

---

**Group 7**

---

**Virtual Chat 3D  
Software Architecture Document**

**Version 1.1**

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

## Revision History

Date	Version	Description	Author
18/11/2024	1.0	Introduction and Architectural Goals and Constraints	Lý Nghị Hoàng
21/11/2024	1.1	Logical View	Lý Nghị Hoàng Châu Vĩnh Đạt Trần Đăng Tuấn

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

## Table of Contents

<b>1. Introduction</b>	<b>4</b>
<b>2. Architectural Goals and Constraints</b>	<b>4</b>
<b>3. Use-Case Model</b>	<b>6</b>
<b>4. Logical View</b>	<b>7</b>
4.1 Component: UserAccount	10
4.2 Component: Admin	12
4.3 Component: Friend	13
4.4 Component: Message	14
4.5 Component: ChatBot	15
4.6 Component: FirebaseDatabase	16
4.7 Component: CharacterCustomize	16
4.8 Component: Network	17
4.9 Component: OnlineRuntime/RoomManagement	19
4.10 Component: OnlineRuntime/Input	20
4.11 Component: OnlineRuntime/CharacterControl	20
4.12 Component: OnlineRuntime/EnvironmentCustomize	21
4.13 Component: OnlineRuntime/Activities/MiniGame	22
4.14 Component: OnlineRuntime/Activities/Voting	23
4.15 Component: OnlineRuntime/Activities/Whiteboard	24
4.16 Component: OnlineRuntime/Message	25
4.17 Component: Server/Network and Server/FirebaseDataModel	26
<b>5. Deployment</b>	<b>27</b>
<b>6. Implementation View</b>	<b>27</b>

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

# Software Architecture Document

## 1. Introduction

### - Purpose

This Software Architecture Document (SAD) outlines the high-level design of the Virtual 3D Chat application. It defines the architectural principles, components, and their interactions, as well as the technical decisions made to achieve the system's functional and non-functional requirements.

### - Scope

This document covers the following aspects of the application architecture:

- System Overview: A high-level description of the system and its components.
- Architectural Patterns: The patterns used to design the system.
- Technology Stack: The programming languages, frameworks, and tools used to implement the system.
- Security Architecture: The security measures implemented to protect the system and user data.
- Deployment Architecture: The deployment strategy and infrastructure requirements.

## 2. Architectural Goals and Constraints

### Architectural Goals:

1. **Performance:**
  - Ensure low-latency communication between clients and the server for smooth and responsive interactions.
2. **Usability:**
  - **Intuitive Interface:** Create a user-friendly interface that is easy to use without learning.
  - **Immersive Experience:** Provide a visually appealing and engaging 3D environment.
3. **Portability:**
  - **Cross-Platform Compatibility:** Ensure the application can run on various platforms: Android, iOS.
4. **Security:**
  - **Secure Authentication:** Implement robust authentication mechanisms to prevent unauthorized access.
  - **Encryption:** Encrypt sensitive data both in transit.

### Architectural Constraints:

1. **Real-time Communication:**
  - **SocketIO:** Use SocketIO for real-time communication between clients and the server.
  - **LAN Hosting Online Room:** Use Unity Netcode for hosting online rooms in LAN.
2. **3D Rendering Engine:**
  - **Unity:** Leverage Unity's powerful 3D rendering capabilities to create immersive environments with cross-Platform compatibility.
3. **User Experience:**
  - **User Interface:** Design an intuitive and user-friendly UI.
  - **Input Methods:** Support various input methods, including keyboard, mouse, and touch.
  - **Social Features:** Implement social features like friend lists, groups, and private messaging.
4. **Security**
  - **Encryption:** Build custom packet encryption in transit based on RSA and AES encryption algorithm.
  - **Authentication:** Use the firebase authentication to verify users and manage user accounts.
5. **Development Tools and Frameworks:**
  - **Client-Side Development**
    - Unity:

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

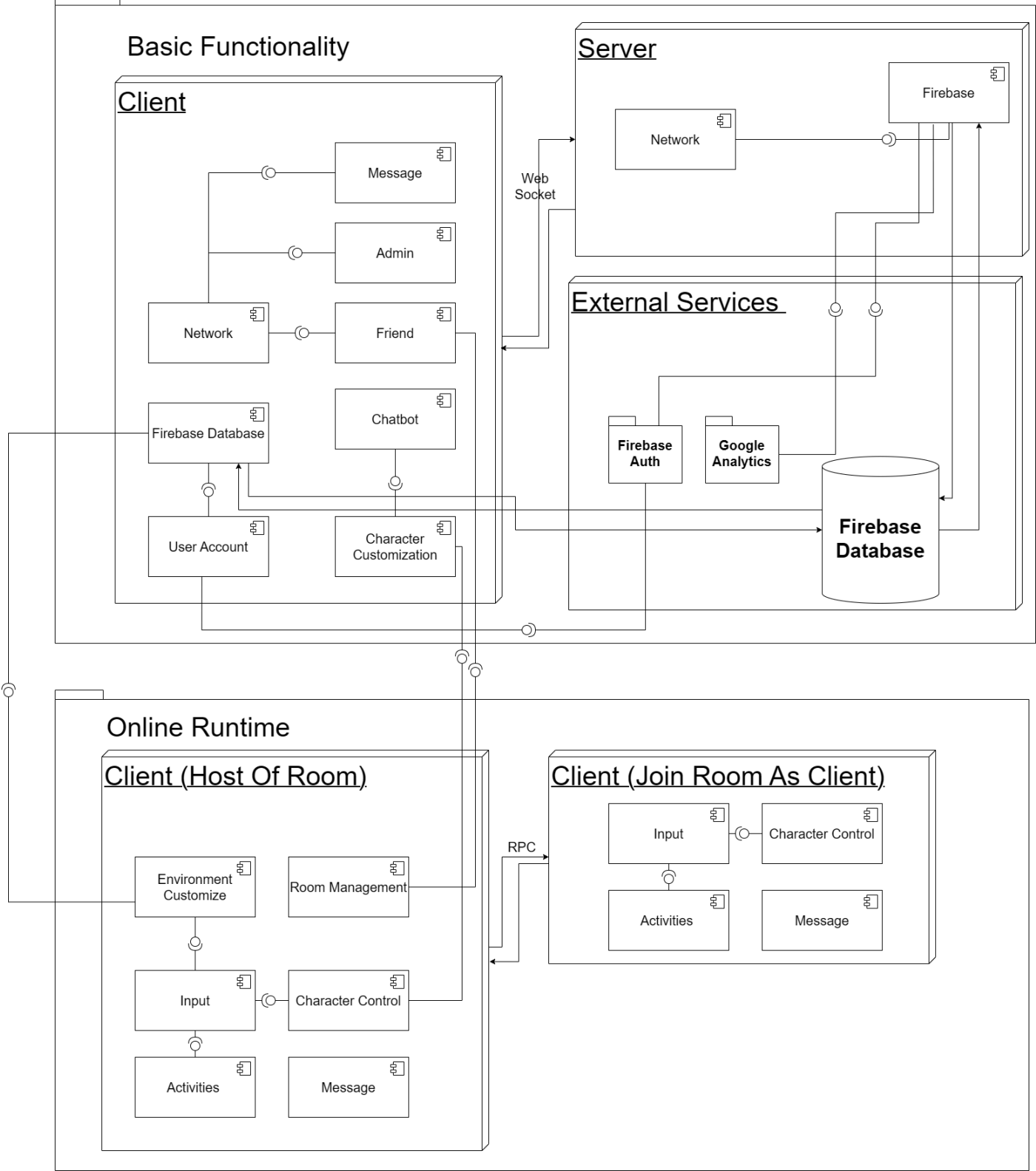
- Core Engine: Used for 3D game development, rendering, physics, and animation.
  - Visual Scripting: For creating game logic without writing code.
  - Asset Store: A marketplace for purchasing or downloading free 3D models, scripts, and other assets.
- JetBrains Rider:
  - C# IDE: For writing and debugging C# scripts.
  - Code Completion and Analysis: Provides intelligent code suggestions and error detection.
  - Integration with Unity: Seamlessly integrates with Unity for efficient development.
- **Server-Side Development**
  - Node.js:
    - JavaScript Runtime: For building scalable and efficient server-side applications.
    - Asynchronous Programming: Enables handling multiple requests concurrently.
    - Large Ecosystem of Modules: Access to a wide range of libraries and frameworks.
  - Visual Studio Code:
    - Lightweight Code Editor: For writing and debugging JavaScript and TypeScript code.
    - Extensions: Supports various extensions for syntax highlighting, code completion, debugging, and version control.
    - Integration with Node.js: Provides tools for debugging, testing, and deploying Node.js applications.
- **Database**
  - Firebase:
    - Real-time Database: For storing and synchronizing data in real-time.
    - Authentication: For user authentication and authorization.
    - Cloud Storage: For storing user-generated content like avatars.
- **Version Control**
  - Git:
    - Distributed Version Control System: For tracking changes to source code and collaborating with other developers.
    - Popular Hosting Platforms: GitHub for hosting and managing code repositories.
- **Additional Tools**
  - Blender: For modify 3D models and animations if necessary.

