HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

**SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING**



**PROJECT REPORT**

**VIRTUAL COMMENTATOR ON SOCIAL NETWORK VIDEOS**

Name of students: Class code: 152564 *–* Team 11

Nguyen Thi Mai Chinh – 20233834

Luu Tri Dung – 20233842

Nguyen Quang Tung – 20233884

Hoang Tieu Yen – 20233888

Name of instructors: Dr. Pham Van Tien

Assoc. Prof. Tran Thi Thanh Hai

Hanoi, 12/2024

Table of Contents

[LIST OF FIGURES iv](#_Toc186741822)

[LIST OF TABLES v](#_Toc186741823)

[ACKNOWLEDGEMENT vi](#_Toc186741824)

[ABSTRACT vii](#_Toc186741825)

[1. Introduction 1](#_Toc186741826)

[1.1. Motivation 1](#_Toc186741827)

[1.2. Objectives 1](#_Toc186741828)

[1.2.1. Main objectives 1](#_Toc186741829)

[1.2.2. Specific objectives 1](#_Toc186741830)

[2. Methodology 2](#_Toc186741831)

[2.1. State of the art 2](#_Toc186741832)

[2.1.1. Methods for virtual commentator 2](#_Toc186741833)

[2.1.2. Existing products for virtual commentator 2](#_Toc186741834)

[2.1.3. Dicussions 4](#_Toc186741835)

[2.2. Application of the 9 steps in engineering design process 4](#_Toc186741836)

[3. Project implementation 4](#_Toc186741837)

[3.1. Step 1: User requirement 4](#_Toc186741838)

[3.2. Step 2: Specifications 5](#_Toc186741839)

[3.2.1. Functionality 5](#_Toc186741840)

[3.2.2. Non functinality 5](#_Toc186741841)

[3.3. Step 3: Planning 6](#_Toc186741842)

[3.4. Step 4: Block Design 7](#_Toc186741843)

[3.5. Step 5: Detail block design 8](#_Toc186741844)

[3.6. Step 6: Best altenatives selection 9](#_Toc186741845)

[3.7. Step 7: Protyping 11](#_Toc186741846)

[3.8. Step 8: Testing 12](#_Toc186741847)

[4. Discussions 12](#_Toc186741848)

[5. Conclusions and future works 13](#_Toc186741849)

[6. Reference 14](#_Toc186741850)

# LIST OF FIGURES

[Figure 1.1: objective of the project 1](#_Toc187679563)

[Figure 2.1: 9 steps in design process 4](#_Toc187679564)

[Figure 3.1: Planning of the project 6](#_Toc187679565)

[Figure 3.2: Main blocks of our proposed system 8](#_Toc187679566)

[Figure 3.3: Upload and process video 9](#_Toc187679567)

[Figure 3.4: Generate output content 9](#_Toc187679568)

# LIST OF TABLES

[Table 2.1: Comparisionof existing products 3](#_Toc186741861)

[Table 3.1: Summary of the main functions of our proposed system / prodcut 6](#_Toc186741862)

[Table 3.2: Summary of task distrubution and completeness 7](#_Toc186741863)

[Table 3.3: Comparison of program language 9](#_Toc186741864)

[Table 3.4: Comparison of AI models 10](#_Toc186741865)

[Table 3.5: Comparision of the interface models 10](#_Toc186741866)

# ACKNOWLEDGEMENT

On the market today, advertising information has immense potential and a wide range of applications. The effective use of innovative techniques is crucial in this dynamic field.

We are deeply grateful to tutor Pham Van Tien and tutor Tran Thi Thanh Hai for their invaluable support throughout our project. They provided our team with the necessary resources and guidance, enabling us to achieve a successful outcome.

While we dedicated significant effort to this project, we acknowledge the limitations of our experience and the inherent challenges of venturing into a new domain. We understand that design errors in content and presentation may have occurred. We eagerly anticipate Dr. Pham Van Tien's feedback and welcome any suggestions for project improvement.

Finally, we would like to express our sincere gratitude to every member of our team for their hard work, dedication, and invaluable contributions to the success of this project.

# ABSTRACT

As the rapid growth of online video content, users often have questions about specific scenes or concepts while watching videos on platforms like YouTube. However, pausing to search for answers interrupts the experience and can be frustrating. To address this, a virtual commentator **chatbot** can be integrated into video platforms, allowing users to ask questions in real time without leaving the video. The chatbot provides instant responses about the content, offering clarifications, background information, or deeper analysis as needed. This solution transforms passive video watching into an interactive experience, helping users gain deeper insights and retain knowledge. By enhancing engagement and convenience, the virtual commentator revolutionizes how users interact with online video content.

