
Air University Multan Campus



SOFTWARE DESIGN DOCUMENT

(SDD DOCUMENT)

for

<VR-Based Job Interview Simulator>

By

Eman Muzaffar (233676)

Amina Hussain (233682)

Abeeha Fatima (233762)

Supervisor

Mr. Hafiz Muhammad Anus Ali

Bachelor of Science in Software Engineering (2023-2027)

Contents

Table of Contents	2
Revisions	3
1. Introduction	4
2. Design methodology and software process model	4
3. System overview	4
3.1 Architectural design	4
3.2 Process flow/Representation	4
4. Design models [along with descriptions]	4
4.1 Models/Diagrams	4

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1. Introduction

This document describes the Software Design for the "VR-Based Job Interview Simulator" project. The goal of the project is to help users practice and improve their job interview skills using virtual reality. The system provides an immersive environment where users can interact with a virtual interviewer, answer questions, and receive feedback on their performance. This solution is designed especially for fresh graduates and job seekers who want to build confidence and get used to real interview scenarios.

The project idea is based on the need for a practical, interactive tool that allows repeated practice in a stress-free, realistic setting. The system focuses on the learning-by-doing approach, where users can engage in different types of interviews (e.g., HR-based, technical) and learn from the instant feedback they receive.

Modules implemented so far include:

- **User Authentication** – to manage secure login and registration.
- **Interview Scenario Selection** – allows users to pick the type of interview they want to practice (HR, Technical, General).
- **VR Interaction Environment** – provides the 3D space and virtual interviewer for an immersive experience.
- **Feedback and Performance Report** – analyzes the answers and provides suggestions and scores based on predefined criteria.

These modules were selected based on the basic flow of a real interview process: starting with user access, choosing a scenario, performing in a virtual session, and finally receiving evaluation. This ensures a complete and realistic experience for the user.

2. Design methodology and software process model

We are using the **Object-Oriented Programming (OOP)** approach in our project because it helps us break down the system into small, reusable parts called objects. Each object represents a real-world entity like a user, a scenario, or feedback. This approach makes our code more organized, easier to test, and scalable when we add more features in the future. For example, our "User" object handles login and registration, while the "Scenario" object deals with different interview types.

The **Software Process Model** we are following is the **Iterative Model**. In this model, we first build a simple version of the system with basic features. Then we test it, get

feedback from users, and improve it in the next version. This process is repeated in cycles until the complete system is ready. This model is useful for our VR project because user experience is very important, and we need to make continuous improvements based on user interaction and feedback.

3. System overview

The system simulates a complete interview process in a VR environment where users interact with a virtual interviewer using predefined questions. The user selects the type of interview (HR, Technical, etc.), responds using speech or input, and receives performance feedback.

3.1 Architectural design

The system architecture consists of:

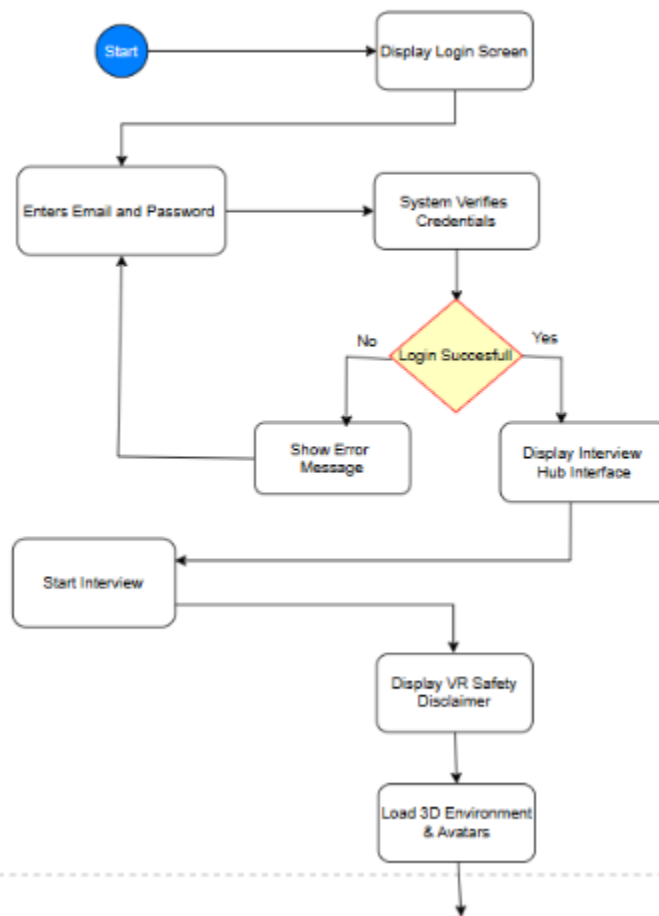
- **Authentication Module** – user login and registration
- **Scenario Manager** – manage and deliver different interview types
- **VR Environment Module** – simulate the interview room and virtual interviewer
- **Response Evaluator** – analyze user responses
- **Database Manager** – manage user data, scenario questions, and results

