**Picnoid : An AR app**

**A Project Report**

Submitted in partial fulfilment of the Requirements for the award of the Degree of

### BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

**By**

Devam Thakar KCTYIT058

**Under the esteemed guidance of Assistant Professor**

**Mr. Kaushal Shah**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

### KISHINCHAND CHELLARAM COLLEGE

***(Affiliated to HSNC University)* MUMBAI, 400020 MAHARASHTRA**

**2024-25**

**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

PNR No.: 2022230210020019 Roll No: KCTYIT058

**Name of the Student: Devam Thakar**

**Title of the Project: Picnoid An AR app**

**Name of the Guide: Mr. Kaushal Shah**

**Teaching experience of the Guide: 24 years**

Is this your first submission? Yes  No 

Signature of the Student Signature of the Guide

Date: ………………… Date: …………………….

Signature of the Co-Ordinator

Date: …………………

### KISHINCHAND CHELLARAM COLLEGE

***(Affiliated to HSNC University, Mumbai)***

**MUMBAI, MAHARASHTRA - 400020**

### DEPARTMENT OF INFORMATION TECHNOLOGY

****

**CERTIFICATE**

This is to certify that the project entitled, **"Picnoid : An AR app"**, is bonafied work of Devam Thakar bearing KCTYIT058 submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from HSNCU

University.

**Internal Guide Co-Ordinator**

**External Examiner**

**Date: College Seal**

**ACKNOWLEDGEMENT**

We would like to take this opportunity to express a deep sense of gratitude to Dr. Tejashree V. Shanbhag (Principal of K.C. College) and Dr. Rakhi Gupta (H.O.D. of B.Sc. I.T. Department) for their cordial support, valuable information and guidance, which helped us in completing this task through various stages. We take this opportunity to express our profound gratitude and deep regards to our guide, Mrs. Pragati Thawani for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her time-to-time shall care us a long way in the journey of life on which we are about to embark.

We are obliged to staff members of K.C. College for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our project.

-Devam.D.Thakar

**DECLARATION**

I hereby declare that the project entitled, **“Picnoid : An AR app”** done at **K.C. College** is done, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted

as final semester project as part of our curriculum.

**Name and Signature of the Student**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Page No** |
| 1 | Introduction | 08 |
|  | 1.1 Background | 09 |
| 1.2 Objectives | 9 |
| 1.3 Purpose, Scope and Applicability | 9-10 |
| 1.4 Achievements | 10 |
| 1.5 Organization of Report | 10-11 |
| 2 | Survey of Technology | 12 |
|  | 2.1 Front-End Features | 13 |
| 2.2 Back-End Features | 13 |
| 2.3 Justification | 14 |
| 3 | Requirement and Analysis | 15 |
|  | 3.1 Problem Definition | 16 |
| 3.2 Requirement Specification | 16 |
| 3.3 Software and Hardware Requirement | 17 |
| 3.4 Planning and Scheduling | 17-18 |
| 3.5 Gantt Chart | 18 |
| 4 | System Design | 19 |
|  | 4.1 Data Flow Diagram | 20-21 |
| 4.2 Use Case Diagram | 23 |
| 4.3 Entity relationship Diagram | 25 |
| * 1. Class Diagram   2. Sequence Diagram   3. Activity Diagram | 27  29  31 |
| 5 | Implementation & Testing | 33 |
|  | 5.1 Implementation  5.2 Testing  5.3 Test Cases | 34  35  36 |
| 6 | Results & Discussion | 37 |
|  | 6.1 Test Result  6.2 User Documentation | 38  39 |
| 7 | Cost analysis | 43 |
| 8 | Conclusion & Future Scope | 46 |
| 9 | Research Paper | 49 |

**LIST OF DIAGRAMS**

|  |  |  |
| --- | --- | --- |
| **Sr no** | **Diagrams** | **Page no** |
| **1.** | Data flow diagram | **20-21** |
| **2.** | Use Case Diagram | **23** |
| **3.** | Entity Relationship Diagram | **25** |
| **4.** | Class Diagram | **27** |
| **5.** | Sequence Diagram | **29** |
| **6.** | Activity Diagram | **31** |

**ABSTRACT**

The proliferation of mobile applications has led to a fragmented user experience where functionalities such as image processing, social media sharing, augmented reality (AR) interactions, and measurement tools are isolated across different platforms. PicNoid aims to address this challenge by consolidating these diverse features into a single, cohesive application for Android users. Leveraging Google’s ARCore API, PicNoid offers an integrated platform that enhances user convenience, engagement, and creativity. Key functionalities include image capture and editing, social media sharing, immersive AR experiences with popular characters and objects, creative face filters, real-time object measurement, image-to-video transformation, and playing videos within AR scenes. Developed using Flutter in Visual Studio Code, and incorporating advanced library like ARCore, PicNoid stands out as a versatile, all-in-one solution. This paper discusses the methodology adopted for requirement analysis, design, development, and testing, and explores the real-world applications of PicNoid in areas such as social media, interior design, education, e- commerce, health, architecture, event management, and the creative arts. PicNoid’s comprehensive approach promises to revolutionize the mobile application landscape by delivering an unmatched user experience through its innovative integration of AR and image processing capabilities.

Keywords: Augmented Reality, Image Processing, Mobile Applications, ARCore, Real-time Measurement, Android Development.

# CHAPTER 1

## Chapter-1 INTRODUCTION

### BACKGROUND

The mobile application landscape is currently fragmented, with users often needing to switch between multiple apps to access functionalities such as image processing, social media sharing, augmented reality (AR) interactions, and measurement tools. This fragmented experience is inefficient and cumbersome. Additionally, many AR applications lack diverse features such as character augmentation, real-time object measurement, and video transformation, limiting the potential for comprehensive AR interactions. To address these challenges, PicNoid aims to consolidate these functionalities into a single cohesive application, enhancing user convenience, engagement, and creativity.

### OBJECTIVES

* Integrate diverse functionalities: Combine image processing, AR interactions, social media sharing, and measurement tools into one application.
* Enhance user experience: Provide a seamless, intuitive, and engaging user interface that simplifies access to multiple features.
* Leverage advanced technologies: Utilize Google's ARCore API, OpenCV, and TensorFlow Lite to deliver cutting-edge AR and image processing capabilities.
* Ensure reliability and performance: Conduct extensive testing and incorporate user feedback to maintain high standards of reliability and performance.

### PURPOSE, SCOPE, AND APPLICABILITY

* + 1. **PURPOSE**

The purpose of PicNoid is to provide a comprehensive and integrated mobile application that combines advanced image processing and augmented reality features. By unifying these functionalities, PicNoid aims to streamline the user experience, reduce the need for multiple apps, and foster creativity and engagement among users.

### SCOPE

PicNoid's scope encompasses the following key features:

* + - * Image capture, editing, and sharing capabilities.
      * Augmented reality interactions with popular characters and objects.
      * Creative face augmentation filters.
      * Real-time object measurement tools.
      * Image-to-video transformation features.
      * Playing videos within AR scenes.

These features are targeted towards Android users and are developed using Flutter, leveraging ARCore for AR functionalities and various APIs and libraries for image processing and machine learning.

### APPLICABILITY

PicNoid is applicable in various domains, including:

* + - * **Social Media and Entertainment**: Enhancing content creation and interactive storytelling.
      * **Interior Design and Visualization**: Virtual placement of furniture and objects.
      * **Education and Training:** Interactive learning and skill training.
      * **E-commerce and Marketing:** Product visualization and AR advertising.
      * **Health and Fitness:** Virtual workouts and health monitoring.
      * **Measurement and Architecture**: Real-time measurements and project planning.
      * **Event Management and Tourism:** Virtual tours and interactive guides.
      * **Creative Arts:** Digital art, animation, and augmented performances.

