Software Engineering department

Braude College

Capstone project phase A – 61998

Video Conferencing Application with Speech Transcription

24-1-D-19

Mentor: Ronen Zelber

Students: Rami Amasha, Moayed Hamze

Github Link : https://github.com/RamiAmasha31/rm-video-call

BRAUDE
College of Engineering, karmiel

Table of Contents

1.Abstract

Our project aims to develop an advanced speech recognition system using the conformer-2 model from AssemblyAI API and stream.io SDK for video chatting. This solution addresses existing challenges by offering accurate speech-to-text conversion, enhancing inclusivity and accessibility in digital communication. We prioritize user experience by sending transcriptions directly to participants after video conversations, promoting convenience and knowledge retention. Our goal is to create web application that seamlessly provides transcription files to all users at the end of each video chat.

2.Introduction

In the contemporary era of digital communication, the demand for efficient speech recognition systems has grown significantly. This surge is not only fueled by the widespread adoption of video communication platforms but is deeply rooted in the changing dynamics of work, education, and collaboration. The need for seamless and accurate speech-to-text solutions has become crucial, especially in the context of remote work, virtual meetings, and online collaboration.

Existing speech recognition systems have made notable strides, but challenges persist. While various platforms offer basic speech-to-text functionality, there is a need for more advanced, context-aware solutions. The current landscape often falls short in delivering robust accuracy, particularly in diverse acoustic environments and with varying accents. Moreover, the computational and memory efficiency of these systems remains a concern, impacting their deployment at scale.

Our project sets out to address these challenges by developing an advanced speech recognition system, leveraging state-of-the-art techniques such as the Conformer-2 model from AssemblyAI. This integration of cutting-edge technology allows us not only to meet but exceed expectations for speech recognition, creating a system that performs exceptionally well in real-world scenarios.

Our envisioned solution holds promise in several domains. Firstly, it contributes to the inclusivity of digital communication by providing an accurate and reliable means of converting spoken words into text. Individuals with hearing impairments can actively participate in video chats, fostering a more inclusive digital environment. Additionally, the educational sector stands to benefit as students gain access to transcriptions, aiding in review, study, and comprehension. In professional settings, our system facilitates efficient information retrieval from video conversations, streamlining workflows in the era of remote work.

Stakeholders who stand to gain from our solution include individuals with hearing impairments, students, professionals engaged in virtual collaboration, and anyone

seeking accurate speech-to-text conversion. The project aims to empower these users, making digital communication more accessible, efficient, and inclusive.

As a key feature of our solution, we prioritize user experience by sending transcriptions directly to participants after the conclusion of video conversations. This ensures that each user retains access to a comprehensive and accurate record of spoken content, promoting convenience and enabling easy reference. By providing users with transcriptions, our solution enhances post-conversation engagement, knowledge retention, and accessibility. This user-centric approach distinguishes our project and contributes to a more enriched and effective communication experience.

Looking ahead, we recognize the importance of continuous improvement and scalability. Future enhancements could include the integration of multilingual support, allowing users from different linguistic backgrounds to communicate effectively. Additionally, optimizing the system for real-time processing in low-bandwidth environments will broaden accessibility. We also aim to explore machine learning techniques that adapt to individual speaking styles and accents over time, further improving accuracy.

Moreover, as our user base grows, ensuring the system's computational efficiency and responsiveness will be critical. We plan to implement cloud-based solutions that can dynamically allocate resources based on demand, ensuring seamless performance even during peak usage. By addressing these future challenges, we aspire to transform our application into a robust platform capable of evolving alongside the needs of our users, making digital communication more effective and inclusive for all.

3.Related Work

The landscape of speech recognition research has undergone significant evolution, encompassing a spectrum of models and methodologies aimed at transcribing spoken language into written text. This section delves into the diverse body of related works, spanning traditional methods to contemporary approaches, shedding light on the progression of the field and contextualizing the current project within this trajectory.

3.1. Traditional Speech Recognition Models.

Historically, Hidden Markov Chains Models and Gaussian Mixture Models formed the bedrock of speech recognition systems. However, these models grappled with capturing intricate linguistic patterns, resulting in limitations regarding accuracy and adaptability across various environments.

3.2. Deep Learning Approaches.

With the advent of deep learning, Recurrent Neural Networks emerged as linchpin in speech recognition. Renowned for their capacity to comprehend temporal dependencies in sequential data, RNNs addressed some of the shortcomings of traditional models. The integration of Connectionist Temporal Classification as a training criterion further enhanced the adaptability of models to variable-length sequences.

3.3. Transformer Architecture.

Vaswani at al.(2017) introduced the transformer architecture, originally designed for natural language processing. Transformers revolutionized speech recognition by capturing long-range dependencies more effectively than RNNs. However, their computational demands, particularly for self-attention mechanisms, raised concerns about their practicality in resource-constrained environments.

3.4. Conformer Architecture.

Recognizing the computational challenges posed by transformers, the conformer architecture emerged as a hybrid solution. By combining Convolutional Neural Networks with transformers, models like conformer-Kd achieved improved efficiency without compromising accuracy. This development showcased the potential for optimizing computational resources while maintaining high-performance levels in speech recognition tasks.

3.5. Attention Mechanisms.

Attention mechanisms assumed a pivotal role in refining the accuracy of speech recognition systems. Enabling models to focus on specific segments of the input sequence, attention mechanisms facilitated the capture of contextual information and nuanced features present in spoken language.

3.6. Multimodal Architecture.

Recent research has explored the fusion of visual information with audio signals to elevate speech recognition capabilities. Multimodal approaches leverage both auditory and visual cues to enhance robustness and accuracy, particularly in challenging environments or amidst noise.

In conclusion, this journey through the evolution of speech recognition models underscores the continuous efforts to address inherent challenges. Each development contributes to the field's advancement, with a focus on improving accuracy, adaptability, and computational efficiency.

4. Background

This section provides a comprehensive understanding of the foundational concepts and technologies relevant to the project. It serves as the knowledge base upon which the proposed solution is built. The background for this project encompasses key elements in speech recognition, including fundamental techniques, models, and the challenges associated with optimizing accuracy and efficiency.

4.1. Speech Recognition Fundamentals.

Speech recognition, also known as automatic speech recognition, is a technology that enables machines to convert spoken language into written text. The primary objective is to develop sophisticated models and algorithms that can accurately understand and transcribe spoken words. This process holds immense value in improving accessibility and enhancing suer interaction with various applications.

4.1.1. Stages involved in the process of converting spoken language into written text.

4.1.1.1 Audio Input.

This is where the process begins, capturing spoken language through a microphone or recording device. The audio signal carries acoustic information like pitch, tone and amplitude.

4.1.1.2. Signal Processing.

Raw audio signals undergo signal processing to extract relevant features. Techniques like Fourier analysis may be applied to convert the audio signal into a frequency-domain representation, highlighting distinct features.

4.1.1.3. Acoustic Modeling.

Acoustic models are pivotal as they map acoustic features to phonetic units. During the training phase, these models learn statistical relationships between acoustic patterns and corresponding speech sounds.

4.1.1.4. Language Modeling.

Language models play a crucial role in understanding the linguistic context of the speech. They provide a probability distribution of word sequences, aiding in selecting the most likely transcription.

4.1.1.5. Decoding.

In the decoding phase, the system generates a sequence of words based on the acoustic and language models. Various algorithms, including Hidden Markov Models or deep learning-based approaches, are employed for decoding.

4.2. Hidden Markov Models.

Stand as keystones in the early chapters of speech recognition systems. These statistical models were meticulously crafted to grapple with the temporal intricacies embedded in sequential data, notably the dynamic nature of spoken language.

4.2.1. Modeling temporal dependencies.

At their core, HMMs emerged as powerful tools for modeling dependencies, adept at capturing the unfolding nuances of speech overtime. Recognizing that spoken language inherently unfolds over a sequence, HMMs were tailored to navigate this sequential landscape. They excelled at representing transitions between different states, creating a dynamic framework mirroring the temporal patterns of speech.

4.2.2. Challenges in handling linguistic complexity.

Yet, the prowess of HMMs faced a formidable challenge – the intricate and varied nature of linguistic patterns. The models, while successful in capturing temporal dependencies, grappled with the intricate dance of language. The rigid structure of HMMs proved somewhat limiting when confronted with the rich and diverse complexities of spoken communication.

4.2.3. Integration with GMMs.

To enhance their capability in handling complex linguistic patterns, HMMs often joined forces with GMMs. This fusion aimed to marry the temporal modeling strengths of HMMs with the probabilistic density estimation abilities of GMMs.

In this symbiotic relationship, GMMs played a crucial role in representing the statistical distribution of acoustic features associated with different phonetic units. The coupling of HMMs and GMMs created a more robust system, where the HMMs navigated temporal dependencies, and GMMs contributed probabilistic density estimates, collectively striving to decode the intricacies of spoken language.

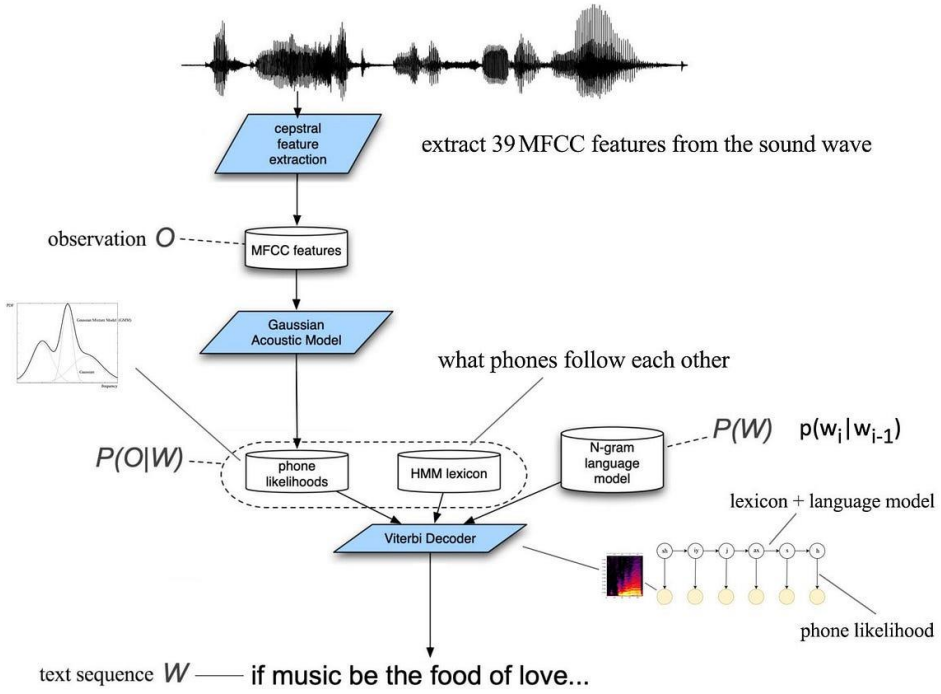


Figure1: process for speech recognition using HMM and GMM.

