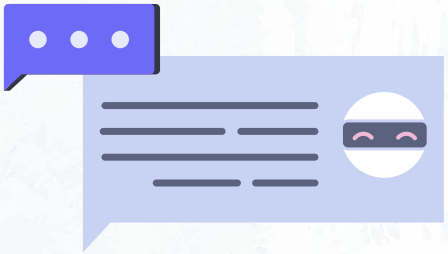
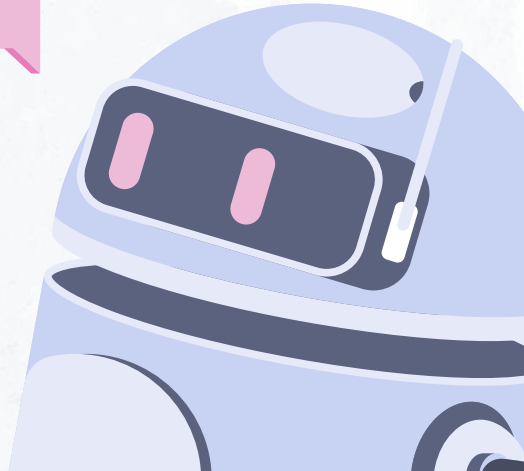
# DATA 515
# Final Presentation

## Team TL;DW

Joobee Jung
Mihir Gathani
Anurag Agarwal
Trisha Prasant





# Agenda

01 Background

02 Data Used

03 Use Cases

04 Design

05 Demo

06 Lessons Learned and Future Work

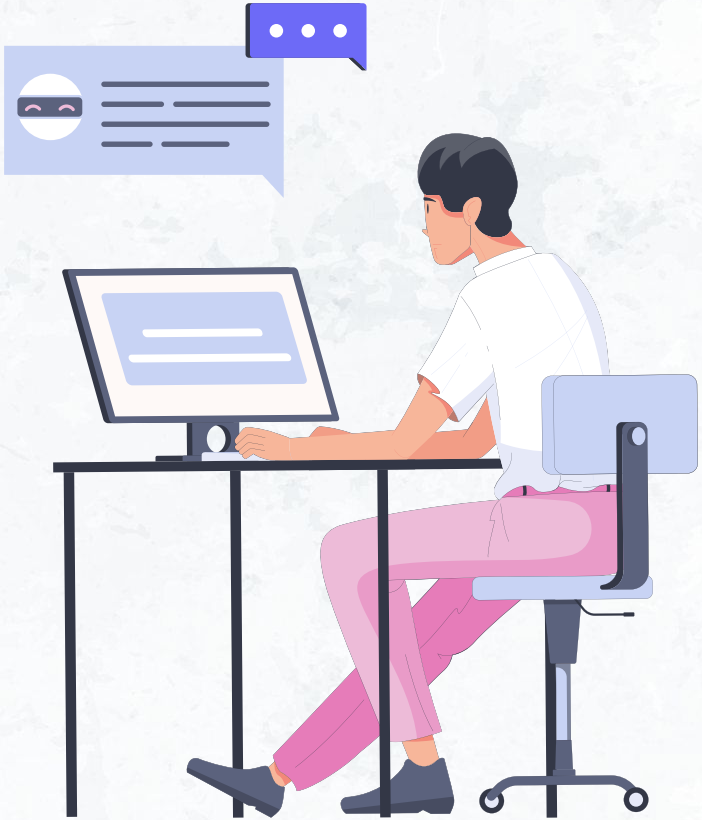
01

# Background

# Background

Users face **challenges** in navigating the content landscape of Youtube for learning purposes. With so many videos available on a wide range of topics, it's **overwhelming** to find the most relevant and high-quality content.

It's time consuming to sift through videos and identify those that would provide us with the **insights** and **knowledge** we are seeking.



# Too Long; Don’t Watch (TL;DW)

TL;DW is an innovative tool designed to address these challenges by summarizing YouTube videos and recommending supplementary TED Talks and podcasts

**Users no longer have to ...**

- spend hours watching lengthy videos to extract key information
- feel overwhelmed by the sheer volume of content available on YouTube.
- rely solely on their own judgment or intuition when selecting what videos to watch.

**Previous work?**


02 →

# Data Used

# Data Used

**Youtube Transcript API**

- To obtain transcripts of YouTube videos based on their links. This API provides access to textual content spoken in videos.

**Gemini API**

- To create summaries from inputted YouTube video links. The Gemini API allows us to easily analyze video content and generate concise summaries automatically.