By carefully considering these architectural goals and constraints, you can build a robust, scalable, and secure virtual 3D chat application.



Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

#### 4. Logical View



Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

There are 2 subsystem in the project:

**a. The Basic Functionality subsystem:**

- This subsystem provides the following features:
  - + Account authentication.
  - + User account profile management
  - + Social functionality: friend management, message with friend.
  - + 3D interactive Chatbot.
  - + It also provides firebase database access on client-side.
- The subsystem contains 2 parts:
  - + The Client - implemented with Unity.
  - + The Server - implemented using NodeJS.
  - + These parts interact with each other through the Interface provided by the Network Component using the Web Socket protocol provided by SocketIO.
- The Components in the Client Side are designed in the **MVC design pattern**:
  - + Model: a class to store all data for the Component.
  - + View: a class to interact with the Object provided by Unity and handle displaying.
  - + Control: a class to handle and control all the logic of the Component.
- Here is a brief description of the components in the Client Side of the subsystem:

Component (Client Side)	Description
UserAccount	Provide authentication and profile management for the user.
Friend	Provide friend management functions by reading data from the Firebase and display on the view. Real-time friend requests are handled by sending requests to the server.
Message	Provide real-time messaging functions by sending and receiving messages from the server. It also read past messages from the Firebase.
Admin	Provide administrator functionalities: lock/unlock users, reading statistics from Firebase.
FirebaseDatabase	Wraps up Firebase SDK for the Client to read directly from Firebase.
Chatbot	Provide interactive chat using Gemini APIs, and enable customization for the Chatbot model.
CharacterCustomization	Handle 3D models importing, manage imported models, provide an interface for the Chatbot Component and the Character Control Component (in the Online Runtime subsystem).
Network	Provide functions for Clients to communicate with the Server in cryptic manner.

- The Server Side utilizes the web socket with the SocketIO library to communicate with the Clients and use the Firebase Admin SDK to manage the Firebase's data, along with the built-in Google Analytics for Firebase to further extract information..

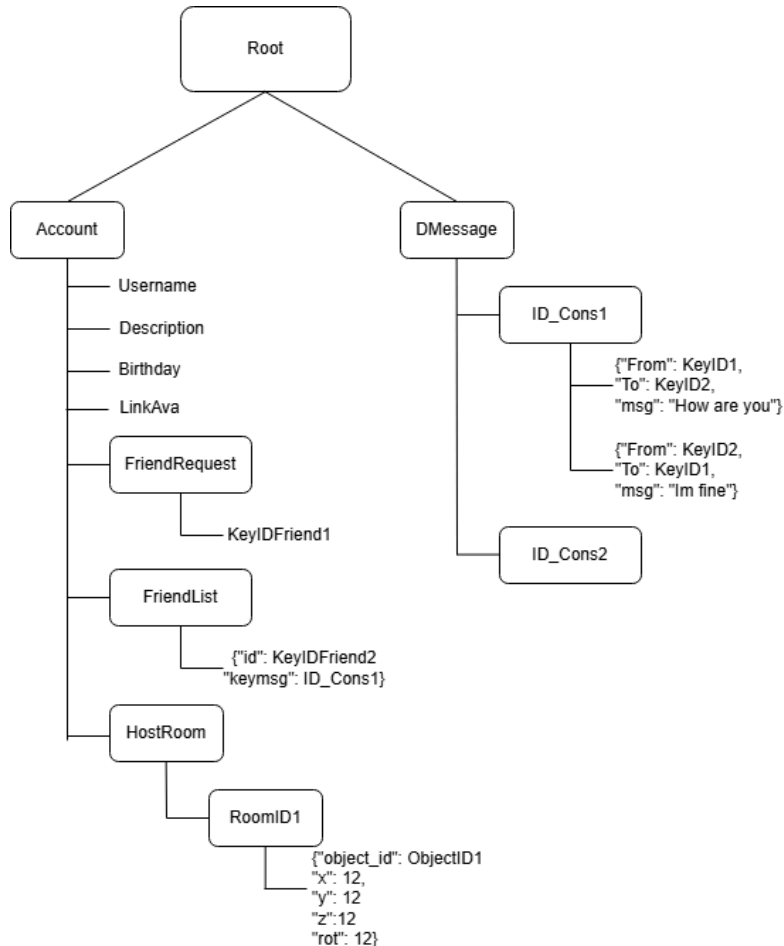
Component (Server Side)	Description
-------------------------	-------------



Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

Network	Provide functions for the Server to process request from Clients in cryptic manner: <ul style="list-style-type: none"> <li>- Direct Messages.</li> <li>- Friend Management.</li> <li>- Admin functionality.</li> </ul>
FirebaseDataModel	Provide: <ul style="list-style-type: none"> <li>- Functions to read/write the database and execute admin tasks</li> <li>- Functions to interact with the Google Analytics APIs and to read information on the server PC..</li> </ul>

Here is the structure of data stored in Firebase:



b. **Online Runtime subsystem:**

- This subsystem runs only on the Client Side.
- It provides the following online room features:
  - + Room management
  - + Environment customization
  - + Character in room control
  - + Messaging/Chatting
  - + Other Activities.
- The Client hosts or joins the chat room by using the RPC provided by Unity Netcode.

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

- Here is a brief description of the components of the subsystem:

Component	Description
RoomManagement	Provide functions for hosting or joining rooms, managing created rooms, room access management, and guests management (view joined users list and kick unwanted users).
EnvironmentCustomize	Provide functions for managing, editing objects in the room. It also utilizes the FirebaseDatabase Component in the Basic functionalities subsystem for storing room details in Firebase.
Message	Provide functions for chatting in a chat room.
Input	Provide the UI for users to control their 3D avatars in a chat room. (such as joystick, touch pad, etc.)
CharacterControl	Control the avatar's movement, animation, and emotion.
Activities	Provide functions for other activities in the chat room such as voting, whiteboard, mini games.

#### 4.1 Component: UserAccount

This component provides Firebase authentication and user profile management functionality, including:

- Sign up, login, sign out, reset password and get login token (use to send login state to server)
- View and edit profile: Avatar, Description, Username, Date of birth.

