A

PROJECT REPORT

ON

“3D Virtual AI Assistant”

SUBMITTED TO

SHIVAJI UNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

MR.	Atharv Mahesh Kulkarni	19UCS067
MR.	Vishal Dineshkumar Joshi	19UCS050
MR.	Shreyas Suhas Kulkarni	19UCS072
MR.	Prashant Krishna Kumbhar	19UCS073
MR.	Shubham Satish Kolap	19UCS061

UNDER THE GUIDANCE OF

Prof. S. S. Sangewar

DKTE

Promoting Excellence in
Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DKTE SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

2022-2023

D.K.T.E.SOCIETY’S

TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN AUTONOMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DKTE
Promoting Excellence in
Teaching, Learning & Research

CERTIFICATE

This is to certify that, project work entitled

“3D Virtual AI Assistant”

is a bonafide record of project work carried out in this college by

MR.	**Atharv Mahesh Kulkarni**	**19UCS067**
MR.	**Vishal Dineshkumar Joshi**	**19UCS050**
MR.	**Shreyas Suhas Kulkarni**	**19UCS072**
MR.	**Prashant Krishna Kumbhar**	**19UCS073**
MR.	**Shubham Satish Kolap**	**19UCS061**

is in the partial fulfillment of the award of degree Bachelor in Technology in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2022-2023.

Prof. Mrs. S. S. Sangewar
(PROJECT GUIDE)

PROF.(DR.) D.V.KODAVADE
(HOD CSE DEPT.)

PROF.(DR.) L.S.ADMOTHE
(DIRECTOR)

EXAMINER: ________________

____

DECLARATION

We hereby declare that, the project work report entitled “3D Virtual AI Assistant” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University,Kolhapur is in partial fulfillment of degree B.E.(CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

MR.	Atharv Mahesh Kulkarni	19UCS067
MR.	Vishal Dineshkumar Joshi	19UCS050
MR.	Shreyas Suhas Kulkarni	19UCS072
MR.	Prashant Krishna Kumbhar	19UCS073
MR.	Shubham Satish Kolap	19UCS061

ACKNOWLEDGEMENT

With great pleasure, we wish to express our deep gratitude to Prof. S. S. Sangewar for his valuable guidance, support, and encouragement in completing this project report.

Also, we would like to take the opportunity to thank our head of department Dr. D. V. Kodavade for his cooperation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Computer Science and Engineering Department for their support, help, and assistance which they extended as and when required.

Thank you,

MR.	Atharv Mahesh Kulkarni	19UCS067
MR.	Vishal Dineshkmar Joshi	19UCS050
MR.	Shreyas Suhas Kulkarni	19UCS072
MR.	Prashant Krishna Kumbhar	19UCS073
MR.	Shubham Satish Kolap	19UCS061

ABSTRACT

A 3D virtual AI assistant is a computer-generated, three-dimensional representation of a personal assistant powered by artificial intelligence. These assistants are designed to understand natural language voice commands and perform various user tasks. They can be integrated with different technologies such as gesture recognition, image/video recognition, speech recognition, and vast dialogue and conversational knowledge base to increase the interaction between humans and machines.

One example of a 3D virtual assistant solution is Pinscreen's AI-driven virtual avatars. Their solution is entirely cloud-based and does not require any game engine on the client side. This ensures high-fidelity renderings independent of which device you are using (mobile, desktop, etc.). All you need is an Internet connection.

One of the goals of Artificial Intelligence (AI) is the realization of natural dialogue between humans and machines. In recent years, dialogue systems, also known as interactive conversational systems, are the fastest-growing area in AI. Many companies have used dialogue systems technology to establish various kinds of Virtual Personal Assistants (VPAs) based on their applications and areas, such as Microsoft's Cortana, Apple's Siri, Amazon Alexa, Google Assistant, and Facebook's M.

There are also 3D virtual assistant solutions that are entirely cloud-based and do not require any game engine on the client-side. This ensures high-fidelity renderings independent of which device you are using (mobile, desktop, etc.). All you need is an Internet connection.

INDEX

1) Introduction:

The 3D AI virtual assistant is a software program that relies on technologies such as natural language processing (NLP) to process voice and commands. They can do many tasks in the same way as human assistants, such as reading, talking, making phone calls. These virtual assistants can interact and communicate with you in a natural way, enhancing your services with human interactions with unprecedented honesty.

3D AI virtual assistants are mostly cloud-based, meaning you can access them from anywhere as long as you have an internet connection. They can be connected to smart devices to integrate them into your daily life. The best 3D artificial intelligence virtual assistant relies on self-learning to be highly personalized. For example, they may learn about your interests or conversation.

These virtual assistants use technologies such as voice recognition and speech synthesis to provide real-time voice and text interaction. They also use deep generative models to create freestyle chatbots that can answer questions and provide information. Real-time lip animation from speech technology allows the 3D avatars to appear more lifelike.

One of the main advantages of using a 3D AI virtual assistant is their ability to provide unparalleled customer service and assistance for e-commerce websites, public spaces, homes, and cars. They are available 24/7 and can answer questions, make recommendations, and find the information you need.

Another advantage of using a 3D AI virtual assistant is its ability to enhance user engagement through its human-like interface. By using data-driven AI approaches that use massive amounts of real-world images to generate lifelike faces in real time, these virtual assistants can create a more engaging and personalized experience for users.

In addition to their ability to provide customer service and enhance user engagement, 3D AI virtual assistants also offer a high level of customization. Companies that offer these services can fully customize the look, voice, personality, and AI of your avatar to meet your specific needs

1.1)Problem Definition:

3D AI Virtual Assistant is a real-time virtual assistant which will be able to communicate with humans based on many domains like day-to-day communication, computer science, engineering, the medical sector, etc. This virtual assistant mainly focuses on analyzing gestures, emotions, faces, voice, and natural language analysis and synthesis. Our main goal is to make such a sophisticated system to integrate all these systems efficiently.

3D AI virtual assistants also offer a high level of security and privacy. Because they are based on the cloud, all data and interactions are securely stored and transmitted. This ensures that your personal information and data are kept safe and secure.

Another advantage of using a 3D AI virtual assistant is their ability to improve efficiency and productivity. By automating many routine tasks such as answering customer inquiries and providing information, these virtual assistants can free up time for businesses to focus on more important tasks.

In summary, 3D AI virtual assistants offer a wide range of benefits including unparalleled customer service, enhanced user engagement, high levels of customization, flexibility, scalability, security, privacy, and improved efficiency and productivity.

1.2)Aim and Objective of the Project:

1) **Aim:**

Developing 3D AI virtual assistant - A multidomain real-time talking virtual assistant with gestures and emotions with multiple features and NLP.

2) **Objective:**

1. To develop User real-time face input data and perform personal assistant tasks.
2. To develop NLP-based User and System interaction.
3. To build Virtual Assistant to perform real-time command tasks, user personal data interaction, and domain-related question answering.
4. To develop a sophisticated real-time 3D model with expression and lip-sync.

1.3)Scope and Limitations of the Project:

1.3.1)Scope:

The scope of 3D virtual AI assistants is vast and they are here to stay. They are becoming more efficient and cost-saving for companies that deploy them. They have potential downsides, including miscommunication and further inequity. Like any form of artificial intelligence (AI), natural language processing (NLP) requires thoughtful governance to ensure it’s used for good, and a framework must be developed to optimize the human-machine interaction.

Virtual assistants using NLP have applications in industries as diverse as healthcare, education, retail, tourism, and others 1. With the growth of the Metaverse and VR technologies, virtual assistants have reached the ultimate 3D AI avatars.

1.3.2)Limitations:

1. **Limited emotional intelligence:** While virtual AI assistants have improved in their ability to understand humans and respond appropriately, they still lack emotional intelligence. They are not able to empathize or understand human emotions, which

can make interactions feel robotic and unresponsive.

2. **Limited customization:** While some virtual AI assistants can be customized to recognize certain voices or preferences, they still lack the ability to fully personalize their interactions. This can make interactions feel impersonal or generic.
3. **Limited knowledge:** Despite being powered by artificial intelligence, virtual assistants still have limitations in their knowledge and ability to understand complex concepts. They may struggle to answer complex or nuanced questions, which can be frustrating for users.
4. **Limited physical interaction:** While virtual assistants can respond to voice commands and perform certain actions, they lack the ability to interact physically with their environment. This limits their usefulness in certain scenarios, such as performing physical tasks or assisting with hands-on projects.
5. **Privacy concerns:** Virtual assistants are always listening for voice commands, which raises privacy concerns for some users. There is a risk that conversations could be recorded and analyzed by third-party companies, which could compromise personal information or sensitive data.

1.4)Timeline of the project:

Phase 1: Planning and Research

Week 1: Define project scope, objectives, and requirements
Week 2: Conduct market research and analyze user needs
Week 3: Finalize project plan and create detailed project schedule
Week 4: prepare project report.
Week 5: First presentation.

Phase 2: Design and Development

Week 5-6: Design and Develop BERT Model and Web Scraper for a virtual assistant.
Week 7-8: Develop the virtual assistant's speech recognition and natural language processing capabilities
Week 9-10: Integrate the whole backend module to work as one unit.
Week 11: Second Presentation of showing code.
Week 12-13: Design and Modelling of 3D character
Week 14-15: Animate 3D character.
Week 16-17: Integration of the whole project
Week 18: Pre-Final/Internal Exam Presentation of the project.

Phase 3: Testing and Launch

Week 19-20: Test thoroughly with Real World Open Domain Questions

1.5)Project Management Plan :

Project initiation:
Define the project objectives, scope, and deliverables
Identify the stakeholders and their requirements
Create a project charter and gain approval from the stakeholders
Form a project team with the necessary skills and expertise

Planning:
Develop a project management plan that outlines the project approach, timeline, budget, and resource allocation
Conduct a risk assessment and develop a risk management plan
Define the project requirements, including the AI algorithms, natural language processing, voice recognition, 3D modeling, and animation
Create a detailed project schedule and task list
Identify the technical and infrastructure requirements, including hardware, software, and data storage
Establish a communication plan to ensure that stakeholders are kept informed of progress and changes

Execution:
Develop the AI algorithms and natural language processing capabilities
Create the 3D model and animation of the virtual assistant
Integrate the AI algorithms, natural language processing, and 3D modeling and animation
Test the virtual assistant to ensure that it functions as expected and meets the project requirements
Train the virtual assistant on a variety of tasks and use cases

Monitoring and control:
Monitor progress against the project schedule and budget
Manage risks and issues as they arise
Conduct regular testing and quality assurance to ensure that the virtual assistant continues to meet the project requirements
Report on project status and progress to stakeholders
Manage changes to the project scope, timeline, and budget as necessary

Project closure:
Obtain sign-off from stakeholders that the project objectives have been met
Archive project documentation and data
Conduct a post-project review to identify lessons learned and areas for improvement
Release the virtual assistant to users, if applicable

1.5.1)Milestone List:

Milestone	Description	Start Date	End Date	Duration In Days	Priority
Literature Review and Research	Conduct a comprehensive review of existing literature and research on 3D AI Virtual assistant	1/9/2022	30/9/2022	30	High
Data Scraping And Preprocessing	Scrape the data for training the model and preprocess the data for further analysis	2/10/2022	23/10/2022	20	Medium
Machine Learning Model Implementation and Training	Implement and train machine learning models for 3D AI virtual assistant using the preprocessed dataset and extracted features.	25/10/2022	10/12/2022	45	High
Evaluation and Performance Testing	Evaluate the performance of the trained classification model using appropriate performance metrics and testing procedures.	15/12/2022	5/1/2023	20	High
3D AI Virtual	Performance improvement	7/1/2023	3/2/2023	25	High

Assistant Development					
Integration and System Testing	Performance improvement related testing	5/2/2023	2/3/2023	25	Medium
Documentation And Report Writing	Document the project's methodology, findings, and outcomes, and write the final project report	3/3/2023	28/3/2023	25	Medium
Project Presentation and Submission	Prepare and deliver a project presentation, and submit the final report and any other required deliverables	1/4/2023	22/4/2023	20	Medium

1.6)Project Cost:

1.6.1)COCOMO Model:

The COCOMO (Constructive Cost Model) is a regression model used to estimate the effort and development time for software projects. It is based on the size of the software product, measured in Kilo Lines of Code (KLOC). For an Embedded project with 1400 Lines of Code (LOC), we first convert LOC to KLOC by dividing by 1000:

KLOC = 1400 / 1000 = 1.4

Then we use the Basic COCOMO model equations for effort and development time:

Effort = a * (KLOC)^b Development Time = c * (Effort)^d

For an Embedded project, the values of the constants a, b, c, and d are:

a = 3.6

b = 1.20

c = 2.5

d = 0.32

Substituting these values and the calculated KLOC value into the equations, we get: Effort = 3.6 * (1.4)^1.20 = 5.29 person-months Development Time = 2.5 * (5.29)^0.32 = 4.22 months So according to the Basic COCOMO model, our Embedded project with 1400 LOC would require an estimated effort of 5.29 person-months and an estimated development time of 4.22 months.

These are rough estimates and actual values may vary depending on various factors specific to our project.

2) Background study and literature overview

2.1) Literature Overview:

There is a lot of literature available on 3D virtual AI assistants. According to a research paper by Shakti Arora et al., virtual assistants are playing a very crucial role in the day-to-day activities of humans 1. They are designed with natural language processing and are used for performing simple tasks. The AI that powers a virtual assistant uses natural language processing (NLP), machine learning, and a speech recognition platform.

With the growth of the Metaverse and VR technologies, virtual assistants have reached the ultimate 3D AI avatars. Cutting-edge technology from companies like Nvidia is capable of producing nearly true-to-life human avatars in real-time.

A 3D virtual AI assistant is a computer-generated avatar that uses artificial intelligence to interact with users. They are designed with natural language processing and can perform simple tasks. With the growth of the Metaverse and VR technologies, virtual assistants have reached the ultimate 3D AI avatars. Cutting-edge technology from companies like Nvidia is capable of producing nearly true-to-life human avatars in real-time.