**Ted Talks Transcript**

- Contains transcripts of TED Talks. Serves as a supplementary data source for recommending TED Talks related to the content of YouTube videos.

**Podcast Transcript Dataset**

- Contains transcripts of podcasts. Provides additional content for recommending podcasts related to topics covered in YouTube videos.

# Data Used - Limitations

- **Transcript Accuracy:** Variability in transcript accuracy from the YouTube Transcript API due to factors like language(en, en-US only), background noise, accents, and speech recognition errors.
- **Limited Content Coverage:** The availability of transcripts and datasets may be restricted to specific topics or speakers, potentially reducing the breadth and diversity of recommended content.
- **API Restrictions:** External APIs such as the YouTube Transcript API and Gemini API are subject to rate limits or restrictions, impacting the application's reliability.
- **Data Bias:** Ted Talks Transcript Dataset and Podcast Transcript Dataset may exhibit biases due to speaker selection, topics, or sources, influencing the recommendations provided.

# Use Cases

## (a) Quick Summary Access

Provides users access to a succinct summary of their inputted video along with relevant keywords they can search for to learn more.

## (b) Obtain Recommendations

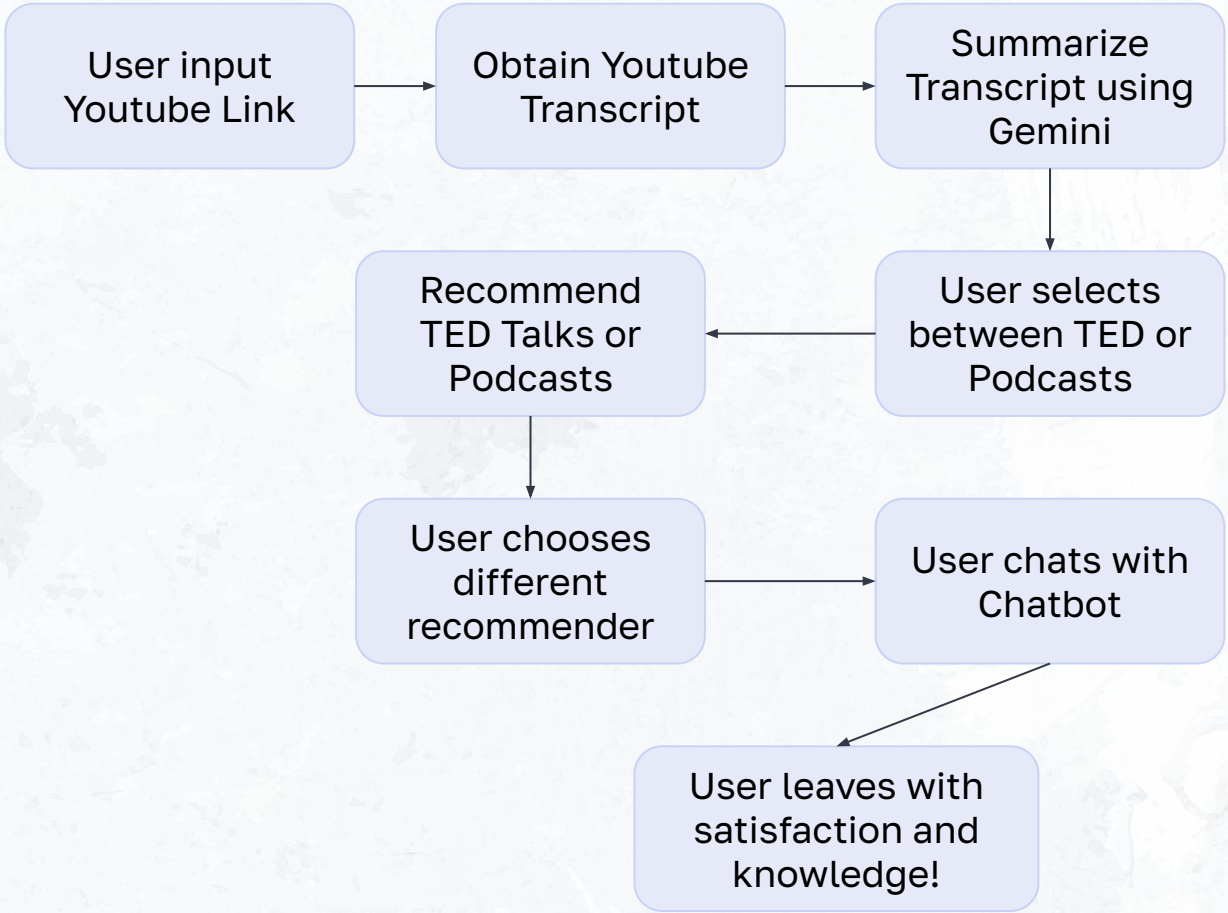
Enables users to get recommendations on relevant TED Talks and/or podcasts for their chosen video using 3 different recommendation engines trained on our datasets.

