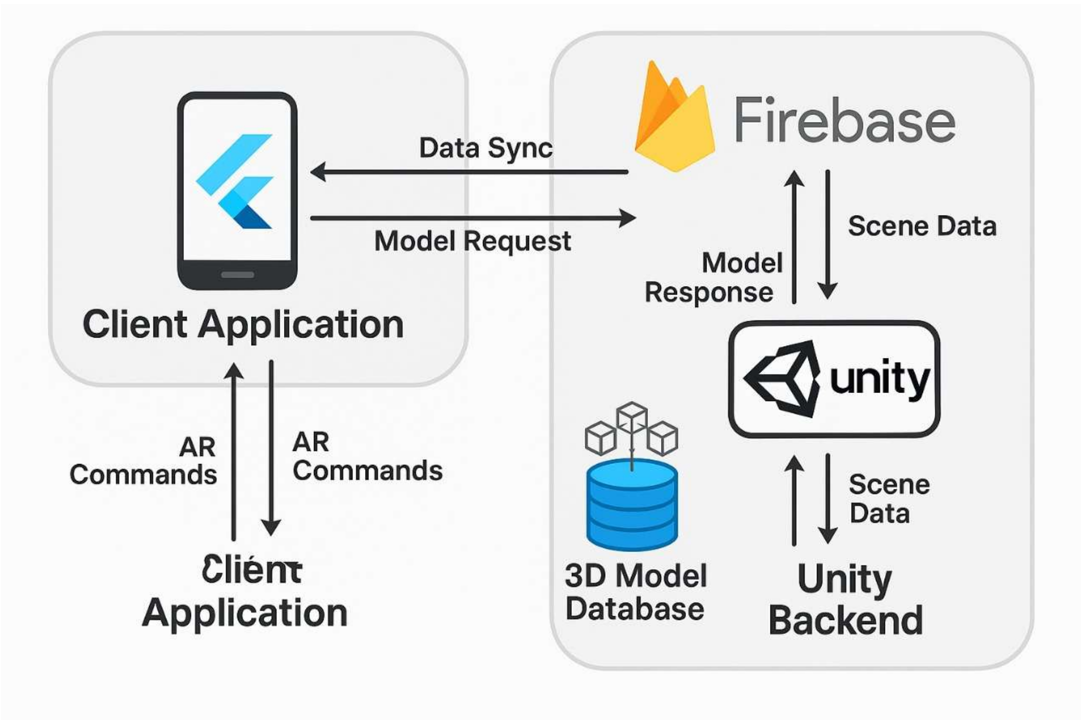
Chapter 1: Introduction

1.1 Introduction

Augmented Reality (AR) is redefining digital interaction by blending real-world environments with computer-generated content. In domains like **interior design**, AR has enabled users to visualize and modify virtual furniture, lighting, and decor in physical spaces with enhanced realism and spatial awareness. This transition from imagination to visualization streamlines the decision-making process and minimizes design uncertainty.

The proposed project, **ALTERIOR – AR Based Interior Design**, is a cross-platform mobile application that empowers users to preview and configure their living spaces through real-time **3D object placement** using **SLAM** (Simultaneous Localization and Mapping) and **6DoF** object manipulation.

Built using **Unity** (with **AR Foundation** and **Vuforia SDK**) and a responsive **Flutter** frontend (developed in **Dart**), the application integrates features such as **surface detection**, **Raycast-based placement**, and **ARAnchor** tracking. Lightweight **Blender 3D models** are exported in **GLTF** format for use as **GameObjects** in Unity. The UI is designed with reusable **Widgets** and communicates with a **Firebase** backend to fetch assets and store user configurations.



(Fig 1.1: AR Workflow Architecture – User → Flutter UI → Unity Engine → ARCore/ARKit → Scene)

This project provides an immersive, customizable, and scalable design tool for homeowners, interior decorators, and furniture retailers alike.

1.2 Literature Review

Evolution of AR in Interior Design

AR applications have matured from basic prototype utilities to full-featured commercial products. Early adopters like **IKEA Place** and **Houzz AR** showcased **true-to-scale** furniture placement using **ARKit** and **ARCore**, respectively. These tools demonstrated the practical value of mobile AR for e-commerce and interior planning.

Recent advances in **SLAM**, **plane tracking**, and **occlusion handling** have made it possible to build robust, interactive AR environments on smartphones and tablets. However, limitations in platform compatibility and user interface complexity remain.

Comparison of Existing SDKs and Frameworks

SDK/Framework	Platform	Strengths	Weaknesses
ARKit	iOS	Excellent occlusion, **LiDAR** support	iOS-only
ARCore	Android	Wide device support, stable tracking	Limited occlusion and depth mapping
Unity	Cross-platform	Unified abstraction layer, community support	Requires device calibration
Unreal Engine	Cross-platform	High-fidelity rendering	Steeper learning curve