### ACHIEVEMENTS

* Development of a comprehensive mobile application: PicNoid successfully integrates multiple functionalities into a single app, enhancing user convenience.
* Utilization of advanced technologies: The application leverages ARCore, OpenCV, and TensorFlow Lite to provide state-of-the-art features.
* Positive user feedback: Iterative development and testing processes have resulted in a highly reliable and user-friendly application.
* Publication and maintenance: PicNoid is available on the Google Play Store, with ongoing updates and improvements based on user feedback.

### ORGANIZATION OF REPORT

1. Introduction
   * Background
   * Objectives
   * Purpose, Scope, and Applicability
2. Literature Review
   * Review of existing mobile applications and their limitations
   * Technological advancements in AR and image processing
3. Methodology
   * Requirement Analysis
   * Design and Prototyping
   * Development Process
   * Testing Procedures
   * Deployment and Maintenance
4. Features and Functionality
   * Detailed description of PicNoid's features
   * User interface and experience design

# CHAPTER 2

## Chapter-2 SURVEY OF TECHNOLOGY

### FRONT-END FEATURES

#### Flutter:

* + - **Cross-Platform Development**: Flutter allows you to create a single codebase that runs on both Android and iOS, ensuring a consistent look and feel across devices.
    - **Customizable Widgets**: Flutter provides a rich set of pre-designed widgets, which can be customized to fit the specific needs of the PicNoid app. This allows for a seamless user interface that is both visually appealing and functionally robust.
    - **Hot Reload**: This feature allows developers to see the effects of changes in real-time, speeding up the development process and making it easier to tweak UI components and fix bugs.
    - **Integration with ARCore**: Flutter can be integrated with ARCore for augmented reality functionalities, enabling the display of AR objects and characters within the app.
    - **High-Performance Rendering**: Flutter’s engine uses Skia for rendering, which ensures smooth animations and high performance, essential for an AR-based application like PicNoid.
    - **Responsive Layouts**: Flutter’s flexible layout system makes it easier to design responsive interfaces that work well on devices of various sizes and resolutions.

### BACK-END FEATURES

#### Firebase:

* + - **Real-time Database**: Firebase’s NoSQL cloud database allows for real-time data synchronization, which can be essential for collaborative or interactive features where users might need to see changes reflected instantly.
    - **Authentication**: Firebase Authentication provides an easy way to manage user sign-ins and registrations, supporting email/password, Google, Facebook, and other social logins.
    - **Cloud Storage**: Firebase Storage allows you to store and serve user-generated content such as images and videos. This is particularly useful for handling the media assets created and edited within the PicNoid app.
    - **Firestore**: Firestore provides a scalable, flexible database for storing app data. It is optimized for storing large collections of small documents, making it suitable for managing the structured data that the PicNoid app will generate.
    - **Cloud Functions**: Firebase Cloud Functions can be used to run backend code in response to events triggered by Firebase features and HTTPS requests, allowing you to offload some processing tasks from the app.
    - **Analytics**: Firebase Analytics offers insights into user behavior and app performance, helping you to optimize the user experience and track the effectiveness of various features.
    - **Push Notifications**: Firebase Cloud Messaging (FCM) allows you to send notifications to users, keeping them engaged with the app by alerting them about new features or reminders.

### JUSTIFICATION - WHY THIS TECHNOLOGY?

**Why Flutter?**

* + - **Ease of Development**: Flutter’s single codebase for both Android and iOS reduces development time and cost, making it an efficient choice for building cross-platform applications.
    - **UI/UX Flexibility**: The customizable widgets and extensive library allow for crafting a unique, engaging user experience that is critical for an app centered on image processing and augmented reality.
    - **Community and Support**: Flutter has a large and active community, with a wealth of tutorials, plugins, and resources that can help overcome development challenges.

#### Why Firebase?

* + - **Scalability**: Firebase’s backend infrastructure is built to scale with your application, from small-scale use cases to large, complex projects, making it ideal for a growing app like PicNoid.
    - **Real-Time Capabilities**: Firebase’s real-time database and synchronization features are crucial for ensuring that users have a seamless experience, particularly when dealing with live data or collaborative functionalities.
    - **Integration and Ecosystem**: Firebase integrates smoothly with Flutter, providing a cohesive environment for both front-end and back-end development. The ecosystem includes analytics, authentication, and cloud storage, all of which are essential for building a robust, data-driven app.
    - **Security**: Firebase offers built-in security features such as secure authentication and role-based access control, ensuring that user data and interactions are protected

# CHAPTER 3

## Chapter-3 REQUIREMENT AND ANALYSIS

### PROBLEM DEFINITION

The PicNoid Augmented Reality (AR) App is designed to address the need for a comprehensive tool that allows users to capture, edit, and enhance images with AR elements and share them on social media platforms. Traditional photo editing apps lack the seamless integration of AR features, real-time image processing, and easy content sharing, limiting user engagement and creativity. PicNoid aims to fill this gap by providing a powerful, user-friendly app that combines the best of photo editing, AR capabilities, and social media integration.

### REQUIREMENTS SPECIFICATION

#### Functional Requirements:

* + - **Image Capture and Selection**: Users must be able to capture images using the device camera or select images from the photo gallery.
    - **Image Editing**: The app should provide tools for cropping, filtering, adjusting brightness, contrast, and adding effects.
    - **AR Features**: Users should be able to overlay AR objects onto captured or selected images.
    - **Measurement Tools**: The app should allow users to measure objects within the AR environment.
    - **Content Sharing**: Users must be able to share edited images and videos directly to social media platforms like Instagram, Facebook, and Twitter.
    - **User Authentication**: The app should support user login and registration through Firebase Authentication.
    - **Data Storage**: Images and videos should be stored securely using Firebase Cloud Storage, with metadata managed via Firestore.

#### Non-Functional Requirements:

* + - **Performance**: The app should offer smooth performance with minimal lag, even when processing high-resolution images or complex AR features.
    - **Usability**: The user interface should be intuitive, with easy-to-navigate menus and clear options for each functionality.
    - **Scalability**: The backend should be capable of scaling to support a growing number of users and data.
    - **Security**: User data, including images and videos, must be stored securely, with appropriate access controls in place.

### SOFTWARE AND HARDWARE REQUIREMENTS

#### Software Requirements:

* + - **Operating Systems**: Android 7.0 (Nougat) and above, iOS 11 and above.
    - **Development Tools**: Android Studio, Xcode, Visual Studio Code.
    - **Programming Languages**: Dart (for Flutter), Swift (for native iOS integration, if needed).
    - **Libraries/Frameworks**: Flutter SDK, Firebase SDK, ARCore (for Android), ARKit (for iOS).
    - **Database**: Firebase Firestore, Firebase Realtime Database.
    - **Version Control**: Git and GitHub for source code management.

#### Hardware Requirements:

* + - **Development Machines**: MacBook (for iOS development), Windows/Linux PC (for Android development).
    - **Mobile Devices**: Smartphones with AR support (e.g., Google Pixel, iPhone 8 or newer) for testing.

### PLANNING AND SCHEDULING

#### Project Phases:

* + - **Requirement Gathering and Analysis**: (1-2 weeks) In this phase, detailed requirements are documented, and user stories are developed.
    - **Design**: (2-3 weeks) This phase includes designing the UI/UX, database schema, and system architecture.