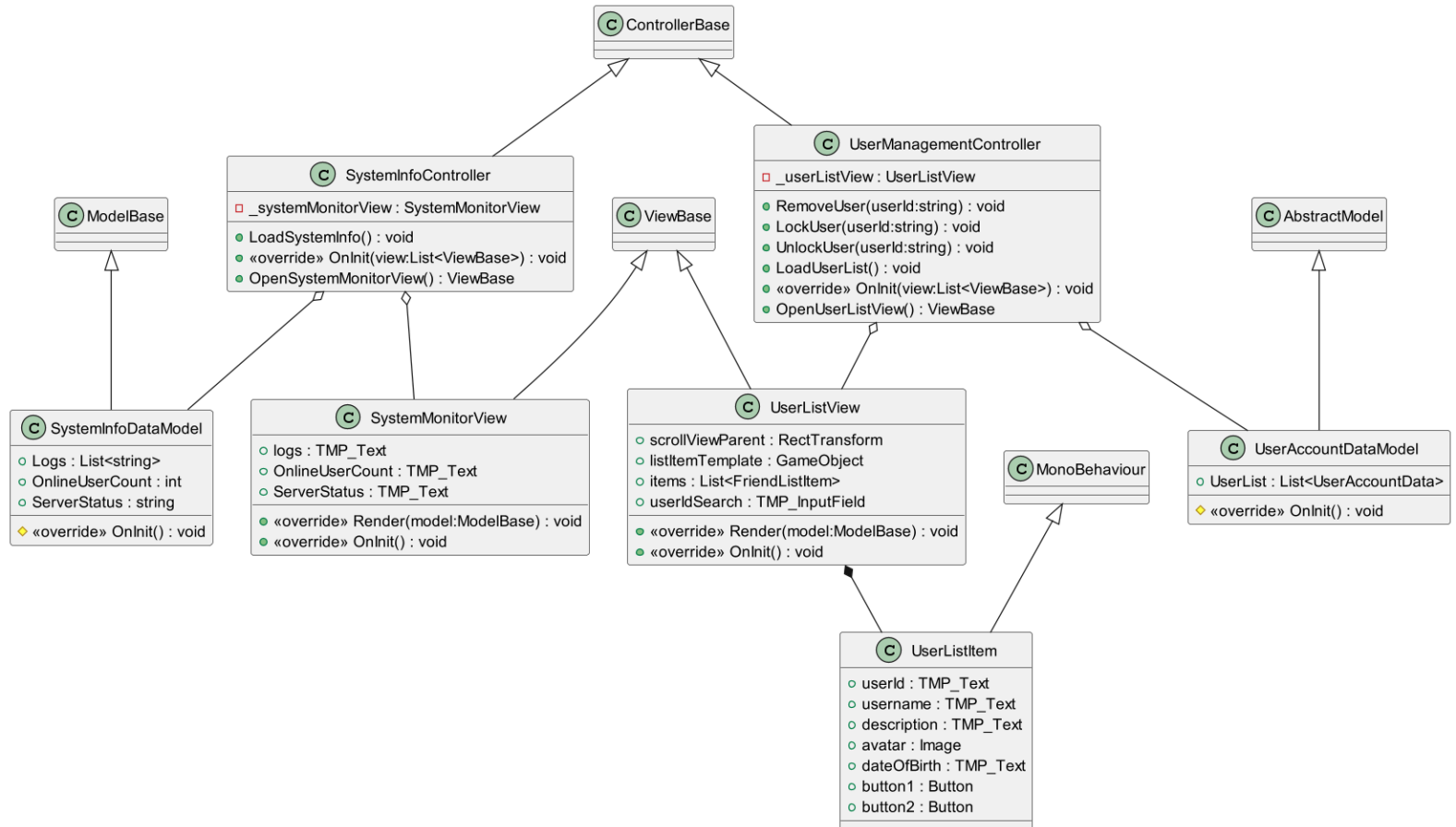


Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

## 4.2 Component: Admin

This component provides user management functionality. Including:

- View user list, lock/unlock user, remove user.
- System state monitoring



Controllers:

- **UserManagementController:**
  - Controls the logic for managing normal users: removing user, locking/unlocking user, loading user list. These functions are executed by calling the server side APIs.
  - Access to the OpenUserListView.
- **SystemInfoController:**
  - Controls the logic for viewing System information by fetching from the server and calling Google Analytics APIs.
  - Access to the OpenUserListView.

Views:

- **UserListView:**
  - Handles the interface for viewing users list functions.
  - Read the Admin's command for execution.
- **SystemMonitorView:**
  - View statistical information about the System and server.

Models:



Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

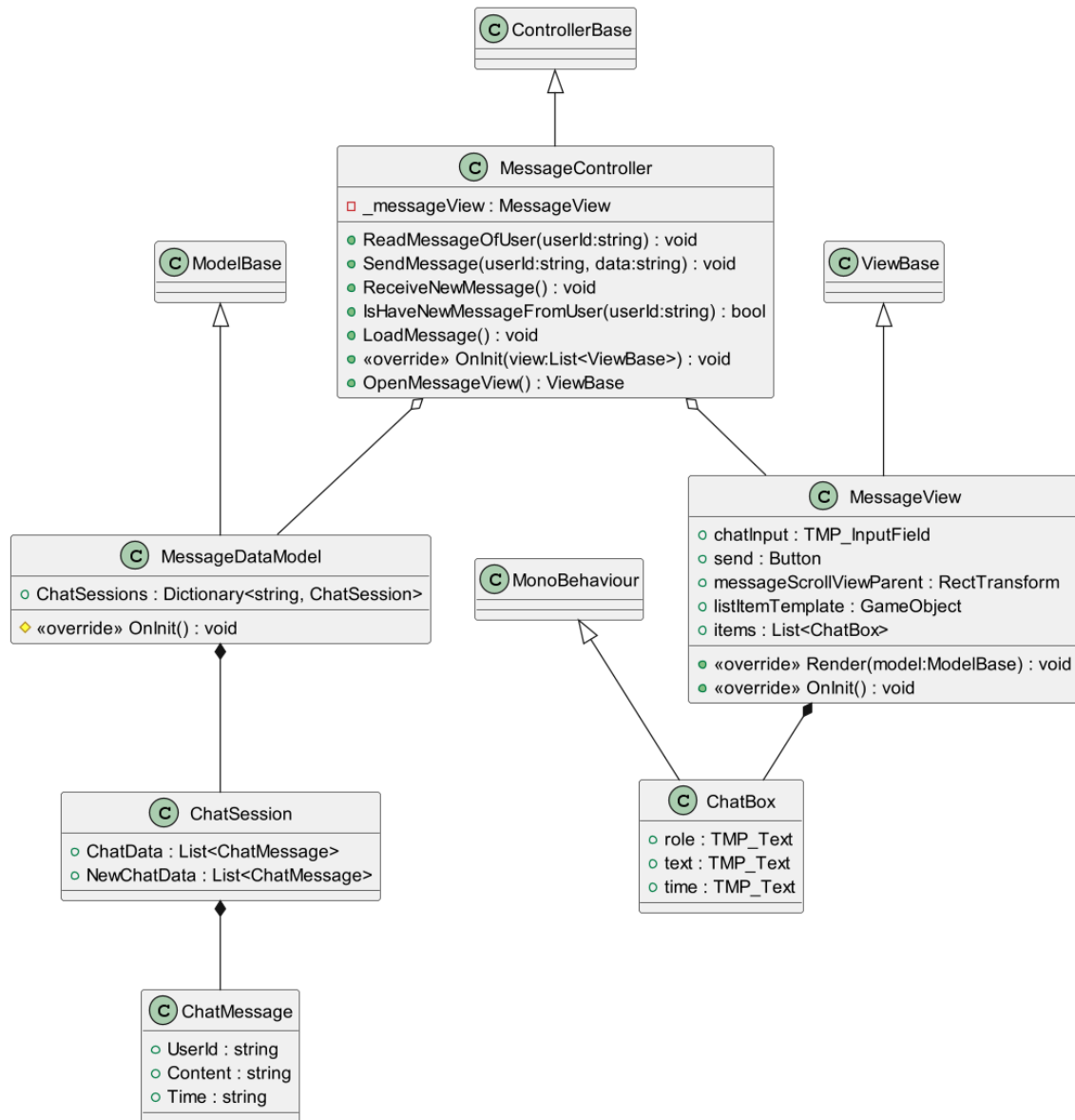
- Reads user input for adding friends and interacting with the friend list.

Models:

- FriendDataModel:
  - Stores the user's friend data, including details like friend IDs, statuses, and other related information.

