This document serves as a comprehensive overview of the project aimed at developing an AI assistant for the VRChat application. Additionally, it outlines detailed debugging procedures to address specific errors that may arise during the implementation or testing phases of this developed system.

## Overview of the System

The system is an AI-powered conversational avatar that can be deployed in the VRChat app.

## Environment Construction Procedure Manual

For the development of this AI assistant for VR Chat we have to consider various system requirements.

**A. Prerequisites for Coding Environment**

**Programming Language**: Python

**Libraries and Tools:**

1. VB-Virtual Audio Cable:
   * VB-Virtual Audio Cable is a software tool used to route audio between applications or devices on Windows operating systems. It allows for the creation of virtual audio devices, enabling audio routing to specific input or output devices.
   * The VB-Virtual Audio Cable tool, available at [<https://vb-audio.com/Cable/>], offers one free audio line. Additional lines, including the A+B VB-Cable for around 5 euros, are needed. VB-Audio's B channel captures user audio in VRChat, while the A channel transmits AI-generated responses through our Python script.
2. SpeechRecognition Python Library:
   * The SpeechRecognition library in Python provides a simple interface for performing speech recognition tasks. It supports multiple speech recognition engines, including Google Speech Recognition, IBM Speech to Text, and CMU Sphinx.
   * SpeechRecognition can be installed via Python Package Index (PyPI) using pip: pip install SpeechRecognition.
3. OpenAI API:
   * The OpenAI API grants developers access to OpenAI's language models, such as the GPT series. It allows for sending text prompts to OpenAI's servers and receiving AI-generated responses, supporting various natural language processing tasks.
   * Access to the OpenAI API is available through registration on the OpenAI website.
4. Pyaudio:
   * Pyaudio is a Python library providing bindings for PortAudio, facilitating cross-platform audio I/O operations. It enables developers to work with audio streams in real-time, including recording from microphones, playback through speakers, and audio manipulation.
   * Pyaudio can be installed via Python Package Index (PyPI) using pip: pip install pyaudio.
5. Wave Module:
   * The wave module in Python offers functions for reading and writing WAV files, a common format for storing raw audio data. It allows developers to manipulate audio files programmatically, extract data, or create new ones.
   * The wave module is included in Python's standard library and does not require additional installation.
6. Pyttsx3:
   * Pyttsx3 is a Python library for text-to-speech (TTS) conversion, providing a simple interface to convert text into spoken words. It is commonly used in applications requiring speech synthesis, such as accessibility tools and voice-enabled applications.
   * Pyttsx3 can be installed via Python Package Index (PyPI) using pip: pip install pyttsx3.

**B. Prerequisites for Running the System:**

1. VRChat: The avatar in this project is meant for VRChat app. Thus, in order to download and install VRChat on PC, please follow the following instruction”

Requirements for VRChat:

* Steam Account (<https://store.steampowered.com/>)
* PC meeting minimum system requirements:

Minimum system requirement:

* + Requires a 64-bit processor and operating system
  + OS \*: Windows 8.1, Windows 10
  + Processor: Intel® i5-4590 / AMD FX 8350 equivalent or greater
  + Memory: 4 GB RAM
  + Graphics: NVIDIA GeForce® GTX 970 / AMD Radeon™ R9 290 equivalent or greater
  + DirectX: Version 11
  + Network: Broadband Internet connection
  + Storage: 1 GB available space
  + VR Support: SteamVR
* If need more detail refer this: (<https://store.steampowered.com/app/438100/VRChat/>)

Downloading and Installing:

1. Install Steam on your PC if you haven't already.
2. Open Steam and create an account if you don't have one. ( Once created, can use the same id to play the game as well)
3. Search for "VRChat" in the Steam Store.
4. Click on "VRChat" in the search results.
5. Click "Install" on the VRChat store page.
6. Follow the on-screen instructions to complete the installation.

Launching VRChat:

1. Once the installation is complete, you'll see "Play" turn into "VRChat" in your Steam library.
2. Click "VRChat" to launch the game.

Account creation:

1. For this project we require two accounts for VRChat, thus make one additional account. One account will be used for AI assistant and another for the user testing the system