## (c) Expanded Learning Outcomes

Powered by Google’s Gemini, our application utilizes a chatbot to foster interactive learning, provide personalized assistance and encourage continuous engagement within the website.

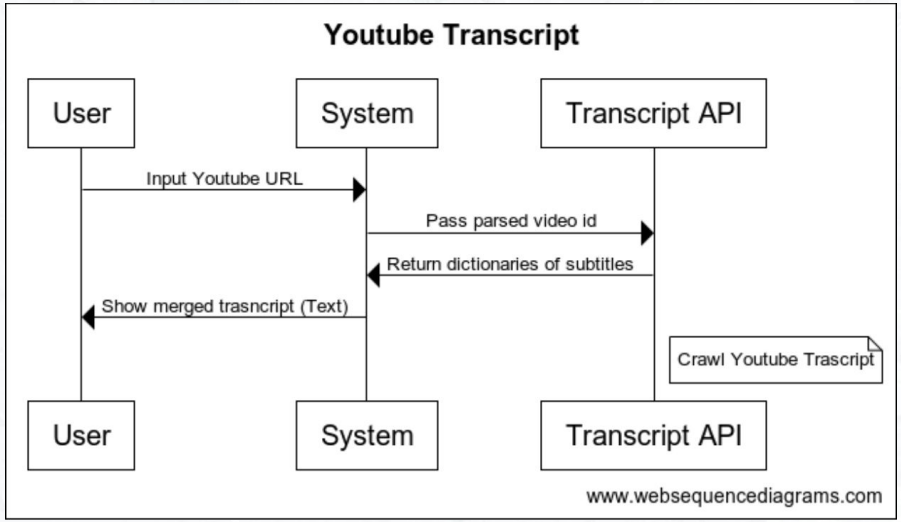
04

# Design


User input Youtube Link
Obtain Youtube Transcript
Summarize Transcript using Gemini
User selects between TED or Podcasts
Recommend TED Talks or Podcasts
User chooses different recommender
User chats with Chatbot
User leaves with satisfaction and knowledge!

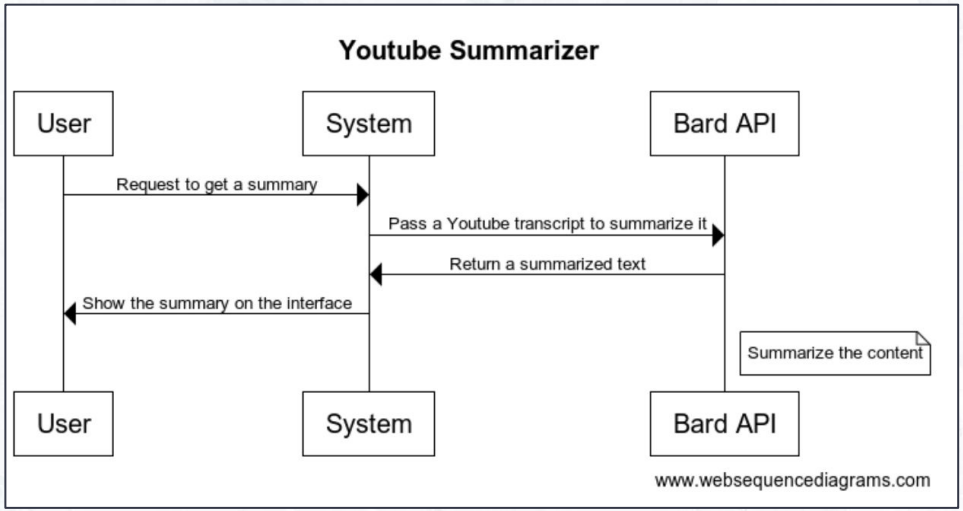
# Design - Youtube Transcript Component

- **What it does**: Getting a YouTube URL to crawl the transcript from users, crawls the transcript, and displays the result
- **Input**: Youtube Video URL
- **Output**: Crawled YouTube Transcript
- **Assumptions**: The URL is valid, and the video has subtitles.
- **How it uses other components**: The output will be used for a summarization component through Gemini API.