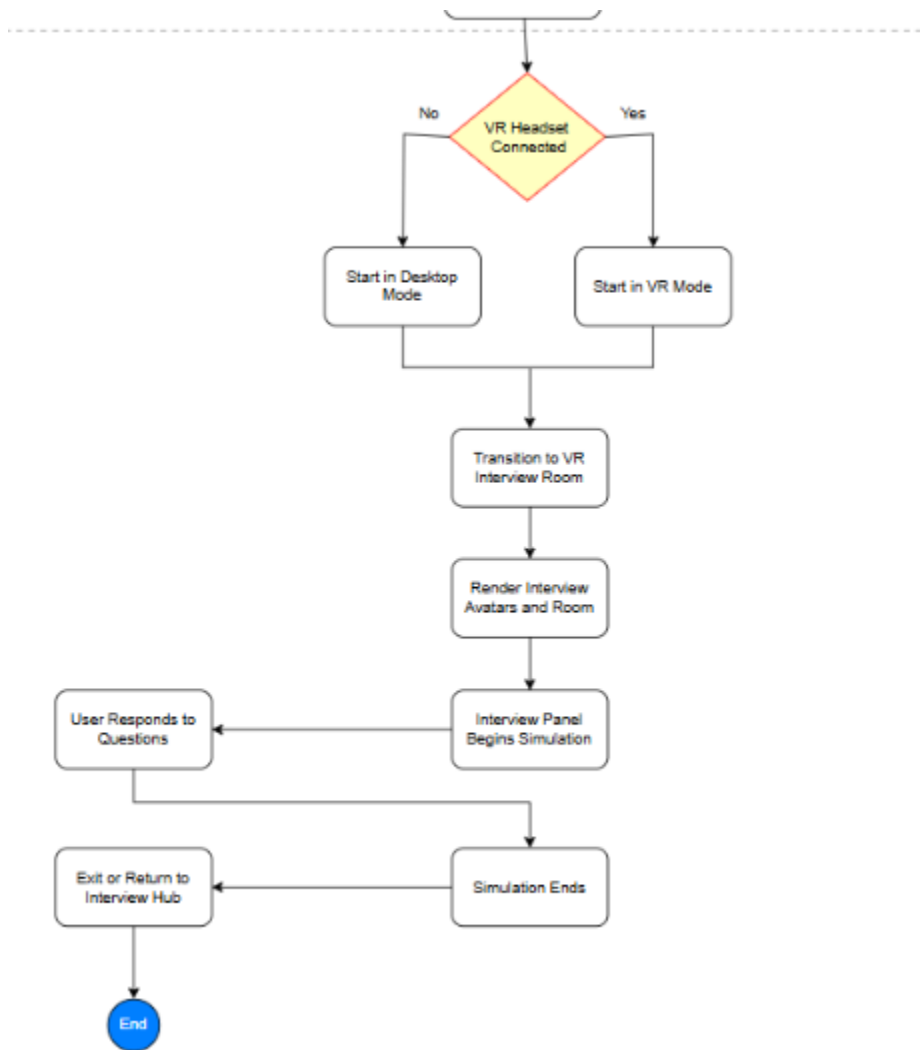
3.2 Process flow/Representation

The core flow is:

1. User logs in
2. Selects scenario
3. VR interview starts
4. Responses are captured
5. Feedback is displayed

Activity Diagram



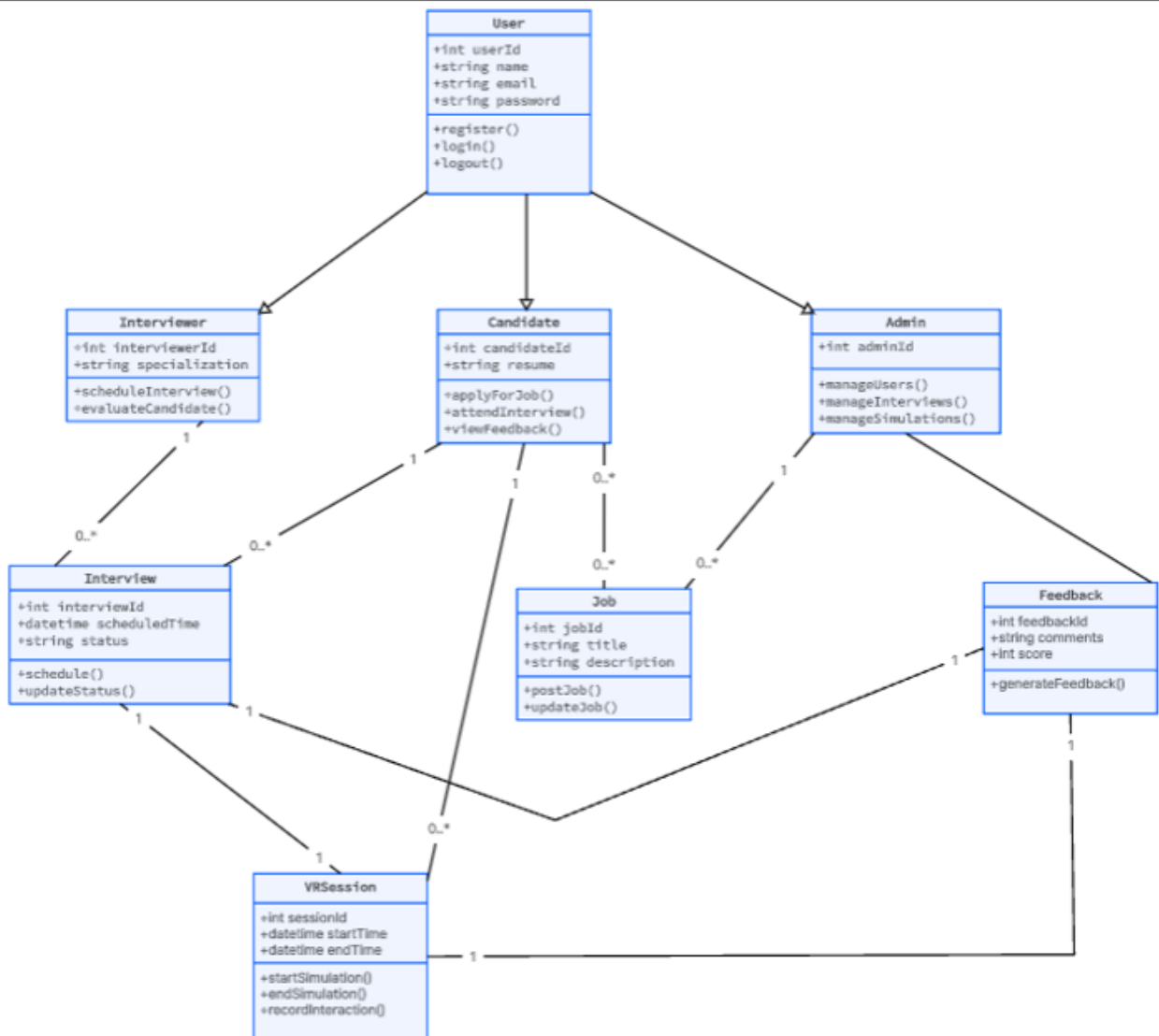