#### Development:

* + - * **Front-End Development**: (4-5 weeks) Developing the Flutter-based UI, including image capture, AR features, and editing tools.
      * **Back-End Development**: (4-5 weeks) Implementing Firebase for authentication, database management, and cloud storage.
    - **Integration and Testing**: (2-3 weeks) This phase involves integrating the front-end with the back-end and performing unit, integration, and user acceptance testing.
    - **Deployment**: (1-2 weeks) Preparing the app for deployment on the Google Play Store and Apple App Store.
    - **Maintenance and Updates**: Ongoing process of bug fixing, performance enhancement, and feature updates based on user feedback.

|  |  |  |  |
| --- | --- | --- | --- |
| Remarks | Defined date of submission | Actual submission date | Signature |
| Selection and approval | 09/06/2024 | 16/06/2024 |  |
| Feasibility Study | 16/06/2024 | 30/06/2024 |  |
| Requirement and Analysis | 30/06/2024 | 21/07/2024 |  |
| System Design | 14/07/2024 | 28/08/2024 |  |
| Implementation & Testing | 20/09/2024 | 11/01/2025 |  |
| Results & Discussions | 11/01/2025 | 30/01/2025 |  |
| Conclusion & Future Scope | 01/02/2025 | 11/02/2025 |  |
| Documentaion | 07/07/2024 | 05/03/2025 |  |

### GANTT CHART

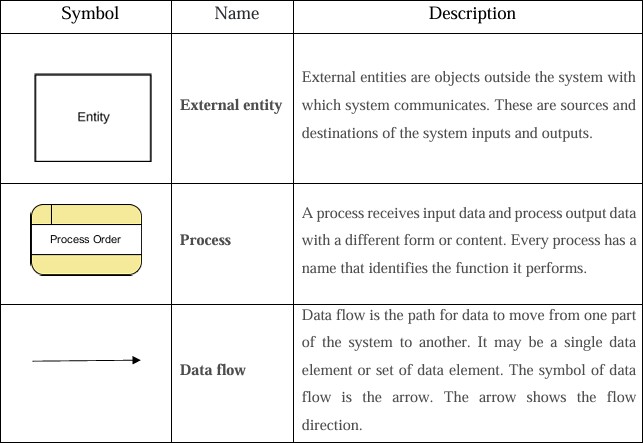
A Gantt chart should be created to visualize the timeline of these phases, showing the start and end dates of each phase and overlapping tasks.

# CHAPTER 4

## Chapter-4 SYSTEM DESIGN

### DATA FLOW DIAGRAM

The flow of data of a system or a process is represented by Data Flow Diagram. It also gives insight into the inputs and outputs of each entity and the process itself.



DFD Level 0

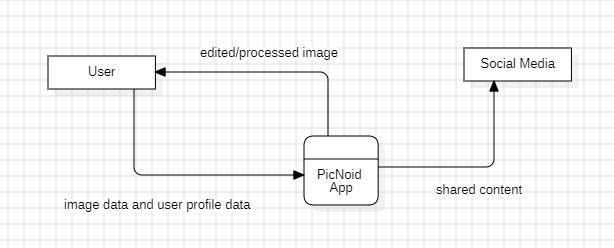


Figure 4.1.1 Data Flow Diagram Level 0

DFD Level 1

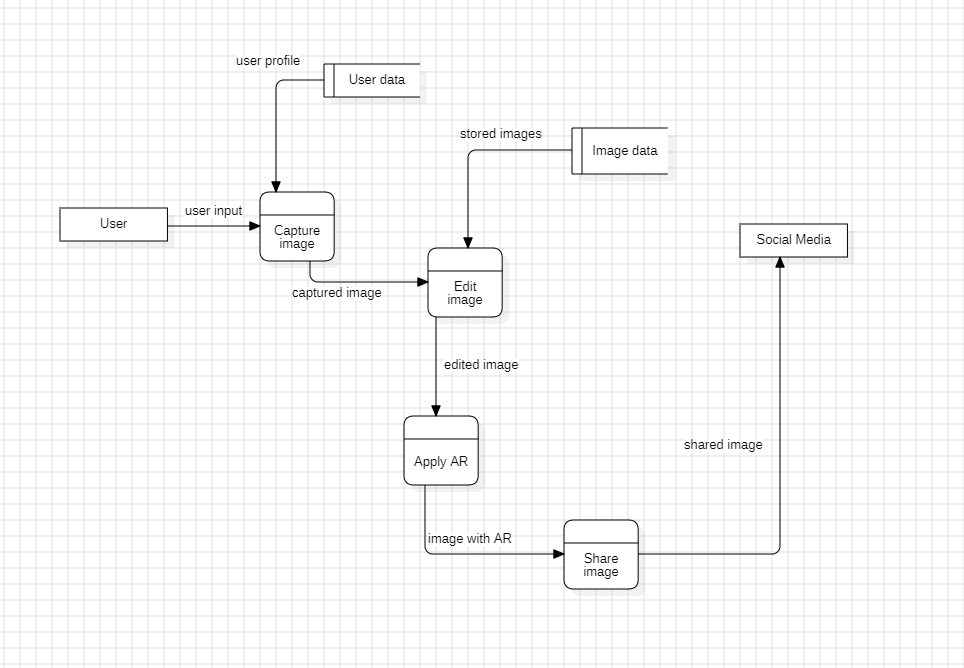
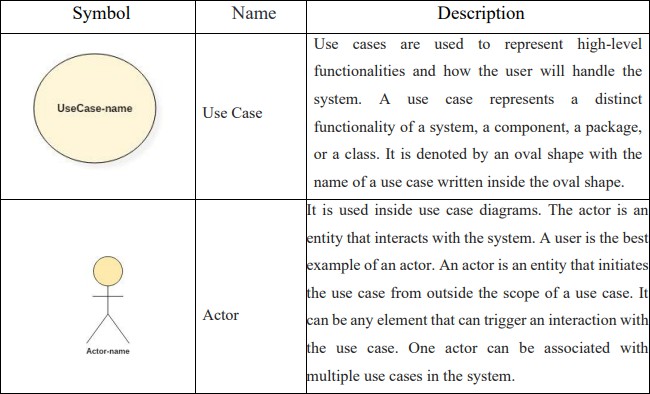


Figure 4.1.2 Data Flow Diagram level 1

### USE CASE DIAGRAM

In this project there is only one user. The user performs the gesture to the system. System then interprets the gesture and display the gesture meaning to the user.