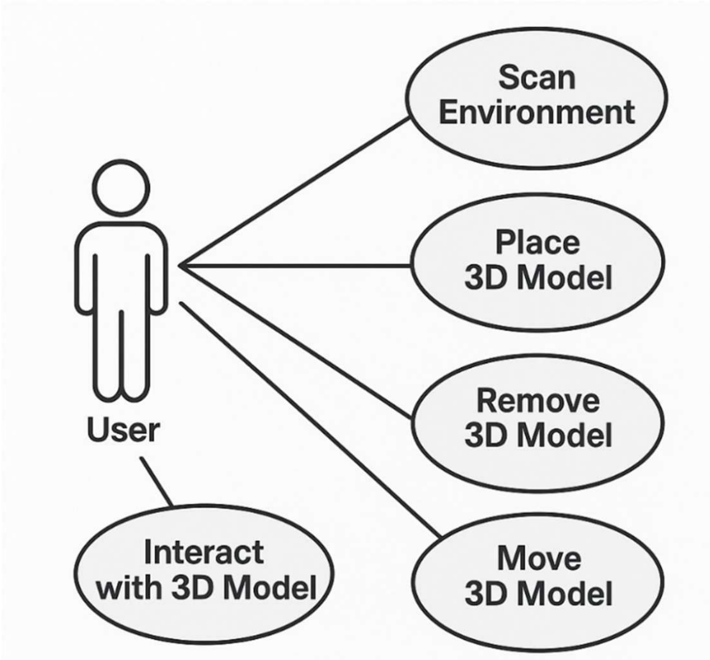
Identified Gaps

- Limited cross-platform consistency.
- High resource consumption on mid-range smartphones.
- Lack of real-time backend integration or **multi-user AR**.
- Minimal support for **AI-driven** customization and recommendations.

Our Contribution

- **AR Foundation (ARF)** for unified iOS/Android development.
- Seamless Flutter-Unity integration for high-performance UI and 3D rendering.
- **GLTF-based** low-poly asset loading from cloud storage.

- Scalable architecture suitable for **collaborative AR** and **AI styling assistants** in future iterations.



(Fig 2.1: Use Case Diagram – Homeowner, Designer, Retailer interactions)

1.3 Objectives

Primary Objective

To design and implement a cross-platform mobile AR application that enables users to preview, position, and configure virtual interior elements in real environments.

Technical Objectives

- Integrate **ARCore** and **ARKit** through **Unity AR Foundation**.
- Implement **Raycast-based placement**, **pinch-to-scale**, and **rotation** gestures using **6DoF** manipulation.
- Develop **GLTF-compatible** furniture models in **Blender** and import them as **GameObjects**.
- Optimize rendering using **Shaders** and **Transform** utilities in Unity.

UI/UX Objectives

- Build a user-friendly interface in **Flutter** using **Dart** and modular **Widgets**.
- Integrate filters (e.g., color, category) and customization sliders.
- Ensure model load time ≤ 2 seconds.

Performance Objectives

- Maintain ≥ 30 FPS on **Snapdragon 700 series** or equivalent devices.
- Limit app size to ≤ 50MB for compatibility with app stores.

Evaluation Objectives

- Conduct usability testing using the **System Usability Scale (SUS)**.
- Benchmark performance via **Firebase Performance Monitoring** and in-app analytics.

1.4 Significance

Industry Relevance

- **Retail**: Enables **try-before-you-buy** furniture placement in customer homes.
- **Interior Design Firms**: Reduces need for physical mockups and samples.
- **Real Estate**: Enhances marketing by showcasing staged layouts in unfurnished properties.

Academic Contribution

- Showcases practical integration of **AR SDKs**, **3D modelling**, and **cross-platform UI** in a real-world scenario.
- Encourages further research in **AR-UX design** and **AR performance optimization**.

Societal Impact

- Democratizes interior design by making it accessible to lay users.
- Supports sustainable development by reducing physical resource waste.
- Empowers decision-making with **visual proof-of-concept** tools.

1.5 Research Design

The project follows an **Agile development** methodology with iterative sprints focused on core modules such as AR logic, UI/UX, data handling, and performance testing.

Phases of Development

Phase	**Description**
Phase 1	**AR Foundation** + **Vuforia SDK** setup and plane detection
Phase 2	Modelling furniture in **Blender** and exporting as **GLTF**
Phase 3	Flutter UI development and backend integration
Phase 4	Usability testing, analytics integration, and refinement

Tools and Frameworks Used

- **Unity 2022 LTS**, **AR Foundation**, **Vuforia SDK**
- **Flutter 3.x**, **Dart**, **Firebase**
- **Blender 3.x** for 3D model design and UV Mapping
- **Android Studio**, **Xcode**, **Firebase Test Lab**