4. Design models [along with descriptions]

4.1 Models/Diagrams

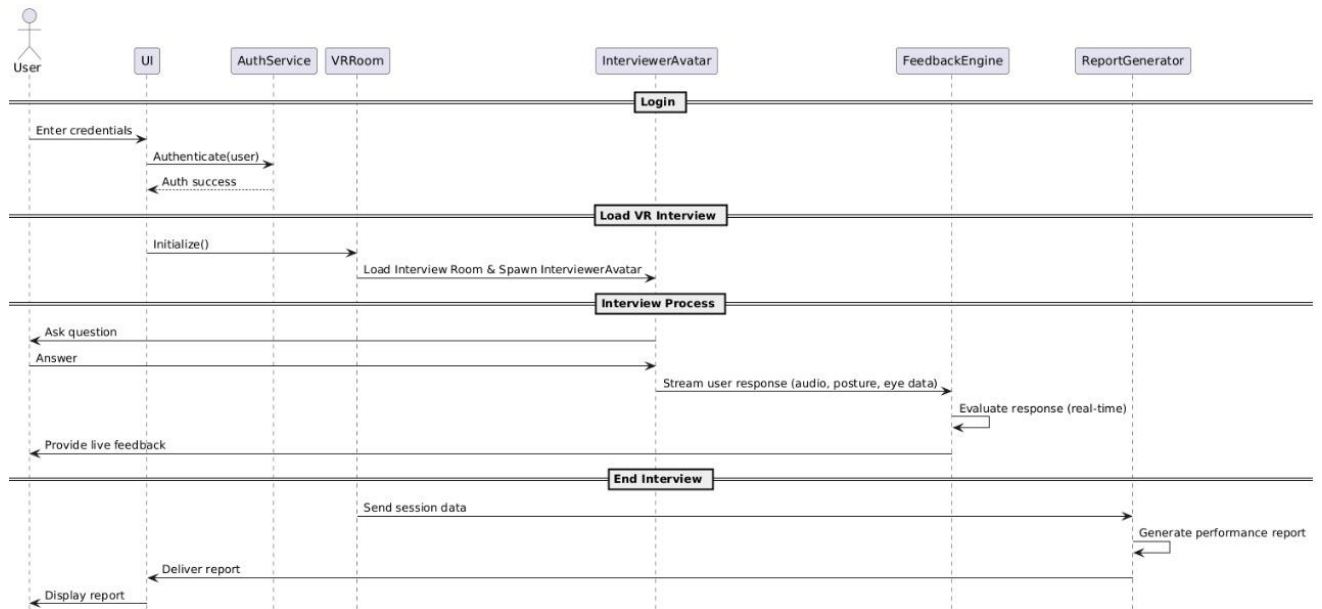
The applicable models may include:

- Class Diagram

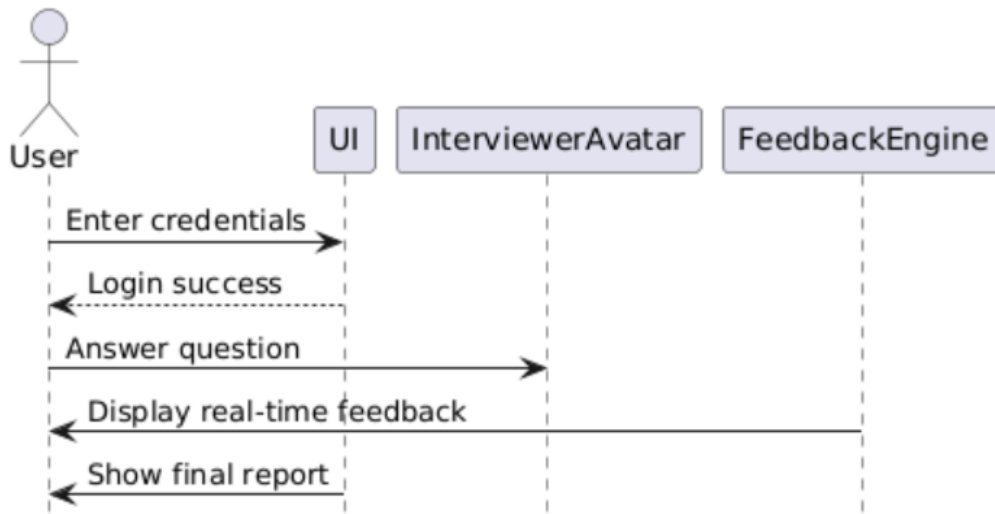


- Sequence Diagram

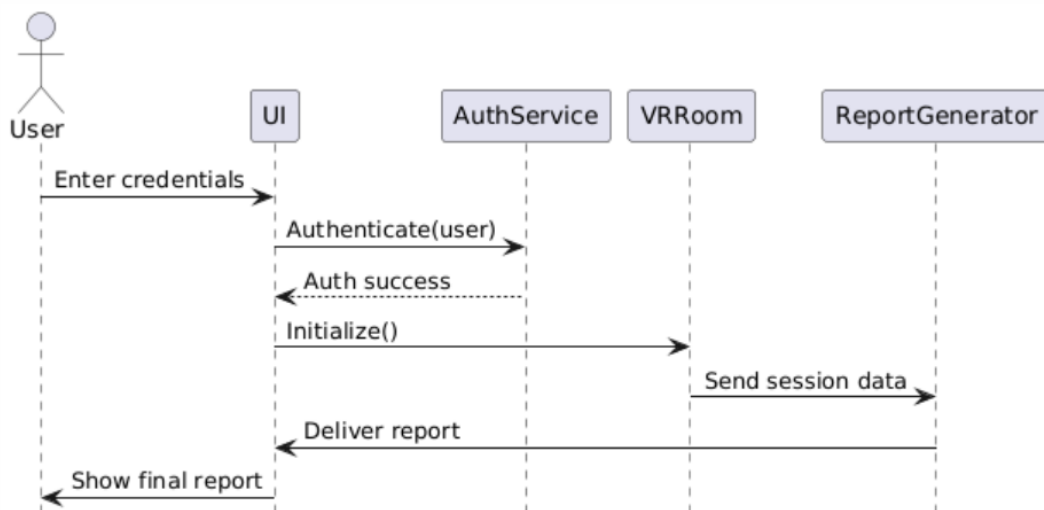
Complete diagram



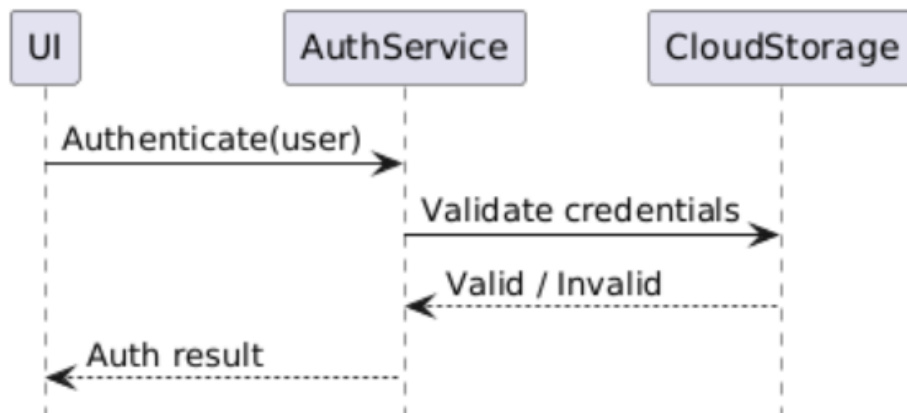
I. User component



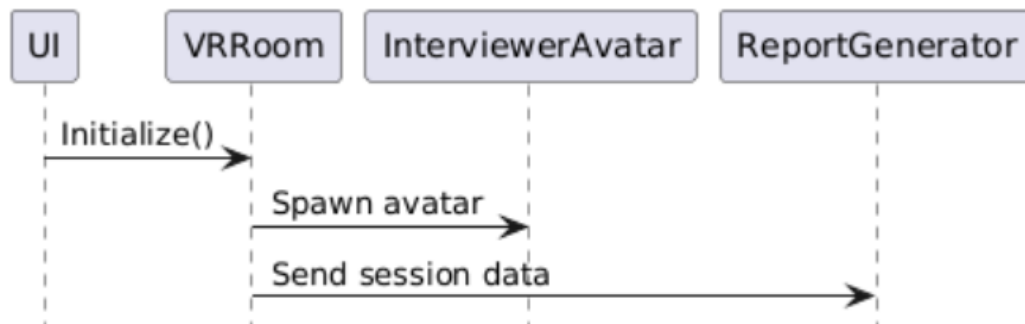
II. UI component



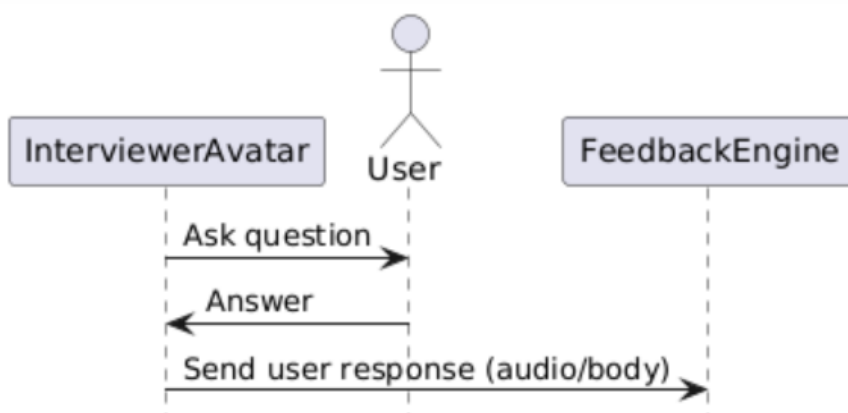