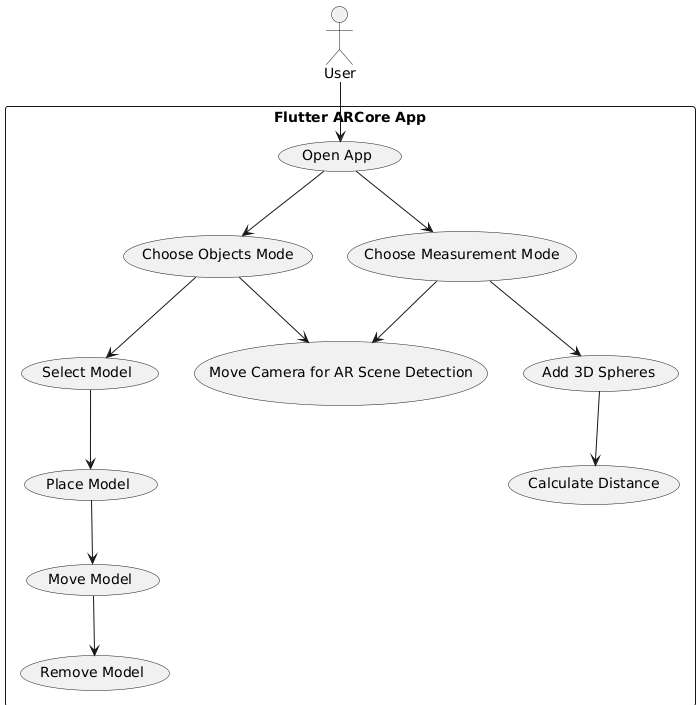
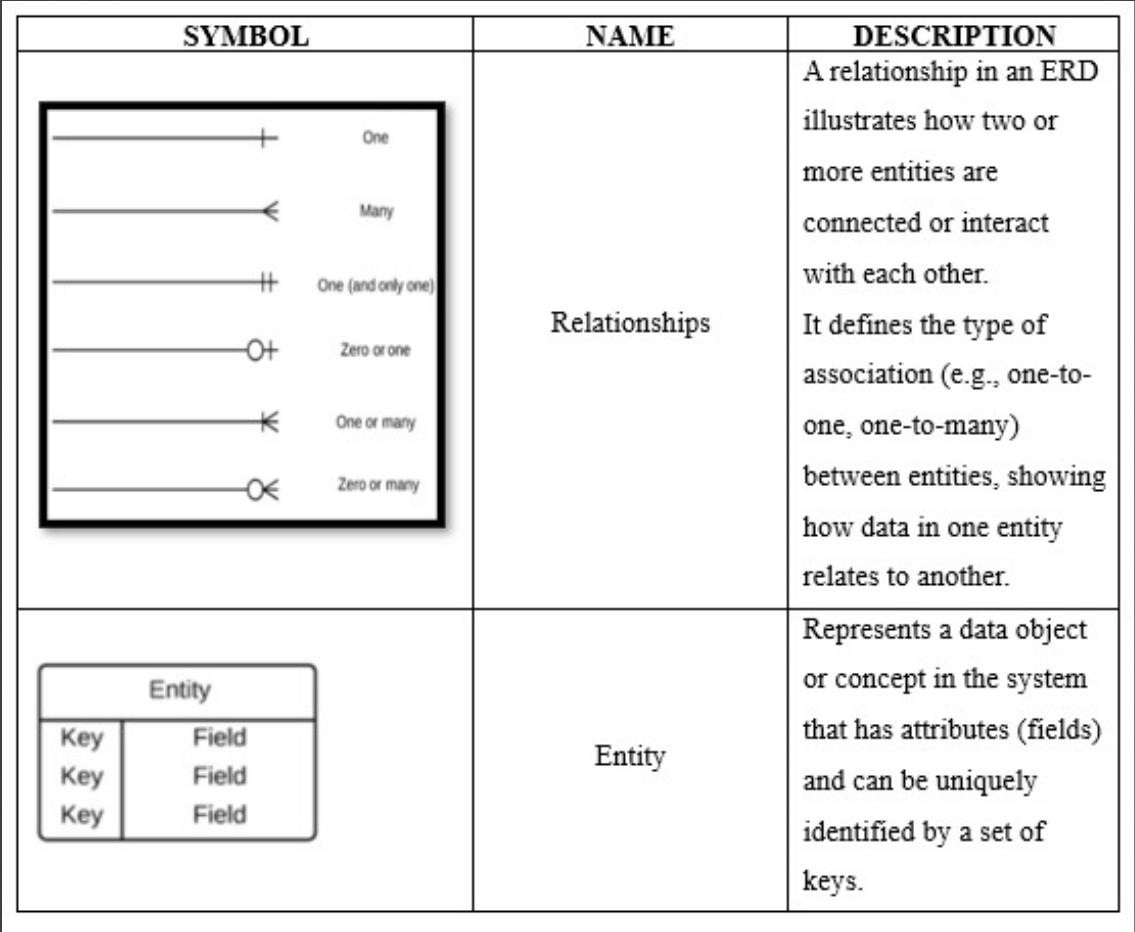


Figure 4.2 Use Case Diagram

### ENTITY RELATIONSHIP DIAGRAM

An Entity-Relationship (ER) Diagram is a visual representation of the relationships between entities in a database. It defines the structure of data by showing entities, their attributes, and the connections between them, helping to design and model the database schema.



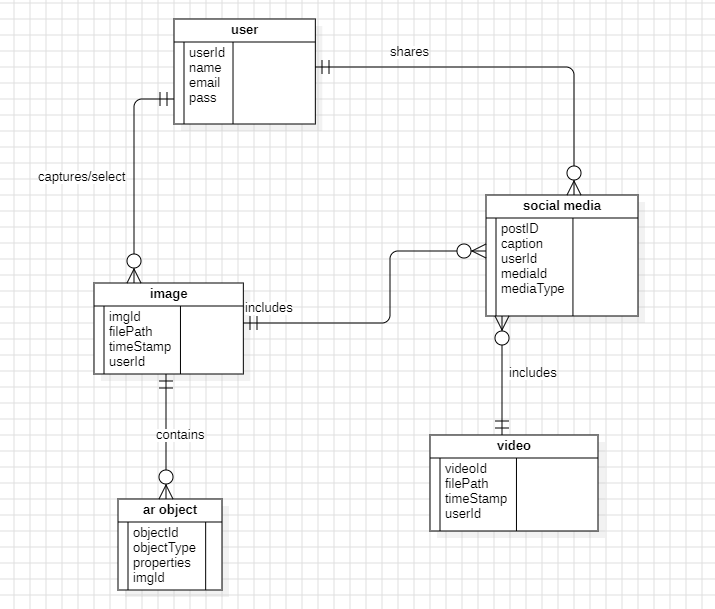
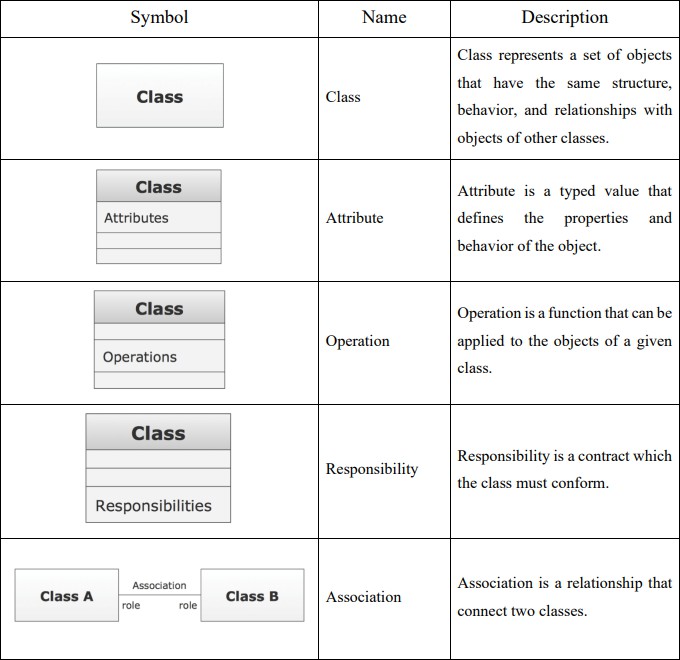


Figure 4.3 Entity Relationship Diagram

### CLASS DIAGRAM

Class diagrams are used to describe the structure of the system. Classes are abstractions that specify the common structure and behaviour of a set of objects. Class diagrams describe the system in terms of objects, classes, attributes, operations and their associations.



