**Software Design Specifications**

**for**

**VR ANXIETY MANAGEMENT APP 1.0**

**Prepared by:**

**Team 11**

**SE22UARI015**

**SE22UARI173**

**SE22UARI068**

**SE22UARI214**

**SE22UARI012**

**SE22UARI064**

**Table of Contents**

**Introduction**

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

**Use Case View**

1.5 Use Cases

Use Case 1: User Authentication

Use Case 2: Environment Selection

Use Case 3: Environment Interaction

Use Case 4: Exit to Selection Page

**Design Overview**

2.1 Design Goals and Constraints

2.2 Design Assumptions

2.3 Significant Design Modules

2.4 Dependencies & External Interfaces

**Logical View**

3.1 Architecture Overview

3.2 Component Breakdown

**Main Menu Scene**

**VR Environment Scenes (Forest/Beach)**

**Settings Menu**

**Data View**

4.1 Persistent Data Storage

4.2 Database (For Future Expansion)

**Quality of Service**

8.1 Availability

8.2 Security & Authorization

8.3 Performance Considerations

8.4 Monitoring & Logging

9.**Conclusion**

**UMLs**

1. Introduction

The Software Design Specifications for the VR Anxiety Coping Application present the system’s architecture, modules, interfaces, and core features in a clear, organized manner. This document defines its purpose—outlining how the app enables user authentication, environment selection, customization, and full VR immersion—and its scope. Key terms, acronyms, and abbreviations are explained for clarity, and references to Unity Documentation, the Google Cardboard SDK, and VR design best practices are provided. An overview of the document’s structure then guides readers through use cases, design and logical views, data management, quality-of-service considerations, and the conclusion.

* 1. Purpose

The purpose of this document is to define the architecture, modules, interfaces, and functionalities of the VR Anxiety Coping Application. This application provides immersive virtual environments designed to help users cope with anxiety by engaging in relaxing experiences.

* 1. Scope

The application will allow users to:

Login using their Unity credentials or create an account if they do not have one. Select a VR environment (Forest or Beach).

Customize environmental settings (brightness, sound, movement speed).

Experience full VR immersion using the XR Cardboard Plugin, which provides stereo vision and head tracking.

* 1. Definitions, Acronyms, and Abbreviations

VR (Virtual Reality) – A simulated experience that can be similar to or different from the real world.

XR (Extended Reality) – An umbrella term for VR, AR, and MR (Mixed Reality). Unity – A cross-platform game engine used to develop the application.

XR Cardboard Plugin – A Unity plugin enabling stereo VR views and head tracking on mobile VR devices.

* 1. References

Unity Documentation: <https://docs.unity3d.com/>

Google Cardboard SDK: <https://developers.google.com/cardboard> VR Design Best Practices: <https://developer.oculus.com/design>

Use Case View

* 1. Use Cases

The application consists of the following primary use cases

Use Case 1: User Authentication

The user logs in using Unity credentials or creates a new account. If authentication fails, an error message is displayed.

Upon successful login, the environment selection screen appears.

Use Case 2: Environment Selection

The user chooses between Forest or Beach environments. Once selected, the VR scene loads.

1. Design Overview
   1. Design Goals and Constraints

The application should be lightweight for mobile VR. The VR experience should be smooth and responsive.

Unity's XR Cardboard Plugin will handle stereo vision and head tracking.

* 1. Design Assumptions

The user has a Google Cardboard-compatible VR headset. The user has an active internet connection for login.

The application will run on Android devices only.

* 1. Significant Design Modules

User Management Module – Handles login/signup.

* 1. Dependencies & External Interfaces

Firebase Authentication – For login and account management.

Google Cardboard SDK – For VR rendering and head tracking. Unity PlayerPrefs – For storing user preferences locally.

1. Logical View
   1. Architecture Overview

The application follows an MVC (Model-View-Controller) architecture:

Model – Handles user data and settings.

View – Displays UI screens and VR environments. Controller – Manages navigation.

* 1. Component Breakdown Main Menu Scene

Login Form

Create Account Button Environment Selection Options

VR Environment Scenes (Forest/Beach) XR Camera (Stereo vision + head tracking) Settings

1. Data View
   1. Database (For Future Expansion)

A database (e.g., Firebase) can be used to store user profiles and progress.**6. Exception Handling**

|  |  |
| --- | --- |
| **Scenario** | **Exception Handling** |
| Invalid Login Credentials | Show error message, prompt retry. |
| Failed Environment Load | Retry loading, show fallback screen. |
| VR Headset Disconnected | Pause the experience, show reconnect message. |
| Settings Not Applied Properly | Reset to default values. |

**7. Configurable Parameters**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Dynamic?** |
| Brightness | Adjusts ambient lighting in VR | Yes |
| Sound Volume | Controls background sound | Yes |
| Movement Speed | Adjusts navigation speed | Yes |

1. Quality of Service
   1. Availability

The application will function offline, except for login.

* 1. Security & Authorization

Unity Authentication ensures secure login.

* 1. Performance Considerations

Optimized VR rendering for smooth frame rates.

* 1. Monitoring & Logging

Logs user activity for debugging.

1. Conclusion

This document provides a detailed design overview of the VR Anxiety Coping Application. The application leverages Unity, XR Cardboard Plugin, and Unity Authentication to create an immersive and relaxing experience for users. Future improvements may include additional environments, guided meditation features, and multiplayer interactions.

UMLs