4.3. RNN.

The advent of deep learning introduced Recurrent Neural Networks, renowned for their proficiency in modeling sequential data. Widely employed in notable applications like Google's voice search and Apple's Siri, RNN excel in processing user input and predicting outputs. Particularly in speech recognition tasks, RNNs demonstrate effectiveness by predicting phonetic segments from audio signals.

4.3.1. Advancements in RNN architecture – E2E models.

Recent progress in RNN architecture focusses on the evolution of end to end models for ASR. These E2E models, replacing traditional hybrid models, exhibit substantial improvements in speech recognition. However, a challenge faced by E2E RNN models lies in synchronizing the input speech sequence with the output label sequence. To address this issue during training, the connectionist temporal classification loss function is commonly used.

4.3.2. Bi-directional RNN.

Bimodal RNN play a pivotal role in advancing audiovisual speech activity detection. BRNNs innovate by integrating separate RNNs for each modality, namely audio and visual. This integration enables BRNNs to capture temporal dependencies with the across modalities, resulting in enhanced performance, especially in challenging and noisy environments.

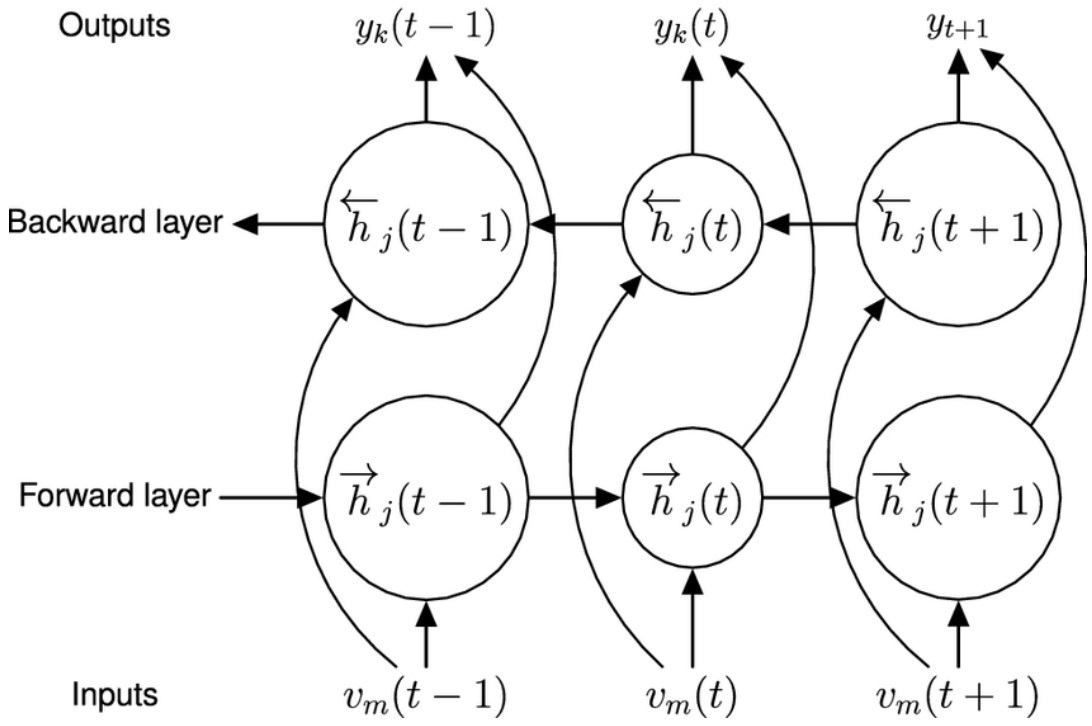


Figure 2:Bi-directional RNN.

4.3.3. LSTM networks in hybrid architectures.

Long Short-Term Memory networks contribute significantly to hybrid architectures, working alongside CNNs to elevate the performance of continuous speech recognition. In these architectures, CNNs extract local features from speech frames, and the processed information is then further refined by LSTMs over time. This collaborative approach enhances the model's capability to understand and transcribe continuous speech with improved accuracy.

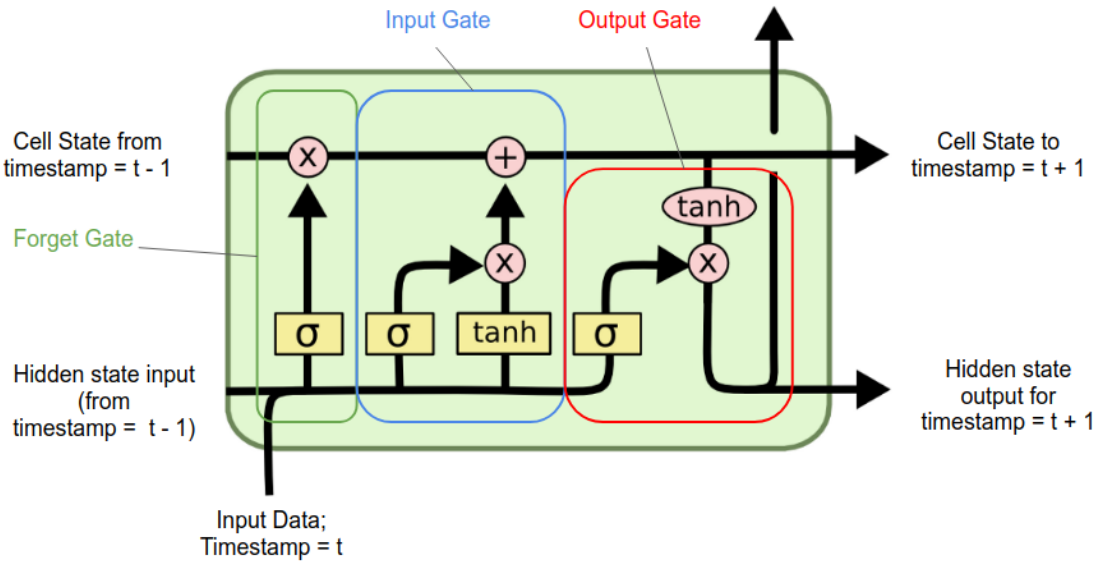


Figure 3: LSTM cell.

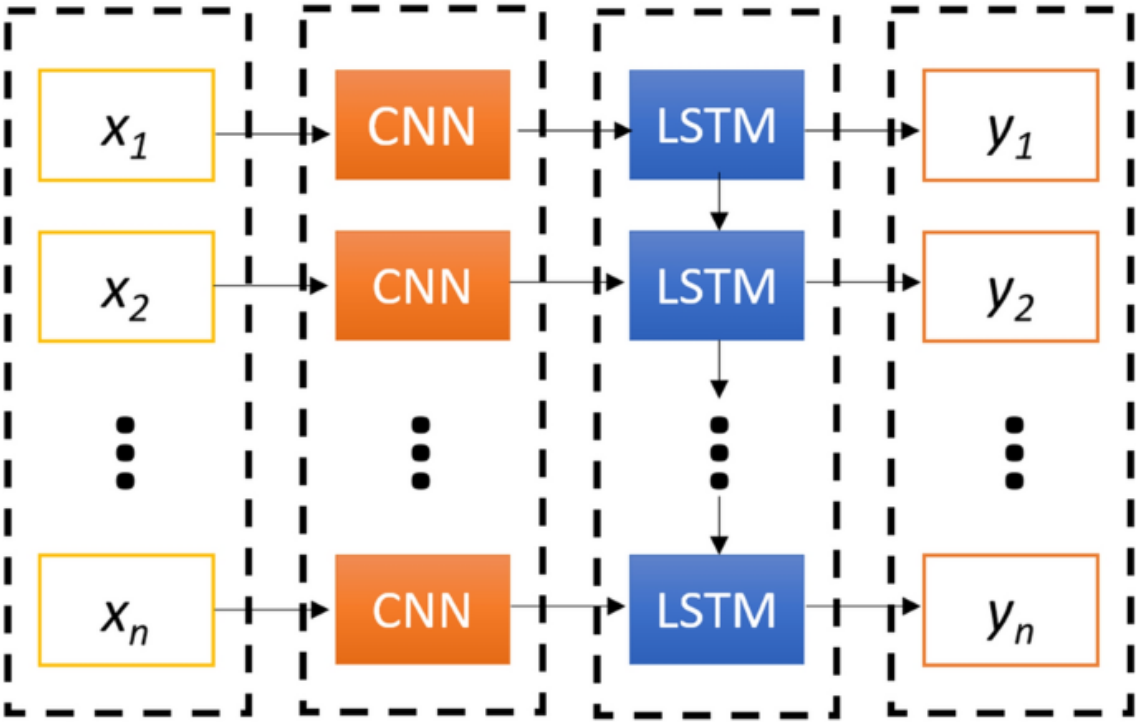


Figure 4: the basic architecture of CNN-LSTM network.

4.4. Attention Mechanisms.

Attention mechanisms in speech recognition play a crucial role in enhancing the accuracy and effectiveness of models. These mechanisms are designed to mimic human attention, allowing the model focus on specific segments or features of the input sequence, enabling the capture of contextual information and nuanced details inherent in spoken language.

4.4.1. Key components and functions attention mechanisms in speech recognition.

4.4.1.1. Contextual Focus.

Attention mechanisms enable the model to dynamically allocate focus to different parts of the input sequence, emphasizing segments that are more relevant for understanding the context. This mimics the way human naturally attend to specific elements during speech comprehension.

4.4.1.2. Capturing Long-Range dependencies.

Unlike traditional models that process input sequences uniformly, attention mechanisms facilitate the capture of long-range dependencies. This is crucial in speech recognition, where understanding the context of a spoken word may require considering information from distant parts of the sequence.

4.4.1.3. Adaptability to variable-length sequences.

Speech signals often vary in duration, making it difficult to align them accurately with corresponding transcriptions. Attention mechanisms address this challenge by allowing the model adaptively focus on different of the input sequence, regardless of if its length, contributing to a more accurate transcriptions.