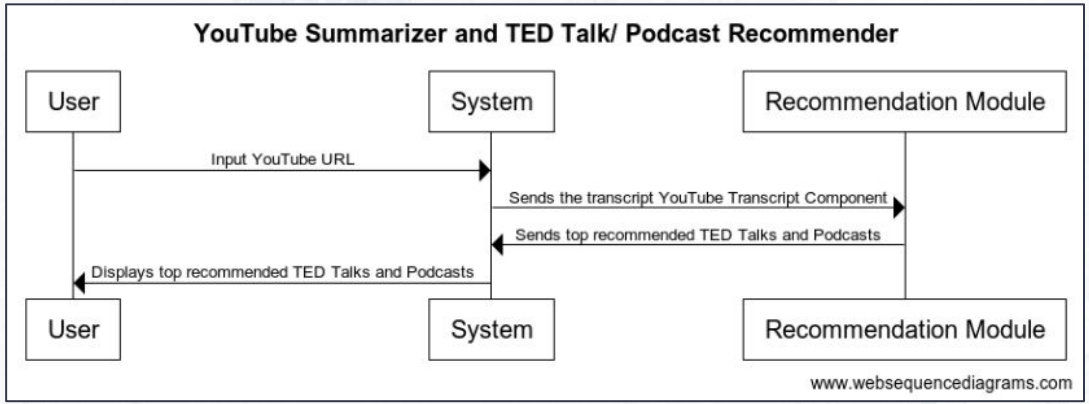
# Design - Summarization Component

- **What it does**: Creates a summary of a YouTube video Transcript
- **Inputs**: Youtube Video Transcript
- **Outputs**: Summary of video
- **Interactions**: Uses the Youtube Transcript Component to obtain YouTube video Transcript to summarize
- **Extra**: Our chatbot uses a similar design to communicate with the Gemini (prev. Bard) API

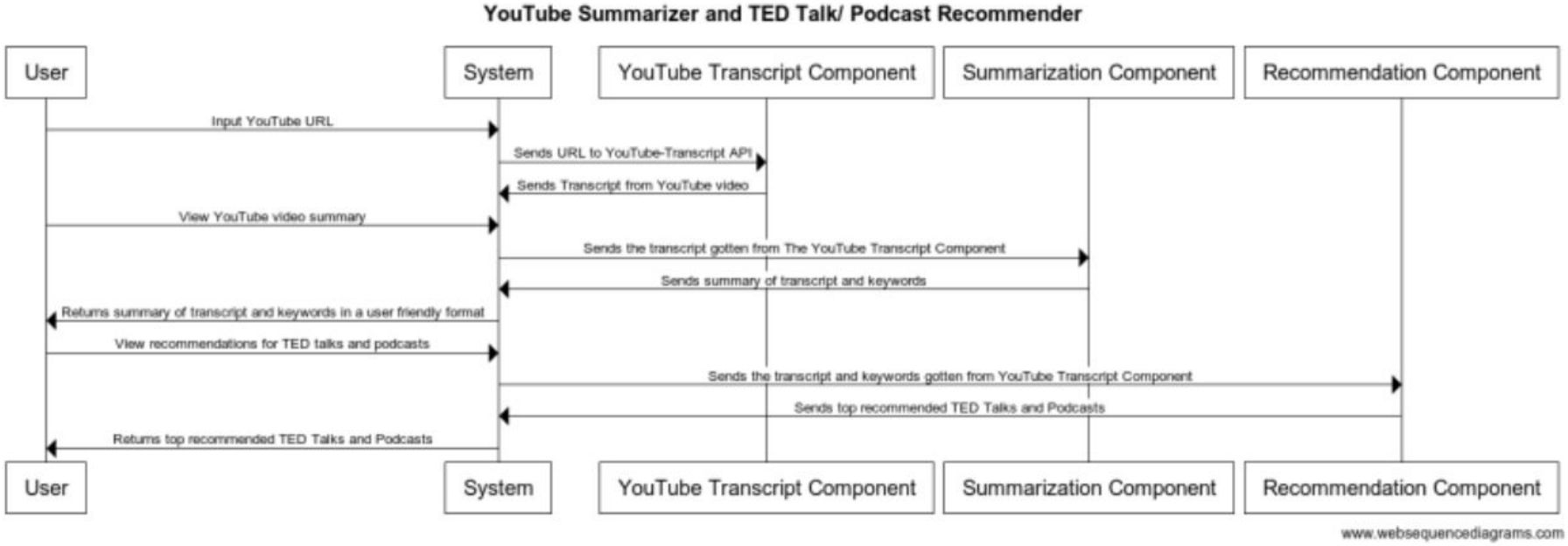


# Design - Recommendation Component

- **What it does:** It selects and shows the TED talks/ podcasts from our database that are most related to the provided YouTube video.
- **Inputs:** Takes the summary from the summarizer
- **Outputs:** A list of TED Talks and/or Podcasts relevant to the user's inputted YouTube Video
- **Interactions:** Uses the summary to generate a similarity score to find and display relevant TED talks and podcasts