# Introduction

## Motivation

The main reason for conducting this project lies in addressing the limitations of current online video platforms. While platforms like YouTube provide an abundance of content, they lack features that allow users to interact with the video in real time to clarify doubts or gain additional insights. Viewers often have to pause videos and search externally for explanations, which disrupts the flow and diminishes the overall experience. This limitation is especially noticeable in educational or complex videos, where understanding specific details is crucial. Recognizing this gap, the motivation for this project is to propose a solution that seamlessly integrates a virtual commentator chatbot. By enabling real-time interaction and instant responses, this solution aims to enhance engagement, provide immediate clarity, and create a smoother and more enriching video-watching experience for users.

## Objectives

### Main objectives

Respond quickly to video while watching

Main objective

Improve user experience

Enhance content understanding

Enable real-time interaction

Figure .: objective of the project

### Specific objectives

* **Enable real-time interaction:** Develop a chatbot that allows users to ask questions and receive instant answers without pausing or leaving the video.
* **Enhance content understanding:** Provide clarifications, background information, and detailed insights to help viewers better comprehend the video content.
* **Improve user experience:** Create a seamless and engaging video-watching process by integrating the chatbot directly into video platforms.

# Methodology

## State of the art

### Methods for virtual commentator

 "LiveBot: Generating Live Video Comments Based on Visual and Textual Contexts" [1]

 "Sentiment-oriented Transformer-based Variational Autoencoder Network for Live Video Commenting" [2]

 "Enhancing Viewing Experience with AI-Driven Commentary" [3]

 "YouTube Tests Generative AI Chatbot and Comment Summaries" [4]

### Existing products for virtual commentator

Table .: Comparisionof existing products

|  |  |  |  |
| --- | --- | --- | --- |
| Product | Main features | Advantages | Drawbacks |
| Murf | · AI-driven text-to-speech technology  · Supports multiple languages  · Voice cloning and background music integration. | · High-quality, customizable voiceovers  · Diverse voice options.  · Easy to use, even for beginners. | · Limited free features; required for advanced options.  · Not suited for live commentary; designed primarily for pre-recorded content. |
| Pictory | · AI-powered video creation and text-to-video capabilities.  · Shortens long-form videos  · Automatic captioning and summarization.  · Integration of voiceovers into videos. | · Simplifies video editing and content repurposing.  · Time-efficient for social media managers and marketers.  · Enhances accessibility with captions. | · Limited customization for voiceovers compared to Murf.  · Primarily focused on video editing, with commentary as a secondary feature. |
| vMix Social | · Social media integration for live broadcasts.  · Displays live comments and interactions.  · Works as an add-on for the vMix live production software. | · Enhances viewer engagement.  · Easy to incorporate real-time audience feedback.  · Complements professional-quality live productions. | · Doesn't generate virtual commentary; depends on user-provided input.  · Requires familiarity with vMix and live production setups. |

### Dicussions

Existing products allow users to engage in video-inspired discussions with AI on a broader range of topics, fostering a more dynamic and immersive environment. This innovation marks a significant step forward in creating more interactive, user-centric, and effective video platforms for a wider audience. However, these applications still face certain limitations, including high subscription fees and limited functionality when using less popular or complex programming languages. Addressing these challenges is essential to ensuring accessibility and inclusivity for a wider audience.