4.4.1.4. Multi-Head attention.

Some attention mechanisms employ a multi-head architecture, where multiple attention heads work in parallel. This enhances the model's ability to capture diverse aspects of the input sequence simultaneously, leading to a more comprehensive understanding of the spoken language.

4.5. Transformer.

The transformer architecture, introduced by Vaswani et al. in 2017, represents a groundbreaking development in natural language processing that has had a profound impact on various fields, including speech recognition. Originally designed for handling sequential data in natural language, transformers have demonstrated remarkable capabilities in capturing long-range dependencies and contextual information, making them well-suited for tasks involving sequential data like speech recognition.

4.5.1 Key components of Transformer architecture.

4.5.1.1. Self-Attention mechanism.

Transformers utilize a self-attention mechanism, allowing each element in the input sequence to focus on other elements, capturing dependencies regardless of their distance from each other. This mechanism enables the model to consider global context information during processing, addressing the challenge of capturing long-range dependencies.

4.5.1.2. Multi-Head attention.

To enhance the self-attention mechanism, transformer employ multiple attention heads. Each head learns different aspects of the input, providing a more comprehensive understanding of relationships within the sequence.

4.5.1.3. Positional Encoding.

Transformers do not inherently understand the sequential order of input data. Positional encoding is introduced to the input embeddings to convey the position of each element in the sequence, allowing the model discern the order of the data.

4.5.2. Concerns and Challenges.

4.5.2.1. Computational Demands.

One notable challenge with transformers is their computational intensity. Training and deploying large transformer models can be resource-intensive, raising concerns about practicality in environments with limited computational resources.

So we thought about using API of an existing transformer model.

4.5.2.2. Optimization for speech recognition.

While transformers were originally designed for text-based tasks, adapting them for speech recognition requires careful optimization. This involves handling audio data representation and ensuring effective modeling of sequential dependencies.

4.6. Conformer.

The conformer architecture represents an innovative solution that emerged to address the need for efficiency in speech recognition. It stands out as a hybrid model, seamlessly integrating CNNs and transformers. This unique combination harnesses the strength of both architectures, aiming to strike a balance between computational efficiency and high accuracy in speech recognition tasks.

4.6.1. Key components and characteristics of the Conformer architecture.

4.6.1.1. Integration of CNNs.

CNNs are renowned for their effectiveness in extracting hierarchical local features from input data. In the Conformer architecture, CNNs play a crucial role in processing the initial audio representations, capturing essential low-level features.

4.6.1.2. Utilization of Transformers.

Building on the success of transformers in capturing long-range dependencies and contextual information, the Conformer architecture incorporates transformer blocks. This allows the model to effectively model intricate sequential patterns present in spoken language.

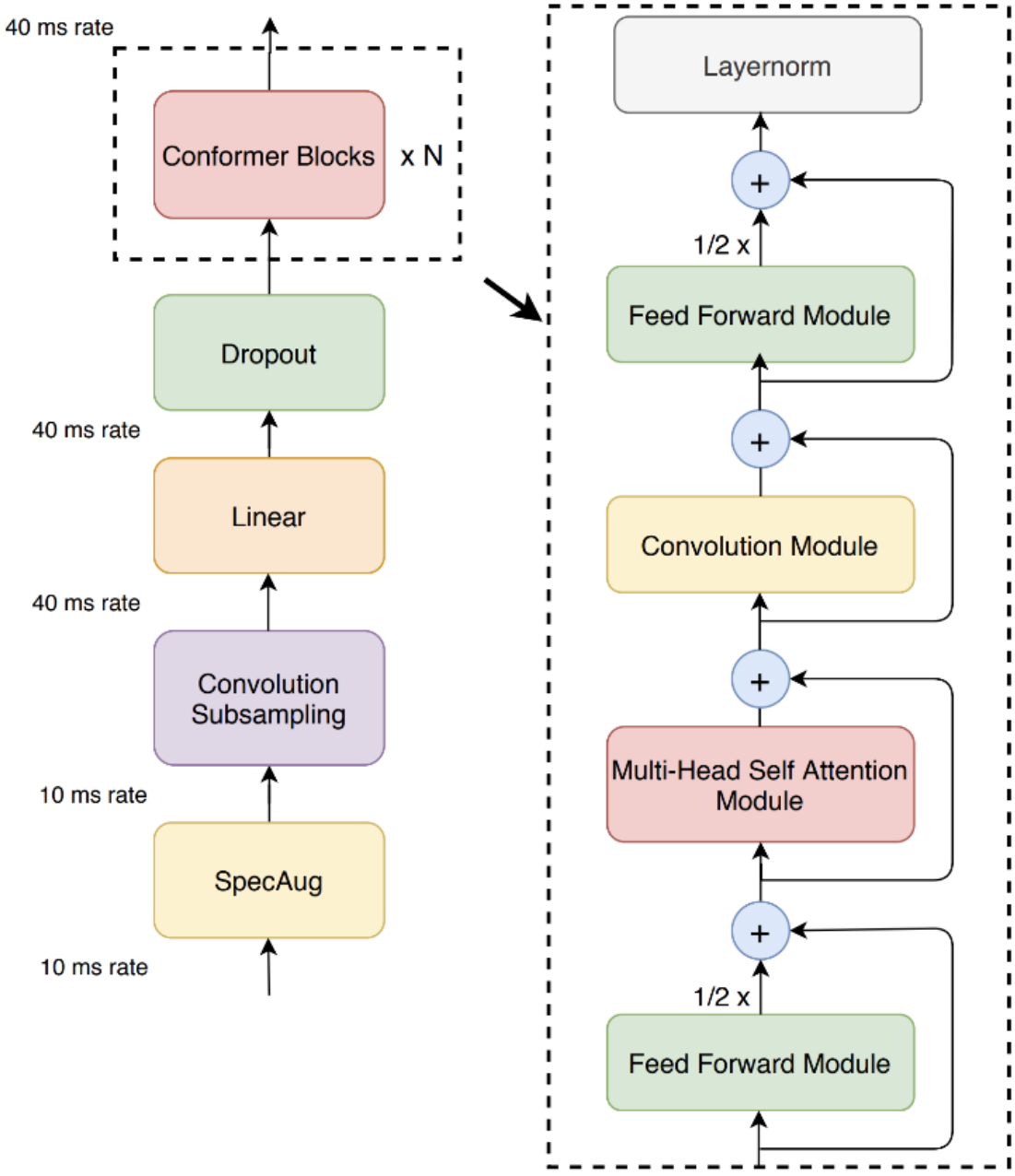


Figure 5: Conformer model architecture.

4.6.2. Conformer-1.

The conformer, introduced by Google brain in 2020, is a neural network designed for speech recognition. It extends the Transformer architecture, renowned for its parallelizability and effective use of attention mechanisms, by incorporating convolutional layers. This integration enables the Conformer to capture both local and global dependencies, maintaining a balance between performance and size efficiency.

While the Conformer has demonstrated cutting-edge performance in speech recognition, it faces challenges in computational and memory efficiency. The extensive use of attention mechanisms becomes a computational bottleneck.

Conformer-1 (introduced by AssemblyAI) addresses these challenges, aiming to create a production-ready speech recognition model capable of deployment at an extremely large scale. The goal is to maximize the outstanding modeling capabilities of the original Conformer architecture while overcoming its computational limitations for practical applications in ASR systems.

4.6.2.1. Modifications.

To enhance the efficiency and performance of the Conformer architecture, Conformer-1 introduces several key modifications. One notable modification is the adoption of the Efficient Conformer, a variant of the original Conformer architecture. The Efficient Conformer incorporates specific technical adjustments aimed at improving computational speed and resource utilization. Two significant modifications brought by Efficient Conformer are:

- Progressive Down sampling. This feature implements a gradual reduction scheme for the length of the encoded sequence. It draws inspiration from the ContextNet approach, which is known for its progressive down sampling technique. The progressive down sampling mechanism efficiently shortens the sequence length, contributing to faster processing and improved computational efficiency.
- Grouped Attention. The attention mechanism, a pivotal component of the original conformer architecture, undergoes modification in Confromer-1. The adaption, referred to as grouped attention, is designed to be agnostic to sequence length. Grouped attention is tailored version of the attention mechanism that enhances

flexibility and adaptability, particularly in scenarios where the sequence length varies.

These modifications result in substantial speed improvements, with Conformer-1 achieving a 29% speedup at inference time and a 36% speedup at training time compared to unmodified Conformer architecture. Importantly, this enhanced efficiency does not compromise accuracy, as Conformer-1 maintains similar or improved word-error-rate accuracy.