#### 4.4 Component: Message

This component provides message functionality. Including sending and receiving messages between users.



Controllers:

- MessageController:
  - Manages the logic for handling messages between users, including reading, sending, and receiving new messages.
  - Handles message loading and checks if there are new messages from specific users.
  - Provides access to the MessageView to display and interact with chat messages.

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

Views:

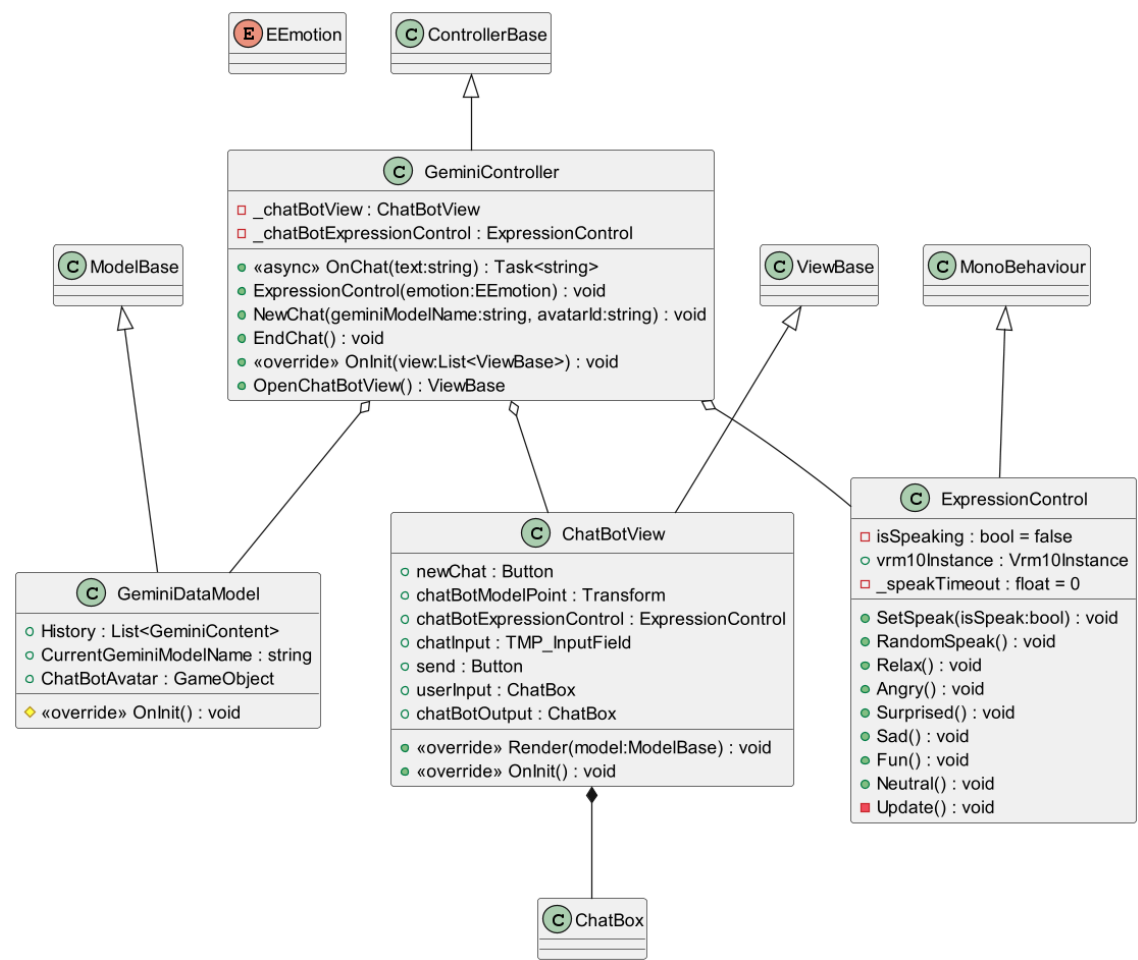
- MessageView:
  - Manages the user interface for chat interactions, including sending and viewing messages.
  - Displays a scrollable list of messages and allows user input through a text field and send button.
  - Dynamically renders chat items using a template for individual chat messages.

Models:

- MessageDataModel:
  - Stores the user's chat sessions as a dictionary, mapping user IDs to chat sessions (ChatSession).
  - Provides message data to the MessageController for operations and updates.

4.5 Component: ChatBot

This component provides ChatBot functionality based on Gemini API. Including interactive chat with Gemini, change Chatbot emotion according to chat content.



Controllers:

- GeminiController:
  - Manages the interaction logic for the Gemini chatbot, including handling user inputs, controlling expressions, and starting new chat sessions.
  - Provides access to the ChatBotView for rendering the chatbot's responses and interactions.
  - Handles chat asynchronously and manages the emotional states of the chatbot through ExpressionControl.
- ExpressionControl:

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

- Unity Component script to controls the chatbot's visual expressions and speaking animations.
- Manages various emotional states (e.g., angry, sad, happy) and updates the expression dynamically during interactions.

Views:

- ChatBotView:
  - Manages the user interface for the Gemini chatbot, including input/output for chat interactions.
  - Displays the chatbot's visual representation and updates its emotional expressions.
  - Dynamically renders user messages and chatbot responses in separate chat boxes.

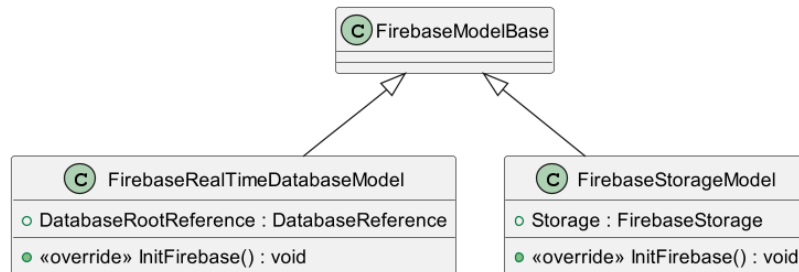
Models:

- GeminiDataModel:
  - Stores the chatbot's data, including a history of chats and the name of the currently active chatbot model.
  - Supplies chatting related data to GeminiController for processing and UI updates.

#### 4.6 Component: FirebaseDatabase

This component provides firebase realtime database functionality. This component wraps Firebase realtime database and firebase storage api provides:

- Read/write/update/delete operation to firebase realtime database (used to store client-side data of the users)
- Read/write/update/delete operation to firebase storage (used to store image/file of the users)



Models:

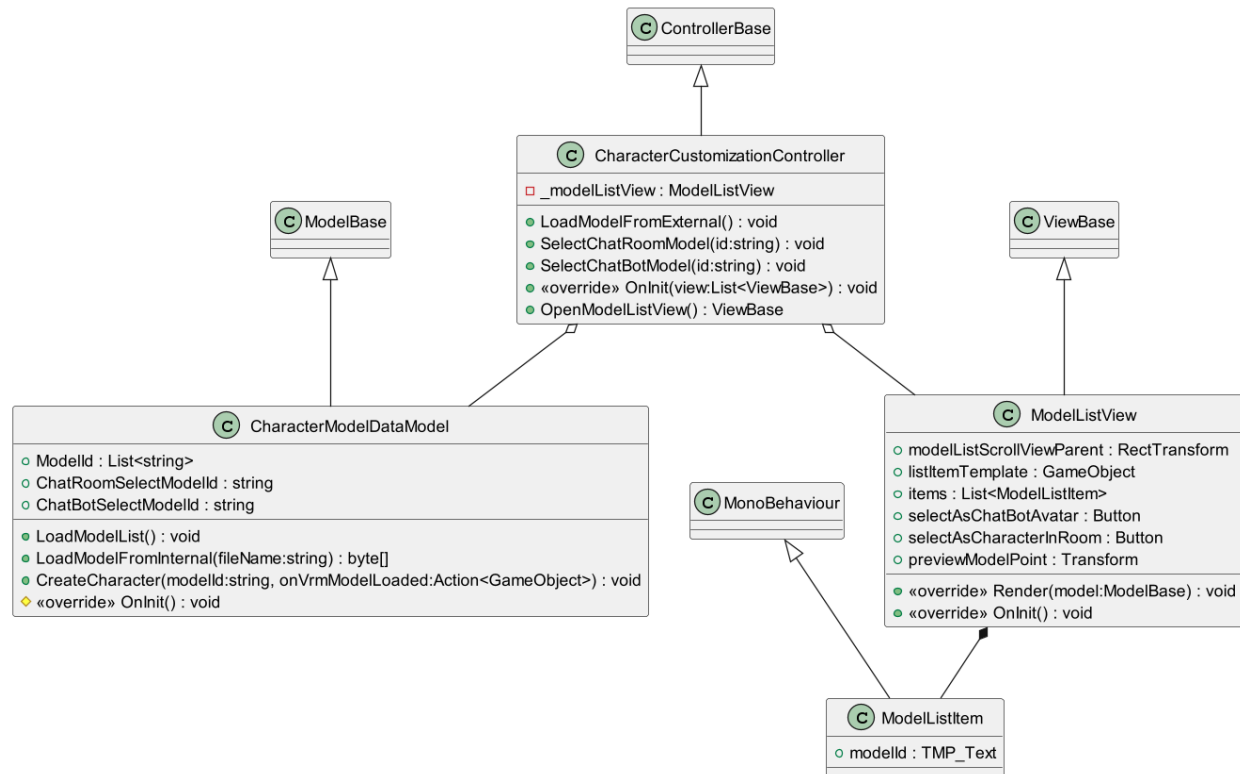
- **FirebaseRealTimeDatabaseModel:**
  - Provides access to the root reference of Firebase Realtime Database for interacting with structured data in real-time.
  - Initializes the database instance for use in the application.
- **FirebaseStorageModel:**
  - Manages Firebase Storage for handling file uploads and downloads.
  - Initializes the storage instance, allowing the application to store and retrieve large files such as images or videos.

#### 4.7 Component: CharacterCustomize

This component provides character customization functionality. Including: import/preview/select model for user's character in online room and for chatbot avatar.



Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	



Controller:

- **CharacterCustomizationController:**
  - Manages logic for character customization, including importing models from external memory of mobile devices and selecting specific models for chat rooms or chatbots.
  - Handles interaction between the user and the ModelListView, enabling model selection and updates.

View:

- **ModelListView:**
  - Displays a scrollable list of character models using a dynamic template.
  - Includes functionality for selecting models and previewing them in a designated area.
  - Allows user interaction via buttons and dynamically renders model-related data

#### 4.8 Component: Network

This component provides security network communication between client and server functionality based on SocketIO. Include connection management, encrypt when sending data, decrypt when receiving data.

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	



- EncryptionProvider:
  - Manages AES and RSA encryption for secure communication.
  - Creates and stores AES and RSA keys, encrypts and decrypts data using these keys.
  - Supports bidirectional encryption between client and server by setting up RSA keys for secure transmission and AES for encrypted communication.
  - Provides functionalities like EncryptAES128ECB, DecryptAES128ECB, EncryptRSAString, and DecryptRSAString for handling encryption tasks.
  - Manages server-side public key setup and ensures encryption protocol initialization.
- SocketIO:
  - Manages WebSocket communication using encryption to ensure secure data transfer.
  - Handles sending encrypted messages using RSA or AES (SendWebSocketMessageRSA and SendWebSocketMessageAES).

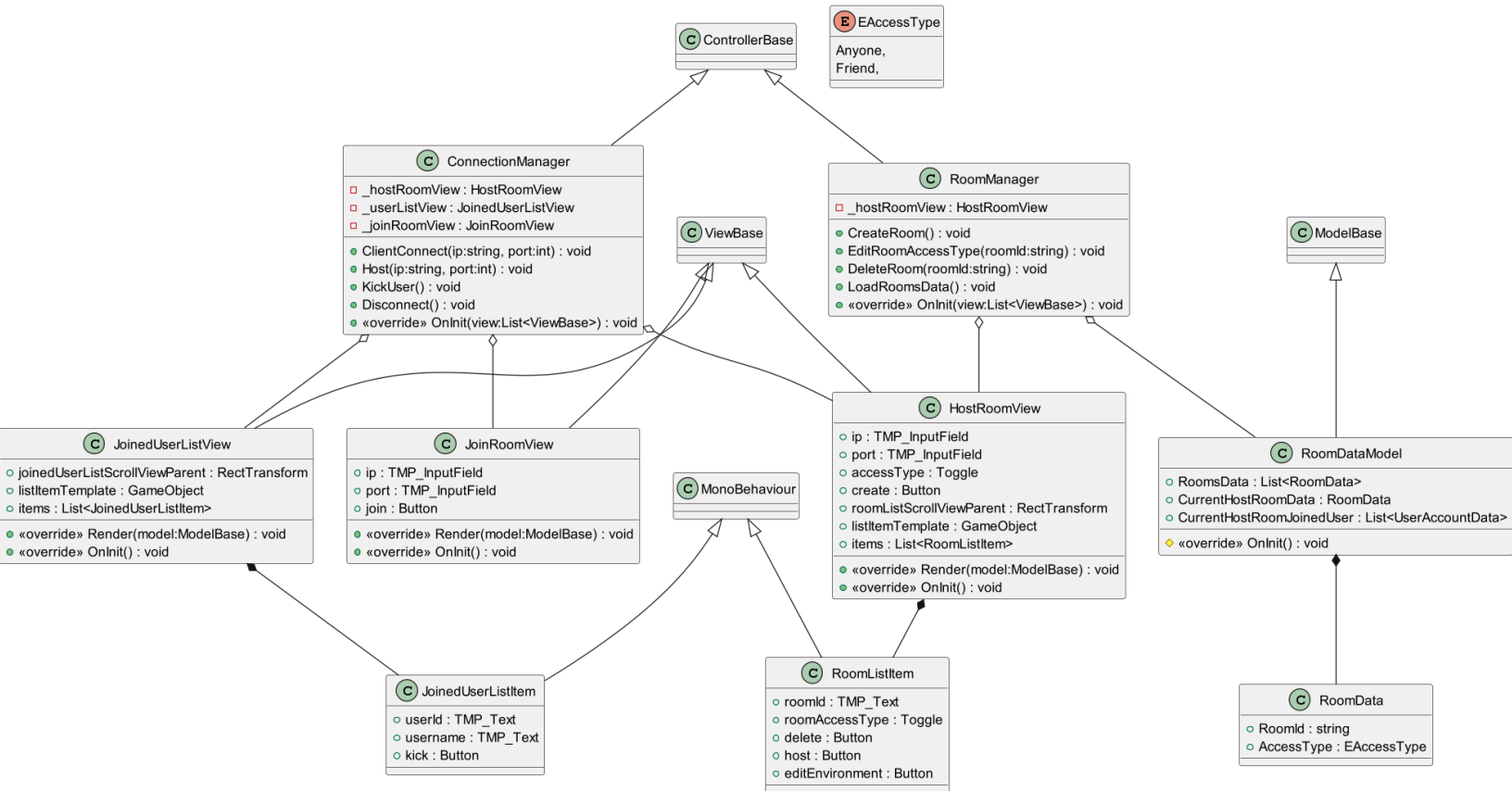
Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

- Establishes WebSocket callbacks for event-driven communication.
- Handles encryption key exchange (SendKey) for secure messaging over the network.
- Manages WebSocket connection lifecycle, including initialization (OnInit) and cleanup (OnDestroy, OnApplicationQuit).

#### 4.9 Component: OnlineRuntime/RoomManagement

This component provides connection and room management functionality. Including:

- Management connection: host room/join to room
- View list of joined users for host, kick user for host.
- Manage room: create/delete room, set room access scope: anyone/only friend.