2.2)Critical appraisal of other people's work:

The study of 3D AI Virtual Assistants has been a topic of interest for researchers in recent years. However, there are some limitations and gaps in the existing literature that need to be addressed. For instance, some studies have focused on using only one type of feature or algorithm for 3D AI Virtual Assistant, which may not be sufficient for accuracy. Additionally, many 3D AI Virtual Assistants are particularly using only some features algorithms, and because of that end users not getting fully satisfied with using AI.also the information generated can be biased and unreliable. Therefore, it is important for future research to explore more diverse and robust approaches to 3D AI Virtual Assistants.

Furthermore, while some studies have investigated the impact of AI is getting increasing, there is still much to be learned about how these factors interact with other variables such as age, gender, and location. Additionally, there is a need for more research on the ethical implications of this topic, particularly regarding issues of privacy and data security. Overall, critical appraisal of the existing literature suggests that there is a need for more interdisciplinary and collaborative research in this field, in order to develop a more effective and inclusive system that helps to reduce much more time as well as manual work, with high accurac**y.**

2.3)Investigation of the current project and related work:

Maya:

Maya is a 3D virtual assistant designed for home automation. She provides users with a more interactive and visually appealing experience when controlling their smart home devices.

Using voice commands or gestures, users can interact with Maya to control various aspects of their home, such as lights, temperature, security systems, and more. Maya's 3D nature allows for a more intuitive and immersive interface.

For example, if a user wants to turn on the lights in their living room, they can simply say, "Maya, turn on the lights in the living room." Maya will respond and visually display the layout of the home, highlighting the living room area. The user can then use gestures or voice commands to control the lights, such as swiping or saying, "Dim the lights to 50%."

Maya can also provide visual feedback by displaying the status of each smart device in the home. For instance, if a user asks, "Is the front door locked?" Maya can visually show the status of the door lock, indicating whether it is locked or unlocked.

Additionally, Maya can offer personalized home automation suggestions and recommendations. She can analyze user preferences, usage patterns, and environmental factors to optimize energy consumption or suggest automation routines based on the user's lifestyle.

The 3D interface of Maya enhances the user's understanding and control of their smart home environment. It allows for a more immersive and visually engaging interaction, making home automation more intuitive and accessible. With Maya's assistance, users can easily manage and customize their smart home devices, creating a more convenient and comfortable living environment

Ava:

Ava is a 3D virtual assistant designed to provide users with an immersive and interactive experience across various tasks and interactions. As a realistic humanoid avatar projected in a virtual environment, Ava offers a visually engaging and human-like interaction.

Users can interact with Ava using voice commands, gestures, or a combination of both, allowing for a natural and intuitive communication process. Ava's 3D capabilities enable her to display visual representations, and interactive elements, and provide a more personalized experience.

Ava's functionalities span a wide range of tasks. For example, she can assist with travel-related inquiries by helping users find and book flights. Through her 3D interface, Ava can display visual flight options, provide 360-degree views of aircraft cabins, and simulate real-time weather conditions at the destination.

Additionally, Ava can support users in organizing their schedules and managing their calendars. She can set reminders, send notifications, and provide visual representations of upcoming events or appointments.

Ava's knowledge base extends to various domains, allowing her to answer general knowledge questions, offer recommendations for restaurants or entertainment, and provide personalized suggestions based on user preferences.

By combining her 3D nature with conversational abilities, Ava creates a more immersive and engaging virtual assistant experience. Her visual representations, interactive features, and natural language processing capabilities contribute to a richer and more human-like interaction for users seeking assistance across a range of tasks.

Sam:

Sam is a 3D virtual assistant designed for customer support in the e-commerce industry. He offers users a more interactive and visually engaging experience when seeking assistance with product-related queries, order tracking, and issue resolution.

Users can interact with Sam through voice commands or text-based communication to address their customer support needs. Sam's 3D avatar provides a human-like representation that enhances the user's engagement and creates a more personalized support experience.

For example, if a user has a question about a specific product, they can ask Sam for assistance. Sam will respond conversationally, providing detailed information about the product, including specifications, features, and customer reviews. He can display images, videos, and virtual demos to help users visualize the product better.

Sam can also assist with order tracking and the resolution of issues. If a user needs to track their order, they can provide Sam with relevant details, such as the order number or tracking ID. Sam can then display the current status of the order and provide real-time updates.

In case of issues or concerns, Sam can guide users through troubleshooting steps or escalate the matter to a customer support representative if necessary. Sam's 3D interface allows him to display visual instructions or diagrams to assist users in resolving common issues.

The 3D nature of Sam creates a more interactive and visually appealing customer support experience. Users can receive personalized assistance, view product details in a more engaging manner, and follow visual instructions for issue resolution. Sam's presence as a 3D virtual assistant adds a human-like touch to the customer support process, fostering a more interactive and satisfying user experience.

3) Requirement analysis:

3.1) Requirement Gathering:

3.1.1). Functional Requirements:

- Determine the scope of the virtual assistant's capabilities. This might include tasks such as scheduling appointments, sending messages, providing information, or controlling objects within the 3D environment.
- Identify the different types of input that the virtual assistant should be able to recognize. This might include voice commands, typed messages, or gestures within the 3D environment.
- Define how the virtual assistant will learn and improve over time. This might include machine learning algorithms or other artificial intelligence techniques.
- Define the user interface for the virtual assistant, including how users will interact with it within the 3D environment.
- Define how the virtual assistant will present information to the user, including visual cues or audio responses.

3.1.2) Technical Requirements:

- Determine how the virtual assistant will recognize and interact with objects and people within the 3D environment. This might include using computer vision techniques or other sensors.
- Determine the natural language processing (NLP) capabilities required for the virtual assistant to generate realistic and natural language responses.
- Define how the virtual assistant will use context to provide relevant responses to the user's requests.
- Define the integration points for the virtual assistant with other systems, such as calendars, email, or messaging platforms.
- Determine the hardware and software requirements for the virtual assistant, including any compatibility considerations.

3.1.3) User Requirements:

- Identify the types of users who will be interacting with the virtual assistant, including their demographic information and specific needs.
- Understand the user's goals and expectations for interacting with the virtual assistant within the 3D environment.
- Identify any specific tasks or workflows that users will perform with the virtual assistant.
- Understand the user's preferred mode of interaction with the virtual assistant, whether it be through voice, text, or gestures.

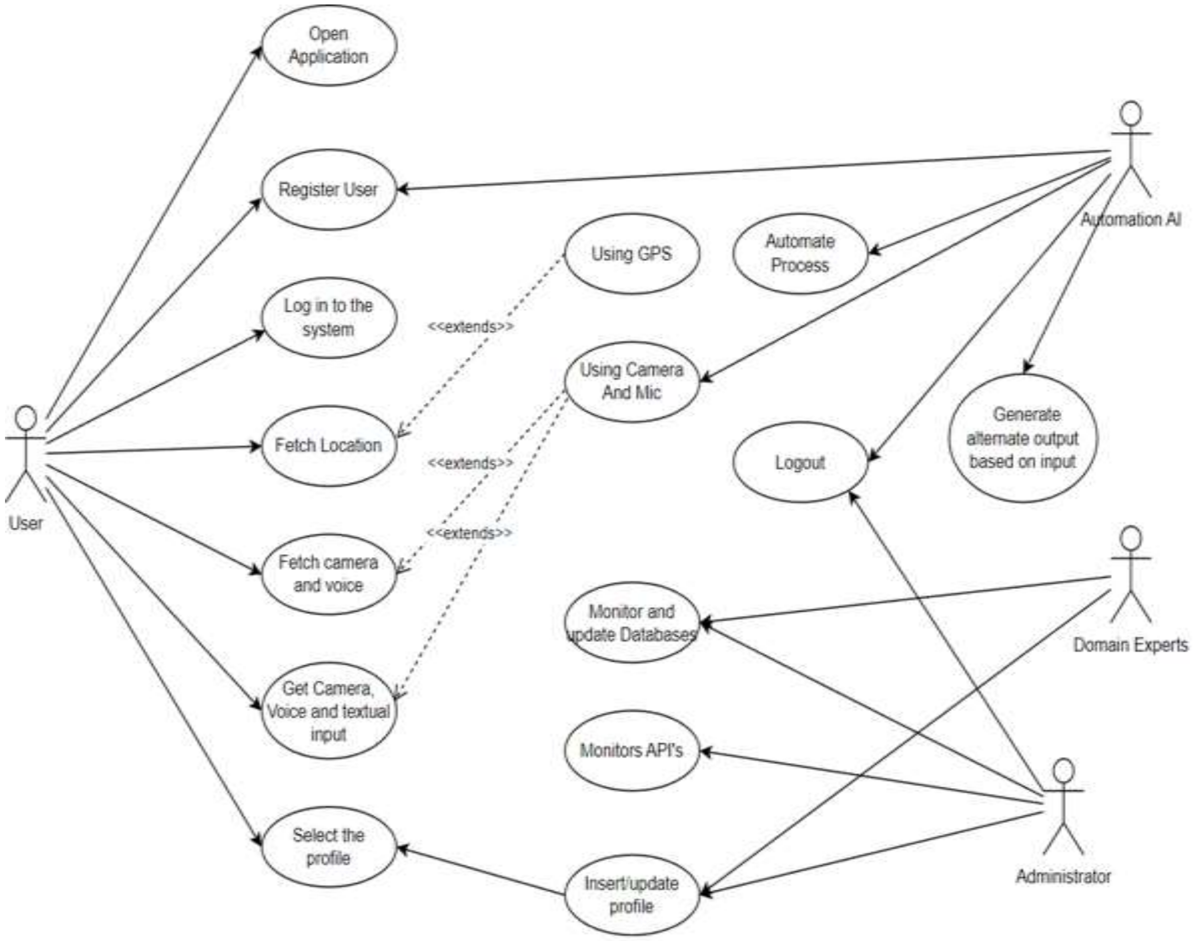
3.2) Requirement Specification:

Personal assistant software is required to act as an interface into the digital world by understanding user requests or commands and then translating them into actions or recommendations based on the agent’s understanding of the world. we focus on relieving the user of entering text input and using voice as the primary means of user input. The agent then applies voice recognition algorithms to this input and records the input. It then uses this input to call one of the personal information management applications such as a task list or calendar to record a new entry or to search about it on search engines like Google, Bing or Yahoo, etc. The focus is on capturing the user input through voice, recognizing the input, and then executing the tasks if the agent understands the task. Software takes this input in natural language, and so makes it easier for the user to input what he or she desires to be done. Voice recognition software enables hands-free use of the applications, and lets users to query or command the agent through a voice interface. This helps users to have access to the agent while performing other tasks and thus enhances the value of the system itself. JIA also have ubiquitous connectivity through Wi-Fi or LAN connection, enabling distributed applications that can leverage other APIs exposed on the web without a need to store them locally. Virtual assistants must provide a wide variety of services.

These include:

- Providing information such as weather, facts from e.g. Wikipedia, etc.
- Set an alarm or making to-do lists and shopping lists.
- Remind you of birthdays and meetings.
- Play music from streaming services such as Saavn and Gaana.
- Play videos, TV shows, or movies on televisions, streaming from e.g. Netflix or Hotstar.
- Book tickets for shows, travel, and movies.

3.3)Use Case Diagram:



Use Case Diagram

3.3)Use case description:

3.3.1)User:

Open Application: The user opens the 3D Virtual Assistant application.

Register User: The user creates a new account to use the 3D Virtual Assistant.

Log in to the system: The user logs in to their account to use the 3D Virtual Assistant.

Fetch Location: The 3D Virtual Assistant fetches the user's location using GPS.

Fetch Camera and Voice: The 3D Virtual Assistant fetches input from the user's camera and microphone.

Get Camera, Voice, and Textual Input: The 3D Virtual Assistant receives input from the user's camera, microphone, and text input.

Select Profile: The user selects a profile to use in the 3D Virtual Assistant application.

Using GPS: The 3D Virtual Assistant uses the user's location to provide location-based services.

Using Camera and Mic: The 3D Virtual Assistant uses the input from the user's camera and microphone to provide services.

3.3.2) Automation AI:

Automate Process: The Automation AI automates certain tasks or processes in the 3D Virtual Assistant.

Using Camera and Mic: The Automation AI uses input from the user's camera and microphone to perform certain tasks.

Logout: The Automation AI logs the user out of the 3D Virtual Assistant.

Generate Alternate Output based on input: The Automation AI generates an alternate output based on the user's input.

3.3.3)Domain Experts:

Monitor and Update Databases: Domain Experts monitor and update the databases used by the 3D Virtual Assistant.

Insert/Update Profile: Domain Experts insert or update user profiles in the 3D Virtual Assistant application.

3.3.4) Administrator:

Monitor and Update Databases: The Administrator monitors and updates the databases used by the 3D Virtual Assistant.

Monitor APIs: The Administrator monitors the APIs used by the 3D Virtual Assistant.

Insert/Update Profile: The Administrator inserts or updates user profiles in the 3D Virtual Assistant application.

Actors:

Actors	Description
Mobile user	Responsible for accessing the facilities provided by the system.
Administrator	Responsible for data entry and data management on the server.
The user	is Responsible for providing the domain as well as additional knowledge to the administrator.
Automation	Responsible for automating registration of the new users and performing required tasks

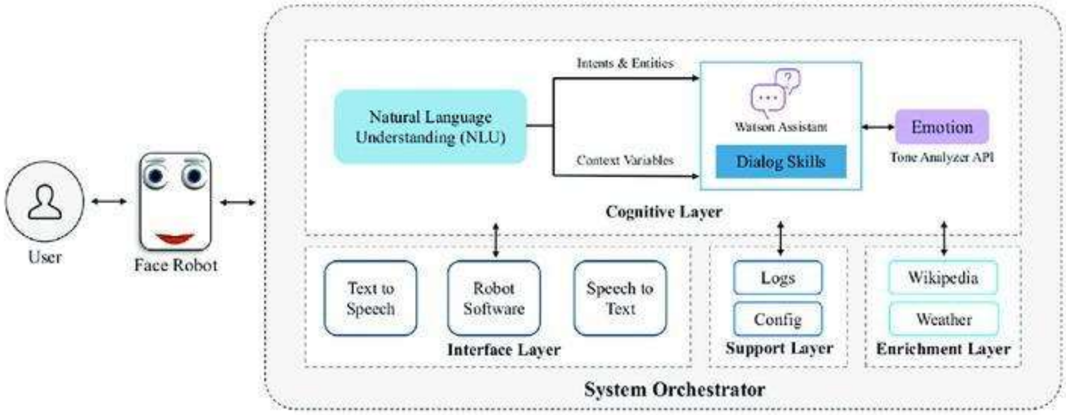
Use case:

Use Case	Description
Open application	The user has to open the application on his system.
Register User	User is registered by using automation.
Log in to the system	User can log in to the application by using the registered credentials.
Fetch Location	The application will automatically fetch the location of the user in the background.
Fetch Camera and voice	Camera and voice are fetched for automation or giving tasks
Select the profile	Profile is required for the administrator use.