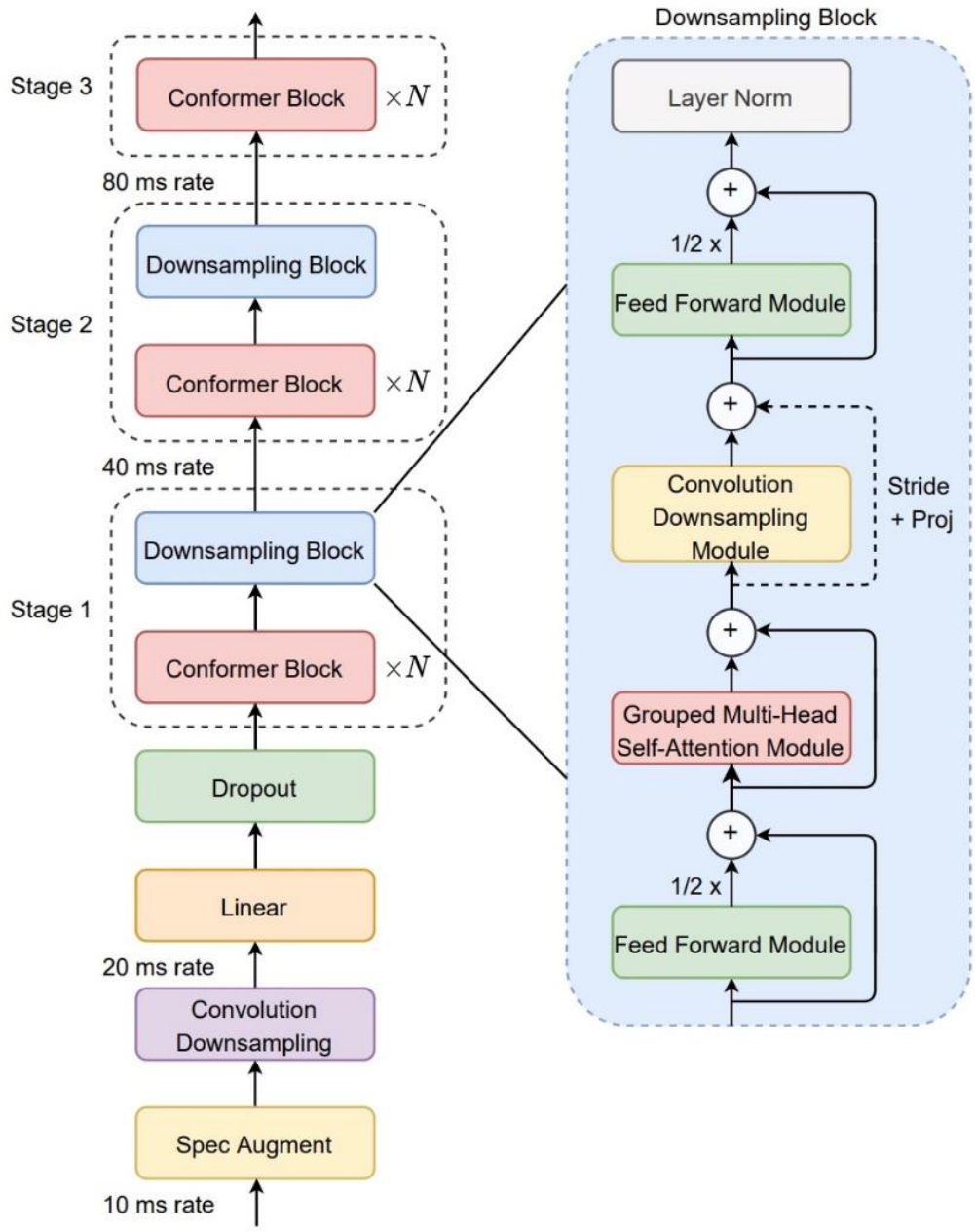


Figure 6 : Efficient Conformer encoder architecture.

4.6.2.2. Enhancements for Robustness.

The conformer-1 model incorporates an enhancement focused on refining accuracy in the presence of noisy audio environments. This improvement involves the implementation of a modified version of Sparse Attention, a pruning technique designed to induce sparsity in the model's weights, thereby providing regularization. The objective of this modification is to bolster performance in scenarios with background noise. By introducing sparsity, the model can reduce the influence of noise at specific timesteps, particularly during the attention steps that capture higher-level features.

In the original sparse attention implementation, the process involves computing the average attention score for a given timestep. The dot-product contributions from timesteps with lower attention scores are then filtered out, enforcing the feature to attend only to the most salient global timesteps. This approach achieves a regularization effect.

With Conformer-1, the method is made more versatile by replacing the average with a moving median. The moving median serves as a quantile threshold for filtering. This modification prevents the filtering threshold from being significantly elevated by very large attention vectors at certain timesteps. Such elevation could lead to excessive pruning. The use of a moving median enhances the generalizability of the sparse attention method, contributing to its effectiveness in addressing noise-related challenges in audio processing.

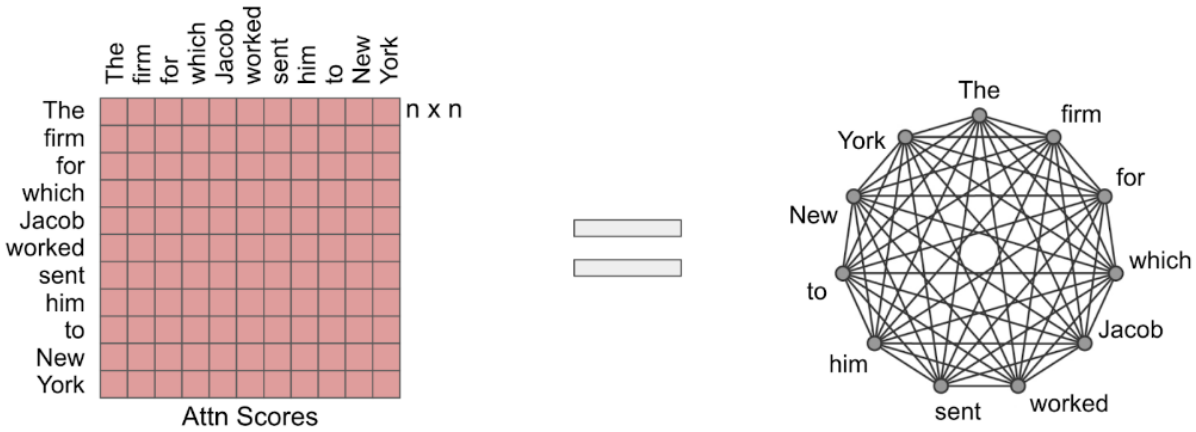


Figure 7: the attention mechanism can be visualized as a fully connected graph.

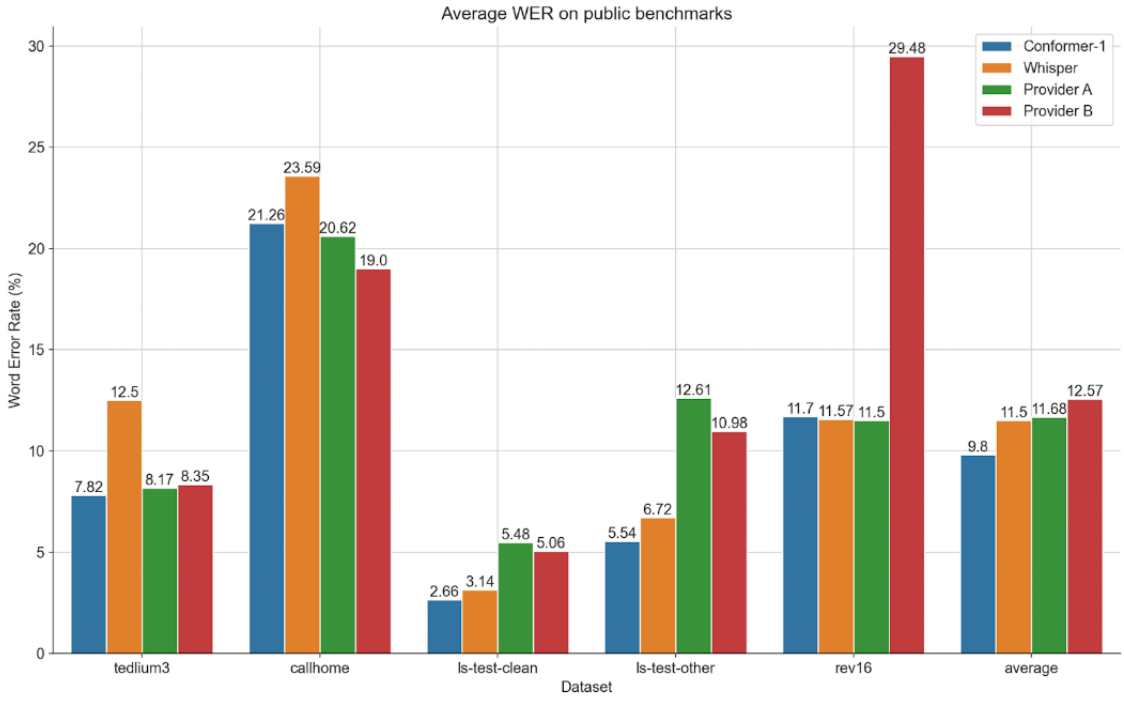


Figure 8: Confomrer-1 results on benchmarks datasets.

4.6.3. Conformer-2.

Conformer-2 represents a significant advancement over Conformer-1, maintaing comparable word error rates while achieving substantial improvements in user-oriented metrics. Inspired from DeepMind's Chincilla paper, this update focuses on enhanced performance and robustness in real-world audio conditions. Conformer-2 exhibits a 31.7% boost in alpha numerics, a 6.8% improvements in Proper Noun Error Rate, and a notable 12.0% increase in noise robustness. These enhancements result from increased training data, scaling up to 1.1 M hours of English audio, and optimizing pseudo labeling with and expand set of models.

4.6.3.1. Model ensembling.

Conformer-2 incorporates model ensembling, building upon the noisy student-teacher training technique used in Conformer-1. In this advanced approach, multiple strong teacher models generate labels for both labeled and unlabeled data, providing a broader distribution of behaviors. Model ensembling is known for reducing variance and enhancing robustness when faced with unseen data during training. By leveraging an ensemble of teacher models, Confromer-2 benefits from a more comprehensive learning experience, where individual model failures are mitigated by the success of others in the ensemble.

4.6.3.2. Model parameters.

Conformer-2 addresses the undertraining issue highlighted in Chinchilla paper by scaling up data and model parameters. Inspired by the paper's scaling laws, which suggest the appropriate training data for different model sizes, Conformer-1 determined a need for 650000 hours of audio data. Conformer-2 adheres to this scaling curve, elevating model size to 450 million parameters and training on an extensive dataset of 1.1 million hours of audio data. This scaling approach aims to optimize the models performance and training depth for enhanced speech recognition capabilities.

4.6.3.3. Speed improvements.

Despite the common trade-off of increased cost and slower speeds associated with larger models, Conformer-2 defies this trend. The implementation of an advanced serving infrastructure allows Conformer-2 to achieve speeds up to 55% faster than Conformer-1, especially noticeable in the processing of longer audio files. This

emphasizes the project's commitment to efficient and cost-effective deployment of advanced speech recognition models.