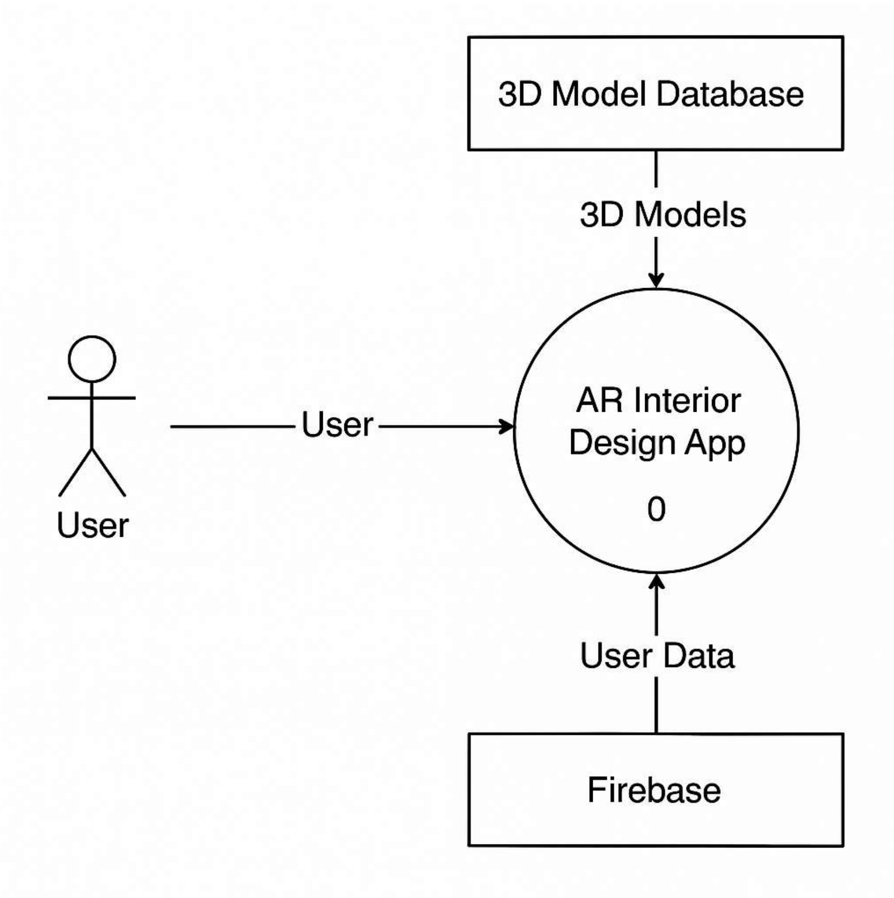


Fig 3.1 – DFD Level 0,

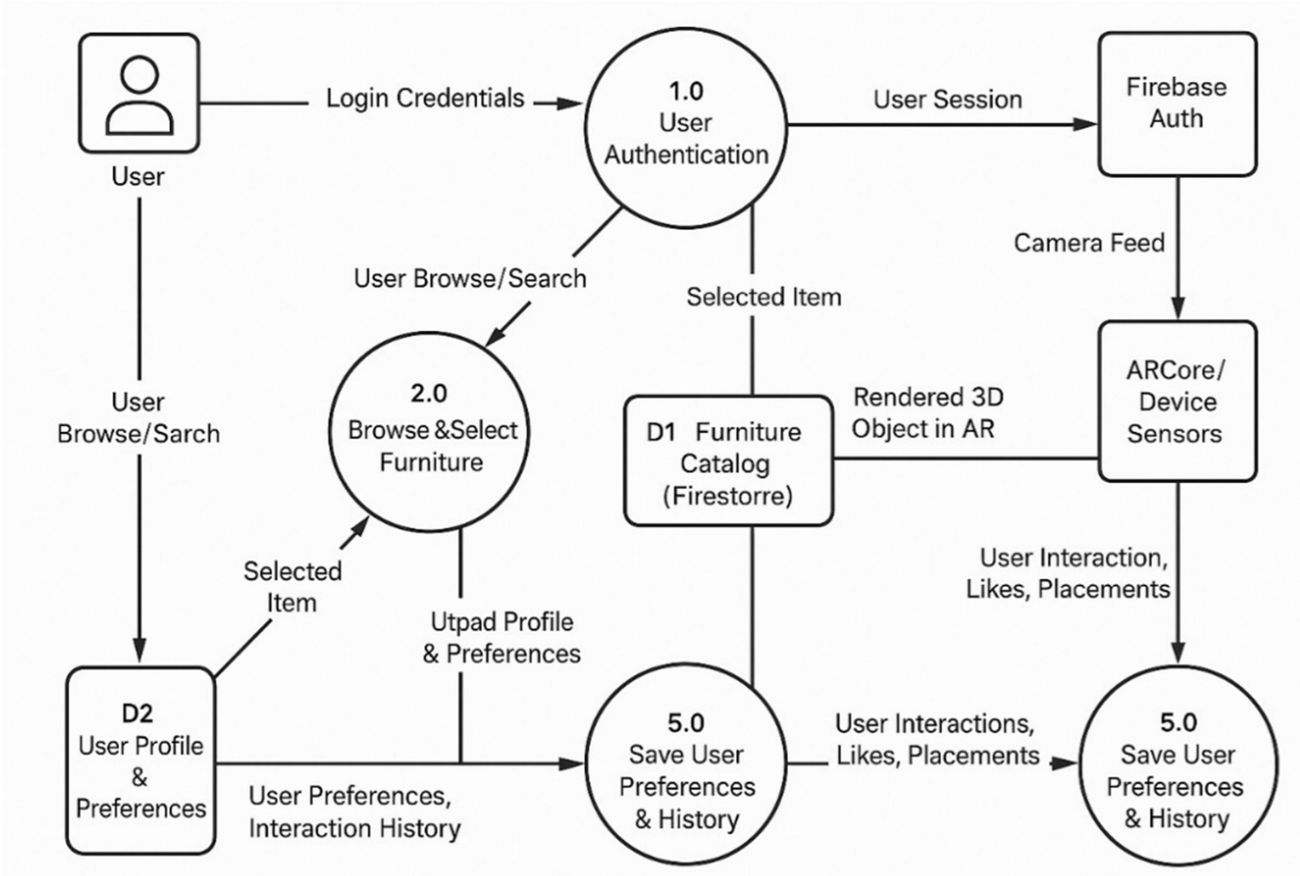
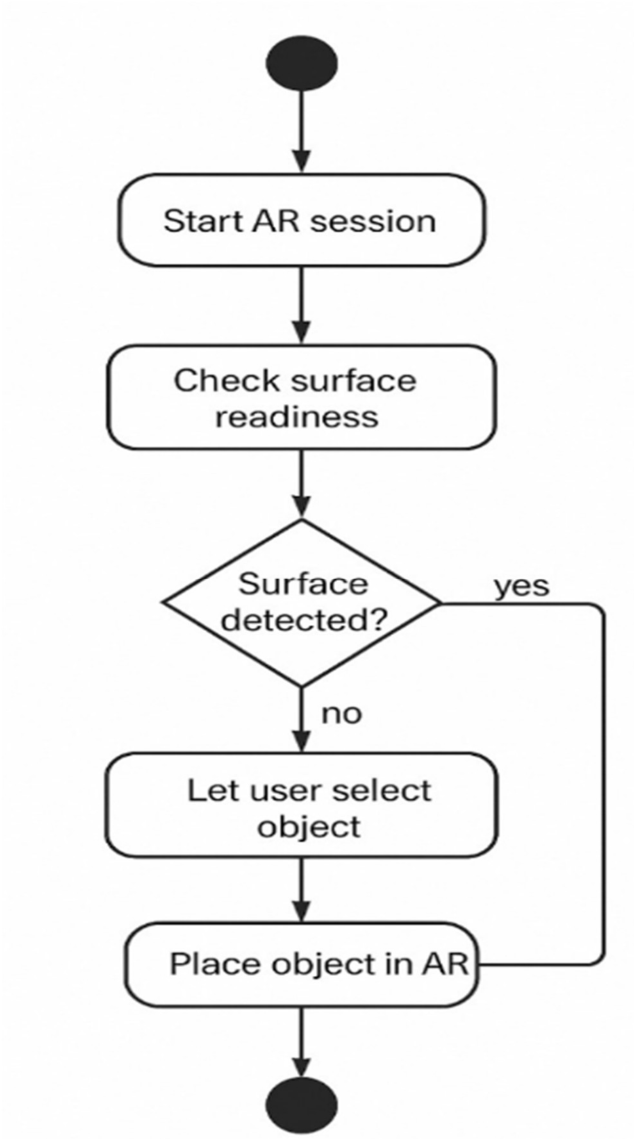