Controller:

- RoomManager:
  - Manages the logic for creating rooms, editing rooms' information and deleting rooms.
  - Handles interaction with the HostRoomView.
- ConnectionManager:
  - Handles the connection between host and client users.
  - Provide a function for the host to view the list of joined users and kick other joined users.

View:

- HostRoomView:
  - Handles the host user's input for managing chat rooms.
  - Displays created chat rooms list and rooms' information.
- JoinRoomView:
  - Provides an interface for the user to access chat rooms via IP and Port.
- JoinedUserView:

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

- Provides an interface for the host to view the list of users who joined the room.

Model:

- RoomDataModel:
  - Contains created room data.

#### 4.10 Component: OnlineRuntime/Input

This component provides keyboard and mouse input to control the character. Including:

- Keyboard and Mouse: move (wasd), look (mouse), jump (space), crouch (c), sprint(shift).
- Touch screen: move (on screen virtual joystick), look (on screen touchpad), jump (on screen button), crouch (on screen button), sprint(on screen button).
- sensor input: look (Gyroscope)

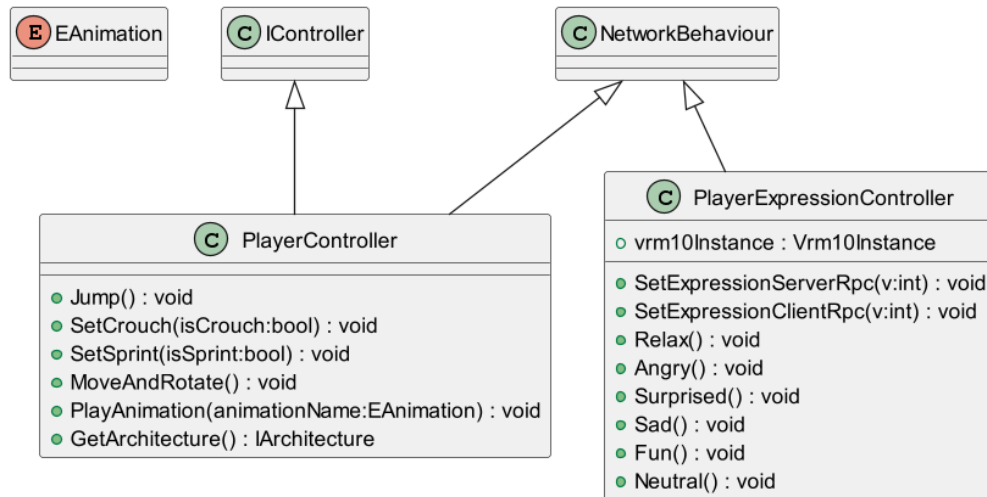


- PlayerInputAction:
  - Container to store input events.
  - Trigger specific event by name.
  - Get event by name (used to register event callback).
- OnScreenButton:
  - Unity component script provides a button component with additional functionality.
- OnScreenJoystick:
  - Unity component script provides a virtual joystick component for the touchscreen.
- OnScreenTouchPad:
  - Unity component script provides a touchpad component for the touchscreen.
- GyroscopeInput:
  - Wrap Unity Gyroscope API to provide input from Gyroscope.
- BasicInput:
  - Wrap Unity input system for keyboard and mouse input.

#### 4.11 Component: OnlineRuntime/CharacterControl

This component provides control to the character on the online room based on Unity netcode. Including: Move, look around, jump, crouch, sprint, play animation, play expression.

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

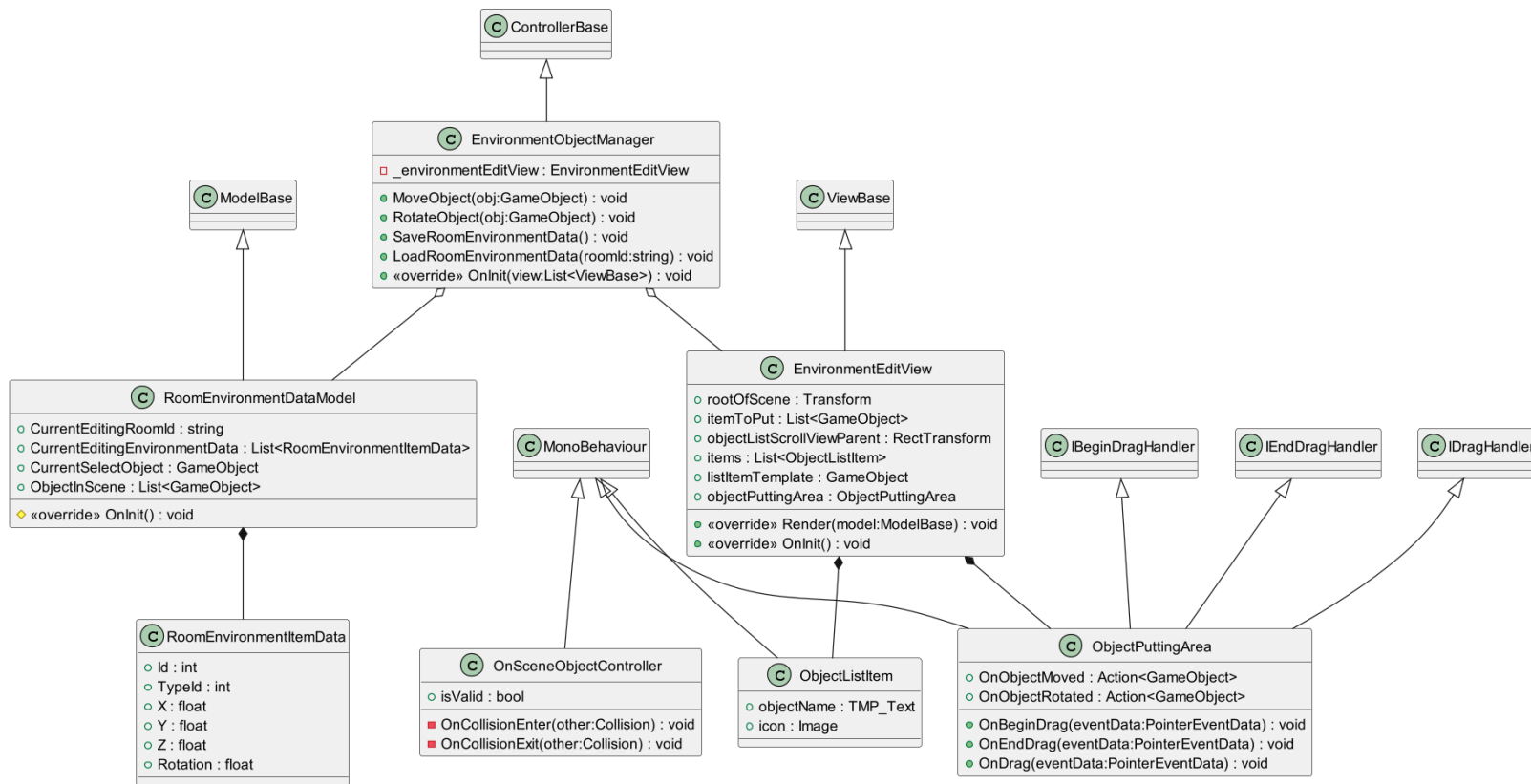


- **PlayerController:**
  - Unity Network object component script provides controlling character action in the room.
  - Use Network Transform and Network Animator component of Unity Netcode to synchronize character in room).
- **PlayerExpressionController:**
  - Unity Network object component script provides controlling character expression in the room.
  - Use RPC to synchronize.

#### 4.12 Component: OnlineRuntime/EnvironmentCustomize

This component provides Environment customization based on Unity netcode. Including: Add/remove pre-build objects, move objects, rotate objects. save/load room environment data to firebase.

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	



#### Controllers:

- **EnvironmentObjectManager:**
  - Provides functions for editing objects in chat rooms.
  - Provides save/load data to/from firebase database.