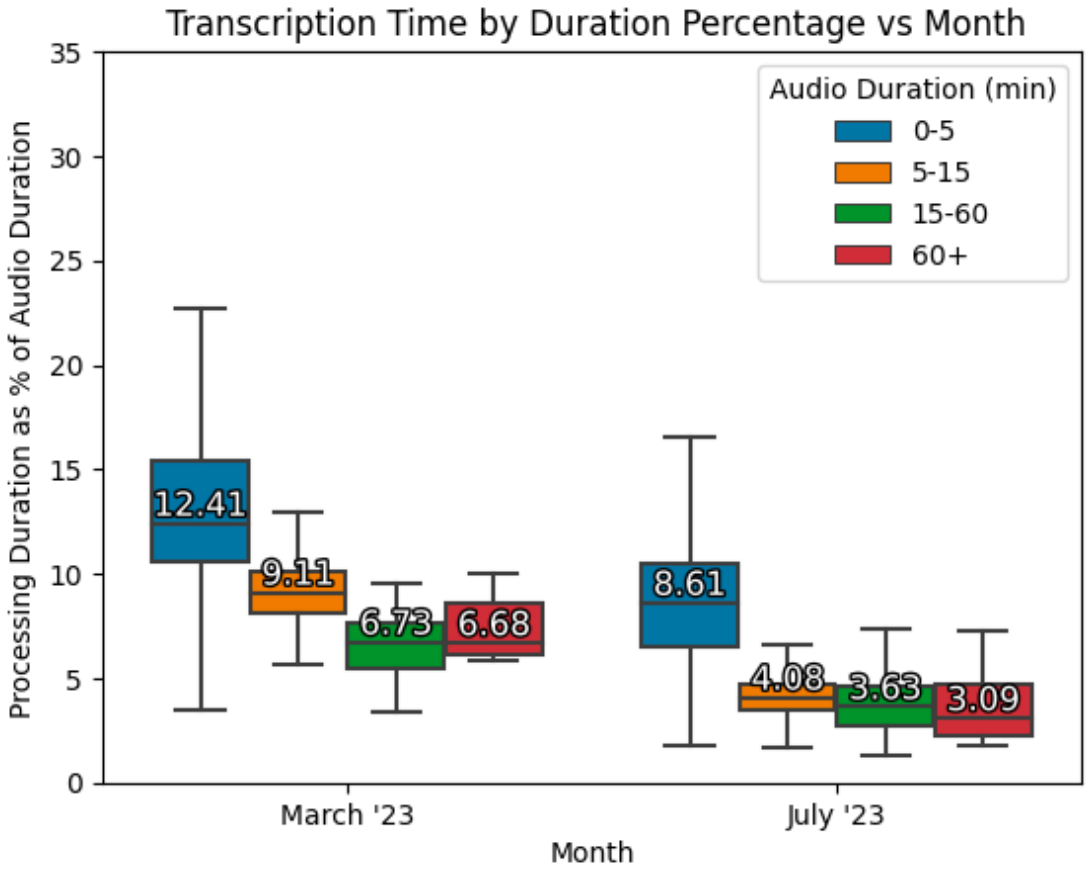


Figure 9: % of File duration taken to transcribe a file in March (Conformer-1 release date) vs in July (Conformer-2 release date). Various file durations were considered and put into the bins shown above. The percentage is calculated using a random sample of 5k files transcribed during each month.

4.7. Multimodal Approaches.

Multimodal approaches in speech recognition involve the integration of both audio and visual information to enhance the robustness and accuracy of the systems. This fusion of modalities provides a more comprehensive understanding of the input data, especially beneficial in challenging environments with background noise or other audio disturbances.

4.7.1. Key Aspects Of Multimodal Approaches In Speech Recognition.

4.7.1.1. Audio-Visual Fusion.

Multimodal approaches leverage information from both audio signals and visual cues. By combining these modalities, the system gains more complete representation of the spoken language, reducing the impact of environmental noise of other audio interferences.

4.7.1.2. Improved Robustness.

The integration of visual information contributes to the robustness of speech recognition systems. Visual cues, such as lip movements or facial expressions, can provide context and aid in disambiguating ambiguous audio signals, especially in challenging acoustic environments.

4.7.1.3. Noise Resilience.

Background noise is a common challenging in many real-world scenarios. Multimodal approaches address this by incorporating visual information, allowing the system to better filter out noise and focus on relevant audio-visual features for accurate transcription.

4.7.1.4. Contextual Understanding.

Visual information adds a layer of contextual understanding to speech recognition. For instance, observing the speaker's mouth movements may help in distinguishing between phonetically similar sound, contributing to improved accuracy.

4.7.1.5. Speaker Diarization.

Multimodal approaches can aid in speaker diarization, the process of distinguishing and attributing speech segments to different speakers in an audio stream. Visual cues, such as facial features can assist in speaker identification and tracking.

4.7.1.6. Human-Computer Interaction.

In applications involving human- computer interaction, multimodal approaches enhance the user experience. Combining auditory and visual feedback enables more natural and intuitive communication with devices, such as voice-activated assistants.

4.7.1.7. Adaptability To Diverse Environments.

The fusion of audio and visual modalities makes speech recognition systems more adaptable to diverse environments. Whether in noisy public spaces, quiet offices, or even situations with multiple speakers, multimodal approaches provide a versatile solution.

4.7.1.8. Gesture Recognition.

Some multimodal systems incorporate gesture recognition along with speech. This combination allows users to convey information through both spoken words and gestures, expanding the range of inputs recognized by the system.

5. Expected Achievements.

5.1. Outcomes.

5.1.1. Web Application Development.

The web application is crucial component of the project, serving as the user interface for video chat interactions and managing the transcription process. Here's an elaboration on the key aspects.

- User-Friendly Interface: We aim to design a clean and intuitive interface that allows users to easily navigate through the applications. Considering a layout that is visually appealing and provides a seamless experience for initiating and participating in video chats.
- Video Chat Capabilities: Implement video chat features within the application, allowing users to initiate, join and end video conversations using STREAM.IO SDK. This way we ensure smooth video streaming, low latency, and high-quality audio to enhance the overall user experience.
- User Authentication: Implement a secure user authentication system to ensure that only authorized users can access the application.
- File Management: Develop a file management system to organize and store transcription files generated by Cnformer-2 AssemblyAI API. Ensuring that users can easily access and review transcriptions from past video chats.

5.1.2. Transcription File Generation.

This component of the project is essential for providing a valuable and accessible record of the spoken content in the video chat sessions. Here's a detailed explanation of the transcription file generation process.

- Post-Video Recording Detection: By using stream.io SDK we can easily detect recording end and then easily can be sent to the transcription API.
- Integration with Confromer-2 Model: The system seamlessly integrates with the Conformer-2 model using the AssemblyAI API. This involves sending audio content of the video chat to the Conformer-2 model.
- Speech-To-Text Transcription: The Conformer-2 model process the audio content and transcribes it into written text. This transcription captures the

spoken words, allowing for a detailed and accurate representation of the conversation.

- File Formatting: The generated transcription is formatted into a PDF file, ensuring clarity, readability, and easy interpretation.
- Automatic Distribution: Once the transcription file generated, the system automatically saves the file for each participant involved in the video chat in his log history.

5.1.3. Scalability.

The scalability aspect of our project is a critical component designed to ensure the robustness and efficiency of the entire system, particularly in response to increasing user engagement and growing demands.

- User Growth Projection: Develop the system architecture with the capacity to seamlessly onboard new users, providing a consistently high-quality experience.
- Concurrent Session Management: Design the system to intelligently mange resources, allocate bandwidth and maintain optimal performance as the number of concurrent video chats increases.

5.2. Success Criteria.

The success of the project will be measured against predefined criteria that encompass technical, user experience and performance aspects. The achievement of these criteria will validate the effectiveness and viability of the web application.

- Transcription Accuracy: The system should achieve a high level of accuracy in transcribing speech to text. So evaluating the accuracy of transcriptions by comparing the generated text with the original spoken content. Success is defined by consistently achieving a predetermined level of accuracy, minimizing errors, and correctly capturing nuances in speech.
- Usability and User Satisfaction: The web application should be user-friendly and meet user expectations. So we need to conduct usability testing and gather user feedback to assess the intuitiveness of the interface, ease of navigation, and

overall user satisfaction. Success is defined by positive user feedback, minimal user-reported issues, and high ratings in usability assessments.

- Automated Transcription Process: the system should seamlessly automate the transcription process after each video chat. So, we need to evaluate the efficiency of the transcription process by analyzing the time taken from the end of a video chat to the delivery of transcription file. Success is defined by timely and automated generation of accurate transcriptions.
- Integration with AssemblyAI API: The application should seamlessly integrate with the AssemblyAI API for transcription services. Verifying the integration by assessing the accuracy and reliability of transcription generated using the Conformer-2 model through the AssemblyAI API. Success is defined by consistent and successful integration without major disruptions.

  By meeting these success criteria, the project aims to deliver a robust, user-friendly and scalable web application that leverages advanced speech recognition capabilities, thereby providing a valuable tool for users engaging in video conversations with automated transcription services.

5.The Process.

5.1 Research – Speech Recognition.

To deepen our understanding of speech recognition and its applications, we embarked on addressing the following key inquiries:

- How does speech recognition technology impact communication and accessibility for individuals with hearing impairments?
- What age groups benefit most from enhanced speech-to-text solutions, particularly in education and professional settings?
- Identifying current limitations in existing speech recognition systems, particularly concerning accuracy and efficiency in diverse environments.
- Reviewing previous attempts to integrate advanced technologies into speech recognition systems and assessing their performance.

Our research encompassed a comprehensive examination of scientific literature, articles, and video resources. Following this exploration, collaborative discussions were held to distill key findings and prioritize development objectives for our project. Notably, our findings emphasized the necessity of implementing a comprehensive questionnaire alongside out evaluation to validate the accuracy and reliability of our proposed speech recognition application.

Moreover, we determined that the primary target demographic for our solution should be professionals and students and people with hearing impairments, given the significance of early intervention and accessibility in enhancing communication and workflow efficiency.

5.1.1 Constraints and Challenges – Speech Recognition

One of the significant challenges encountered during our project was the substantial computational power required by advanced speech recognition models to maintain optimal performance. To address this challenge effectively, we opted to leverage API models such as the Conformer-2 from AssemblyAI. This decision allows us to harness state-of-the-art technology without the need for extensive computational resources. The Conformer-2 API provides exceptional accuracy and efficiency in English speech-to-text conversion, meeting our requirements for scalability and accessibility within this language constraints. In addition, STREAM.io SDK's provides several services for video chatting that would simplify our coding process and make it more efficient with higher performances, one of these problems is scalability which can be solved by utilizing stream.io SDK's that can handle large amount of users and video room simultaneously.

Utilizing API-based models presents a constraint as it limits our system's language support to English. Despite this constraint, our approach enhances overall performance and usability, accommodating diverse speech patterns and environmental variables. This strategy ensures that our speech recognition system delivers reliable and effective results, contributing to the success of our project goals.