Using GPS	For accurate tracking of the user’s location GPS is used.
Monitor and update databases	Administrator or automation can insert and updates the information of the user in the database.
Monitor APIs	Multiple APIs are needed for fetching the required information.
Insert and Update Profile	Administrator or Automation can insert and updates the information on the database.
Automation AI	It generates alternate output based on input and it is also used in registering the user.
Domain Expert,	inserts and updates the database as well as monitor and updates it.
Administrator	It can perform tasks like registration, log out, and all tasks of domain experts.

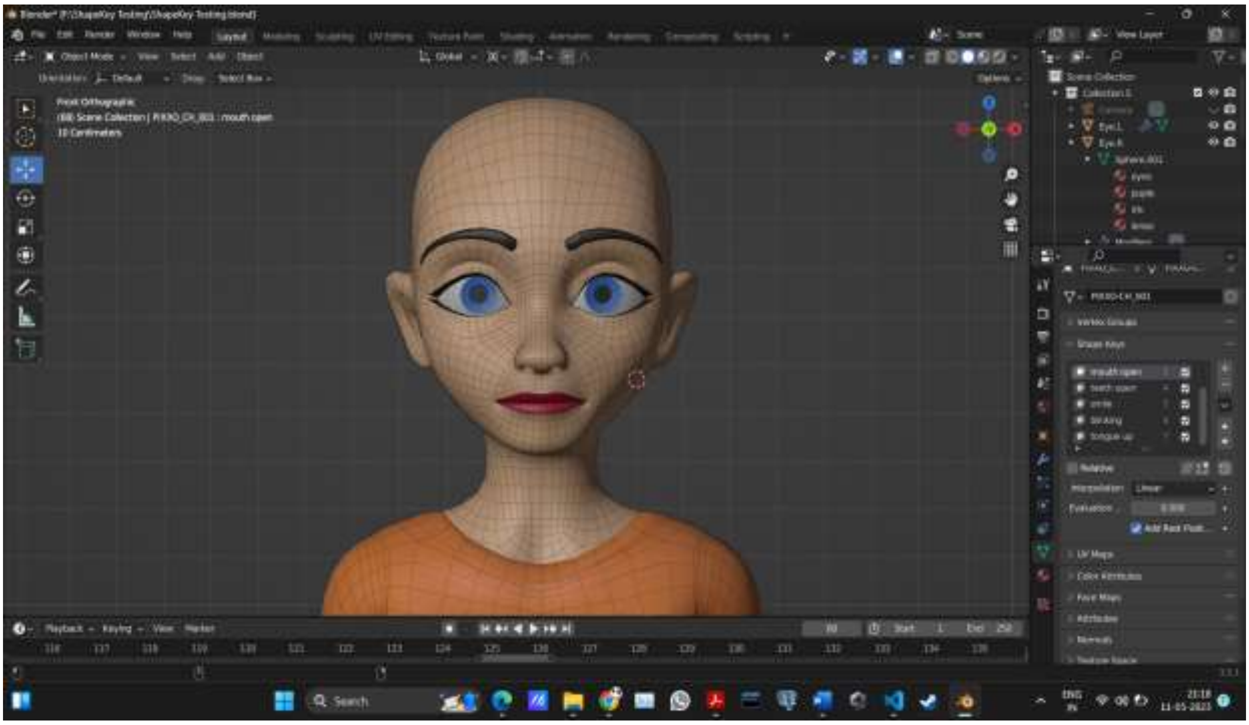
4. System design:

4.1)Architectural Design:



4.1.1)Architectural Design

4.2) User Interface Design:



User interface Design

4.2.1) Description:

The screenshot shows the user interface of the 3D Virtual Assistant application. The user interface consists of several sections and elements that are designed to provide an intuitive and user-friendly experience.

At the top of the screen, there is a navigation bar that provides access to various sections of the application. The navigation bar includes options to access the user's profile, settings, help, and logout. The profile section allows the user to view and edit their profile information, while the settings section provides options to configure various settings of the application.

Below the navigation bar, there is a search bar that allows the user to search for specific content or services within the application. The search bar is followed by a carousel of featured services or content that the user can access quickly.

Further down on the screen, there is a section for location-based services. This section displays the user's current location on a map and provides options to search for nearby services or businesses.

Below the location-based services section, there is a section for voice commands. This section allows the user to give voice commands to the 3D Virtual Assistant, which will perform various tasks or provide information based on the user's input.

At the bottom of the screen, there are quick action buttons that provide access to frequently used features or services. The quick action buttons include options to make a call, send a message, take a picture, and access the camera.

Overall, the user interface of the 3D Virtual Assistant application is designed to provide a seamless and intuitive experience for the user, with easy access to various features and services.

4.3) An algorithmic description of each module:

4.3.1)Google Maps.py:

i) Identify the goal of the script:

The first step in creating an algorithm is to clearly define the problem that needs to be solved. In this case, the goal of the script appears to be to scrape information about a route between two locations from Google Maps.

ii) Choose appropriate tools and libraries**:**

Once the goal of the script has been identified, the next step is to choose the tools and libraries that will be used to achieve this goal. In this case, the script uses Selenium for web automation and BeautifulSoup for web scraping.

iii) Navigate to the Google Maps website:

The script uses Selenium to open a web browser and navigate to the Google Maps website.

iv) Enter the desired search query:

The script takes in a string data as an input and uses it to generate a URL for a Google Maps search. The URL is constructed by replacing spaces in the source and destination strings with + symbols and appending them to the base Google Maps URL. The script then navigates to this URL.

v) Parse the page source:

Once the page has loaded, the script uses BeautifulSoup to parse the page source and extract information about the route between the two locations.

vi) Extract relevant information:

The script uses BeautifulSoup's find method to locate the div element that contains information about the route. It then extracts this information and returns it.

4.3.2) Google_Scraper.py:

i) Identify the goal of the script:

The first step in creating an algorithm is to clearly define the problem that needs to be solved. In this case, the goal of the script appears to be to scrape information from Google search results.

ii) Choose appropriate tools and libraries:

Once the goal of the script has been identified, the next step is to choose the tools and libraries that will be used to achieve this goal. In this case, the script uses Selenium for web automation and BeautifulSoup for web scraping.

iii) Take user input for a search query:

The script prompts the user to enter a search query.

iv) Construct a Google search URL using the query:

The script takes the user's search query and constructs a Google search URL by replacing spaces in the query with + symbols and appending it to the base Google search URL.

v) Navigate to the constructed URL using Selenium:

The script uses Selenium to open a web browser and navigate to the constructed Google search URL.

vi) Parse the page source using BeautifulSoup:

Once the page has loaded, the script uses BeautifulSoup to parse the page source and extract information from the search results.

vii) Extract information from search results:

The script uses BeautifulSoup's find and find_all methods to locate specific elements on the page that contain information about the search results. It then extracts this information and stores it in variables.

viii) Store extracted information in a CSV file:

The script stores the extracted information in a CSV file for later analysis.

4.3.3) Google_Speech_to_text.py:

i) Import the Google Cloud Speech-to-Text library:

The first step in this algorithm is to import the speech module from the google.cloud library. This module provides access to the Google Cloud Speech-to-Text API.

ii) Instantiate a SpeechClient object:

The next step is to create an instance of the SpeechClient class. This object provides methods for interacting with the Speech-to-Text API.

iii) Specify the location of the audio file:

The script specifies the location of the audio file to transcribe by setting the value of the gcs_uri variable. This variable contains a string that represents the URI of an audio file stored in Google Cloud Storage.

iv) Create a RecognitionAudio object:

The script creates an instance of the RecognitionAudio class using the gcs_uri variable as an argument. This object represents the audio data that will be transcribed.

v) Set transcription configuration options:

The script creates an instance of the RecognitionConfig class and sets various configuration options for the transcription, such as the audio encoding, sample rate, and language code.

vi) Transcribe the audio file:

The script calls the recognized method of the SpeechClient object, passing in the RecognitionConfig and RecognitionAudio objects as arguments. This method sends a request to the Speech-to-Text API to transcribe the specified audio file.

vii) Print the resulting transcript:

The script iterates over the results returned by the recognized method and prints the transcript of each result.

4.3.4) Youtube_Scraper.py:

i) Import necessary libraries:

The first step in this algorithm is to import the necessary libraries, including BeautifulSoup from the bs4 library for web scraping, web driver from the selenium library for web automation, and other libraries such as CSV, time, and web browser.

ii) Configure Selenium options:

The script sets various options for the Selenium web driver, such as ignoring certificate errors, running in incognito mode, and running headless.
Take user input for a search query: The script prompts the user to enter a search query for a YouTube video.

iii) Check for certain keywords in the search query:

The script checks if the search query contains certain keywords such as “latest video” or “shorts” and sets a flag variable accordingly.

iv) Construct a YouTube search URL based on the search query:

Depending on whether certain keywords were found in the search query, the script constructs a YouTube search URL by appending the appropriate path to the base YouTube URL.

v) Navigate to the constructed URL using Selenium:

The script uses Selenium to open a web browser and navigate to the constructed YouTube search URL.

vi) Parse the page source using BeautifulSoup:

Once the page has loaded, the script uses BeautifulSoup to parse the page source and extract information about the search results.

vii) Extract information about the first search result:

The script uses BeautifulSoup’s find method to locate the first search result on the page and extracts its URL.

viii) Open the first search result in a web browser:

The script uses Python’s built-in web browser module to open the URL of the first search result in a web browser.

4.3.5) Face_Recognition_algorithm:

i) import necessary libraries:

The first step in creating a face recognition algorithm using dlib is to import the necessary libraries, including dlib for face detection and recognition, and other libraries such as NumPy and cv2 for image processing.

ii) Load pre-trained models:

The next step is to load pre-trained models for face detection and recognition. Dlib provides pre-trained models for detecting faces in images and for extracting facial features to generate face embeddings.

iii) Read input image:

The algorithm reads an input image using a library such as cv2 or PIL.

iv) Convert image to grayscale:

The input image is converted to grayscale to reduce the amount of data that needs to be processed.

v) Detect faces in the image:

Given the grayscale input image, the algorithm uses the loaded face detection model to detect faces in the image. This can be done using dlib's get_frontal_face_detector function. Extract face regions: Once the faces have been detected, the algorithm extracts the regions of the image that contain each face.

vi) Align faces:

The extracted face regions are aligned to ensure that the facial features are in a consistent position across all faces.

vii) Extract facial features:

Once the faces have been aligned, the algorithm uses the loaded facial feature extraction model to extract facial features from each face. This can be done using dlib's shape_predictor function.

viii) Generate face embeddings:

The extracted facial features are then used to generate face embeddings, which are numerical representations of the faces. This can be done using dlib's face_recognition_model_v1 function.

ix) Normalize face embeddings:

The generated face embeddings are normalized to ensure that they have a consistent scale.

x) Store face embeddings:

The normalized face embeddings are stored in a database or data structure for later use.

xi) Read test image:

The algorithm reads a test image containing a face that needs to be recognized. Convert test image to grayscale: The test image is converted to grayscale to reduce the amount of data that needs to be processed.

xii) Detect face in test image:

The algorithm uses the loaded face detection model to detect the face in the test image.

xiii) Extract face region from test image:

The region of the test image containing the detected face is extracted.

xiv) Align face in test image:

The extracted face region from the test image is aligned to ensure that its facial features are in a consistent position with the stored faces.

xv) Extract facial features from the test image:

The algorithm uses the loaded facial feature extraction model to extract facial features from the aligned face region in the test image.

xvi) Generate face embedding for test image:

The extracted facial features from the test image are used to generate a face embedding for the test image.

xvii) Normalize face embedding for test image:

The generated face embedding for the test image is normalized to ensure that it has a consistent scale with the stored embeddings.

xviii) Compare face embeddings:

The normalized face embedding for the test image is compared with the stored embeddings to determine if it matches any of them. This can be done by calculating the Euclidean distance between the embeddings and setting a threshold value for determining if two faces are a match.

4.3.6) Text to Speech:

i) Import necessary libraries:

The first step in this algorithm is to import the necessary libraries, including gTTS from the gtts library for text-to-speech conversion and pyglet for playing audio files.

ii) Take user input for text to be converted:

The script prompts the user to enter the text that they want to convert to speech.

iii) Create a gTTS object:

The script creates an instance of the gTTS class using the user’s input text as an argument.

iv) Set language for text-to-speech conversion:

The script sets the language for the text-to-speech conversion by passing in a language code as an argument when creating the gTTS object.

v) Generate speech from the text:

The script calls the save method of the gTTS object to generate speech from the input text and save it as an MP3 file.

vi) Load AVbin library:

The script attempts to load the AVbin library using pyglet.lib.load_library. This library is required for playing audio files with pyglet.

vii) Set AVbin flag:

If the AVbin library is successfully loaded, the script sets a flag variable to indicate that AVbin is available.

viii) Load saved MP3 file:

The script uses pyglet.media.load to load the saved MP3 file containing the generated speech.

ix) Play-loaded MP3 file:

The script calls the play method of the loaded MP3 file to play it.

x) Run pyglet app:

The script calls pyglet.app.run to run the pyglet app and play the audio file.

xi) Handle exceptions:

If an exception occurs while loading the AVbin library or playing the audio file, the script catches it and prints a message indicating that it is being ignored.

xii) Stop pyglet app:

Once the audio file has finished playing, the pyglet app is stopped and control is returned to the user.

xiii) Delete saved MP3 file:

Optionally, after playing the audio file, you can delete it from your system using Python’s built-in os. remove function.

xiv) Exit program:

Once all steps have been completed, control is returned to the user and they can choose to exit or run again.

xv) Repeat process:

Optionally, you can repeat this process by prompting for new user input and generating new speech from it.