## Application of the 9 steps in engineering design process

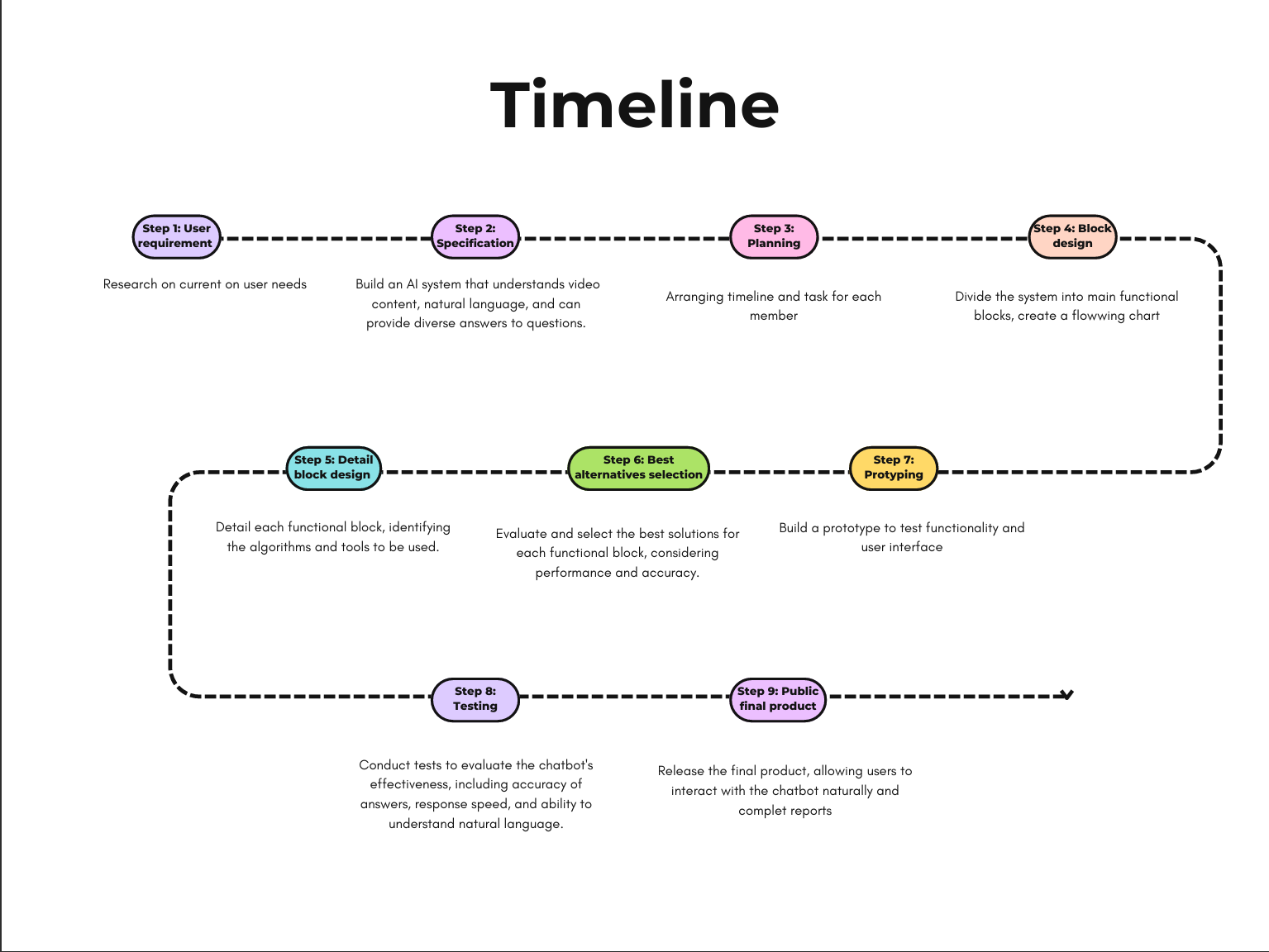


Figure .: 9 steps in design process

# Project implementation

## Step 1: User requirement

We launch a survey to investigate the demands of having an assistant while watch a video which helps them understand the content of video[[1]](#footnote-1). We observe that 30% of the users get misunderstood while watching a video. Moreover, 60% of the users needs an assistant to help them understand the video content.

User requirement: The user wants a chatbot that can answer any question related to the video they are watching.

## Step 2: Specifications

### Functionality

* Function 1: Real-Time Q&A Interaction

Allow users to ask questions and receive instant answers while watching videos without pausing playback.

* Function 2: Contextual Video Analysis

Provide answers based on video metadata, subtitles, or scene content for accurate and relevant responses.

* Function 3: Recommendations and Insights

Suggest related videos, topics, or external resources to enhance the user’s understanding and engagement.

### Non functinality

* Non- Function 1: Scalability

The system should handle a large number of simultaneous users without performance degradation

* Non- Function 2: User Privacy

Ensure that user data and interactions are securely handled and comply with data privacy regulations.

Table .: Summary of the main functions of our proposed system / prodcut

|  |  |  |
| --- | --- | --- |
| Function | Description of the function | Priority (Requỉed / Optional) |
| Real-Time Q&A Interaction | Allows users to ask questions and get instant answers without disrupting the video experience. | Required |
| Contextual Video Analysis | Uses metadata, subtitles, or scene analysis to generate accurate responses. | Required |
| Recommendations and Insights | Provides suggestions for related videos or additional learning materials. | Optional |