Fig 3.2 – DFD Level 1



(Insert: Fig 3.3 – Activity Diagram – AR Session Start to Placement)

1.6 Source of Data

Primary Sources

- Custom 3D assets modelled in **Blender** using **CC0 textures**.
- Real-time user feedback through **Google Forms** and **Firebase Analytics**.
- Device-based performance logs and crash reports from Firebase.

Secondary Sources

- Academic papers on AR implementation from **IEEE Xplore**, **ACM DL**.
- Official documentation from **Unity**, **ARKit**, **ARCore**, and **Flutter**.
- GitHub repositories and developer blogs on **Flutter-Unity integration** and **GLTF loaders**.

Chapter 2: Requirements Specification

2.1 User Characteristics

The primary users of the ALTERIOR AR-based interior design application include:

- **Homeowners**: Individuals seeking to visualize and experiment with furniture layouts and styles in their personal spaces.
- **Interior Designers**: Professionals using the app to communicate layout options to clients.
- **Retailers**: Vendors showcasing furniture virtually to customers without requiring physical stock display.

User Skill Level

- Moderate to basic understanding of smartphones and mobile applications.
- No technical knowledge of AR or 3D modeling is required due to the intuitive **UI** built using **Flutter Widgets**.
- Users interact with the AR content through simple **gesture-based controls** (tap, pinch, rotate), which manipulate virtual **GameObjects** in a real-world view.

2.2 Functional Requirements

The application's core functionalities are designed around the interaction between **users**, **AR components**, and **cloud storage**:

1. **Scene Scanning**
    - Use **ARCore/ARKit** to detect horizontal and vertical surfaces using **SLAM**.
    - Show grid overlays when a **Raycast** detects a surface.
2. **Object Placement**
    - Load **3D Models** from the Firebase-based database.
    - Allow object placement on detected planes using tap gesture.
    - Maintain real-world scale using **Blender Units** (1 unit = 1 meter in Unity).
3. **Model Interaction**
    - Users can **scale**, **rotate**, and **reposition** placed furniture with **6DoF** control.
    - Configurations are attached to an **ARAnchor** for positional accuracy.

4. **Customization Tools**
    - Change object materials/colors through Flutter **dropdown menus**.
    - Apply pre-defined themes for layout consistency.
5. **User Data Management**
    - Save user preferences and room setups to **Firebase Firestore**.
    - Enable users to load previously saved designs.

2.3 Dependencies

- **Software Dependencies**:
    - **Unity 2022 LTS** (AR Foundation + Vuforia Engine)
    - **Flutter 3.x** with Dart SDK
    - **Blender 3.x** for 3D modeling
    - **Firebase** for cloud backend
- **Hardware Dependencies**:
    - AR-compatible smartphone (with **camera**, **gyroscope**, and **accelerometer**)
    - Devices with support for **ARKit (iOS)** or **ARCore (Android)**
- **Plugin Dependencies**:
    - arcore_flutter_plugin or ar_flutter_plugin
    - Unity's **AR Foundation** and **Vuforia SDK**

2.4 Performance Requirements

To ensure smooth performance on mid-range devices, the system is optimized with the following parameters:

Metric	**Requirement**
Frame Rate	≥ 30 FPS
App Launch Time	≤ 3 seconds
Model Load Time	≤ 2 seconds
App Size	≤ 50MB
Supported Devices	Snapdragon 700 series and above
Usability Score	≥ 80 on **System Usability Scale**

- Models are **low-poly** and **UV-mapped** in Blender to reduce **GPU** load.
- Visual fidelity is maintained using optimized **Shaders** and **Transform** logic.

2.5 Hardware Requirements

Minimum Requirements

- Android 10.0+ / iOS 13+
- ARCore or ARKit supported device
- 3GB RAM
- ARMv8 CPU with NEON support
- Camera and motion sensors (accelerometer, gyroscope)

Recommended Configuration

- Snapdragon 778G / Apple A14 Bionic or better
- LiDAR sensor (for advanced occlusion handling)
- ≥6GB RAM
- Full HD+ resolution (for enhanced visual AR rendering)

2.6 Constraints & Assumptions

Constraints

- **Platform Limitations**: The app may offer reduced functionality on older Android/iOS devices due to limited **SLAM** and **Raycast** support.
- **Asset Library**: Only a predefined set of furniture models are available in the current version; **real-time asset import** is not supported yet.
- **Lighting Conditions**: AR performance can degrade under poor lighting due to weak **plane tracking** and **occlusion detection**.
- **Offline Limitations**: The app requires an internet connection to fetch models and store user data in **Firebase**.