4.3.7) 3D Face Animation code:

i) Import necessary libraries:

The first step in this algorithm is to import the necessary libraries for working with Blender and audio files.

ii) Load 3D model:

The algorithm loads the 3D model of the character that will be animated.

iii) Set up shape keys:

The algorithm sets up shape keys for the character’s facial expressions and lip movements. Shape keys are used to define different shapes for the character’s face, which can be blended together to create complex expressions.

iv) Define visemes:

The algorithm defines a set of visemes, which are visual representations of speech sounds. Each viseme corresponds to a specific shape of the character’s mouth.

v) Load audio file:

The algorithm loads an audio file containing speech that will be used for lip-syncing.

vi) Extract audio features:

The algorithm extracts relevant features from the audio file, such as the timing and amplitude of speech sounds.

vii) Map audio features to visemes:

The algorithm maps the extracted audio features to the defined visemes to determine which visemes should be displayed at which times.

viii) Generate keyframes for lip sync:

The algorithm generates keyframes for the character’s lip movements based on the mapped visemes. Each keyframe specifies the shape of the character’s mouth at a specific point in time.

ix) Add keyframes to animation:

The generated keyframes are added to the character’s animation to create lip sync. Define facial expressions: The algorithm defines a set of facial expressions for the character, such as smiling or frowning.

x) Map facial expressions to audio features:

The algorithm maps the defined facial expressions to relevant audio features, such as changes in pitch or volume.

xi) Generate keyframes for facial expressions:

The algorithm generates keyframes for the character’s facial expressions based on the mapped audio features. Each keyframe specifies the shape of the character’s face at a specific point in time.

xii) Add keyframes to animation:

The generated keyframes are added to the character’s animation to create facial expressions.

xiii) Preview animation:

The algorithm previews the animation to ensure that it looks natural and believable.

xiv) Adjust keyframes as needed:

If necessary, the algorithm adjusts the generated keyframes to improve the quality of the animation.

xv) Render animation:

Once all adjustments have been made, the algorithm renders the final animation.

xvi) Export animation:

The rendered animation is exported in a format that can be played back on other devices.

xvii) Load new audio file:

Optionally, you can load a new audio file and repeat this process to generate new animations for different speech inputs.

xviii) Adjust shape keys and visemes as needed:

Optionally, you can adjust the defined shape keys and visemes to improve the quality of future animations.

xix) Repeat process:

Optionally, you can repeat this process by loading new audio files and generating new animations for them.

4.3.8) Mathematics.py:

i) Import necessary libraries:

The first step in this algorithm is to import the necessary libraries, including the custom myfunctions module that contains the implementation of various mathematical functions.

ii) Enter an infinite loop:

The mathematics.py script enters an infinite loop that prompts the user to enter a command on each iteration.

iii) Take user input for a command:

The script prompts the user to enter a command, such as “addition” or “subtraction”.

iv) Extract numbers from the command:

The script extracts any numbers present in the user’s command using a list comprehension and stores them in a list.

v) Check for specific keywords in the command:

The script checks if the user’s command contains specific keywords such as “addition” or “subtraction” by iterating over a list of possible keywords and using the in operator to check if each keyword is present in the user’s command.

vi) Call the appropriate function from myfunctions module:

Depending on which keyword was found in the user’s command, the script calls the appropriate function from the myfunctions module and passes in the extracted numbers as arguments. For example, if the user’s command contains the keyword “addition”, the script calls the addition function from the myfunctions module and passes in the extracted numbers as arguments.

vii) Print the result of the function call:

The script prints the result returned by the called function using Python’s built-in print function.

viii) Repeat process:

The script repeats this process by prompting for new user input and calling the appropriate function from the myfunctions module on each iteration of the loop.

4.3.9) separate algorithm for modules in functions.py:

4.3.9.1)addition function:

1. Define a function named addition that takes a single argument, a list of numbers.
2. Initialize a variable sum to 0 to keep track of the sum of the numbers.
3. Iterate over the list of numbers using a for loop.
4. For each number in the list, add it to the sum variable.
5. After all numbers have been added, return the value of the sum variable.

4.3.9.2)subtraction function:

1. Define a function named subtraction that takes two arguments, a list of numbers and a counter variable.
2. Check the value of the counter variable using an if statement.
3. If the counter variable is 0, initialize a variable sub to the last element in the list of numbers and iterate over all but the last element in the list using a for loop.
4. For each number in the list, subtract it from the sub variable.
5. If the counter variable is 1, initialize a variable sub to the first element in the list of numbers and iterate over all but the first element in the list using a for loop.
6. For each number in the list, subtract it from the sub variable.
7. After all numbers have been subtracted, return the value of the sub variable.

4.3.9.3)multiplication function:

1. Define a function named multiplication that takes a single argument, a list of numbers.
2. Initialize a variable multi to 1 to keep track of the product of the numbers.
3. Iterate over the list of numbers using a for loop.
4. For each number in the list, multiply it by the multi variable.
5. After all numbers have been multiplied, return the value of the multi variable.

4.3.9.4)division function:

1. Define a function named division that takes a single argument, a list of numbers.
2. Initialize a variable divi to the last element in the list of numbers.
3. Iterate over all but the last element in the list using a for loop.
4. For each number in the list, divide it by the current value of the divi variable and update its value with this result.
5. After all numbers have been divided, return the value of the divi variable.

4.3.9.5)average function:

1. Define a function named average that takes a single argument, a list of numbers.
2. Call the previously defined addition function with this list as an argument and store its result in a variable named summ.
3. Divide this result by the length of this list and store it in another variable named ‘avg’.
4. Return this result.

4.3.9.6)factorial function:

1. Define a function named factorial that takes a single argument, a list containing a single number.
2. Initialize a variable fact to 1 to keep track of the factorial of the number.
3. Iterate over the range from 1 to the number + 1 using a for loop.
4. For each value in the range, multiply it by the fact variable.
5. After all values have been multiplied, return the value of the fact variable.

4.3.9.7)DecimalToBinary function:

1. Define a recursive function named DecimalToBinary that takes two arguments, a decimal number and an initially empty list.
2. Check if the decimal number is greater than or equal to 1 using an if statement.
3. If it is, call the DecimalToBinary function recursively with the decimal number divided by 2 and the list as arguments.
4. Append the remainder of the decimal number divided by 2 to the list.
5. Return the list.

4.3.9.8)square function:

1. Define a function named square that takes a single argument, a list containing a single number.
2. Return the square of this number.

4.3.9.9)cube function:

1. Define a function named cube that takes a single argument, a list containing a single number.
2. Return the cube of this number.

4.3.9.10)power function:

1. Define a function named power that takes a single argument, a list containing two numbers.
2. Return the first number raised to the power of the second number.

4.3.9.11)sin function:

1. Define a function named sin that takes a single argument, a list containing an angle in degrees.
2. Convert this angle to radians by multiplying it with pi/180 using math.pi
3. Calculate and return sine of this angle rounded to 2 decimal places using math.sin

4.3.9.12)cos function:

1. Define a function named cos that takes a single argument, a list containing an angle in degrees.
2. Convert this angle to radians by multiplying it with pi/180 using math.pi
3. Calculate and return cosine of this angle rounded to 2 decimal places using math.cos

4.3.9.13)tan function:

1. Define a function named tan that takes a single argument, a list containing an angle in degrees.
2. Convert this angle to radians by multiplying it with pi/180 using math.pi
3. Calculate and return tangent of this angle rounded to 2 decimal places using math.tan

4.3.9.14)cot function:

1. Define a function named cot that takes a single argument, a list containing an angle in degrees.
2. Convert this angle to radians by multiplying it with pi/180 using math.pi
3. Calculate and return cotangent of this angle rounded to 2 decimal places by dividing 1 by math.tan of this angle.

4.3.9.15)sec function:

1. Define a function named sec that takes a single argument, a list containing an angle in degrees.

2. Convert this angle to radians by multiplying it with pi/180 using math.pi
3. Calculate and return secant of this angle rounded to 2 decimal places by dividing 1 by math.cos of this angle.

4.3.9.16)cosec function:

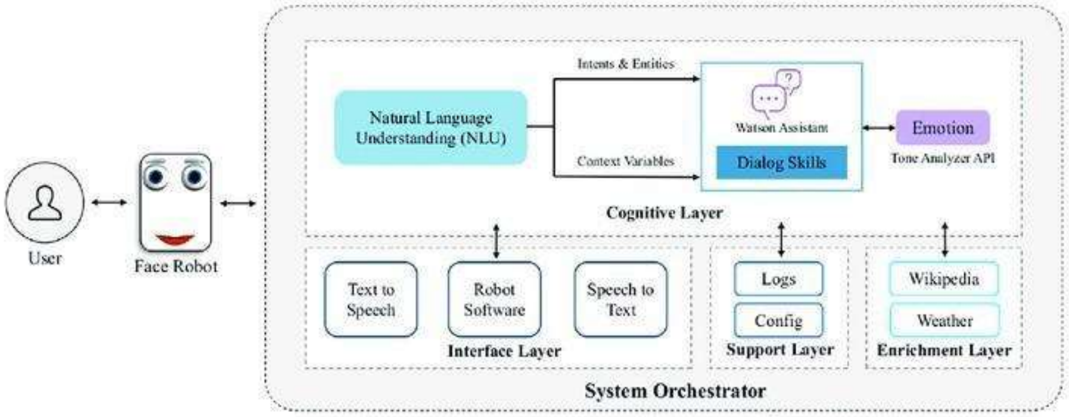
1. Define a function named cosec that takes a single argument, a list containing an angle in degrees.
2. Convert this angle to radians by multiplying it with pi/180 using math.pi
3. Calculate and return cosecant of this angle rounded to 2 decimal places by dividing 1 by math.sin of this angle.

4.3.9.17)derivative function:

1. Define a function named derivate that takes a single argument, a list containing an expression and the variable with respect to which we want to take derivative.
2. Prompt user for input for the expression and the variable with respect to which we want to take derivative.
3. Use sympy.diff method to calculate derivative of the expression with respect to the variable.
4. Return the result.

4.4) System Modeling

4.4.1) Architectural Diagram:



Architectural Diagram

Description:

The architectural diagram shows the different layers and components of the 3D Virtual Assistant system. The system is designed using a three-tier architecture, consisting of the presentation layer, application layer, and data layer.

The presentation layer is responsible for handling the user interface and displaying the information to the user. This layer consists of various components, including the web server, mobile application, and voice assistant. The web server handles the web-based user interface, while the mobile application provides a native mobile experience. The voice assistant allows the user to interact with the system using voice commands.

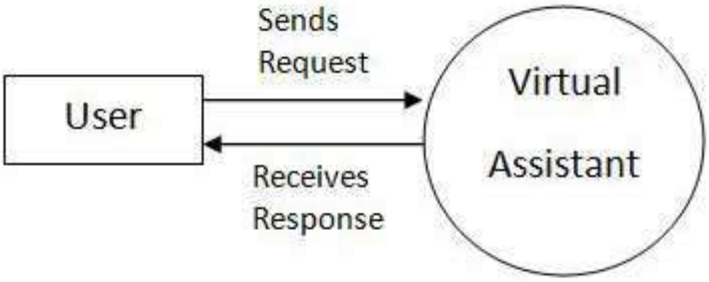
The application layer is responsible for handling the business logic and processing the user requests. This layer consists of various components, including the API gateway, authentication service, speech recognition module, natural language processing (NLP) module, and media storage service. The API gateway serves as the entry point for all requests coming into the system. The authentication service is responsible for verifying the user's credentials and validating their session tokens. The speech recognition module converts the user's voice input into text. The NLP module analyzes the user's intent and determines the appropriate response. The media storage service stores the user's media files, such as pictures and videos.

The data layer is responsible for storing and managing the system's data. This layer consists of various components, including the database server, location-based services, and external APIs. The database server stores the user's profile information and media files. The location-based services allow the system to determine the user's location and provide information on nearby services or businesses. The external APIs allow the system to retrieve information from third-party services, such as weather or news APIs.

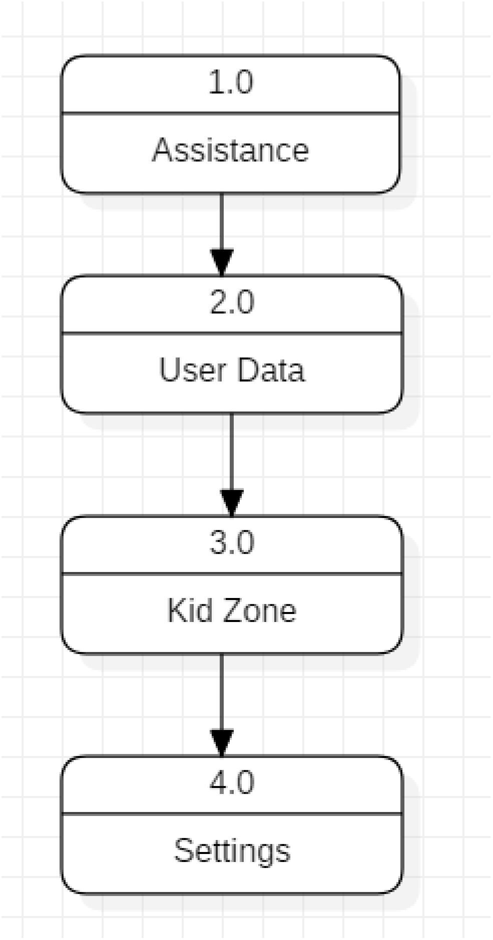
Overall, the architectural diagram shows how the different components of the 3D Virtual Assistant system are organized and interact with each other. The three-tier architecture provides a scalable and maintainable solution for the system's development and deployment.