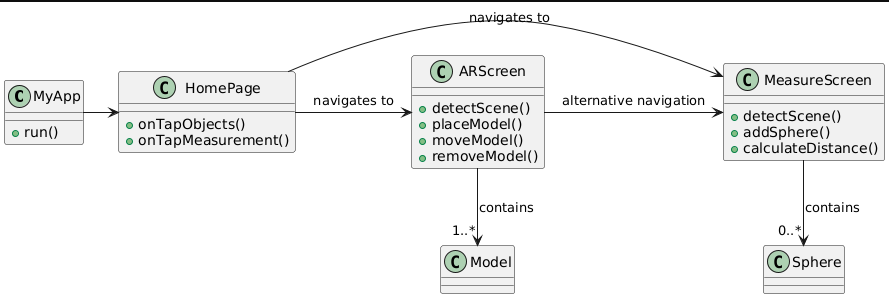
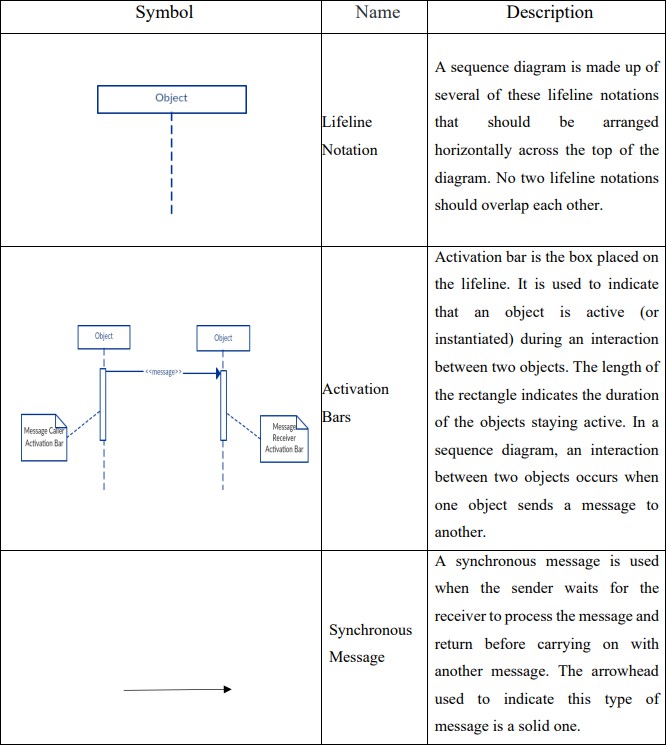


Figure 4.4 Class Diagram

### SEQUENCE DIAGRAM

The purpose of a sequence diagram in UML is to visualize the sequence of a message flow in the system. The sequence diagram shows the interaction between two lifelines as a timeordered sequence of events



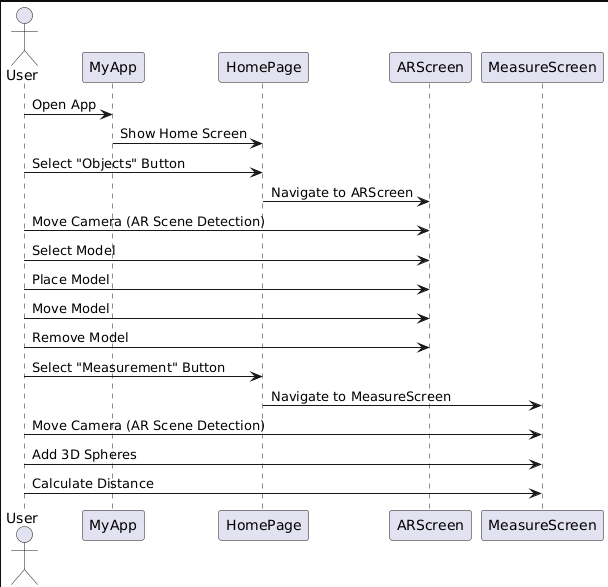
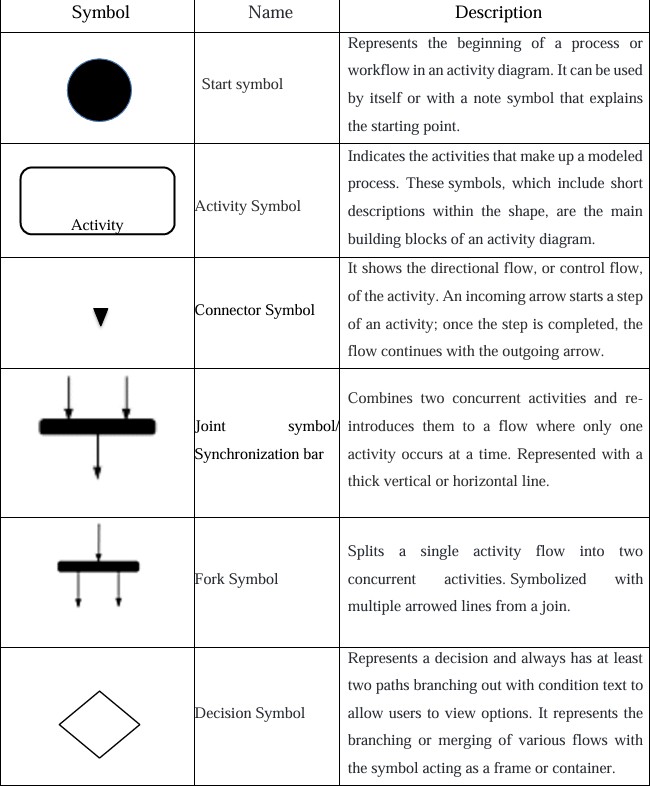
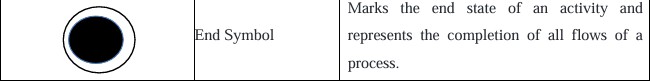


Figure 4.5 Sequence diagram

### 4.7 ACTIVITY DIAGRAM

An activity diagram visually represents the workflow of a system, showing the sequence of activities, decision points, and parallel processes in a clear, step-by- step manner.





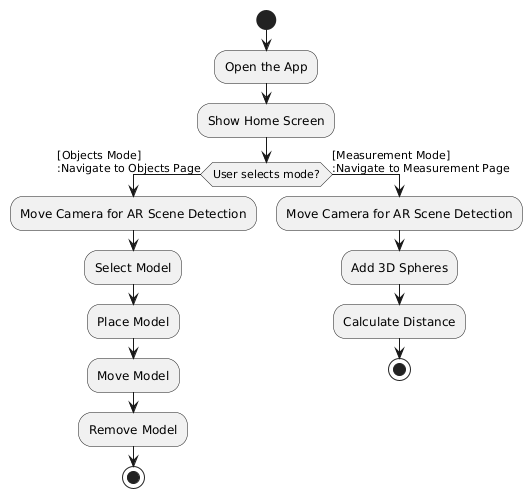
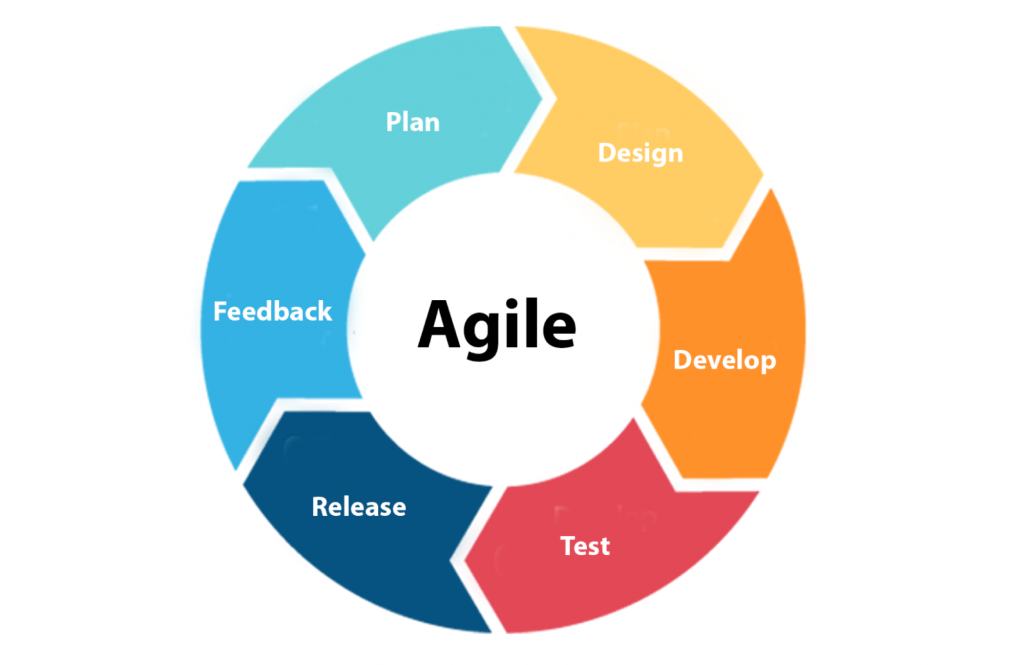