#### Views:

- **EnvironmentEditView:**
  - Provides interfaces for the host to use functionalities in EnvironmentObjectManager.
- **ObjectPuttingArea:**
  - Unity component script provides interface to putting/editing object on scene.

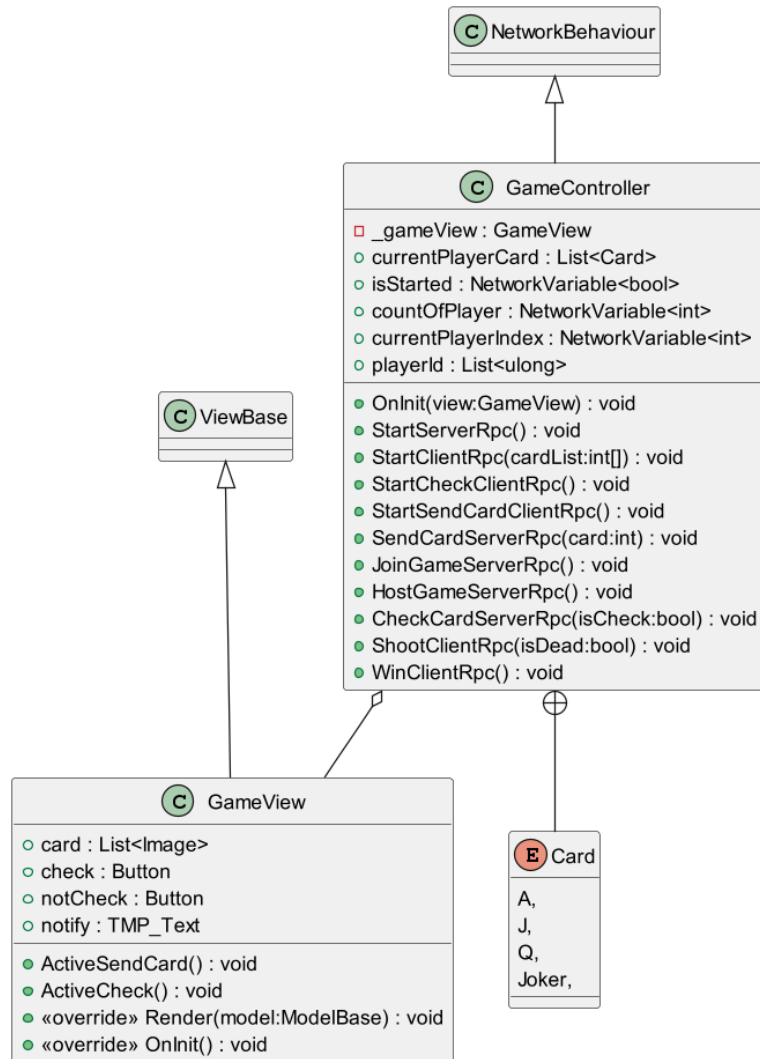
#### Models:

- **RoomEnvironmentDataModel:**
  - Store information about the objects in the chat room.

### 4.13 Component: OnlineRuntime/Activities/MiniGame

This component provides Minigame functionality based on Unity netcode. This Component provides a card game like Liar's Bar ([https://store.steampowered.com/app/3097560/Liars\\_Bar/](https://store.steampowered.com/app/3097560/Liars_Bar/)). This Component will be further designed in later phases of the development process.

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

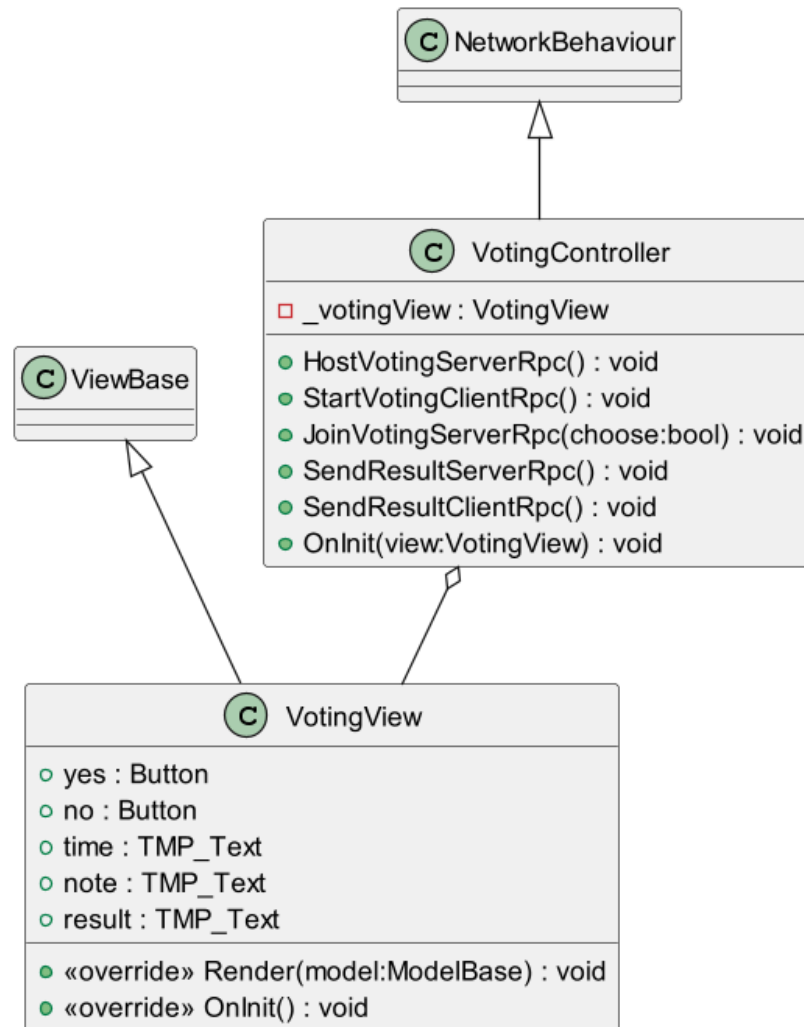


- GameController:
  - Unity network object component script management game rule, game logic, game state, synchronize using RPC.
- GameView:
  - View of Game.

#### 4.14 Component: OnlineRuntime/Activities/Voting

This component provides hosting voting based on Unity netcode, including: hosting a voting, voting settings (vote duration), voting, display result to all users joined in room.

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	



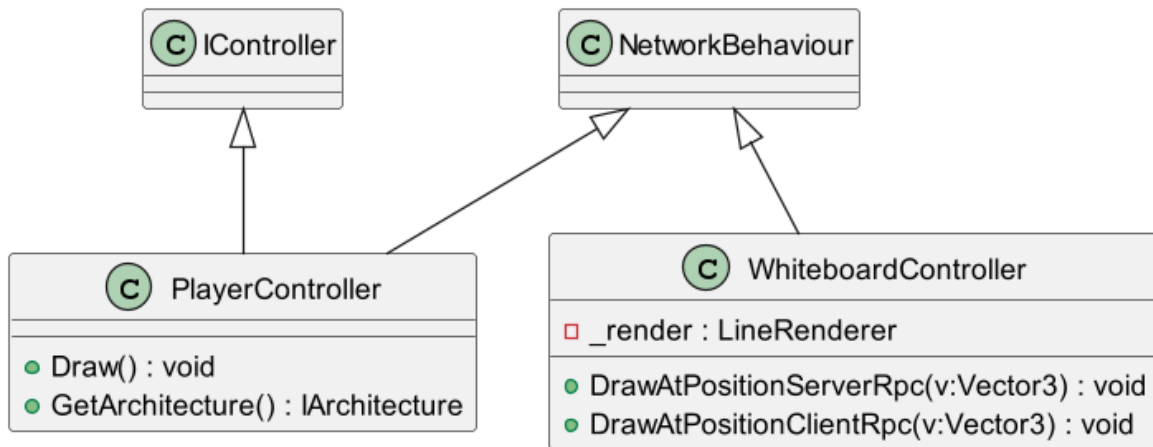
- VotingController:
  - Unity network object component script management Voting process, synchronize using RPC.
- VotingView:
  - View of voting

#### 4.15 Component: OnlineRuntime/Activities/Whiteboard

This component provides a whiteboard based on Unity netcode. Any users joined in the room can draw/erase on the whiteboard.