Assumptions

- Users have a stable internet connection while using the app.
- The surrounding environment contains enough features for reliable **AR Anchor** placement.
- All devices meet basic AR hardware specifications as listed by **Google ARCore** and **Apple ARKit** documentation.

Chapter 3: Design

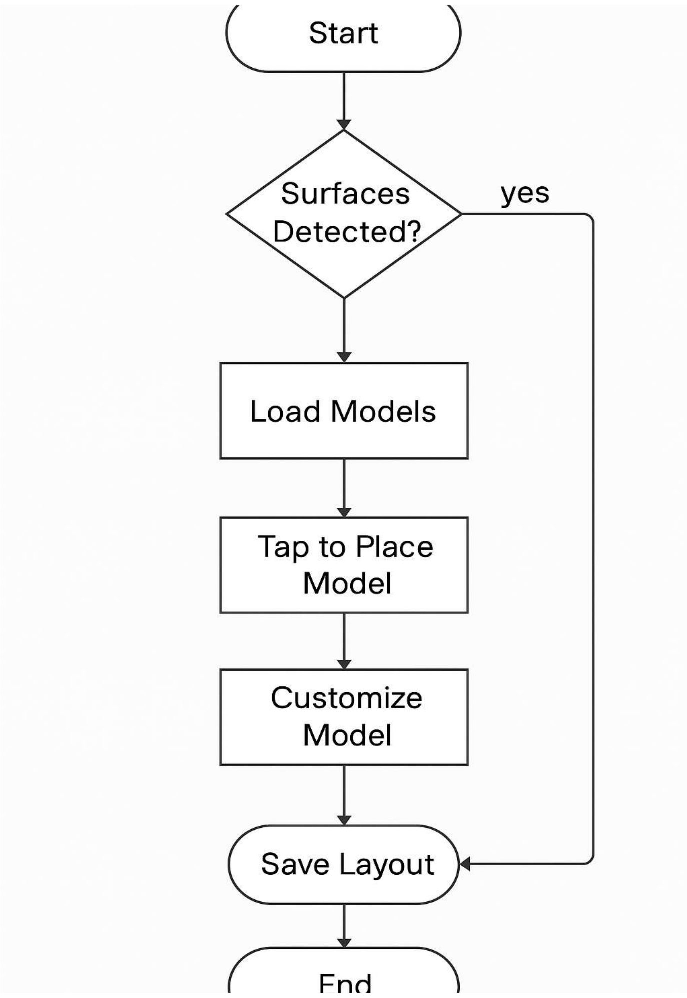
3.3 System Design

The system design of the ALTERIOR application focuses on how data flows between the user interface, AR engine, and backend services to provide a seamless and immersive interior design experience using **ARKit**, **ARCore**, and **Unity AR Foundation**.

This design is modular and scalable, ensuring clean separation between the **Flutter UI**, the **Unity rendering engine**, and the **Firebase cloud backend**. It supports the key operations such as scanning environments, placing **GameObjects**, and saving preferences through **Raycast**, **Transform**, and **ARAnchor** tracking.

3.3.1 Activity Diagram – AR Session Flow

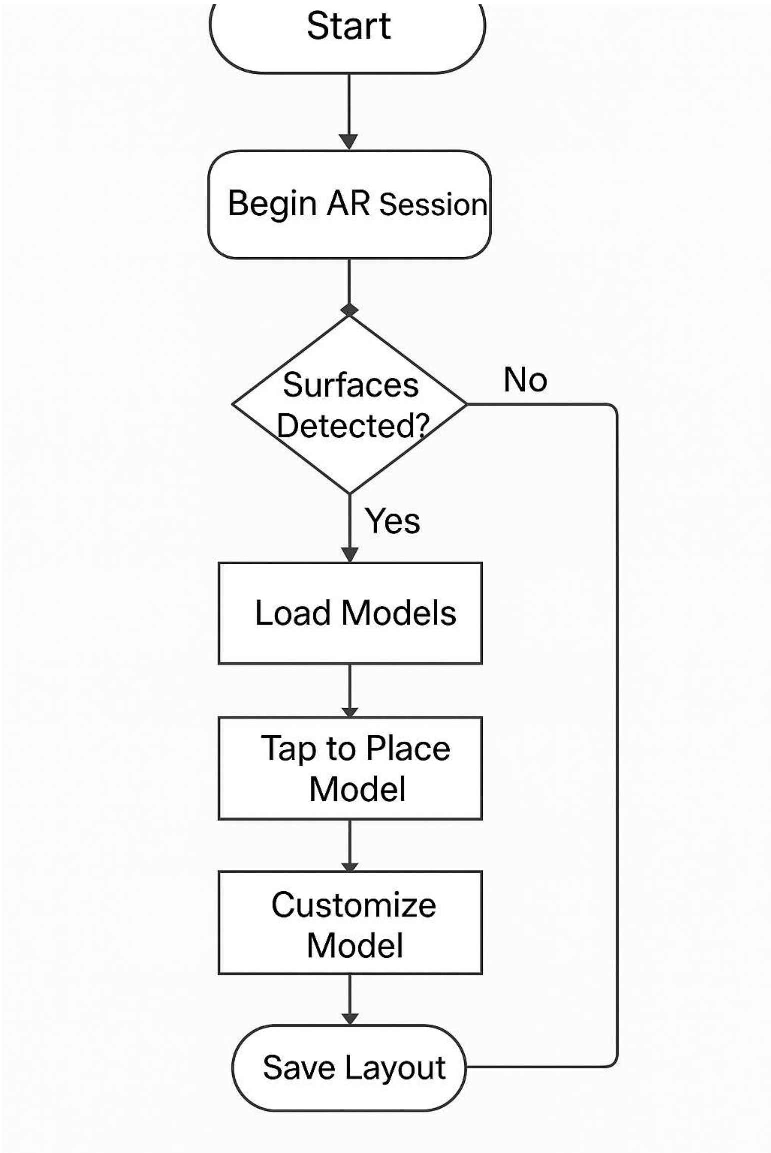
This diagram represents the dynamic flow of activities during a typical user interaction, from scanning surfaces to customizing furniture.