Figure 4.7 Activity Diagram

CHAPTER 5

**5.1 Implementation**

The project follows the **Agile Software Development Model** .



Here's why:

**1. Incremental Development:**

* Features are developed incrementally, such as:
  + Adding basic AR plane detection.
  + Placing nodes on the detected planes.
  + Calculating and displaying distances.
  + Adding interactive UI components like buttons for clearing nodes, capturing the scene, etc.
* Each feature is built and tested before moving to the next.

2. Iterative Approach:

* The project progresses in iterations:
  + Start with a simple AR environment setup.
  + Add marker placement functionality.
  + Add distance measurement and visual line rendering.
  + Continuously refine and test the implementation for performance and usability.

3. Focus on User Feedback:

* The iterative and incremental approach allows integrating user feedback at each stage to enhance usability, such as UI refinements, button placement, or performance improvements.

4. Flexibility:

* Adjustments can easily be made to accommodate changes, such as:
  + Adding additional functionality (e.g., capturing screenshots, sharing measurements).
  + Improving the visual representation of the measured line.

5. Prototype-Driven Development:

* Early prototypes or proof-of-concepts are likely developed to test ARCore functionality and UI designs before moving to full implementation.

**5.2 Testing**

**5.2.1 Integration Testing**

Integrated Testing, also known as Integration Testing, is a level of software testing where individual units or components of a software application are combined and tested as a group. The purpose of integrated testing is to verify that the interactions between these components work as expected and to uncover any defects that may arise due to integration issues.

Integrated Testing can be further categorized into different approaches:

* Top-Down Integration Testing: In this approach, testing starts from the top of the application's hierarchy and gradually moves downwards, integrating and testing components along the way.
* Bottom-Up Integration Testing: This approach begins with testing individual units or components at the
* lowest level of the hierarchy and gradually integrates higher-level components, testing them together.
* Big Bang Integration Testing: In this approach, all components are integrated simultaneously, and the application is tested as a whole. This method is typically used when the software application is small or when integration issues are minimal.

Integration Testing helps ensure that different parts of the software application work together seamlessly, and it is crucial for identifying and resolving integration issues early in the development process.

**5.2.2 Widget tests**

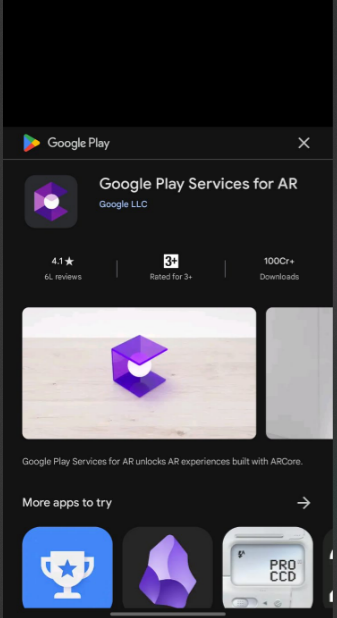
A *widget test* (in other UI frameworks referred to as *component test*) tests a single widget. The goal of a widget test is to verify that the widget's UI looks and interacts as expected. Testing a widget involves multiple classes and requires a test environment that provides the appropriate widget lifecycle context. For example, the Widget being tested should be able to receive and respond to user actions and events, perform layout, and instantiate child widgets. A widget test is therefore more comprehensive than a unit test. However, like a unit test, a widget test's environment is replaced with an implementation much simpler than a full-blown UI system.

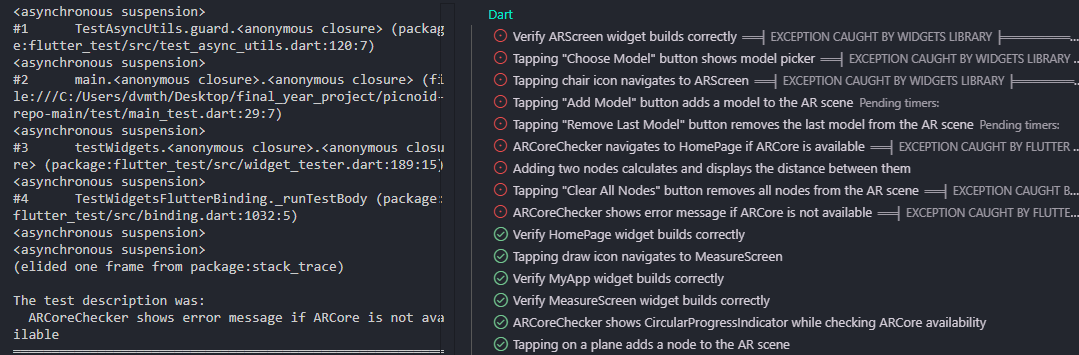
**5.3 Test Cases**

| **Test Case ID** | **Test Type** | **Test Case Description** |
| --- | --- | --- |
| **TC001** | **Widget Test** | **Verify that MyApp widget builds correctly and contains ARCoreChecker as the home.** |
| **TC002** | **Widget Test** | **Check that ARCoreChecker shows a CircularProgressIndicator while checking ARCore availability.** |
| **TC003** | **Widget Test** | **Ensure ARCoreChecker navigates to HomePage if ARCore is available.** |
| **TC004** | **Widget Test** | **Ensure ARCoreChecker shows an error message if ARCore is not available.** |
| **TC005** | **Widget Test** | **Verify that HomePage widget builds correctly and contains the expected UI elements.** |
| **TC006** | **Widget Test** | **Check that tapping the chair icon navigates to ARScreen.** |
| **TC007** | **Widget Test** | **Check that tapping the draw icon navigates to MeasureScreen.** |
| **TC008** | **Widget Test** | **Verify that ARScreen widget builds correctly and contains the expected UI elements.** |
| **TC009** | **Widget Test** | **Ensure that tapping the "Choose Model" button shows the model picker.** |
| **TC010** | **Widget Test** | **Ensure that tapping the "Add Model" button adds a model to the AR scene.** |
| **TC011** | **Widget Test** | **Ensure that tapping the "Remove Last Model" button removes the last model from the AR scene.** |
| **TC012** | **Widget Test** | **Verify that MeasureScreen widget builds correctly and contains the expected UI elements.** |
| **TC013** | **Widget Test** | **Ensure that tapping on a plane adds a node to the AR scene.** |
| **TC014** | **Widget Test** | **Ensure that adding two nodes calculates and displays the distance between them.** |
| **TC015** | **Widget Test** | **Ensure that tapping the "Clear All Nodes" button removes all nodes from the AR scene.** |
| **TC016** | **Integration Test** | **Verify the entire app flow from launching the app to navigating between screens.** |
| **TC017** | **Integration Test** | **Verify the AR model selection and placement functionality.** |
| **TC018** | **Integration Test** | **Verify the functionality from adding nodes to calculating distances.** |