4.4.2)Dataflow Diagram:

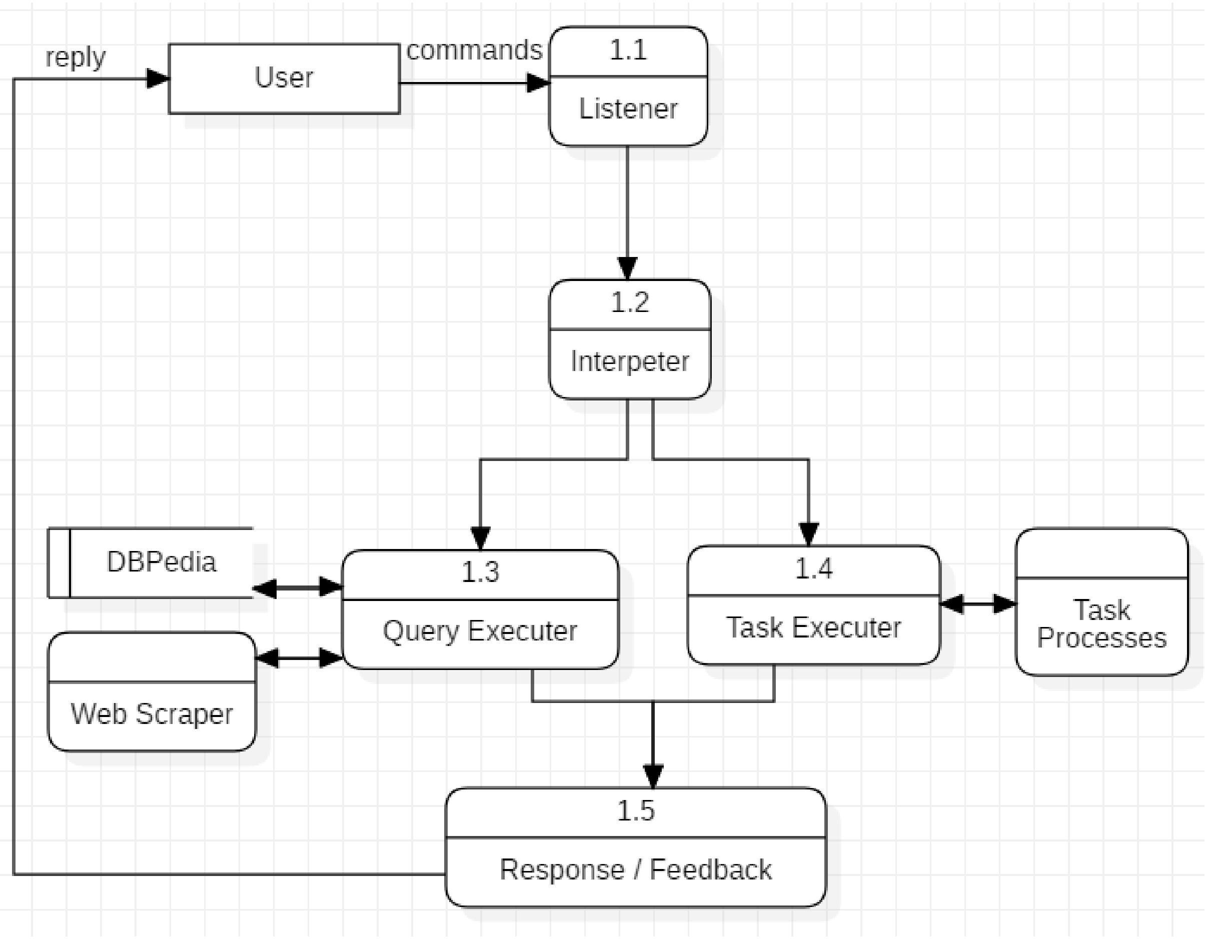
4.4.2.1)Dataflow diagram level 0 (Context level Diagram):



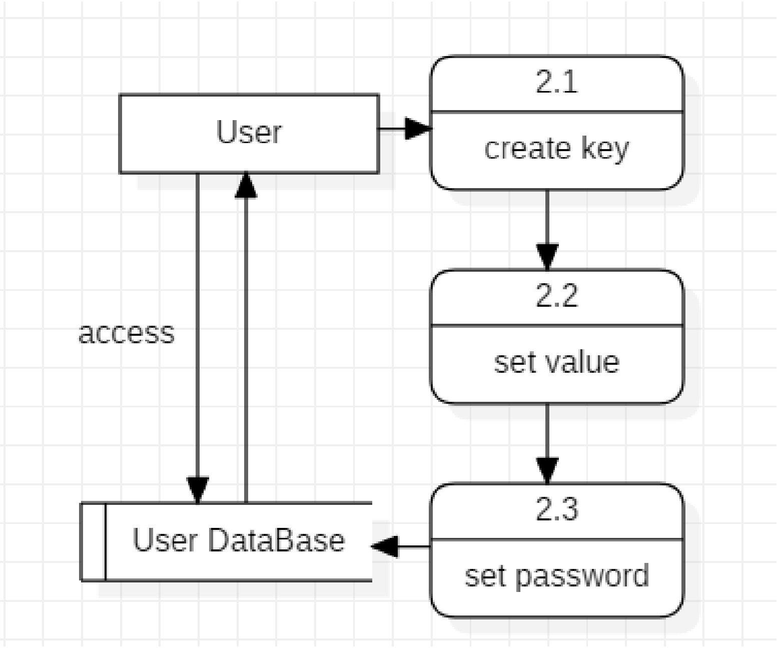
4.4.2.2)Data Flow Diagram level 1:



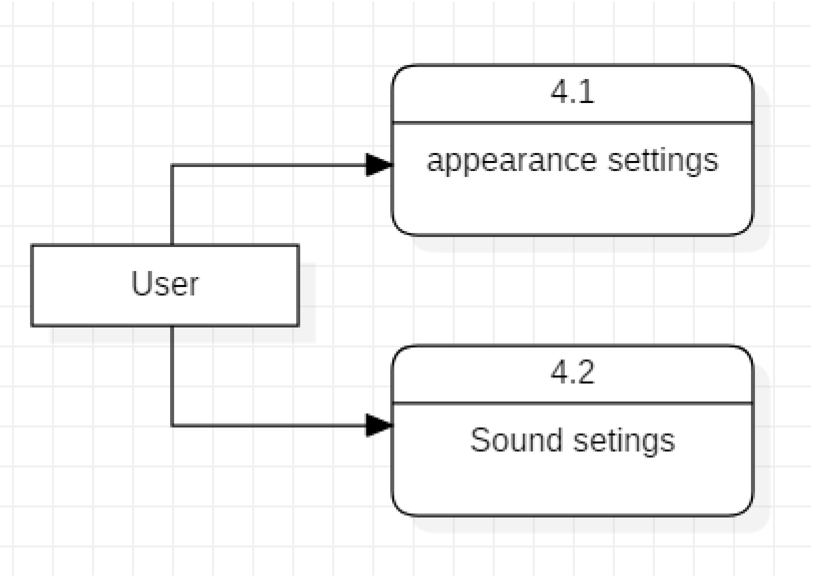
4.4.2.3)Dataflow diagram level 2:



Dataflow in Assistance



Managing User Data



Setting of Virtual Assistant

Description:

Dataflow Diagram level 0:

A level 0 data flow diagram (DFD), also known as a context diagram, is similar to a level 1 DFD in that it provides a high-level overview of a system and its interactions with external entities. However, a level 0 DFD differs from a level 1 DFD in that it does not show any internal processes or data flows within the system.

In a level 0 DFD, the system is represented as a single process, typically drawn as a circle or rounded rectangle in the center of the diagram. External entities are represented by squares or rectangles and are connected to the system process by data flows, which are depicted as arrows. These data flows show the inputs to and outputs from the system.

A level 0 DFD provides a high-level view of the system and its interactions with external entities. It does not show any internal details of the system or its data processing. Instead, it provides an overview of the system’s inputs and outputs and how it interacts with its environment.

Dataflow Diagram level 1:

A data flow diagram (DFD) can be drawn at different levels of detail to represent different aspects of a system. A level 1 DFD, also known as a context diagram, provides a high-level overview of the system and its interactions with external entities. It shows the system as a single process and represents its inputs, outputs, and external entities.

In a level 1 DFD, the system is represented by a single process, typically drawn as a circle or rounded rectangle in the center of the diagram. External entities are represented by squares or rectangles and are connected to the system process by data flows, which are depicted as arrows. These data flows show the inputs to and outputs from the system.

A level 1 DFD provides a high-level view of the system and its interactions with external entities. It does not show the internal details of the system or its data processing. Instead, it provides an overview of the system’s inputs and outputs and how it interacts with its environment

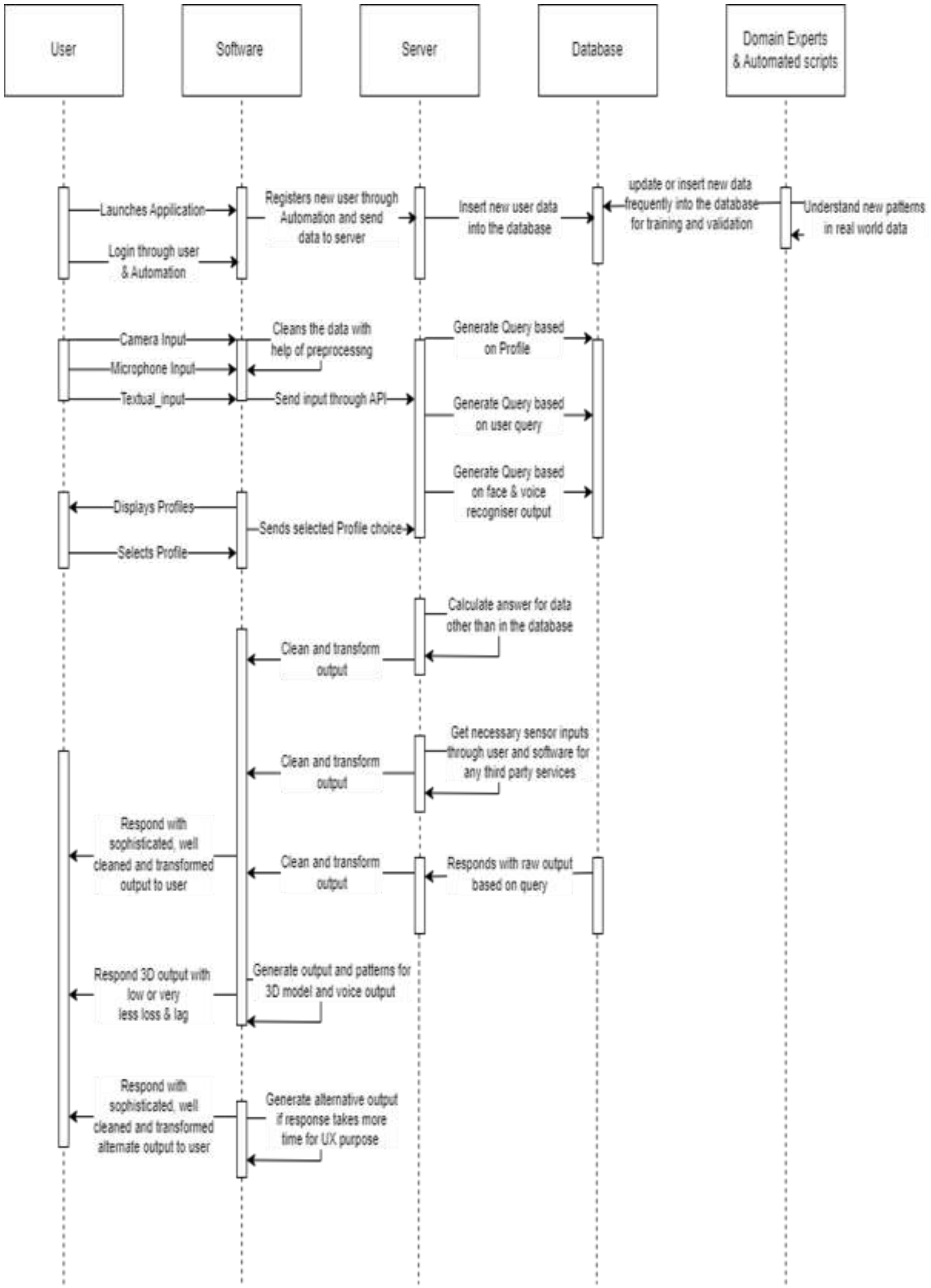
Dataflow Diagram level 2:

A level 2 data flow diagram (DFD) provides a more detailed view of a system than a level 1 DFD. It shows the major processes and data stores within the system and how they interact with each other. A level 2 DFD decomposes the single process shown in a level 1 DFD into its sub-processes and represents the data flows between them.

In a level 2 DFD, processes are represented by circles or rounded rectangles and are connected by data flows, which are depicted as arrows. Data stores are shown as open-ended rectangles and represent the storage of data within the system. External entities, which represent sources or destinations of data outside the system, are also shown as squares or rectangles.

A level 2 DFD provides a more detailed view of the system than a level 1 DFD. It shows the major processes and data stores within the system and how they interact with each other. This can help in understanding the internal workings of the system and its data processing.

4.4.3)Sequence Diagram:



User
Software
Server
Database
Domain Experts & Automated scripts
Launches Application
Login through user & Automation
Registers new user through Automation and send data to server
Insert new user data into the database
update or insert new data frequently into the database for training and validation
Understand new patterns in real world data
Camera Input
Microphone Input
Textual_input
Cleans the data with help of preprocessng
Send input through API
Generate Query based on Profile
Generate Query based on user query
Generate Query based on face & voice recogniser output
Displays Profiles
Selects Profile
Sends selected Profile choice
Calculate answer for data other than in the database
Clean and transform output
Get necessary sensor inputs through user and software for any third party services
Clean and transform output
Respond with sophisticated, well cleaned and transformed output to user
Clean and transform output
Responds with raw output based on query
Respond 3D output with low or very less loss & lag
Generate output and patterns for 3D model and voice output
Respond with sophisticated, well cleaned and transformed alternate output to user
Generate alternative output if response takes more time for UX purpose

Sequence Diagram

4.4.4) Sequence Diagram Description:

The sequence diagram represents the interaction between the user and the 3D Virtual Assistant application. It starts with the user sending a request to the application to open it. The application then responds by displaying the login page, which allows the user to enter their login credentials.