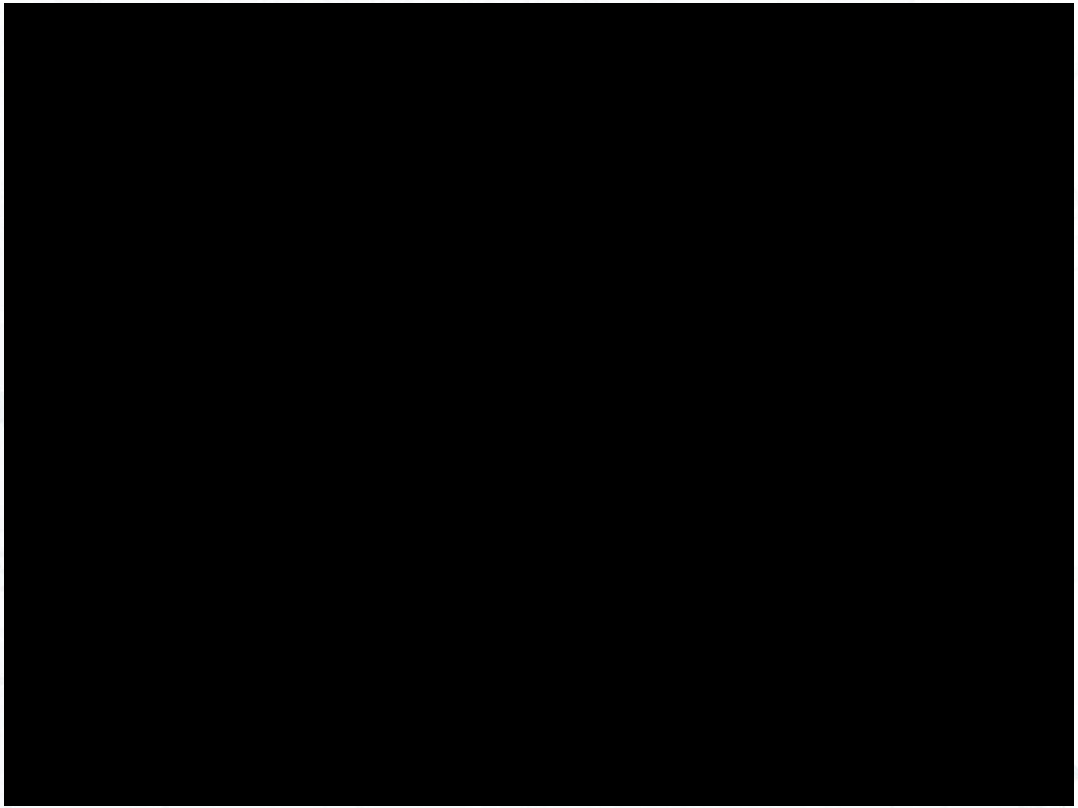


# Design - Final Design Diagram



05

# Demo



06

# Lessons Learned + Future Work

# Lessons Learned

- **Effective API Integration:**
  - Integrating external APIs efficiently by understanding documentation, managing authentication, and implementing error handling.
- **Designing Recommendation Systems:**
  - Developing robust recommendation systems involves careful algorithm selection, data preprocessing, and model evaluation.
- **Developing User-Friendly Web Applications:**
  - Creating user-friendly web applications with frameworks like Streamlit, emphasizing simplicity, responsiveness, and accessibility.
- **Agile Collaboration and Iterative Development:**
  - Embracing agile methodologies and iterative development for collaboration, adaptability, and continuous improvement throughout the software lifecycle.

# Future Work

- **Expanding variety of user input data:**
  - Additional user input formats such as PDF files or URLs containing text data for summarization
- **Expanding variety of recommendation content:**
  - Additional resources such as relevant reading materials and courses
- **Expanding language variety of application:**
  - Allowing multiple languages for YouTube video and recommendation
- **Further Chatbot Improvement:**
  - Implementing Retrieval-Augmented Generation (RAG) to optimize the output of a large language model by referencing transcripts

# Thanks!

Any questions?