## Step 3: Planning

Put the figure showing the tasks, subtasks, the person in chages, the timelines

A table with numbers and a few days

Description automatically generated

Figure .: Planning of the project

Table .: Summary of task distrubution and completeness

|  |  |  |
| --- | --- | --- |
| Member | Tasks | Completeness |
| Nguyen Thi Mai Chinh | - Survey users  - Allow the system to receive more format of inputs  - Complete report | 90% |
| Luu Tri Dung | - Survey non-function  - Research for efficient alternatives selections  - Complete report | 90% |
| Nguyen Quang Tung | - Survey technology  - Add functions, fix errors and test products  - Complete report | 90% |
| Hoang Tieu Yen | - Interface(design)  - Page setup and interface(code)  - Complete report | 90% |

## Step 4: Block Design

* Video uploading block: This block receives video uploaded by users then store them on the system. This block also displays the video directly on the interface.
* Video processing block: This block uses the API to upload the video to Google Generative AI model, where such data type as video can be processed.
* Video content generating block: We design a text input box where system could acquire prompt from users. Then we send it as prompt to AI model and receive text-based answer.
* Output block: Finally this block would display the answer texts on the interface. Then, users could still watching and asking more questions.

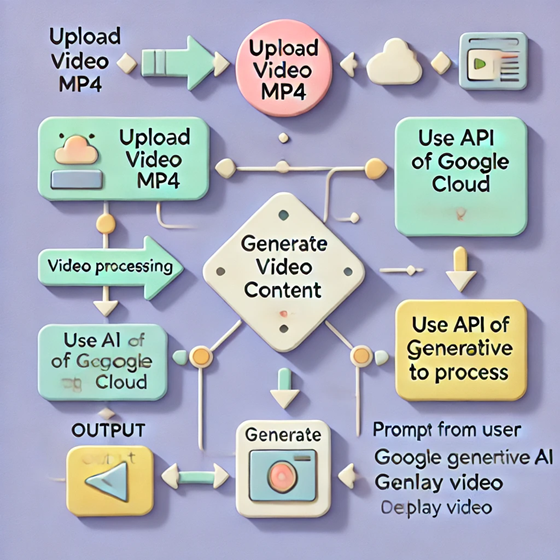


Figure .: Main blocks of our proposed system

## Step 5: Detail block design

* Upload video: We allow users to upload different format of video file such as MP4, MOV, AVI,… To handle this uploaded video file, the binary content is written to a temporary file using Python’s tempfile.NamedTemporaryFile() function. This ensures the video is safely stored on the system in a temporary location. The delete=False argument keeps the file on disk until the program explicitly removes it. Then the video is displayed in the Streamlit interface.
* Video processing: Use the API of Google Generative AI to upload the video to the AI system. The AI model here is free to upload such data type as video.

A diagram of a company

Description automatically generated

Figure .: Upload and process video

* Generate video content: The system provides an input field where users can enter prompts or questions related to the video content. Once the user submits a prompt, the Google Generative AI model is instantiated (e.g., using genai.GenerativeModel()). The generate\_content() method is used to pass both the uploaded video and the user's prompt to the AI. The model processes these inputs and generates text-based content in response.
* Finally, content is displayed on the app as plain texts by function st.write.

A diagram of a company

Description automatically generated

Figure .: Generate output content

## Step 6: Best altenatives selection

Table .: Comparison of program language

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Criteria 1  Complexity | Criteria 2  Reliability | Criteria 3  Appearance |
| Python | Easy to learn and use | Auto memory management | Lower performance, suit for data processing |
| Java | Complex but easy to learn | Auto memory management | High performance, suit for web |
| C++ | Hard to learn | Manual memory management | Highest performance |

Table .: Comparison of AI models

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Criteria 1  Price | Criteria 2  Reliability | Criteria 3  Appearance |
| Gemini | Free | High but still lack of comments | Good at processing multimedia |
| ChatGPT | Cost | High | Creativity, process complex problem |
| Azure | Cost | High, match with companies | Same as ChatGPT |

Table .: Comparision of the interface models

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Criteria 1  Price | Criteria 2  Reliability | Criteria 3  Appearance |
| Streamlit | Free | Good at processing | Easy to design, to use |
| Tkinter | Free | Old | Too basic |
| Dash | Free | Complex web | Modern, complex to use |

## Step 7: Protyping

## **Objective:** Test the system’s functionality by simulating key processes before developing the prototype.

## **Process:**

# ****3.7.1. Simulation Tools and Data:****

* **Tools:** Python, Streamlit, Google Generative AI.
* **Input:** Video files in various formats (e.g., MP4, AVI, MOV) with diverse content and user-generated prompts.