After the user logs in successfully, the application sends a request to the server to fetch the user's profile information. Once the server responds with the profile information, the application displays it on the user interface.

The user can then perform various actions such as searching for nearby locations or services, taking pictures or videos, and using the voice assistant to perform tasks. When the user performs any of these actions, the application sends a request to the server to fetch the required information or perform the necessary tasks.

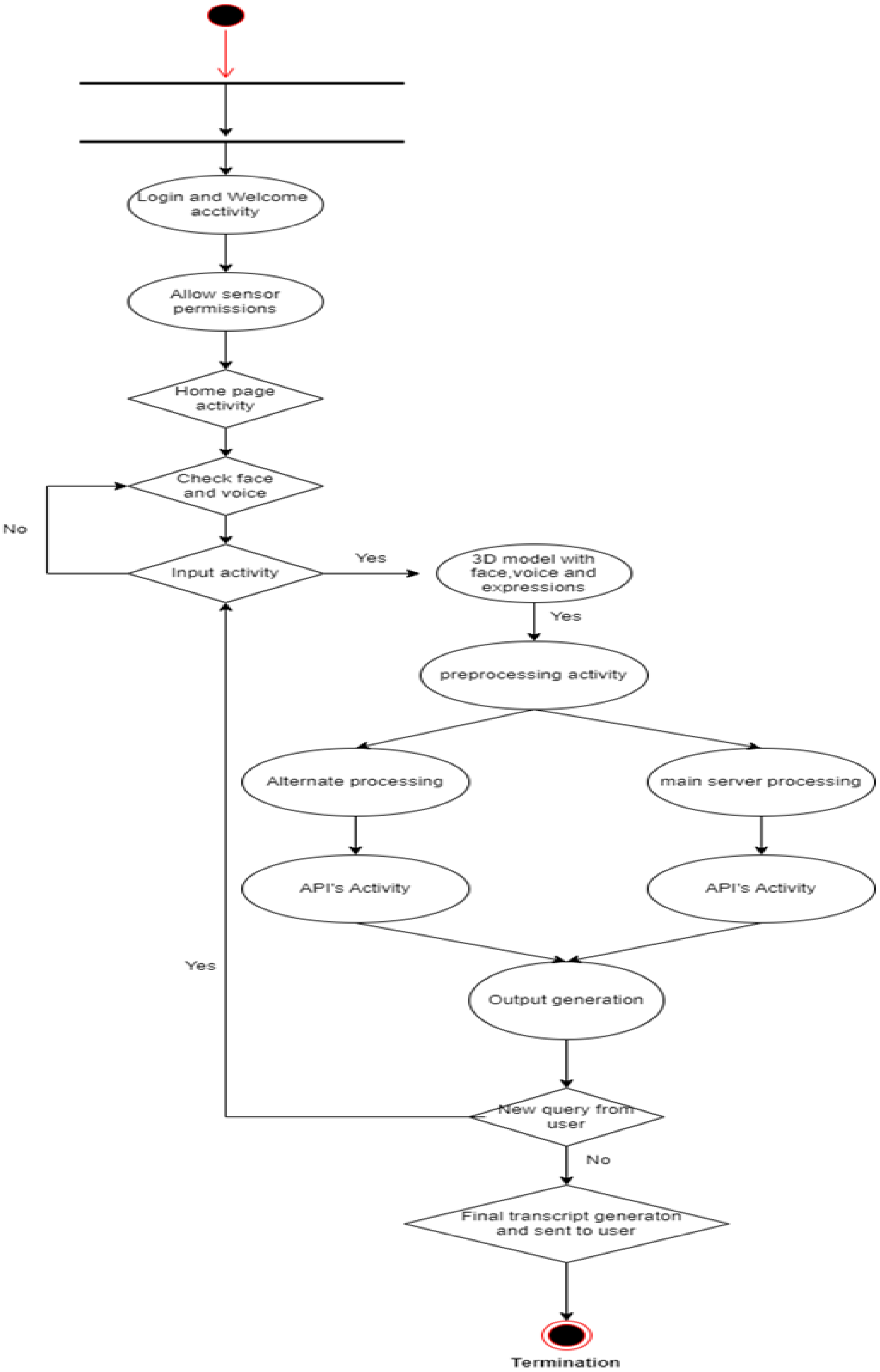
If the user uses the voice assistant to perform a task, the application first sends the voice input to the speech recognition module, which converts it into text. The text is then sent to the natural language processing (NLP) module, which analyzes the user's intent and sends a response to the application with the appropriate action to be taken.

If the user takes a picture or video, the application sends a request to the server to store the media files in the database.

Finally, when the user logs out of the application, the application sends a request to the server to terminate the session and invalidate the user's token.

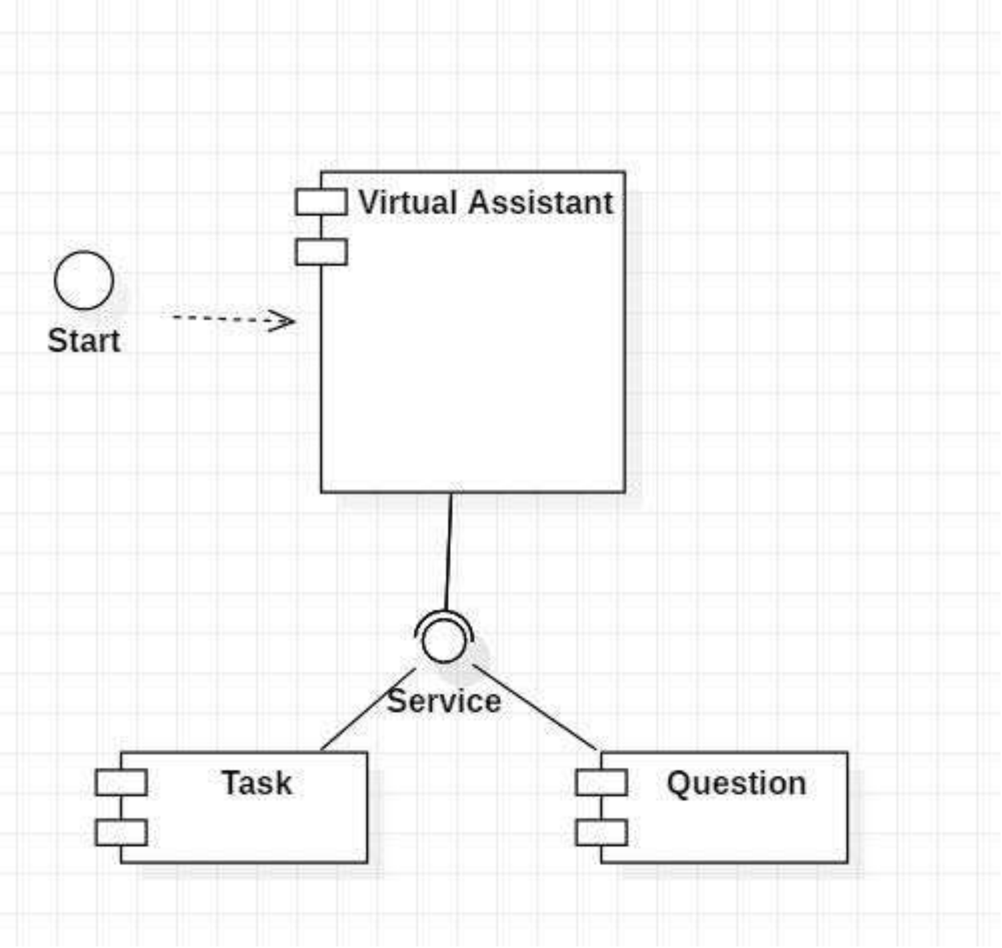
Overall, the sequence diagram represents the flow of communication and interaction between the user, the application, and the server in the 3D Virtual Assistant system.

4.4.5)Activity Diagram:



Activity Diagram

4.4.6) Component Diagram



Component Diagram

4.4.6.1) Component Diagram Description

The User Interface component represents the interface that users will interact with to use the system. The Automation AI component represents the artificial intelligence algorithms and software that will automate certain processes within the system. The Domain Experts component represents the individuals or teams with the necessary domain expertise to develop and maintain the system.

The Database component represents the system's underlying database that stores information relevant to the system's operations. The APIs component represents the various APIs that the system may interact with or provide to external systems.

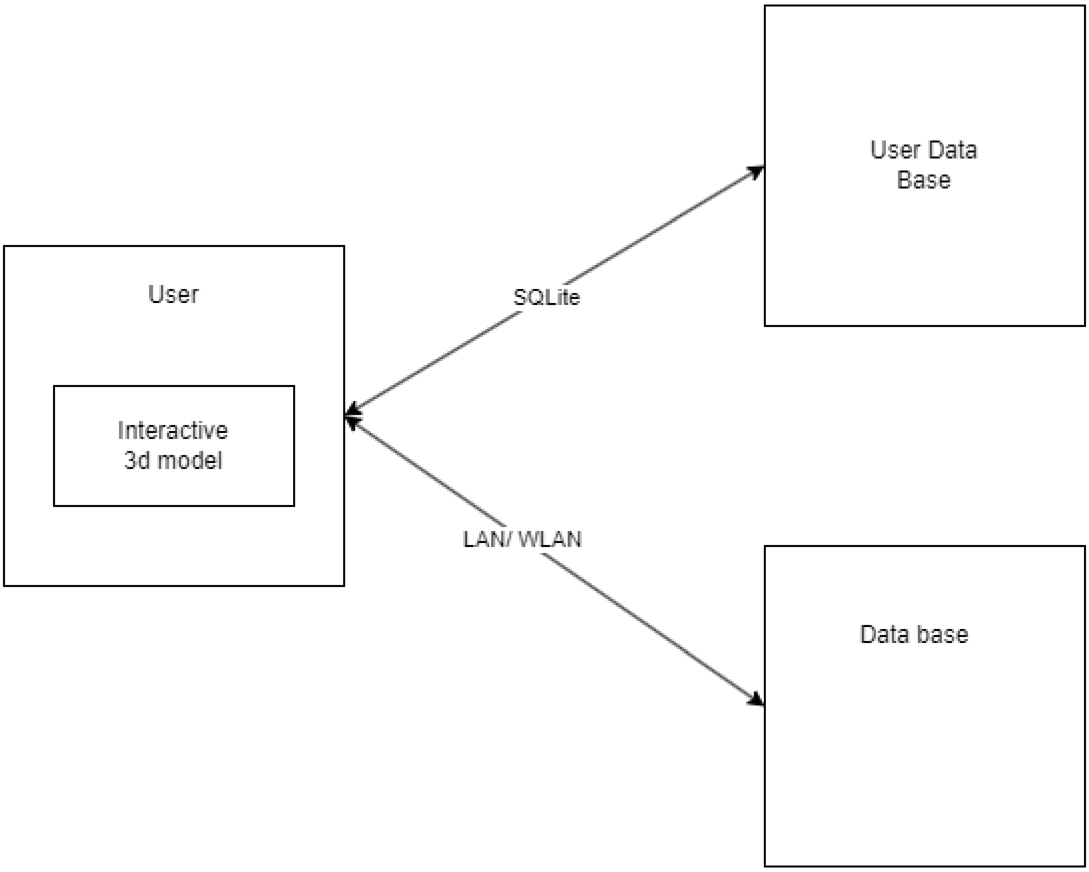
The component diagram also includes interfaces, which are represented by the small circles on the components. For example, the User Interface component has interfaces such as "Login," "Select Profile," "Get Camera and Voice Input," and "Fetch Location." Similarly, the Automation AI component has interfaces such as "Automate Process," "Logout," and "Generate alternate Output based on input."

The diagram also shows the dependencies between the components. For example, the User Interface component depends on the Automation AI component, which in turn depends on the Domain Experts component. Additionally, the Database component is connected to the Domain Experts and API's components, indicating that they interact with the system's underlying database.

Finally, the diagram includes the deployment target, which is represented by the cloud icon on the right side of the diagram. This indicates that the system is deployed on a cloud-based server or infrastructure.

In summary, the component diagram provided in the drive link represents the high-level architecture of a 3D Virtual Assistant system, including its various components, interfaces, dependencies, and deployment target.

4.4.7)Deployment Diagram



Deployment Diagram

4.4.7.1)Deployment diagram Description:

A deployment diagram is a type of UML diagram that models the physical architecture of a system. It shows how the components of the system are distributed across hardware nodes and how they interact with each other. Deployment diagrams are used to represent the hardware topology of a system, its middleware infrastructure, software distribution, and runtime processing nodes.

