SDD DOCUMENT FOR AUGMENTED REALITY

Introduction:

Purpose:

- The purpose of this document is to outline the design aspects of the AR modeling project, which involves creating an application for augmented reality modeling.

Scope:

- This document covers the architectural and design considerations for developing an AR modeling application using Unity platform for frontend development, Firebase for backend database management, and Figma for UI/UX design.

Definitions, Acronyms, and Abbreviation:

- AR: Augmented Reality
- Firebase: Google's mobile and web application development platform
- Figma: Collaborative interface design tool

- Unity: A cross-platform game engine

Reference:

- IEEE std 1016-2009 SDD Document.

Overview:

- This document provides an overview of the system architecture, user interface design, data design, and deployment design for the AR modeling application.

System Overview:

System Context:

- The AR modeling application allows users to create and manipulate 3D models in augmented reality using their mobile devices.
- Users can place the 3D models in the real life scenario to check whether its compatible or not.

System Functionality:

- Users can interact with the application to place, rotate, and scale 3D models in real-world environments.
- Models created by users are stored and managed in the Firebase database.

- Users can tap on the place where he wants the model image to be placed.

User Characteristics:

- Target users include hobbyists, designers, and anyone interested in creating AR models.
- Users should have basic information about the application, how to use the AR models and how to address them.
- Users should have basic information about how to use the application.

Constraints:

- Performance may vary based on the hardware capabilities of the user's device.
- Network connectivity is required for accessing and saving models to the Firebase database.

Assumptions and Dependencies:

- The application assumes users have basic familiarity with mobile devices and augmented reality technology.
- The availability of stable internet connectivity for Firebase database access.

- The user authentication should be necessary inorder to access the application.

Architectural Design:

Overview:

- This section describes the overall architecture of the AR modeling including hardware and software components, interfaces, and data flow between them.

Hardware/Software Mapping:

- Hardware: Mobile devices with ARCore or ARKit support.
- Software: Unity for frontend development, Firebase for backend database management.

Data Architecture:

- User-created models are stored in the Firebase Realtime Database or Firestore.
- User related information is also stored in Firebase Realtime Database.

Access Control and Security:

- Firebase Authentication is used for user authentication and access control to ensure data security.

User Interface Design:

Overview:

- The user interface design focuses on simplicity and intuitive interaction for creating and manipulating 3D models in AR.
- We will be using Figma for the creation of the user Interface Design.

Interface Design:

- UI elements are designed using Figma, following modern design principles and guidelines for mobile applications.

Interface Features:

- Tap to place: Users can tap on surfaces in the AR environment to place models.

- Swipe gestures: Navigation between menu options and model manipulation features.
- If the user doesn't have any 3D models to place he can use our 3D models which are available in our database.

Interface Identification and Description :

- Main menu: Allows users to create new models, access saved models, and adjust settings.
- Model manipulation controls: Gestures for placing, and scaling 3D models in AR space.

Data Design:

Overview:

- This section describes the data design of AR modeling. It includes a description of the data used by the system, a data dictionary, and information about data storage, access, and manipulation.

Data Description:

- Unity frontend components: UI elements, AR scene rendering, model manipulation scripts.

- Firebase backend components: Authentication, database management, cloud functions for server-side logic.

Data Storage:

- Models and metadata are stored in Firebase Realtime Database or Firestore, with backups and data integrity measures in place.
- The user information will be stored in the firebase servers and the security will be assured.

Component Design:

Component Description:

- This section provides a detailed description of the components that make up the AR modeling. The system consists of the following components.

User interface component:

- This component is responsible for handling all user interactions with the system, including registering for using the AR models and giving the user information which is useful for user authentication.

Storage component:

- This component is responsible for the storage of data if the user does not provide any ar models he can access the models which are present in our database.

Component Functionality:

- Each component performs specific tasks related to its functionality, following the principles of separation of concerns.

Interface Description:

- Components communicate through well-defined interfaces, adhering to established protocols for data exchange.

Account Registration:

- This allows users to create an account and provide necessary information.

Display Information:

- This allows users to display their models in real life by just tapping on the screen.

Component Implementation:

- Unity components are implemented using C# scripts, while Firebase components are configured using Firebase SDKs and APIs.

Software Infrastructure:

- Database management system: Firebase
- Programming languages: Python,C#
- Frontend:Figma,Java script,HTML and CSS

Deployment Design:

Overview:

- The deployment design outlines the process of distributing the application to end-users and managing server-side infrastructure.

Security Infrastructure:

- To ensure the security the following measures has been taken for ar modeling application:

- User Authentication:

All users of the system will be required to authenticate themselves using a username and password.