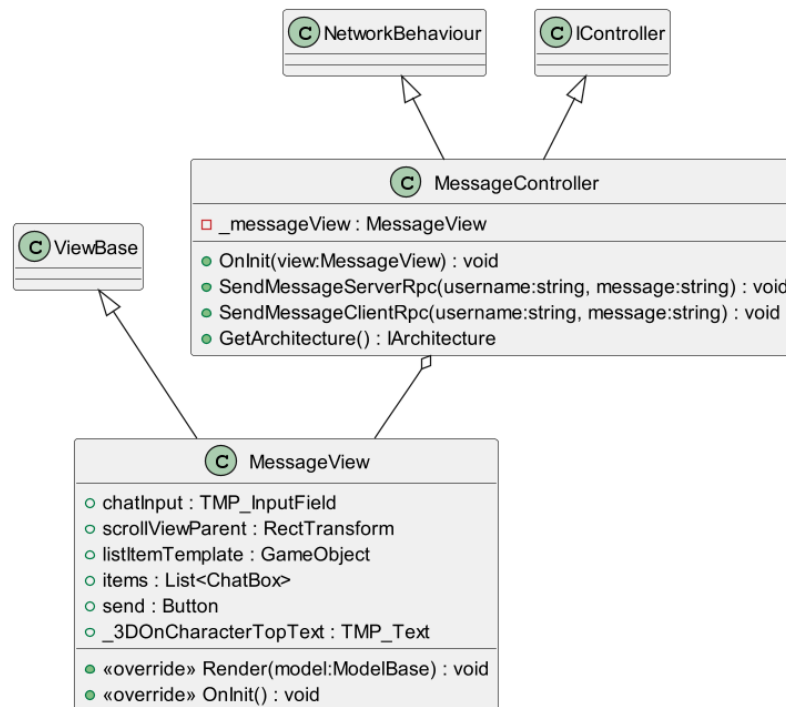
Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	



- PlayerController:
  - Unity network object component script to provide whiteboard drawing.
- Whiteboard controller:
  - Render lines on the whiteboard and synchronize to all clients using RPC.

#### 4.16 Component: OnlineRuntime/Message

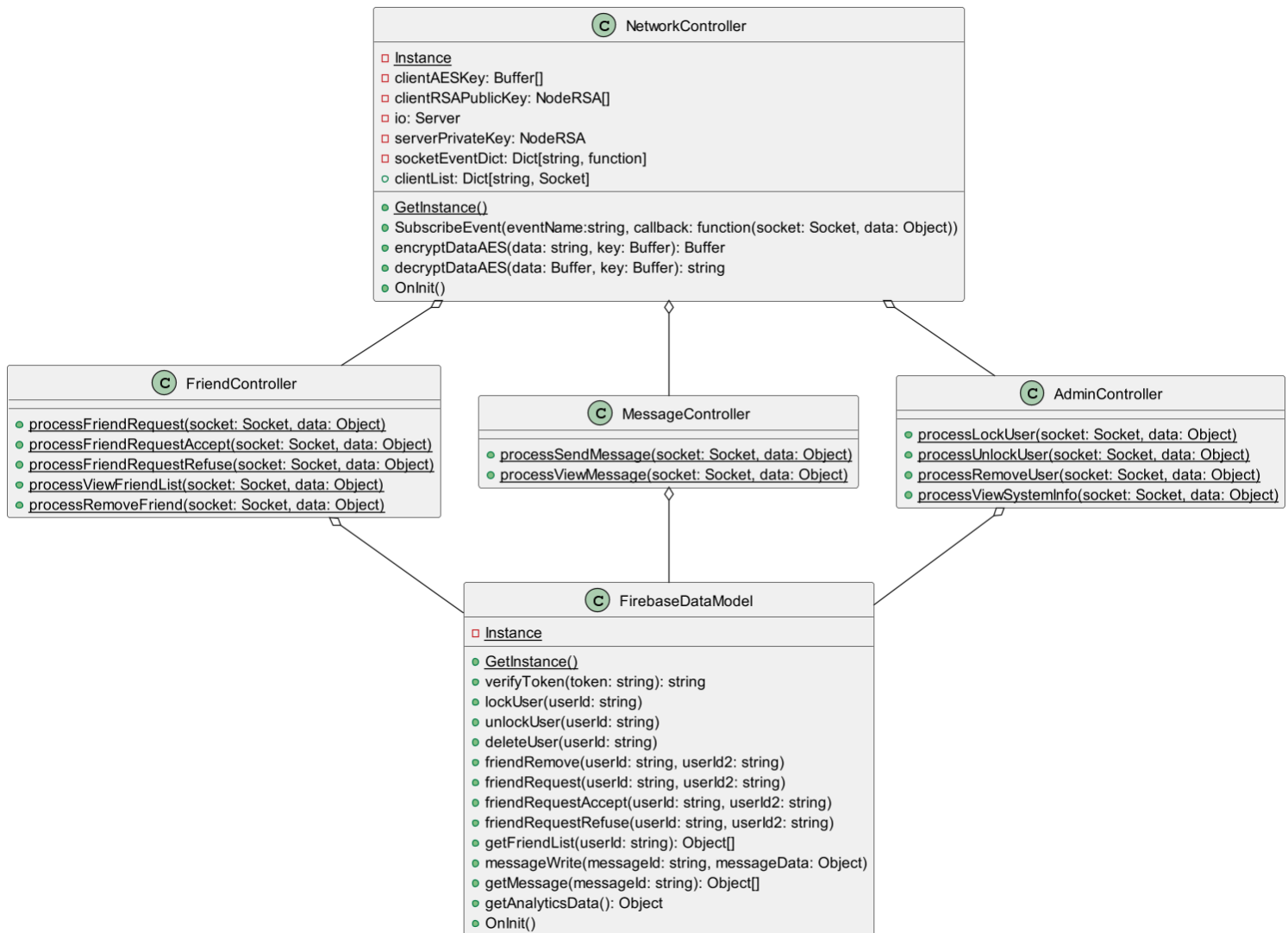
This component provides message communication based on Unity netcode, including send/receive messages on chatbox. Messages of users are also displayed on the head of 3D characters who sent the message.



- MessageController:
  - Provides send/receive real time messages functionality in the room.
  - Use RPC to synchronize messages.
- MessageView:
  - View to display messages on chat box and display messages on top of character.

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

#### 4.17 Component: Server/Network and Server/FirebaseDataModel



##### Server/Network:

This component is used for the server to process requests from the clients via SocketIO:

- Direct Messages.
- Friend Management.
- Admin functionality

The data sent between the two sides is encrypted with RSA and AES.

Explain:

- NetworkController:
  - Management connection with clients.
  - Provides data encryption/decryption.
  - Interface to process comes in request.
- FriendController:
  - Process Friend functionality request.
- MessageController:
  - Process Message functionality request.
- AdminController:

Virtual Chat 3D	Version: 1.1
Software Architecture Document	Date: 21/11/2024
Software-Architecture-Document-v1.0	

- Process Admin functionality request.

#### **Server/FirebaseDataModel:**

This component is used to manage data in the firebase database and get data from google analytics.

Explain:

- FirebaseDataModel:
  - Manage data in the firebase database for Message, Friend, Admin functionality.
  - Get data from google analytics.

## **5. Deployment**

*[Leave this section blank for PA3 if you are writing this document for PA4.]*

*In this section, describe how the system is deployed by mapping the components in Section 4 to machines running them. For example, your mobile app is running on a mobile device (Android, iOS, etc), your server runs all components on the server side including the database]*

## **6. Implementation View**

*[Leave this section blank for PA3 if you are writing this document for PA4.]*

*In this section, provide folder structures for your code for all components described in Section 4. ]*