CHAPTER 6

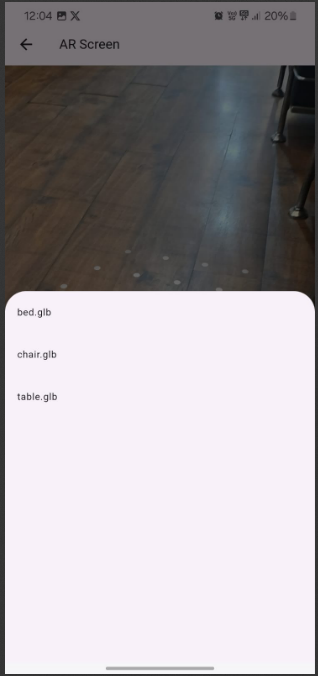
**6.1 Tests resuls**





* 1. **User documentation**

1. Open the app
2. Choose either one : object’button or measurement button
3. Once you go to either page make sure to move your camera around for a while in circular motion for AR scene detection.
4. When objects button gets clicked, it navigates to the page where you can interact with the models like select them, place, move and remove them.

1. When measurement button gets clicked, it leads to page where you can add 3D spheres to calculate the distance.



CHAPTER 7

**7.1 Cost Analysis**

Function Point Analysis (FPA) is a structured technique for estimating the cost of a software project based on the functionalities it provides. This method is useful for your **Flutter + ARCore project** because it helps estimate development effort in terms of function points (FP), which can then be converted into cost.

**Step 1: Identify Function Types and Assign Weights**

Function Points are calculated based on different functional components in the system. These are divided into **five categories**:

| **Function Type** | **Count** | **Complexity (Low/Avg/High)** | **Weight (FPs)** | **Total FP** |
| --- | --- | --- | --- | --- |
| **External Inputs (EI)** | 3 (User Inputs) | Average | 4 | 3 × 4 = **12** |
| **External Outputs (EO)** | 3 (AR Scene, 3D Models, Distance Calculation) | High | 7 | 3 × 7 = **21** |
| **External Inquiries (EQ)** | 2 (Fetching AR Data, Measuring Objects) | Average | 4 | 2 × 4 = **8** |
| **Internal Logical Files (ILF)** | 1 (User Preferences) | Low | 7 | 1 × 7 = **7** |
| **External Interface Files (EIF)** | 2 (ARCore API, Firebase) | High | 5 | 2 × 5 = **10** |

**Total Function Points (Unadjusted)**

12+21+8+7+10=∗∗58FP∗∗12 + 21 + 8 + 7 + 10 = \*\*58 FP\*\*12+21+8+7+10=∗∗58FP∗∗

**Step 2: Adjust Function Points Based on Complexity Factors**

We apply a complexity factor (0.65 to 1.35) based on project characteristics. Since **this project with moderate complexity**, we assume a **complexity factor of 1.0**.

Adjusted FP=58×1.0=58

**Step 3: Estimate Effort (Person-Hours)**

Using industry standards:

Effort (Hours) = Adjusted FP × Productivity Factor

* Productivity Factor = 3 Hours per FP
* Effort Required = 58 × 3 = 174 Hours

**Step 4: Calculate Development Cost**

Using an estimated **developer rate of ₹200 per hour (for a college project)**:

Total Cost = 174×200 = ₹34,800

**2️⃣ Final Cost Estimation**

| **Category** | **Estimated Cost (INR)** |
| --- | --- |
| **Development Effort Cost** | ₹34,800 |
| **Software Tools (Firebase, Cloud, ARCore Testing)** | ₹5,000 |
| **Testing & Miscellaneous Costs** | ₹2,000 |
| **Documentation & Report** | ₹3,000 |
| **Total Estimated Cost** | **₹45,000** |

CHAPTER 8

**Conclusion**

The project successfully integrates augmented reality (AR) technology with an intuitive user interface to deliver an engaging experience. Features such as model placement and interactive measurements demonstrate the practical application of AR in solving real-world problems. Built using Flutter and ARCore, the project showcases cross-platform capabilities and efficient resource management, making it accessible and scalable.

This application highlights the growing potential of AR technologies to enhance user interaction and productivity, setting the stage for further development in innovative and impactful directions.

Future Scope

1. Advanced AR Features:
   * Object Recognition: Incorporate object recognition to interact dynamically with real-world objects.
   * Persistent AR Experiences: Enable surface mapping and cloud anchors for multi-user shared AR environments.
2. Cross-Platform Support:
   * Extend compatibility with ARKit to support iOS devices, broadening the user base.
3. Integration with Firebase:
   * Real-Time Database: Store and synchronize AR scenes or measurement data in real-time.
   * Cloud Firestore: Enable efficient data management for storing user preferences and custom AR settings.
   * Authentication: Provide user sign-in functionality using Firebase Authentication for a personalized experience.
   * Cloud Storage: Allow users to upload, store, and retrieve custom 3D models securely.
   * Crashlytics: Monitor and improve app stability with real-time crash reporting.
4. Enhanced User Interaction:
   * Introduce gesture-based controls for scaling, rotating, and moving AR models.
   * Enable voice-command functionality for better accessibility.
5. Machine Learning Integration:
   * Use ML models for object detection and scene understanding.
   * Provide real-time recommendations based on user activity or recognized environments.
6. Measurement Enhancements:
   * Expand measurement tools to include area and volume calculations.
   * Introduce accuracy improvement techniques for industrial use cases.
7. E-Commerce Integration:
   * Develop AR solutions for virtual try-on experiences, such as furniture placement or fashion items.
   * Integrate AR-based advertisements or product promotions to attract businesses.
8. Customization Options:
   * Enable users to personalize their AR experience by uploading and interacting with their own 3D models.
9. Collaborative AR:
   * Implement multi-user interactions in AR environments using Firebase cloud anchors and real-time synchronization.

By combining ARCore with Firebase and expanding its functionalities, the project can evolve into a versatile and impactful platform suitable for various industries, such as e-commerce, education, and architecture. These enhancements will drive higher user engagement and enable new business opportunities.

# RESEARCH PAPER

PicNoid: A Comprehensive Augmented Reality and Image Processing Application for Android

#1Devam Thakar , #2Pragati Thawani *#BSc IT Department, HSNC University Colaba, Mumbai, India*

[1dvmthakar@gmail.com](mailto:1dvmthakar@gmail.com)

[2pragati.thawani@kccollege.edu.in](mailto:2pragati.thawani@kccollege.edu.in)

##### ABSTRACT

The proliferation of mobile applications has led to a fragmented user experience where functionalities such as image processing, social media sharing, augmented reality (AR) interactions, and measurement tools are isolated across different platforms. PicNoid aims to address this challenge by consolidating these diverse features into a single, cohesive application for Android users. Leveraging Google’s ARCore API, PicNoid offers an integrated platform that enhances user convenience, engagement, and creativity. Key functionalities include image capture and editing, social media sharing, immersive AR experiences with popular characters and objects, creative face filters, real-time object measurement, image-to-video transformation, and playing videos within AR scenes. Developed using Flutter in Android Studio, and incorporating advanced libraries like OpenCV and TensorFlow Lite, PicNoid stands out as a versatile, all-in-one solution. This paper discusses the methodology adopted for requirement analysis, design, development, and testing, and explores the real-world applications of PicNoid in areas such as social media, interior design, education, e- commerce, health, architecture, event management, and the creative arts. PicNoid’s comprehensive approach promises to revolutionize the mobile application landscape by delivering an unmatched user experience through its innovative integration of AR and image processing capabilities.