In a deployment diagram, nodes represent physical elements such as servers and devices while components represent modular parts of the system such as executables and libraries. These elements are connected by relationships that show how they interact with each other

5) Implementation

5.1.1)Pseudo Code:

```
@ recognise faces on a live webcam feed:
# import required python libraries\
# load the face recognition ML model
data = pickle.loads(open('face_enc', "rb").read())
# capture real time face image
video_capture = cv2.VideoCapture(0)
# capture image until face is infront of the camera and convert in grey color
while True:
ret, frame = video_capture.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
# convert the input frame from BGR to RGB
rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
# loop over the recognized faces.
# rescale the face coordinates and draw the predicted face name on the image.
```

```
@ recognise faces on a live webcam feed:
# import required python libraries\
# load the face recognition ML model
data = pickle.loads(open('face_enc', "rb").read())
# capture real time face image
video_capture = cv2.VideoCapture(0)
# capture image until face is infront of the camera and convert in grey color
while True:
ret, frame = video_capture.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
# convert the input frame from BGR to RGB
rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
# loop over the recognized faces.
# rescale the face coordinates and draw the predicted face name on the image.user_audio_db.add(r);
except:
query.out('please speak again');
# recoginize_() method will throw a request error if the API.
```

5.1.2)LOGIN:

```
session_start();
if($_SERVER["REQUEST_METHOD"] == "POST")
// username and password sent from form
$myusername =
mysqli_real_escape_string($db,$_POST['username'])
;
$mypassword =
mysqli_real_escape_string($db,$_POST['password'])
;$sql = "SELECT name FROM employee
WHERE email = '$myusername' and pass =
'$mypassword'";
$result = mysqli_query($db,$sql);
$row =
mysqli_fetch_array($result,MYSQLI_ASSOC);
$active = $row['active'];
$count = mysqli_num_rows($result);
// If result matched $myusername and
$mypassword, table row must be 1 row
if($count == 1)
session_register("myusername");
$_SESSION['login_user'] = $myusername;
header("location: welcome.php");
else
$error = "Your Login Name or Password is
invalid";
```

5.1.3) Logout:

```
session_start();
session_unset();
header("Location: index.html")
```

5.1.4)Model API Algorithm:

```
class main_logic:
if (user.is_authenticated):
camera_input = request.get(camera);
voice_input = request.get(microphone_input);
text_input = request.get(text_input);
if(camera_input ) :
recog_face=model.face_recognition (camera_input)
recog_emotion=model.emotion_recognition (camera_input)
if(voice_input ) :
recog_voice = model.voice_recognition(voice_input)
```

```
if(text_input ) :
cleaned_text = clean.text(text_input);
```

5.1.5) Model API Algorithm:

```
class main_logic:
if (user.is_authenticated):
camera_input = request.get(camera);
voice_input = request.get(microphone_input);
text_input = request.get(text_input);
if(camera_input ) :
recog_face=model.face_recognition (camera_input)
recog_emotion=model.emotion_recognition (camera_input)
if(voice_input ) :
recog_voice = model.voice_recognition(voice_input)
if(text_input ) :
cleaned_text = clean.text(text_input);
Class query_cal {
If final_ans in db_based_on_profile:
db.call(final_ans)
If final_ans in calculation :
calculator.call(final_ans);
# model calling continues as per text input…….
}
```

5.2) Environmental Setting for Running the Project:

To run a 3D AI virtual assistant project in Blender and deploy it on the web, you need to ensure that your environment has the necessary software and hardware requirements. Here are some key environmental settings you should consider:

5.2.1)Hardware Requirements:

To run a 3D AI virtual assistant project smoothly, you will need a powerful computer with a fast processor, plenty of RAM, and a good graphics card. Additionally, if you plan to deploy the project on the web, you will need a reliable and fast internet connection.

5.2.2)Software Requirements:

The key software required for this project are Blender, a 3D modeling and animation software, and a web development framework such as Flask, Django or Node.js for deploying the project on the web.

5.2.3)Operating System:

Blender runs on Windows, macOS, and Linux operating systems. You should ensure that you have a compatible operating system and that it meets the minimum requirements for running Blender.

5.2.4)Development Environment:

You will need an Integrated Development Environment (IDE) such as Visual Studio Code, PyCharm or Sublime Text to develop and test the code for the virtual assistant.

5.2.5)Libraries and Dependencies:

You will need to install any libraries and dependencies required by your project, such as Python packages, 3D models, and audio files.

5.2.6)Web Server:

You will need to deploy your project on a web server to make it accessible on the web. You can use a cloud-based hosting service such as AWS, Google Cloud or Azure to host your web application.

5.2.7)Security:

You should consider the security implications of running a 3D AI virtual assistant project on the web. You need to ensure that the web application is secure and protected against potential attacks such as SQL injection, cross-site scripting, and cross-site request forgery.

Overall, you need to ensure that your hardware, software, and development environment meet the necessary requirements to run and deploy a 3D AI virtual assistant project on the web. It is also essential to ensure that your project is secure and protected against potential attacks**.**

5.3) Detailed Description of Methods:

5.3.1) Natural Language Processing (NLP):

NLP is a technique that allows computers to interpret and understand human language. This is a crucial component of a 3D AI virtual assistant, as it enables the assistant to understand spoken or typed commands from the user. NLP algorithms can analyze the input text or speech, recognize the intent behind the message, and provide an appropriate response.

5.3.2) Machine Learning (ML):

Machine learning algorithms are used in 3D AI virtual assistants to enable them to learn from data and improve their performance over time. By analyzing data sets, the assistant can learn to recognize patterns and make more accurate predictions. For example, an assistant that helps with interior design might use machine learning to analyze different furniture styles and suggest matching pieces.

5.3.3) Animation:

Animations are an essential aspect of a 3D AI virtual assistant, as they help to create a more engaging and interactive experience for the user. Animations can be used to simulate the assistant's movements, facial expressions, and other gestures to create a more human-like appearance.

5.3.4) Voice Recognition:

Voice recognition is a technology that allows a computer to recognize spoken words and phrases. This is a critical component of a 3D AI virtual assistant, as it enables the assistant to respond to voice commands. Voice recognition algorithms use machine learning techniques to analyze spoken words, recognize patterns, and convert them into text.

5.3.5) 3D Modeling and Animation Software:

3D modeling and animation software such as Blender are essential tools for creating the 3D environment in which the virtual assistant operates. This software allows developers to create 3D models of objects, characters, and environments, and to animate them to simulate movement and interaction with the virtual assistant.

5.3.6) Dialogue Management:

Dialogue management is the process of controlling the flow of conversation between the user and the virtual assistant. This involves using algorithms to understand the context of the conversation, and to choose appropriate responses that maintain the flow of the conversation and achieve the desired outcome.

5.3.7) Computer Vision:

Computer vision is a field of AI that enables computers to analyze and interpret visual data, such as images and videos. This can be used in 3D AI virtual assistants to enable them to recognize objects, gestures, and facial expressions. Computer vision techniques can be used to analyze video input from a camera to track the user's movements and gestures, and to respond appropriately.

5.4) Implementation Details:

5.4.1)Backend development:

The first step in building a 3D AI virtual assistant is to develop the backend using Python. Python is an excellent language for developing the backend of the virtual assistant because of its simplicity and versatility. The backend is responsible for handling the user requests, processing the natural language, performing actions, and sending responses back to the user. In this project, BERT was used for natural language processing (NLP).

5.4.2)Modeling and animation:

The next step is to create the 3D models and animations for the virtual assistant using Blender. Blender is an open-source 3D modeling and animation software that is widely used in the industry. The virtual assistant model can be created by designing and customizing a pre-existing character model or creating a new one from scratch.

5.4.3)Integration with Google APIs:

To enhance the functionality of the virtual assistant, various Google APIs can be used. In this project, the Google Maps API, Google Search API, Google Calendar API, and Voice API were used. The APIs provide access to various features such as location data, search results, calendar events, and text-to-speech functionality.

5.4.4)Web scraping:

To provide more relevant information to the user, web scraping can be used to extract data from various websites. Beautiful Soup and Scrapy are two popular Python libraries used for web scraping. In this project, web scraping was used to search for information on the web based on user requests

5.4.5)Dialogue management:

In addition to NLP processing, the virtual assistant also requires dialogue management to handle the flow of the conversation. Dialogue management involves keeping track of the conversation history, identifying the user's intent, generating responses, and handling fallback scenarios.

5.4.6)Speech recognition:

To allow users to interact with the virtual assistant using their voice, speech recognition can be integrated into the backend. Python libraries such as SpeechRecognition and PocketSphinx can be used for this purpose.

5.4.7)Text-to-speech:

To provide responses to the user using audio, text-to-speech functionality can be integrated into the backend. The Google Text-to-Speech API and the pyttsx3 library are two options for this.

5.4.8)Automation:

To make the virtual assistant more efficient, automation can be used. This involves automating tasks such as sending emails, scheduling events, and playing music. In this project, Spotify API was used to play music based on user requests.

5.4.9)Frontend development:

The final step is to develop the frontend of the virtual assistant. This involves creating a user interface that interacts with the backend to provide responses to the user. The frontend can be developed using a web framework such as Flask, Django, or React. The user interface can be designed using HTML, CSS, and JavaScript.

5.4.10)Deployment:

Once the virtual assistant is developed, it needs to be deployed to a server so that it can be accessed by users. Cloud-based hosting services such as AWS, Google Cloud, and Microsoft Azure are popular options for deploying web applications.

5.4.11)Testing and debugging:

Testing and debugging are essential steps in the implementation process to ensure that the virtual assistant is working correctly. Tools such as pytest and PyCharm can be used for testing and debugging.

6) Integration and Testing

6.1) Description of the Integration Modules:

6.1.1)To integrate the 3D AI virtual assistant project modules such that they will work as a one unit, you will first design and model the virtual assistant's appearance in Blender and create animations for specific tasks or responses. The model and animations will be exported in a compatible format for use in the final project. You will then integrate the NLP module using BERT to enable the virtual assistant to understand and respond to natural language commands. This involves processing the user's input using BERT and generating appropriate responses. You can also use web scraping to search through the web for relevant information to provide better responses to the user's queries.

6.1.2)Additionally, you can integrate automation capabilities to enable the virtual assistant to perform tasks automatically, such as sending emails or scheduling appointments. This involves utilizing APIs such as Google Calendar and Voice API to interact with other services and perform actions based on user input.

6.1.3)Finally, you can use APIs such as Google Maps, Google Search, and Spotify to provide additional features and functionality to the virtual assistant. This can include finding directions, playing music, and searching for information related to the user's query. By integrating all of these different modules together, you can create a seamless and interactive 3D AI virtual assistant that can provide a wide range of helpful services to the user.

6.1.4)Another important consideration when integrating the different modules is the user interface. The virtual assistant should be intuitive and easy to use, with clear prompts and feedback to guide the user through each step of the process. This may involve designing a custom interface using a web development framework like Flask or Django, or using existing tools like the Google Assistant API or Amazon Alexa Skills Kit to create a voice interface.

6.1.5)In addition to the core modules discussed earlier, there are a wide range of additional technologies and tools that can be used to enhance the functionality of the virtual assistant. For example, you might use sentiment analysis algorithms to detect the emotional state of the user and adjust the response accordingly, or integrate computer vision tools to enable the virtual assistant to recognize objects and respond appropriately.

6.1.6)Another important consideration is performance and scalability. As the virtual assistant becomes more popular and handles more requests, it will be important to ensure that the underlying infrastructure can handle the increased load. This may involve using cloud-based services like AWS or Google Cloud to scale the application dynamically, or optimizing the code for better performance and faster response times.

6.1.7)As we considered using REST API's for most of our Integration part here is detailed description how we are achieving that:

6.1.7.1) Define endpoints:

You should define endpoints for each module or service in your virtual assistant project that you want to expose via REST API. Each endpoint should correspond to a specific function or feature of your virtual assistant, such as searching the web or playing music. You can define endpoints using a web framework like Flask or Django.

6.1.7.2)Implement endpoints:

Once you have defined the endpoints, you need to implement them in your virtual assistant project. For each endpoint, you should define the corresponding function or method that will handle the request and return the response.

6.1.7.3)Serialize data:

When handling the request, you should convert the data into a JSON format that can be sent over the network. You can use a serialization library like JSON or Pickle to do this.

6.1.7.4)Use HTTP methods:

You should use the appropriate HTTP methods for each endpoint, such as GET, POST, PUT, or DELETE. For example, a search endpoint might use GET to retrieve search results, while a music playback endpoint might use POST to start or stop playback.

6.1.7.5)Add authentication:

To ensure that only authorized users can access your REST API, you should add authentication and authorization to your endpoints. This can be done using tokens, OAuth, or other authentication methods.

6.1.7.6) Deploy the API:

Once you have implemented and tested your endpoints, you can deploy your REST API to a web server or cloud platform. You can use a platform like Heroku, AWS, or Google Cloud Platform to deploy your API and make it available to other users.

In summary, creating a 3D AI virtual assistant requires careful integration of a wide range of different tools and technologies, along with a thoughtful approach to user interface design, performance, and scalability. By taking the time to carefully plan and test each module, and continually iterating and improving on the overall design, you can create a powerful and effective virtual assistant that provides real value to the user.

6.2) Testing

Test Case ID	Description	Input	Expected output	Actual Output	Status
TC01	Verify that the virtual assistant can recognize voice commands.	Voice command	Virtual assistant recognizes voice command	recognizes voice command	Passed
TC02	Verify that the virtual assistant can recognize hand gestures.	Hand gesture	Virtual assistant recognizes hand gesture	recognizes hand gesture	Passed
TC03	Verify that the virtual assistant can recognize facial expressions.	Facial expression	Virtual assistant recognizes facial expression	recognizes facial expression	Passed
TC04	Verify that the virtual assistant can provide accurate responses to user queries.	User query	Virtual assistant provides an accurate response to user query	provides accurate response to user query	Passed
TC05	Verify that the virtual assistant can perform actions based on	User request	Virtual assistant performs action based on user request	performs action based on user request	Passed

	user requests.				
TC06	Verify that the virtual assistant can provide accurate locations using maps.	User Query	Should Provide the accurate locations using maps	provides the accurate locations	Passed
TC07	Verify that the virtual assistant can schedule appointments and set reminders for users.	Schedule appointments and set reminders for users	Virtual assistant schedules appointments and sets reminders for users	Schedules appointments and sets reminders for users	Passed
TC08	Verify that the virtual assistant can provide driving directions and traffic updates in real time.	Driving directions and traffic updates in real-time	Virtual assistant provides driving directions and traffic updates in real-time	provides driving directions and traffic updates in real-time	Passed
TC09	Verify that the virtual assistant can provide accurate information about local businesses and services.	Local businesses and services information	Virtual assistant provides accurate information about local businesses and services	provides accurate information about local businesses and services	Passed
TC10	Verify that the virtual assistant can	Nearby restaurants recommendati	Virtual assistant provides	provides recommendations for nearby	Passed

	provide recommendations for nearby restaurants based on user preferences.	ons based on user preferences	recommendations for nearby restaurants based on user preferences	restaurants based on user preferences	
TC11	Verify that the virtual assistant can provide accurate information about local events and activities.	Local events and activities information .	virtual assistant provides Accurate information about local events and activities	provides Accurate information about local events and activities	Passed

7)Performance Analysis:

Performance analysis of a 3D AI virtual assistant involves evaluating its ability to process user input and provide accurate responses in a timely manner, as well as its overall usability and user experience. Here are some key performance metrics to consider:

7.1) Response Time:

The time it takes for the virtual assistant to respond to user input is critical to the user experience. The response time should be quick and consistent, ideally within a few seconds.

7.2)Accuracy:

The virtual assistant's ability to understand user input and provide accurate responses is essential. The accuracy of the responses should be measured by comparing them to the expected output.