5.1.2 Conclusions From Research – Project Direction

Our research findings have strongly influenced our project direction, leading us to prioritize the development of comprehensive application for video chats integrated with

advanced speech recognition capabilities using the AssemblyAI API, specifically leveraging the Conformer-2 model, also using stream.io SDK for video chatting and recording rooms functionality due its ability to handle more users simultaneously which improves our web application scalability. Through our investigation, we have identified the critical importance of context-awareness and adaptability in addressing the complexities of speech variability and environmental conditions.

By incorporating the AssemblyAI Conformer-2 API into our solution, we aim to enhance accuracy and usability across diverse acoustic environments and speech variations within the context of video communication experience, empowering users with seamless speech-to-text conversion during video interactions. Our goal is to surpass conventional capabilities by integrating state-of-the-art technology into a user-friendly application tailored to for real-world use.

5.2 Research – Machine Learning

Our exploration into machine learning encompassed fundamental aspects essential for developing our advanced speech recognition system:

Hardware:

- Computational Requirements: We extensively studied the computational demands of deploying machine learning models for real-time speech processing. This investigation focuses on identifying hardware specifications and capabilities necessary to achieve optimal performance.
- Role of Specialized Hardware Accelerators: We evaluated the effectiveness of specialized hardware accelerators in optimizing models inference and efficiency. This include examining GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units) to leverage parallel processing for enhanced speed and efficiency.

Software:

- Surveying Machine Learning Frameworks and Libraries: We conducted a comprehensive survey of available machine learning frameworks and libraries suitable for developing a speech recognition algorithms. This included popular frameworks such as TesorFlow, PyTorch, and Keras, assessing their compatibility and features for our specific application.
- Investigating Data Preprocessing and Feature Engineering Techniques: We explored various data preprocessing methods and feature engineering techniques to enhance the performance of our machine learning models. This involved analyzing methods for cleaning and transforming raw speech data into suitable input features, ultimately improving the accuracy and robustness of our speech recognition system.

- Examining API Performance: In addition to frameworks and libraries, we examined the performance of different APIs (Application Programming Interfaces) available for speech recognition tasks. This evaluation included assessing the accuracy, latency, and scalability of APIs such as Google Cloud Speech-to-Text, Microsoft Azure Speech Services, and AssemblyAI, to determine the most suitable solution of our project.

Our research in machine learning provided foundational insights crucial for the successful implementation and optimization of our advanced speech recognition system. By understanding the interplay between hardware capabilities, software frameworks, and API performance, we have positioned ourselves to leverage state-of-the-art technologies effectively in developing a scalable and efficient solution.

5.3 Methodology and Development Process

To effectively develop our advanced speech recognition system utilizing AssemblyAI API (Conformer-2) and using stream.io SDK's, we have adopted an agile methodology tailored to our project's requirements. Our development process encompasses the following stages:

- Designing and Prototyping System Architecture: We will design and prototype the speech recognition system architecture, integrating the AssemblyAI Conformer-2 API and stream.io SDK's into our solution. This involves defining data flow, processing pipelines, and integration points within the system.
- Integrating User Feedback and Optimization: Throughout development, we will gather user feedback and integrate it into the system. We will optimize the speech recognition system based on performance metrics obtained directly from the AssemblyAI Conformer-2 API, ensuring it meets the specific needs and expectations of our target users.
- Developing User-Friendly web Application: In parallel with the speech recognition system, we will develop a user-friendly web application. This application will seamlessly integrate the speech-to-text

functionality, allowing users to participate in video chats and receive transcriptions post-conversation.

Throughout the development lifecycle, we will emphasize continuous integration and iterative improvements based on the outputs and insights provided by the AssemblyAI Conformer-2 API. This iterative approach ensures the delivery of a robust, scalable, and user-centric speech recognition solution integrated into a user-friendly web application. By leveraging the AssemblyAI Conformer-2 API and agile methodology, we aim to create a cutting-edge system that enhances communication and accessibility for our users.

6. Product.

6.1. Requirements.

- Functional.

1	Users can initiate one-on-one video calls with other participants.
2	User can create and join group video calls with multiple users.
3	The application provides controls for users to toggle microphone and camera settings during video call.
4	Speech-to-text transcription using AssemblyAI conformer-2 API is integrated into the application.
5	Users have control over their data privacy, including the ability to delete stored transcriptions.
6	Users have the ability to view previous transcription files.
7	Transcription files are organized to the date of its creation.

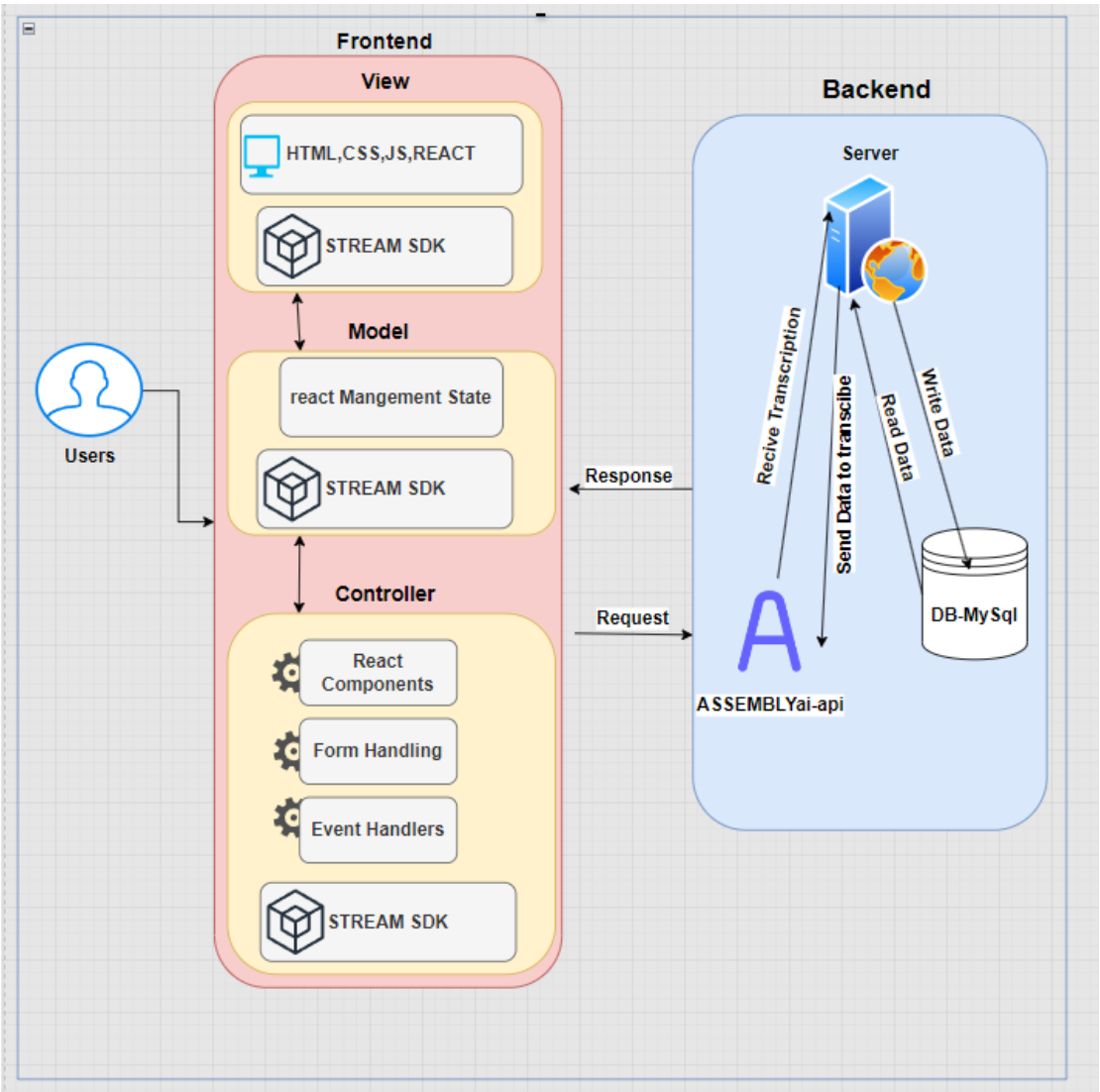
- Non-Functional.

1	The application shall maintain low latency and high responsiveness during video calls.
2	The system architecture should be designed to scale efficiently to accommodate increasing numbers of users and concurrent video sessions.
3	The application shall be robust and stable, minimizing downtime and ensuring consistent availability for users.
4	The application should be compatible with wide range of devices and operating systems to ensure broad accessibility for users.
5	The user interface should be user friendly.
6	Comprehensive error handling and recovery mechanisms should be implemented to gracefully mange unexpected system failures.
7	The codebase should be well structured and documented, facilitating ease of maintenance, updates, and future enhancements.

6.2. Architecture Overview.

Our architecture consists of several key components:

- Frontend, which is what the user sees.
- We also have our backend server for handling the requests from clients, also to communicate with external third parties.
- Our database (MySql) for storing all the necessary data.
- Also we have the API's and SDK's that will simplify the coding process.

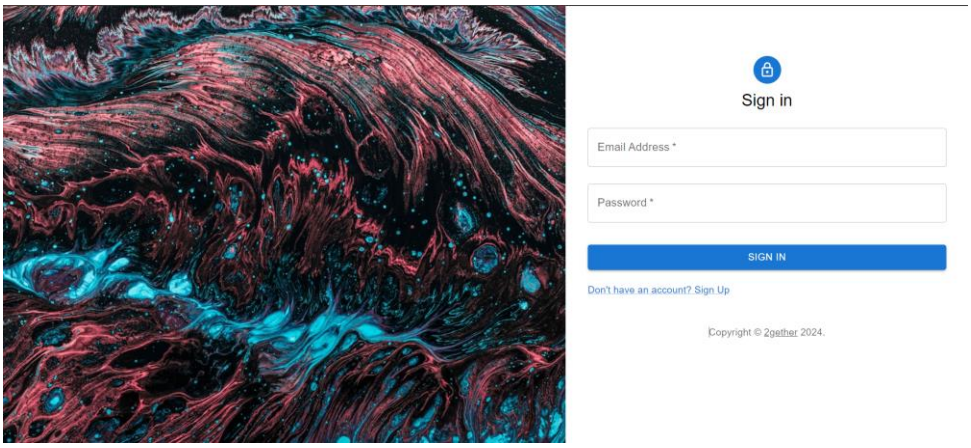


6.2.1. Interfaces.

This section will present the key component interfaces in our application, we strive to have user-friendly interface, simple, organized and easy to use.