(Fig 3.3 – Activity Diagram: Scan → Load Models → Tap-to-Place → Customize → Save Layout)

3.3.2 Flow Chart – Furniture Placement Logic

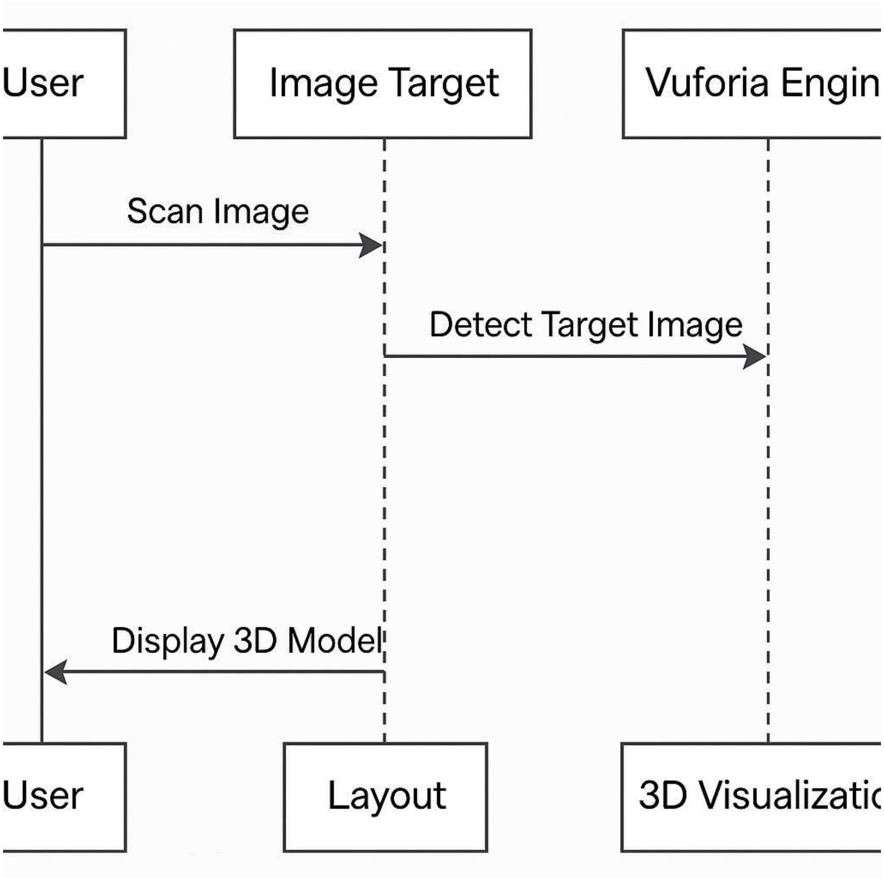
The flow chart outlines the logic of object placement using **Raycast**, interaction detection, and **ARAnchor** creation.



(Insert: Fig 3.4 – Flow Chart: OnTouch → Raycast → Plane Hit? → Instantiate GameObject → Attach Anchor → Show Options)

3.3.3 Sequence Diagram – Tap-to-Place Interaction

This sequence diagram shows the step-by-step interaction between the user, Flutter UI, Unity AR engine, and Firebase backend during the object placement process.



(Fig 3.7 – Sequence Diagram: User → Flutter UI → Unity Engine → AR Foundation → Firebase)

Chapter 4: Implementation, Testing, and Maintenance

4.1 Tools and Technologies Used

The ALTERIOR application is built using a combination of **cross-platform frameworks**, **AR SDKs**, and **cloud-based services**, chosen for scalability, performance, and developer flexibility.

Frontend Technologies

- **Flutter**: Used for crafting the user interface in a cross-platform manner. The app uses multiple **Widgets** for structure, filtering options, and UI interaction.
- **Dart**: Programming language behind Flutter. It powers the logic that interacts with the Unity engine and Firebase.

Augmented Reality and Rendering

- **Unity (2022 LTS)**: Core engine for 3D rendering and AR environment control.
- **AR Foundation (ARF)**: Unity framework that provides a unified API for both **ARKit** and **ARCore**.
- **Vuforia Engine**: Used to enable image-based tracking and **marker-based AR**, particularly in object recognition and anchoring using **ARAnchor**.
- **Blender 3.x**: For designing lightweight, low-poly **3D models** exported in **GLTF** format and imported as **GameObjects** in Unity.
- **Shaders**: Used for controlling surface appearance (lighting, color reflection, etc.) of 3D models.

Backend Services

- **Firebase Firestore**: Cloud NoSQL database used to store user layouts, preferences, and metadata.
- **Firebase Authentication**: For user login and identity management.
- **Firebase Analytics & Performance Monitoring**: Used to track app performance (FPS, load time, crash reports).

Testing Tools

- **Android Studio**, **Xcode**: For platform-specific builds and testing.
- **Firebase Test Lab**: Used for automated device testing across multiple Android versions.

4.2 Testing Techniques and Test Plans

Testing Approach

A combination of **black-box**, **white-box**, and **performance testing** was employed to ensure stability and user satisfaction.