# ****3.7.2. Steps:****

1. **Video Upload and Display:**
   * Use Streamlit to enable users to upload video files.
   * Display the uploaded video within the web interface for seamless playback.
2. **Prompt Input and Processing:**
   * Integrate a chatbox using Streamlit for users to input prompts or queries related to the video.
   * Pass the uploaded video and prompt to Google Generative AI for processing.
3. **AI-Driven Content Generation:**
   * Generate contextual textual insights based on the video content and the user’s input.
4. **Output Presentation:**
   * Display the AI-generated results in real-time within the chatbox.

**3.7.3. Results:**

* Smooth video playback on the interface for various file formats.
* Interactive chatbox functionality with real-time input and response.
* Contextual and accurate textual results generated by AI.
* Enhanced user experience through seamless integration of video analysis and prompt-based AI responses.

**3.7.4. Challenges and Improvements:**

* **Challenge:** Videos with low resolution or unclear scenes affected AI accuracy.
* **Improvement:** Implement video preprocessing techniques to enhance video quality and clarity.
* **Challenge:** High latency in generating real-time AI responses for longer videos.
* **Improvement:** Optimize video and prompt processing pipelines for faster response times.

## Step 8: Testing

* We suppose that some of the main factors affecting the product are the hardware of the devices, the environment condition, and the input video.
* We tested for some types of input video including the large memory size one, the bad quality(break, noise) with the normal one.
* The videos being tested have the same content, and they would be processed under the same conditions. We consider a short football clip with commentator as the base video, a larger clip whose size is three times larger than the base one as the example 1, and a fancam clip at the same scenario with a lot of noise as the example 2.
* After testing for ten times, we observed that the example 1 required three times longer time to be processed than the base one. Besides that, the example 2 generated less accurate results, which is 40% different to the rest.
* We then conclude that the size of the input would affect the time that the system requires to process while the quality of audio and frames would affect the accuracy of the results.

# Discussions

Compared to the initial goal, the project is still quite basic. So far, we've only managed to create a simple website that helps users directly answer questions while watching videos on it. The interface is rather simple and not visually appealing enough to attract viewers. We haven’t yet integrated many functions into the website. However, getting the code to run and the website to function took quite a bit of time for debugging and exploration. Although the progress is slightly behind the original expectations, the team has remained united, made great efforts, and completed a functional product that can help users answer questions and provide additional information while watching videos directly.

# Conclusions and future works

Implementing a chatbot AI that allows users to ask questions directly while watching a video is a promising endeavor, but it comes with several challenges. One major limitation is the AI's capacity to understand natural language, especially in complex contexts or when dealing with nuances like slang or sarcasm. Additionally, the quality and diversity of the training data significantly impact the AI's ability to provide accurate and relevant responses. Moreover, the AI's capacity for reasoning and inference is often limited, making it difficult to answer questions that require deep understanding or complex problem-solving. Furthermore, ensuring data privacy and security is crucial, as user queries may contain sensitive information. Finally, the development and maintenance of such a system can be costly and time-consuming, requiring continuous updates to keep up with evolving technologies and user needs.

The project has yielded positive results with the creation of a near-complete product that enables users to interact with videos in a more intuitive and intelligent manner. The chatbot's response time has been significantly reduced, providing a seamless user experience. Looking ahead, we plan to continue improving the product by expanding the range of questions the chatbot can answer and enhancing the AI's natural language understanding capabilities.

# Reference

[1] [Shuming Ma, Lei Cui, Damai Dai, Furu Wei, Xu Sun, LiveBot: Generating Live Video Comments Based on Visual and Textual Contexts, 2018](https://arxiv.org/abs/1809.04938?utm_source=chatgpt.com)

[2] [Fengyi Fu, Shancheng Fang, Weidong Chen, Zhendong Mao, Sentiment-oriented Transformer-based Variational Autoencoder Network for Live Video Commenting, 2024](https://arxiv.org/abs/2404.12782?utm_source=chatgpt.com)

[3] [University of Toronto’s Students, Enhancing Viewing Experience with AI-Driven Commentary: A Chatbot Companion for Movies and Videos](https://usra.cs.toronto.edu/public/project/567?opportunity_id=15&sort_by=0&utm_source=chatgpt.com)

[4] [Eric Hal Schwartz, YouTube Tests Generative AI Chatbot and Comment Summaries, 2023](https://voicebot.ai/2023/11/07/youtube-tests-generative-ai-chatbot-and-comment-summaries/)

1. User [survey](https://husteduvn-my.sharepoint.com/:x:/g/personal/tung_nq233884_sis_hust_edu_vn/ETZEzjqdJ4VFq3SoXfr2_QMB6IfJbv3rsew6_rbEZZXHVg?e=UE4YTA) [↑](#footnote-ref-1)