**C. Step-by-Step Installation Guide**

Installing Dependencies

1. Python Installation:
   * If Python is not already installed on your system, download the latest version from the official Python website (<https://www.python.org/downloads/>) and follow the installation instructions for your operating system.This will also install pip for managing the packages.
2. Installing Required Libraries:
   * Open a command prompt or terminal.
   * Use pip, Python's package manager, to install the required libraries. The necessary libraries are already listed in the “requirements.txt” file in the project folder. To download and install dependencies, navigate to the project folder in the terminal and type:
     + pip install -r requirements.txt
   * This will automatically download the required dependencies. Alternatively, you can download them individually by typing:
     + pip install SpeechRecognition==3.10.3 openai==1.16.2 pygame==2.5.2 pyaudio==0.2.14 pyttsx3==2.90 python-osc==1.8.3 setuptools==69.5.1
   * Ensure successful installation by checking for any error messages during installation.
3. Download and Install VB-Virtual Audio Cable:
   * Visit the VB-Audio website [(https://vb-audio.com/Cable/)](https://vb-audio.com/Cable/) and download the VB-Virtual Audio Cable software.
   * Download the VB-Cable A+B as both channels are required for this project. A donation of 5 euros is required.
   * Follow the installation instructions provided by the software installer if needed.
   * Step to install:
     + After downloading, extract the zip file.
     + Inside, there will be separate folders for A and B channels installers named “VBCABLE\_B\_Driver\_Pack43” and “VBCABLE\_A\_Driver\_Pack43”.
     + Navigate into these folders individually and install the drivers by double-clicking “VBCABLE\_Setup\_x64” and following the installer instructions.
     + Upon completion, restart your computer to apply any changes made during the installation process.
4. b. OpenAI API Authentication:
   * Register for access to the OpenAI API on the OpenAI website (<https://openai.com/>).
   * Obtain your API key after registration.
   * Store your API key securely and do not share it publicly.
   * In the project folder, locate the python file named “default.py” and set the value of your OpenAPI key in the variable named “open\_ai\_key”.
5. Configuring Settings:a. Setting Up VB-Virtual Audio Cable:
   * Configure the virtual audio cables as needed:
     + Set one cable to capture user audio from VRChat (input).:
     + Set another cable to transmit AI-generated responses (output).
6. Use CABLE B to capture user audio from VRChat (input) : To make it happen configuration should be done both in Windows Sound settings as well as in python file as follows:
   1. Setting in Windows Audio: Go to volume mixture in windows and then select the VRChat option (VR chat should be open to set it for the first time). On the output device audio of VRchat option choose CABLE-B Input (VB- Audio Cable B)
   2. Setting in Python: In the project folder’s “default\_values.py” file set the values for “VB\_Cable\_B\_channel” to the index of CABLE-B Input (VB-Audio Cable B)
7. Use CABLE A to transmit AI-generated audio responses (output) to VRChat. To make it happen configuration should be done both in python file as well as VRChat app as follows:
   1. Setting in python file: In the project folder’s “default\_values.py” file set the values for “VB\_Cable\_A\_channel” to the index of CABLE-A Input (VB-Audio Cable A)
   2. Setting in VRChat app: Open VRchat app and go to settings for audio and in there change the Microphone to CABLE-A Output (VB-Audio Cable A)

Note: For details about the index of available audio channels for audio setups, navigate to the project folder in the terminal and run:

“python audio\_dev.py --all\_channels”

This will provide information about the available virtual channels for A and B input and outputs, which can be set in the “default.py” file. However, the project has already automated the setting for audio channels in the “default.py” file. Nevertheless, in the audio settings for the VR-Chat app and in Windows, the values should be set manually as instructed above.

The final audio configuration visualization is like this:

→ Pass audio from VR chat to python



→ Pass audio from python to VR chat ( let’s Use Cable A)



1. Testing the Overall Working:

To ensure the environment is set up correctly and the functions are working together, you'll need two separate VRChat accounts: one for logging in as the AI Avatar and another for testing purposes, and follow the steps below.

Step 1: Setting Up the AI Avatar Account

1. In the first terminal (PC), launch VRChat and log in with the AI Avatar account.
2. In the same terminal, navigate to the project folder and execute the run.py script using the command python run.py.
3. While executing this, ensure that you've configured the Windows audio settings and VRChat app microphone settings as outlined in the Step-by-Step Installation Guide (referencing point 5).
4. Once the script is running, enter and create an instance of any world associated with the AI Avatar account, ensuring it has public access.

Step 2: Testing with a Secondary Account

1. On another device, launch VRChat and log in with the secondary test user account.
2. Navigate to the same instance of the world previously created or entered by the AI Avatar account.
3. Inside the world, approach the AI avatar and vocalize the wake word, "Hello".
4. The AI avatar should respond with "Hello! How can I help you?".
5. From this point, you can engage in conversation with the AI avatar using the test user account.

## System Configuration Diagram

* + Overview of System Architecture

The overall architecture is represented below



Fig: Overall System Diagram.

When the run.py file is executed, it initiates a continuous listening process within VRChat until a specific wake word, "Hello", is detected via the VB-Cable B. This functionality is managed by the vr\_listen.py file within the project folder.

1. Continuous Audio Monitoring: The vr\_listen.py script continuously monitors the audio stream within VRChat, awaiting the wake word signal.
2. Speech Recognition: Upon detection of the wake word, the speech recognition system implemented within vr\_listen.py converts the spoken text by the user into a textual format.
3. OpenAI API Interaction: The recognized text is then forwarded to the OpenAI API for generating an appropriate response. This interaction is handled by the ai\_chat.py file.
4. Response Conversion to Audio: Once the response is received from the OpenAI API, it is converted back into audio format using the vr\_speak.py script.
5. Audio Transmission to VRChat: Utilizing the PyAudio library, the audio output is fed back into VRChat through the VB-Cable A, enabling the AI avatar to articulate the response.
6. Iterative Process: Upon completion of the response, the AI avatar resumes its listening mode, awaiting further interactions from the user. This iterative process ensures seamless and dynamic communication between the user and the AI avatar within the VRChat environment.

## Error handling: If things go wrong

The system is tested for errors and fulfilling the above mentioned steps one might not get errors. However, in the context there are points where things may go wrong:

1. Error in audio channel settings: Error may come due to mismatch of audio channels and sampling frequency given by the user. So, in that case please run the “audio\_dev.py” file using “python audio\_dev.py” to get the information about the available audio channels for both VB-audio A and B. Then set them accordingly as specified in the above sections.
2. Error due to python versions: Please set the python version to 3.12.0 if any new updates may depreciate dependencies. Also if you use 3.12.0 please install setup tools manually if not automatically installed using “pip install setuptools” (ref: https://stackoverflow.com/questions/69919970/no-module-named-distutils-but-distutils-installed)