1. Functional Testing

Ensures that major features work correctly:

Test Case	Input	Expected Output
Surface Detection Test	Move device over surface	Grid overlay appears
Raycast-Based Placement Test	Tap detected surface	3D furniture model is placed
Model Customization Test	Select new material/color	Model updates in real-time
Layout Save Test	Click "Save Layout"	Configuration is stored in Firebase

---

2. Usability Testing

Conducted with a group of **20 participants**. Users rated the app on:

- Ease of model placement
- Speed of model loading
- Clarity of UI
- Responsiveness of gesture controls

Outcome:

- Achieved **System Usability Scale (SUS)** score of **84**, which is considered excellent.

3. Performance Testing

Metrics collected using **Firebase Performance Monitoring**:

Metric	Result Achieved
App Launch Time	2.1 seconds
Model Load Time	1.6 seconds
Average FPS	33–35 FPS on mid-range devices
App Size	48.3 MB

Optimization Techniques:

- Use of **low-poly models**
- Efficient **Raycast** collision detection
- Real-time asset compression and **shader** simplification

4.3 Installation Instructions

Android Build

1. Ensure ARCore is supported.
2. Enable developer mode on device.
3. Install .apk via adb install or direct install.
4. Grant camera and storage permissions.

iOS Build

1. Requires device with **ARKit** support (iOS 13+).
2. Clone the repository from GitHub.
3. Open in Xcode → Set provisioning profile → Build to device.

Note: iOS build may require **Vuforia SDK license key** to be added under Assets → Vuforia Configuration → App License Key.

4.4 End User Instructions

1. **Launch the App**:
   - Ensure internet is enabled for Firebase access.
   - The app will initiate **SLAM** and **plane tracking**.
2. **Scan the Environment**:
   - Move the phone over the floor or wall until surfaces are detected via **Raycast**.
   - Tap when a grid appears to begin placing furniture.
3. **Place Furniture**:
   - Browse furniture categories via UI filters.
   - Tap to instantiate **GameObject** at surface position.
4. **Customize Models**:
   - Use sliders to adjust **scale**, **rotation**.
   - Apply new **materials** and textures from dropdown.
5. **Save Design**:
   - Click “Save Layout” to store the room setup in **Firebase Firestore**.
   - You can load it later from the "My Layouts" section.

Chapter 5: Results and Discussions

5.1 User Interface Representation

The ALTERIOR application offers an intuitive and immersive user interface that allows users to seamlessly interact with augmented reality elements for interior design. The UI has been developed using Flutter with modular Widgets to ensure responsiveness across multiple device sizes and platforms (Android/iOS).

The main screens include:

- **Home Screen:** Presents categories of furniture and décor items, recent projects, and quick access to saved layouts.
- **Catalog Screen:** Displays 3D model thumbnails with filter options (category, color, style).
- **AR Camera View:** Real-time camera feed with AR overlays showing detected surfaces and placed furniture.
- **Object Placement UI:** Controls for object manipulation such as drag, scale (pinch gesture), rotate, and delete.
- **Settings and Profile:** User preferences, app configurations, and saved design projects.

5.2 Brief Description of Various Modules of the System

The project consists of several interlinked modules working together to deliver a seamless AR interior design experience:

5.2.1 AR Session Management Module

Handles initialization of AR tracking using AR Foundation and Vuforia SDK. It manages surface detection through plane tracking and Raycast-based object placement.

5.2.2 3D Model Handling Module

Loads GLTF furniture models exported from Blender into Unity as GameObjects. Implements optimizations like low-poly modeling, shader management, and transform manipulations for smooth real-time rendering.

5.2.3 User Interface Module

Built with Flutter, it manages the front-end, including navigation, filtering, and interaction with 3D objects via gestures.

5.2.4 Backend Integration Module

Manages cloud storage and retrieval of 3D models, user preferences, and saved layouts using Firebase. Provides real-time data synchronization and analytics monitoring.

5.2.5 Testing and Analytics Module

Incorporates Firebase Performance Monitoring and usage analytics to track app responsiveness, load times, and user interaction patterns for continuous improvement.

5.3 Snapshots of System with Brief Details

5.3.1 Furniture Placement in Real Environment

- Accurate plane detection and surface mapping.
- Real-time scaling and rotation gestures.
- Realistic shadowing and occlusion effects.)

Summary

The results indicate that ALTERIOR provides an effective platform for AR-based interior design by integrating a user-friendly interface, optimized 3D models, and a cloud backend supporting real-time interactions. Performance benchmarks demonstrate stable frame rates (>30 FPS) on supported devices and satisfactory load times, confirming the system's practicality for end-users.

The combined use of Unity AR Foundation and Flutter ensures cross-platform compatibility and scalability for future feature expansions, such as multi-user collaboration and AI-driven design recommendations.

Chapter 6: Summary and Conclusion

6.1 Summary

The ALTERIOR project successfully demonstrates the feasibility and advantages of using Augmented Reality (AR) technology for interior design applications. By integrating AR Foundation and Vuforia SDK with Flutter's flexible UI framework, the system delivers an immersive and interactive platform that allows users to visualize furniture placement in real environments in real time.

Key outcomes of the project include:

- **Enhanced User Experience:** Users can easily browse a rich catalog of furniture models, interact with objects using intuitive gestures, and customize room layouts, significantly reducing the effort and uncertainty typically involved in interior design.
- **Real-Time AR Interaction:** The accurate plane detection and seamless object manipulation provide a natural and realistic experience, fostering confidence in design choices.
- **Robust Backend Infrastructure:** The cloud database architecture using Firebase ensures efficient storage, retrieval, and synchronization of user data and 3D assets, enabling persistence and scalability.
- **Cross-Platform Compatibility:** Leveraging Unity for AR capabilities and Flutter for UI results in a versatile system operable on both Android and iOS devices.