7.3)Natural Language Processing (NLP) capabilities:

NLP is critical to the ability of the virtual assistant to understand user input. The system's NLP capabilities should be evaluated for accuracy and efficiency.

7.4)Multitasking:

The virtual assistant should be able to handle multiple tasks and commands simultaneously without experiencing performance issues or slowdowns.

7.5)User Interface:

The user interface should be intuitive and easy to navigate, with clear and concise instructions and feedback. The design and layout of the interface should be evaluated for ease of use and overall user experience.

7.6)Integration with other systems:

The virtual assistant should be able to integrate seamlessly with other systems and platforms, such as messaging apps, email clients, calendars, and task managers.

7.7)Security:

The virtual assistant should be designed with robust security features to protect user data and prevent unauthorized access.

Overall, a 3D AI virtual assistant's performance can be evaluated by measuring its response time, accuracy, NLP capabilities, multitasking, user interface, integration with other systems, and security features. A high-performing virtual assistant can enhance the user experience and improve productivity.

8)Future Scope:

The future of AI-powered virtual assistants is very promising. According to Gartner, by 2025, 50% of knowledge workers will use a virtual assistant on a daily basis, up from 2% in 2019. Virtual assistants are expected to provide a new level of personalization in an all-digital world.

Virtual assistants using natural language processing (NLP) are here to stay as they become more efficient and cost-saving for companies that deploy them. The potential for such virtual assistants is enormous and many benefits are already being realized.

However, like any form of artificial intelligence (AI), NLP requires thoughtful governance to ensure it’s used for good, and a framework must be developed to optimize human-machine interaction.

9) Applications:

9.1) Healthcare:

AI can automate administrative tasks, such as appointment scheduling and medical coding, freeing up medical professionals to focus on patient care. AI virtual assistants can also provide patients with 24/7 access to medical information and guidance.

9.2) Education:

Chatbots can deliver AI-enabled education across the world.

9.3) Retail:

Virtual assistants can be used to provide customer service and support, answering routine questions and providing product information.

9.4) Tourism:

Virtual assistants can be used for scheduling things like airport shuttles and rental cars.

9.5) Finance:

Virtual assistants can provide customers with account information and assist with transactions.

9.6) Human Resources:

Virtual assistants can assist with tasks such as scheduling interviews and answering routine HR questions.

9.7) Customer Service:

Virtual assistants can provide 24/7 customer support, answering routine questions and providing information.

9.8) Marketing:

Virtual assistants can assist with tasks such as lead generation and customer engagement.

9.9) Sales:

Virtual assistants can assist with tasks such as lead qualification and appointment scheduling.

9.10) Personal Assistance:

Virtual assistants can assist individuals with tasks such as scheduling appointments and reminders**.**

13) Plagiarism Report:

Similarity Index

13%

test

By: test

As of: Jun 10, 2023 3:01:05 PM
14,177 words - 128 matches - 73 sources

sources:

303 words / 2% - Internet from 16-Jun-2022 12:00AM

www.coursehero.com

150 words / 1% - Internet from 10-Mar-2022 12:00AM

www.weforum.org

139 words / 1% - Internet from 16-Sep-2020 12:00AM

pinscreen.com

106 words / 1% - Internet from 10-Nov-2022 12:00AM

www.unite.ai

79 words / 1% - Internet from 16-Oct-2022 12:00AM

repo.darmajaya.ac.id

32 words / < 1% match - from 04-Apr-2023 12:00AM

WWW.coursehero.com

22 words / < 1% match - Internet from 28-Feb-2021 12:00AM

www.coursehero.com

17 words / < 1% match - Internet from 31-Jan-2023 12:00AM

www.coursehero.com

10 words / < 1% match - Internet from 01-Jun-2022 12:00AM

www.coursehero.com

74 words / < 1% match - Internet from 22-Feb-2023 12:00AM

www.researchgate.net

21 words / < 1% match - Internet from 07-Feb-2023 12:00AM

www.researchgate.net

64 words / < 1% match - Internet from 10-Jan-2023 12:00AM

morioh.com

52 words / < 1% match - Internet from 09-Nov-2022 12:00AM

saxon.ai

19 words / < 1% match - Internet from 26-Jan-2023 12:00AM

www.slideshare.net

16 words / < 1% match - Internet from 05-Apr-2020 12:00AM

www.slideshare.net

14 words / < 1% match - Internet from 21-Aug-2022 12:00AM

www.slideshare.net

44 words / < 1% match - Internet from 22-Dec-2022 12:00AM

blog.actorsfit.com

41 words / < 1% match - Internet from 22-Aug-2022 12:00AM

mobidev.biz

38 words / < 1% match - Internet

Abdul, Raqeeb. "Enhancing Cloud Security by a Series of Mobile Applications That Provide Timely and Process Level Intervention of Real-Time Attacks", The Repository at St. Cloud State, 2017

37 words / < 1% match - Internet from 12-Jan-2021 12:00AM

findanyanswer.com

31 words / < 1% match - from 25-Apr-2023 12:00AM

Www.zdnet.com

31 words / < 1% match - from 13-Apr-2023 12:00AM

yellow.systems

20 words / < 1% match - from 29-May-2023 12:00AM

www.termpaperwarehouse.com

9 words / < 1% match - Internet from 08-Dec-2022 12:00AM

www.termpaperwarehouse.com

25 words / < 1% match - Internet

Nedim Dedić, Clare Stanier. "An Evaluation of the Challenges of Multilingualism in Data Warehouse Development", 'Scitepress', 2016

25 words / < 1% match - Crossref

Sangeetha Manoharan, Parth Natu. "Development of a framework for a collaborative and personalised voice assistant", Electronic Government, an International Journal, 2021

25 words / < 1% match - Internet from 19-Dec-2021 12:00AM

www.igi-global.com

24 words / < 1% match - Crossref

Nigel Hall. "Colorectal cancer: features and investigation", Medicine, 2007

23 words / < 1% match - Internet from 01-Nov-2022 12:00AM

site.ieee.org

22 words / < 1% match - Crossref

"New Trends in Computational Vision and Bio-inspired Computing", Springer Science and Business Media LLC, 2020

20 words / < 1% match - Internet from 04-Nov-2022 12:00AM

marinemanagement.org

20 words / < 1% match - Internet from 22-Mar-2018 12:00AM

www.scribd.com

19 words / < 1% match - Internet from 17-Jun-2019 12:00AM

dktes.com

17 words / < 1% match - Internet from 07-Nov-2022 12:00AM

files.core.ac.uk

16 words / < 1% match - Crossref

Bharti Rana, Yashwant Singh, Pradeep Kumar Singh, Kayhan Zrar Ghafoor, Sachin Shrestha. "A hybrid approach for energy-efficient routing in IoT using duty cycling and improved ant colony", IET Communications, 2022

16 words / < 1% match - Internet from 29-Jul-2022 12:00AM

gitlab.sliit.lk

15 words / < 1% match - Crossref

"Data Management, Analytics and Innovation", Springer Science and Business Media LLC, 2023

15 words / < 1% match - Internet from 05-Jan-2023 12:00AM

www.hausarbeiten.de

14 words / < 1% match - Internet from 01-Nov-2022 12:00AM

jozilla.net

14 words / < 1% match - Internet from 10-Jan-2023 12:00AM

www.ibm.com

14 words / < 1% match - Internet from 05-Aug-2021 12:00AM

www.ijrte.org

13 words / < 1% match - Crossref

Bintu Harwani. "jQuery Recipes", Springer Science and Business Media LLC, 2022

13 words / < 1% match - from 16-May-2023 12:00AM

blog.enterprisedna.co

13 words / < 1% match - Internet from 17-Jul-2019 12:00AM

mbessaywrite.blogspot.com

13 words / < 1% match - Internet

NANDA SYAHPUTRA, -. "PENERAPAN ALGORITMA BACKPROPAGATION UNTUK MEMPREDIKSI JUMLAH PENUMPANG BUS TRANS METRO PEKANBARU", 2021

12 words / < 1% match - Crossref

K. Murugesh Babu, M. Selvadass, R. Somashekar. "Characterization of the conventional and organic cotton fibres", Journal of the Textile Institute, 2013

12 words / < 1% match - Crossref

Thamali Madhushani Adhikari, Yan Wu. "Classifying Software Vulnerabilities by Using the Bugs Framework", 2020 8th International Symposium on Digital Forensics and Security (ISDFS), 2020

11 words / < 1% match - Internet from 13-Feb-2020 12:00AM

blog.logikonlabs.com

11 words / < 1% match - Internet from 18-Mar-2003 12:00AM

quicken.buyerzone.com

11 words / < 1% match - Internet from 11-Feb-2023 12:00AM

wanivipin.files.wordpress.com

10 words / < 1% match - Crossref

Jia-Wei Chang. "Enabling progressive system integration for AIoT and speech-based HCI through semantic-aware computing", The Journal of Supercomputing, 2021

10 words / < 1% match - Internet from 12-Jun-2021 12:00AM

askinglot.com

10 words / < 1% match - Internet

Syahputra, Eswin, Nursukmi, Irpan, Putra, Sony, Sani, Bayu Sukma, Wijaya, Rian Farta. "EYE ASPECT RATIO ADJUSTMENT DETECTION FOR STRONG BLINKING SLEEPINESS BASED ON FACIAL LANDMARKS WITH EYE-BLINK DATASET", 'UIN Sumatera Utara Medan', 2023

10 words / < 1% match - Internet from 13-Apr-2022 12:00AM

link.springer.com

10 words / < 1% match - from 06-Apr-2023 12:00AM

www.cigniti.com

10 words / < 1% match - Internet from 04-Feb-2023 12:00AM

www.electronicwings.com

10 words / < 1% match - from 06-Jun-2023 12:00AM

www.frontiersin.org

9 words / < 1% match - Crossref

Learn Android App Development, 2013.

9 words / < 1% match - Internet from 26-Dec-2022 12:00AM

dokumen.tips

9 words / < 1% match - Internet

Permana, Roby. "RANCANG BANGUN APLIKASI E-COMMERCE JAMBU AIR CAMPLONG PRODUK UNGGULAN KABUPATEN SAMPANG KECAMATAN CAMPLONG MADURA (STUDI KASUS DI KECAMATAN CAMPLONG MADURA)", 2020

9 words / < 1% match - from 30-Mar-2023 12:00AM

eprints.utar.edu.my

9 words / < 1% match - Internet from 31-Dec-2009 12:00AM

tu-dresden.de

9 words / < 1% match - Internet from 17-Oct-2010 12:00AM

www.nitk.ac.in

8 words / < 1% match - Crossref

"Computer Vision – ECCV 2020 Workshops", Springer Science and Business Media LLC, 2020

8 words / < 1% match - Crossref

Lecture Notes in Computer Science, 2004.

8 words / < 1% match - Crossref

Ronakkumar Patel. "How to talk to a hologram", Proceedings of the 11th international conference on Intelligent user interfaces - IUI 06 IUI 06, 2006

8 words / < 1% match - ProQuest

Teli, Mohammad Nayeem. "Face detection using correlation filters.", Proquest, 2014.

8 words / < 1% match - Crossref

Vladimir A. Kulyukin. "On natural language dialogue with assistive robots", Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction - HRI 06 HRI 06, 2006

8 words / < 1% match - Internet from 12-May-2015 12:00AM

biz101websites.org

8 words / < 1% match - from 06-May-2023 12:00AM

d-nb.info

8 words / < 1% match - Internet from 09-Feb-2014 12:00AM

oraserv.cs.siena.edu

7 words / < 1% match - Crossref

"Knowledge Management and Acquisition for Smart Systems and Services", Springer Science and Business Media LLC, 2010

6 words / < 1% match - Crossref

"Tourism Management, Marketing, and Development", Springer Science and Business Media LLC, 2014

14) Ethics:

14.1) Transparency:

Users should be able to understand how the technology works and how their data is being used. This includes providing clear and concise information about the virtual assistant’s capabilities and limitations.

14.2) Accountability:

There should be clear lines of accountability for the actions of the virtual assistant and any potential harm it may cause. This includes having mechanisms in place to address any issues or concerns that may arise.

14.3) Privacy:

The virtual assistant should respect users’ privacy and protect their personal data. This includes implementing strong security measures to prevent unauthorized access to user data.

14.4) Fairness:

The virtual assistant should be designed to prevent bias and discrimination. This includes ensuring that the algorithms used by the virtual assistant are fair and unbiased.

14.5) Consent:

Users should be able to provide informed consent for the use of their data by the virtual assistant. This includes providing clear and concise information about how user data will be used and allowing users to opt-out if they choose.

14.6) Safety:

The virtual assistant should be designed to ensure the safety of its users. This includes implementing measures to prevent harm or injury to users.

14.7) Social Responsibility:

The virtual assistant should be designed with social responsibility in mind. This includes considering the potential impact of the technology on society and taking steps to minimize any negative effects

15) References:

15. 1) Joseph Weizenbaum. ELIZA—a computer program for the study of natural lan-guage communication between man and machine. Communications of the ACM,9(1):36–45, jan 1966.

15.2) Brian Kingsbury Xiaodong Cui, Vaibhava Goel. Data Augmentation for Deep Neu-ral Network Acoustic Modeling. 2014 IEEE International Conference on Acoustic,Speech and Signal Processing (ICASSP), pages 5619–5623, 2014.

15.3) Chatbots In Education: Applications Of Chatbot Technologies - eLearning Indus-try.

15.4) Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, KorayKavukcuoglu, and Pavel Kuksa. Natural Language Processing (almost) from scratch. mar 2011.

15.5) Debasatwa Dutta. Developing an Intelligent Chat-bot Tool to assist high school students for learning general knowledge subjects. Technical report

15.6) Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (almost) from Scratch. mar 2011

15.7) Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, KorayKavukcuoglu, and Pavel Kuksa. Natural Language Processing (almost) from scratch. Mar 2011

15.8) Luciana Benotti, Mar´ıa Cecilia Mart´ınez, and Fernando Schapachnik. Engaging high school students using Chatbots. In ITICSE 2014 - Proceedings of the 2014 Innovation and Technology in Computer Science Education Conference, pages 63–68. Association for Computing Machinery, 201