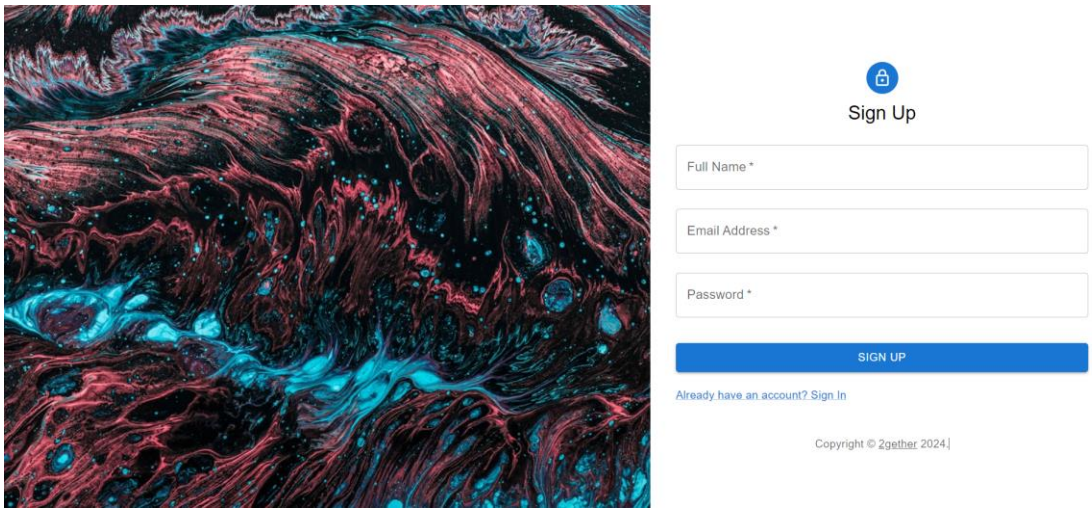
6.2.1.1. Login Page

This is the first page user will see after he enters the website, form for logging in and random image will be generated once the user refreshes the screen. The user will be asked for email and password.



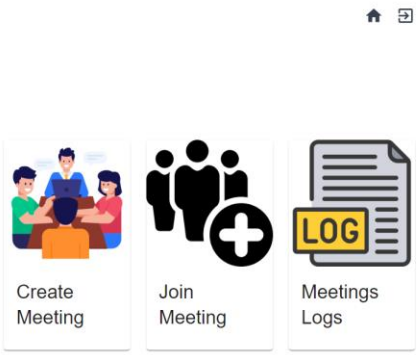
6.2.1.2. Sign up Page.

If the user new and not signed up, he would click on sign-up button and will be redirected to the following screen, which is similar to login screen but here we also ask for full name field.



6.2.1.3. Hompage.

After successful login process, the user will land on our homepage, we wanted to keep it simple and clear to make it easier for the user, in this page the user can easily pick what he want and he also can logout.

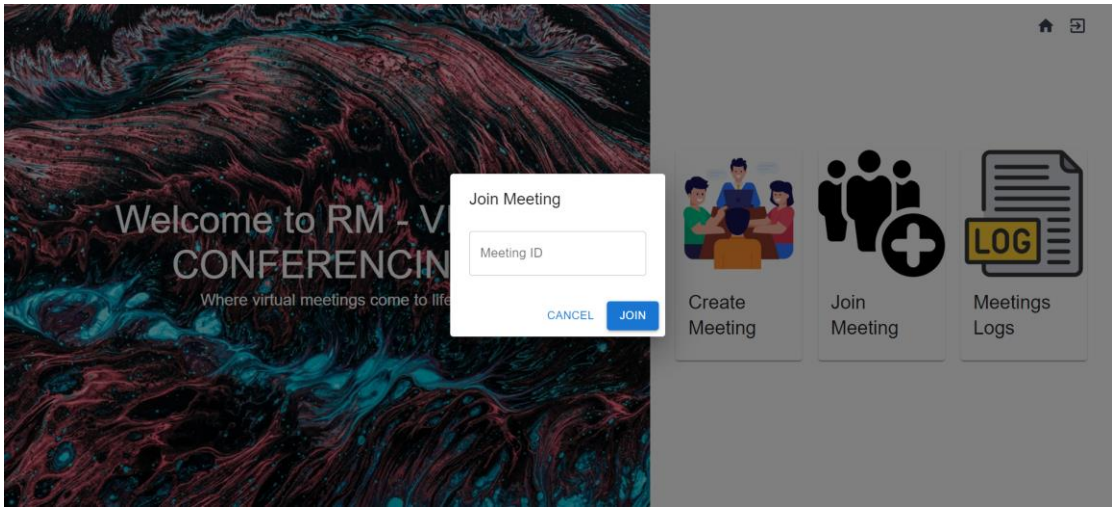


6.2.1.4. Video room.

If user has clicked on create meeting in the homepage then he will be immediately redirected to the meeting room, here we used the stream SDK UI components for keeping it simple and because this structure is in common for all video chat applications so we don’t want to confuse the u

6.2.1.5. Joining meeting.

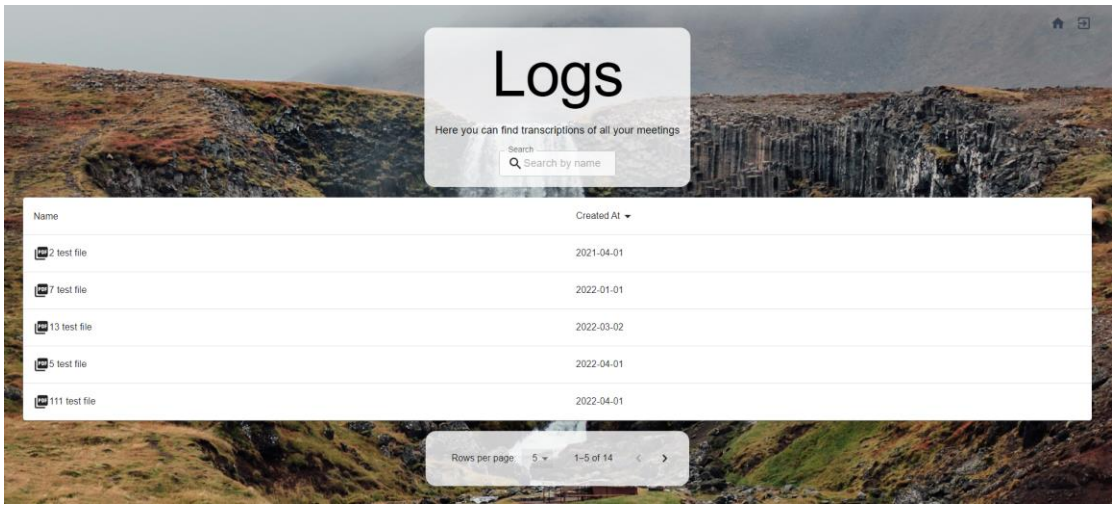
When user clicks on join meeting, a dialog will pop out and ask for the meeting ID, after that he will be redirected to the meeting room.



6.2.1.5.1. Redirecting to room meeting after successful Meeting ID check.

6.2.1.6. Logs screen.

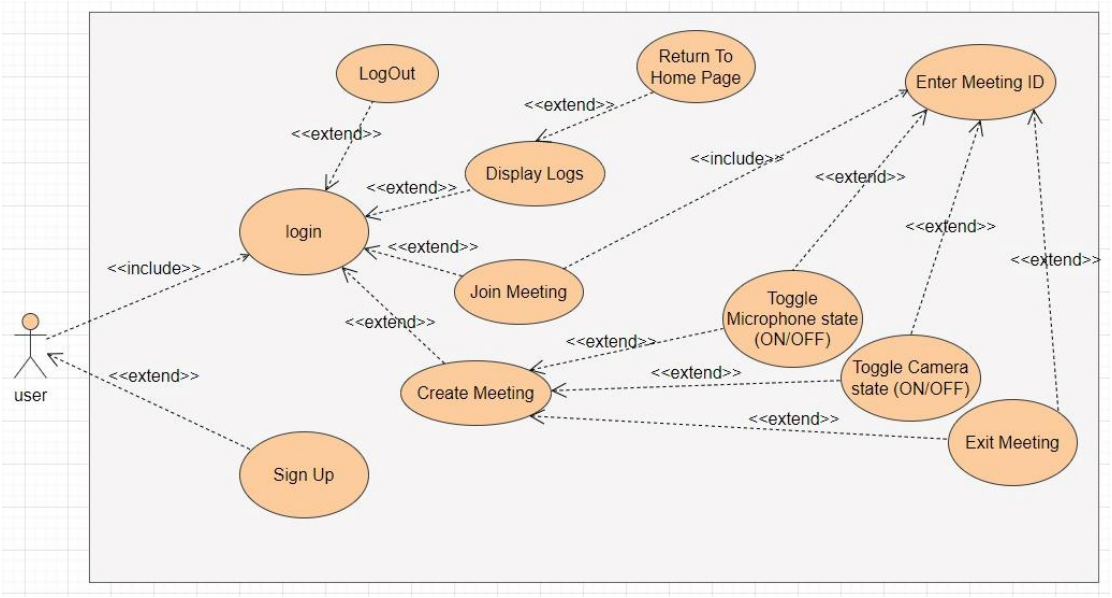
The following screen is for displaying all the transcription files for the specific user meetings, he can search for specific file name, sort them by date and also when clicking on file name it will be downloaded automatically to the user's computer.