Keywords: Augmented Reality, Image Processing, Mobile Applications, ARCore, Social Media, Real- time Measurement, Face Filters, Android Development.

##### INTRODUCTION

The current landscape of mobile applications is characterized by a fragmented experience, where users must switch between multiple apps to access functionalities such as image processing, social media sharing, augmented reality (AR) interactions, and measurement tools. This leads to inefficiencies and a disjointed user experience. PicNoid addresses this issue by consolidating these diverse functionalities into a single, cohesive application for Android users. By leveraging advanced technologies such as Google’s ARCore API, OpenCV, and TensorFlow Lite, PicNoid offers a comprehensive platform that enhances user convenience, engagement, and creativity. This report explores the development, features, real-world applications, and critical issues associated with PicNoid.

##### ROLE

PicNoid plays a crucial role in redefining the mobile application landscape by integrating multiple functionalities into one application. It serves various roles, including:

**Content Creation Tool**: Allows users to capture, edit, and share images and videos seamlessly on social media platforms.

**Augmented Reality Platform**: Provides immersive AR experiences with popular characters, objects, and interactive video scenes.

**Creative Enhancement**: Offers creative face filters and image-to-video transformation features, encouraging user creativity and engagement.

**Measurement Tool**: Facilitates real-time object measurement, making it useful for practical applications in architecture, interior design, and DIY projects

**Entertainment and Education**: Acts as an interactive tool for entertainment and educational purposes, enhancing user experiences through augmented reality.

##### APPLICATIONS

PicNoid's integrated features have broad applications across various domains:

* + - 1. **Social Media and Entertainment**:

**Enhanced Content Creation:** Users can create and share engaging content using advanced image processing, face filters, and AR characters.

Interactive Storytelling: Integrates AR characters and objects for creating interactive and engaging stories and videos.

* + - 1. **Interior Design and Visualization**:

**Virtual Placement of Furniture**: Similar to IKEA’s AR app, users can visualize furniture and objects in their real-world space before purchasing.

* + - 1. **Education and Training**:

**Interactive Learning**: Utilizes AR characters and objects to create engaging educational content for students.

**Skill Training**: Supports practical training scenarios with real-time measurement tools and AR simulations.

* + - 1. **E-commerce and Marketing**:

**Product Visualization**: Allows customers to visualize products in their own environment, enhancing confidence in online purchases.

**AR Advertising:** Enables marketers to create immersive AR advertisements that capture user attention more effectively.

* + - 1. **Health and Fitness**:

**Virtual Workouts**: Users can engage in virtual workout sessions with AR trainers or characters, making fitness routines more interactive and enjoyable.

**Health Monitoring**: Adaptable for monitoring certain health metrics through image processing capabilities.

* + - 1. **Measurement and Architecture**:

**Real-time Measurements**: Architects and DIY enthusiasts can take precise measurements of spaces and objects without physical tools.

**Project Planning**: Assists in planning renovations and construction projects by visualizing modifications and measuring spaces.

* + - 1. **Event Management and Tourism**:

**Virtual Tours**: Users can take virtual tours of tourist destinations or event venues through AR, enhancing their planning and experience.

**Interactive Guides**: Provides interactive guides and information at museums, historical sites, and events, enriching visitor experiences.

* + - 1. **Creative Arts**:

**Digital Art and Animation**: Supports artists in creating digital art and animations by combining image processing with AR features.

**Augmented Performances**: Allows performers to incorporate AR elements into live shows for an immersive audience experience.

##### CRITICAL ISSUES

Despite its innovative features, PicNoid faces several critical issues:

* + - 1. **Technological Challenges**:

**Performance and Optimization**: Ensuring smooth performance and optimal resource usage on various Android devices

**Integration of APIs and Libraries**: Effective integration of ARCore, OpenCV, TensorFlow Lite, and other libraries without conflicts.

* + - * 1. **User Experience**:

**Intuitive Design**: Creating a user interface that is easy to navigate despite the complexity of the integrated features.

**User Feedback and Adaptation**: Continuously adapting the app based on user feedback to improve usability and functionality.

* + - * 1. **Data Privacy and Security**:

**Handling Sensitive Data**: Ensuring that user data, especially images and videos, is handled securely and in compliance with privacy regulations.

**Secure Sharing**: Providing secure options for sharing content on social media platforms.

* + - * 1. **Market Competition**:

**Standing Out**: Differentiating PicNoid from other apps in a saturated market with similar functionalities.

**Continuous Innovation**: Keeping the app updated with the latest technological advancements to stay ahead of competitors.

* + - * 1. **User Adoption**:

**Initial Adoption**: Encouraging users to switch from multiple apps to PicNoid for a unified experience.

**Retention and Engagement**: Ensuring that users remain engaged with the app through regular updates and new features

V : Work Flow



Social Media

User

Picnoid

Figure. Flow diagram

VI. Future Scope

**Image Editing Enhancement**

**Advanced Features:** Introduce professional-grade tools like color correction, noise reduction, and object removal.

**AI Integration:** Utilize AI for automated tasks such as background removal and style transfer.

**User Customization:** Allow users to create and share custom filters and presets.

**Augmented Reality Expansion**

**Interactive Experiences:** Develop AR games, quizzes, and virtual try-ons.

**Rich Object Library:** Offer a vast collection of 3D models for AR integration.

**Collaborative AR:** Enable real-time AR project collaboration.

**Social and Commercial Growth**

**Community Building:** Foster a user community for sharing and discovering AR creations.

**Strategic Partnerships:** Collaborate with social media platforms and influencers.

**Monetization:** Explore in-app purchases, subscriptions, and advertising.

**Diversification**

**Video Focus:** Expand to video editing with AR effects.

**Immersive Experiences:** Develop VR and AR experiences beyond image editing.

**New Market Ventures:** Explore e- commerce integration and educational AR content.

##### CONCLUSION

PicNoid integrates augmented reality and image processing into a single mobile application, enhancing user experience by consolidating diverse functionalities. Leveraging advanced technologies

like ARCore and OpenCV, PicNoid offers versatile applications in social media, education, and e- commerce. While facing challenges such as performance optimization and market competition, PicNoid is well-positioned to redefine mobile interactions. Continued innovation and user feedback will be key to its success, ensuring it remains a leading solution for engaging and creative mobile experiences.

##### ACKNOWLEDGMENT

This research was supported Kishinchand Chellaram College, HSNC University.

REFERENCES

https://developers.google.com/ar/develop/fundame ntals

https[://www.tensorflow.org/](http://www.tensorflow.org/)

https[://www.ikea.com/global/en/newsroom/innovati](http://www.ikea.com/global/en/newsroom/innovati) on/ikea-launches-ikea-place-a-new-app-that- allows-people-to-virtually-place-furniture-in-their- home-170912/

https[://www.lenskart.com/eyeglasses/promotions/ar](http://www.lenskart.com/eyeglasses/promotions/ar)

-eyeglasses.html https://firebase.google.com/

https://appleinsider.com/articles/21/07/22/apples- measure-app-may-gain-instant-automatic- measurements-with-ar