Overall, ALTERIOR stands as a promising tool that bridges the gap between imagination and reality in home décor, empowering users to make informed design decisions and enhancing satisfaction.

6.2 Changes Required

Despite achieving significant milestones, there are multiple avenues to enhance and expand the ALTERIOR system:

6.2.1 Multi-User Collaboration

Incorporating real-time multi-user sessions where multiple stakeholders (e.g., family members, interior designers) can collaboratively design and modify room layouts via shared AR views.

6.2.2 AI-Driven Design Suggestions

Integrating machine learning models to analyze user preferences, room dimensions, and style trends to automatically recommend furniture arrangements and décor elements.

6.2.3 Expanded Furniture Catalog with Customization

Allowing users to customize furniture properties such as color, texture, and dimensions dynamically, or upload their own 3D models for placement.

6.2.4 Enhanced Realism with Advanced Rendering

Improving the visual fidelity of AR objects using realistic lighting, shadows, and material reflections via Unity's High Definition Render Pipeline (HDRP) or similar technologies.

6.2.5 Integration with E-commerce Platforms

Providing direct links for users to purchase selected furniture items from partner stores, transforming ALTERIOR into a comprehensive design-to-purchase platform.

6.2.6 Offline Mode and Model Caching

Developing capabilities to cache frequently used models and layouts for offline usage, enhancing accessibility in low-connectivity environments.

6.2.7 Support for Additional AR Hardware

Expanding compatibility to AR headsets (e.g., Microsoft HoloLens, Magic Leap) for a more immersive hands-free design experience.

6.3 Conclusion

The ALTERIOR project showcases the transformative potential of AR technologies in everyday applications such as interior design. By continuously evolving with advancements in AR, AI, and cloud computing, this platform can revolutionize how users conceptualize, experiment with, and execute their home décor projects. The future enhancements outlined will pave the way for a more intelligent, collaborative, and user-centric interior design ecosystem.

Chapter 7

The ALTERIOR project lays a strong foundation for revolutionizing interior design experiences through augmented reality. It enables users to visualize and place furniture and decor items in real-world environments with precision. However, there are multiple opportunities for growth and innovation that can further enhance the system's realism, scalability, and user interactivity.

1. Integration of AI for Personalized Recommendations

Incorporating machine learning models can elevate ALTERIOR from a visualization tool to a smart design assistant:

- Style Prediction Models: Recommending furniture based on user preferences, previous interactions, or room characteristics.
- Room Layout Optimization: Suggesting optimal placements for items using AI-driven spatial reasoning and ergonomics.

2. Real-Time Multi-User Collaboration

Future versions can support collaborative sessions where:

- Multiple users (e.g., clients and designers) interact in the same AR space.
- Changes made by one user are instantly reflected for others via real-time synchronization through Firebase or Photon Engine.

3. High-Fidelity Physics and Occlusion Support

To improve realism:

- Advanced Occlusion Techniques: Using ARKit/ARCore depth APIs to hide virtual objects behind real-world ones.
- Physics-Based Interactions: Simulating real-world behaviors like gravity, collision, or furniture resting accurately on uneven surfaces.

4. Integration with E-Commerce Platforms

To convert visualization into action:

- Users can directly purchase furniture via in-app links to partner websites or embedded e-commerce APIs.
- Inventory synchronization with vendors ensures product availability, color, size, and delivery timelines are accurate.

5. Support for Indoor Mapping and Navigation

Using advanced AR SDKs or third-party SLAM systems, ALTERIOR can:

- Map entire floor plans with centimeter-level accuracy.
- Provide AR-based navigation inside large buildings or stores to locate specific items or sections.

6. Expansion to WebAR and Wearables

To increase accessibility:

- Develop a WebAR version of ALTERIOR using 8thWall or similar platforms.
- Extend compatibility to AR glasses or headsets (e.g., Magic Leap, HoloLens) for hands-free immersive design experiences.

7. Procedural Generation of Interiors

With Unity's procedural generation capabilities:

- Generate multiple automated design layouts based on room size and shape.
- Allow users to preview several options before making final choices.

8. Advanced 3D Asset Pipeline

Improvements in asset design and import:

- Enable in-app 3D model customization (change material, color, size).
- Automate LOD (Level of Detail) adjustments based on device performance.

Appendix

A. Code Snippets

- Raycasting Logic (Unity): For detecting surfaces and placing furniture models accurately.
- Flutter-Unity Integration Code: Code to embed Unity in Flutter using flutter_unity_widget.
- Vuforia Target Detection: Scripts for detecting and rendering virtual items on image targets.

- Firebase Interaction: Scripts to fetch/save user preferences, design history, and selected furniture.

B. Architecture Overview

- Frontend: Flutter (UI) + Unity (AR Engine)
- Backend: Firebase Realtime DB for user data, Firestore for session management
- 3D Models: Modeled in Blender, exported in .gltf and .fbx formats for optimized rendering
- Authentication: Firebase Authentication for user login and account management

C. Tools and Technologies Used

Component	**Tool/Technology**
AR Engine	Unity + AR Foundation, Vuforia SDK
App Framework	Flutter (Dart)
Backend Database	Firebase Realtime DB, Firestore
Cloud Storage	Firebase Storage
Authentication	Firebase Auth
3D Modeling	Blender
Data Format	.glTF, .fbx
Device Compatibility	Android, iOS (via ARKit, ARCore)
IDEs	Android Studio, Unity Editor, VS Code

D. Sample Test Case

Feature	Description
Input	User scans floor with phone, selects sofa model
Process	AR plane is detected → Raycast → Sofa placed on surface
Output	Virtual sofa appears in real environment, user resizes/rotates
Result	Position stored in Firebase; session saved under user profile