6.3. Diagrams.

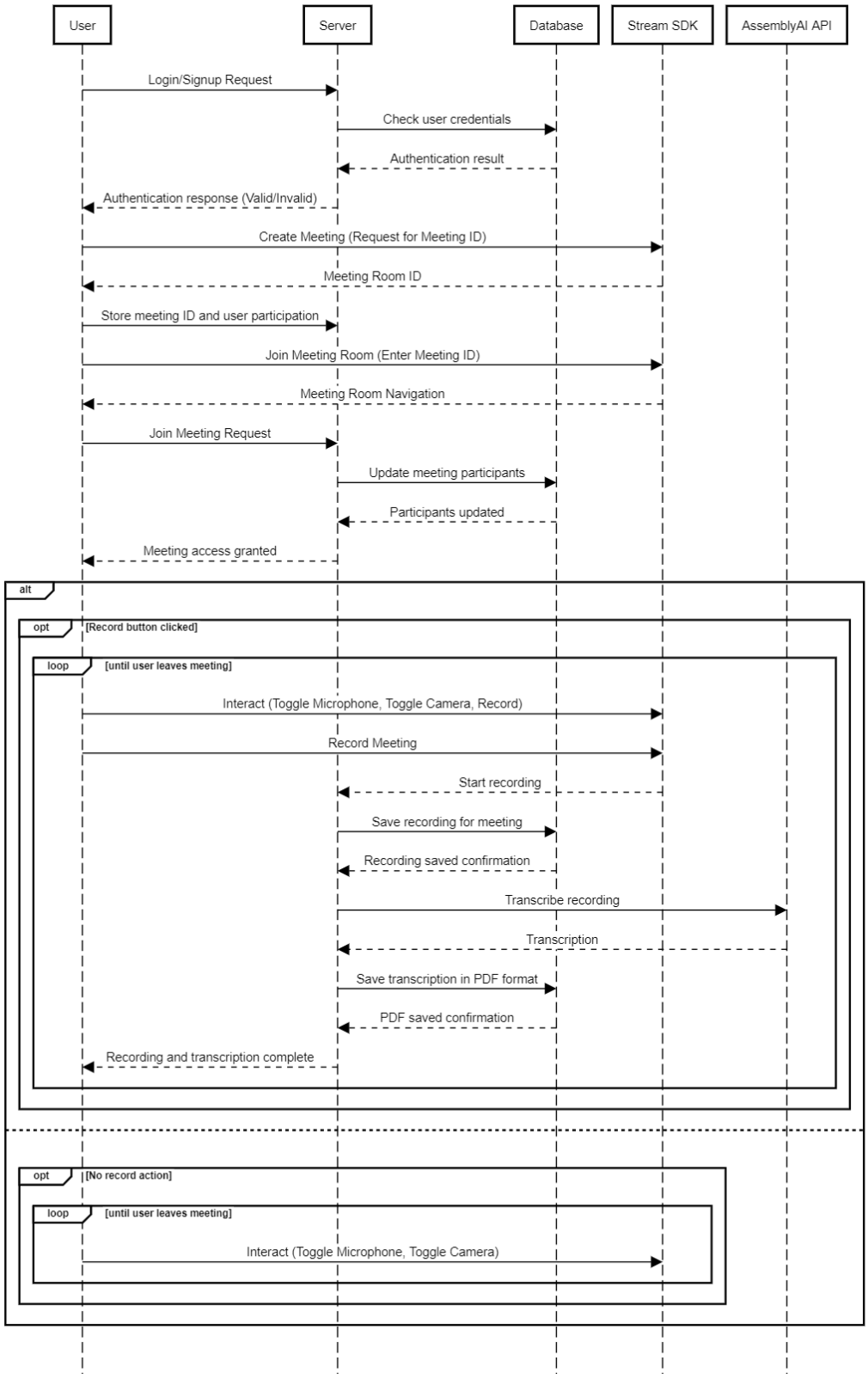
6.3.1. Use Case Diagram.

The following use case shows all possible user interaction with the web application.



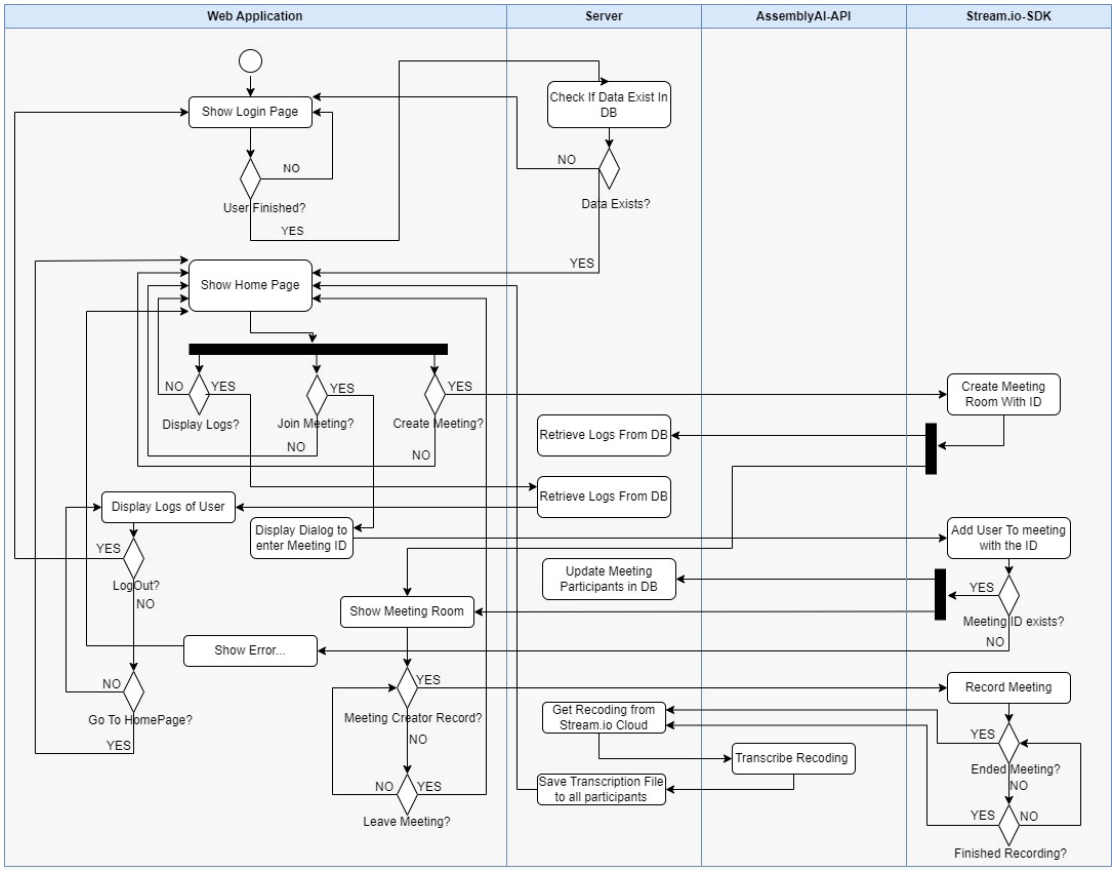
6.3.2. Sequence Diagram.

The following sequence diagram shows the logical flow between the user, server, web components and external API's and SDK's.



6.3.3. Activity Diagram.

The following activity diagram shows the whole process from when the user opens the web application and until he finishes doing all what he wants in the application.



7. Verification and Evaluation.

- We will evaluate our web application based on its ability to capture high quality recordings and sending them appropriately for AssemblyAI API, and writing the transcription as an organized pdf file and saving the generated file for each participant.
- Due to the nature of our project architecture and development process, we will perform testing on 4 modules to ensure right functionalities and good user serving. The modules are: Web application, server AssemblyAI API and stream.io SDK.

Test	*Module*	*Tested Function*	*Expected Result*
1	Web application	Page load	First page load<2s
2	Web application	UI	Responsive and easy to use.
3	Web application	Logs page	Correctly show files and enable user to perform filtering to results.
4	Web application	display meeting	Correctly display the meeting room.
5	Web application	Join meeting	Display pop out asking for meeting ID.
6	Web application	Log out	Correctly navigate to login page.
7	Web application	Sign in	Correctly navigate to homepage.
8	Server	Login	Retrieve correct result from DB.
9	Server	Sign-Up	Correctly insert new user to DB.
10	Server	Navigation	Correct navigation between routes.

11	Server	Update participants	Correctly update participant for specific meeting.
12	Server	Transcription file saving	Correctly save and structure transcription files in DB for each participant.
13	AssemblyAI-API	Transcription returning	Correctly returns transcription for the sent recording.
14	Stream.io SDK	Creating meeting	Correctly create meeting room connection with specific ID.
15	Stream.io SDK	Meeting recording	Correctly record the meeting audio.
16	Stream.io SDK	Recording sending	Correctly send the recording to the server.

8. Resources.

- On speech Recognition Algorithms Shaun V. Ault, Rene J. Perez, Chloe A. Kimble, and Jin Wang, International Journal of Machine Learning and Computing, Vol. 8, No. 6, December 2018.
- A review of Deep Learning Techniques for Speech Processing AMBUJ MEHRISH, Singapore University of Technology and Design, Singapore NAVONIL MAJUMDER, Singapore University of Technology and Design, Singapore RISHABH BHARDWAJ, Singapore University of Technology and Design, Singapore RADA MIHALCEA, University of Michigan, USA SOUJANYA PORIA, Singapore University of Technology and Design, Singapore.
- A Real-Time End-To-end Multilingual Speech Recognition Architecture Javier Gonzales-Dominguez, Member, IEEE, David Eustis, Ignacio Lopez-Moreno, Member, IEEE, Andrew Senior, Senior Member, IEEE, Françoise Beaufays, Senior Member, IEEE, and Pedro J. Moreno, Senior Member, IEEE.
- SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORKS Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton Department of Computer Science, University of Torento.
- Qifeng Zhu, Barry Chen, Nelson Morgan, and Andreas Stolcke, “Tandem connectionist feature extraction of conversational speech recognition,” in International Conference on Machine Learning For Multimodal Interaction, Berlin, Heidelberg, 2005, MLMI’04, pp. 223-231, Springer-Verlag.
- Towards End-to-End Speech Recognition with Recurrent Neural Networks Alex Graves, Google DeepMind London, United Kingdom.
- Using Speech Recognition for Real-Time Captioning and Lecture Transcription in the Classroom Rohit Ranchal, Member, IEEE, Teresa Taber-Doughty, Yiren Guo, Keith Bain, Heather Martin, J. Paul Robinson, Member, IEEE, and Bradley S. Duerstock.
- Trends and developments in automatic speech recognition research Douglas O’Shaughnessy INRS-EMNT (University of Quebec), Montreal, Quebec H5A 1K6, Canada.

- Conformer-1: Robust ASR via Large-Scale Semisupervied Bootstrapping Kevin Zhang* , Luka Chkhetiani* , Francis McCann Rmirez* , Yash Khare, Andrea Vanzo, Michael Liang, Sergio Ramirez Martin, Gabriel Oexle, Roben Bousbib, Taufiquzzman Peyash, Michael Nguyen, Dillon Pulliam, Domenic Donato AssemblyAI Inc.