III. AuthService component



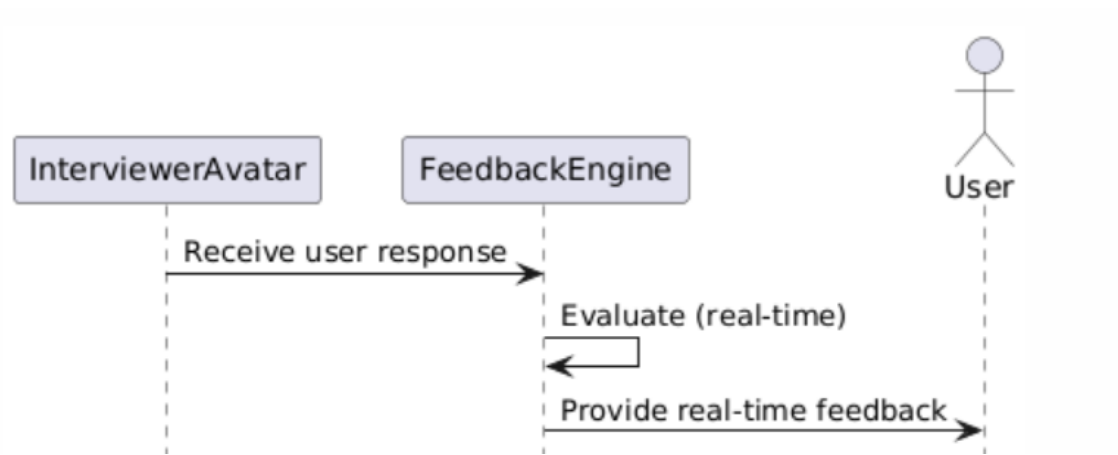
IV. VR room component



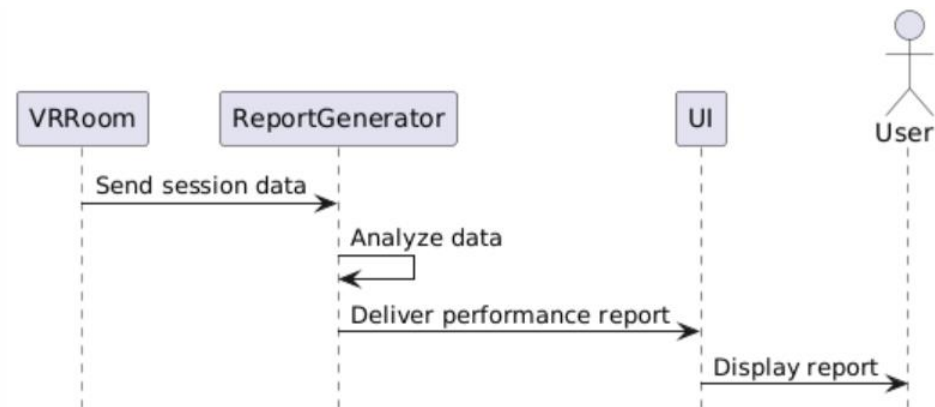
V. Interview avatar component



VI. Feedback engine component



VII. Report generator component



- Activity Diagram (Swim Lane.)

