2022 - 2023

# GRADUATION PROJECT

## NATIONAL ENGINEERING DEGREE

SPECIALTY : TWIN

# Enhancing Workplace Efficiency: Developing AI-Driven Chatbot and Real-Time Transcription Solutions

**By:** *OUHIBI Mohamed Charafeddine*

**Academic supervisor:** *Mme. HLEL Jihene*

**Corporate Internship Supervisor:** *Mr. GHOMRI Ilyes*

Je valide le dépôt du rapport PFE relatif à l'étudiant nommé ci-dessous / I validate the submission of the student's report:

- Nom & Prénom /Name & Surname : Ouhibi Mohamed Charafeddine

## Encadrant Entreprise/ Business site Supervisor

- Nom & Prénom /Name & Surname : Ilyes Ghomri

**Cachet & Signature / Stamp & Signature**

Sté ACCRETIO TUNISIE
Rue
Lac Leman
Imm, OMRANE
Les Berges du Lac
- 1053 -
MF: 1602166D/A/M/000 - Tél: 71 860 337

## Encadrant Académique/Academic Supervisor

- Nom & Prénom /Name & Surname : Jihen Hlel

**Signature / Signature**

Ce formulaire doit être rempli, signé et **scanné**/This form must be completed, signed and **scanned**.

Ce formulaire doit être introduit après la page de garde/ This form must be inserted after the cover page.

# Dedication

*I dedicate this work:*

*To my parents, who have provided all the support i needed to achieve success. Your belief in my ability continues to inspire me to reach for the stars. Your wisdom and strength are the beacon that lights up my path.*

*To my brother, who has been a source of comfort and encouragement. Your words of advice and encouragement fill me with determination.*

*To my friends, for their unyielding support and understanding. Your companionship and humor have kept me afloat.*

*With absolute gratitude to all who have helped me throughout this journey.*

**Mohamed Charafeddine Ouhibi**

# Acknowledgments

The successful completion of this project would not have been possible without the support and contributions of several individuals and organizations. I would like to express my heartfelt gratitude to all those who have played a role in making this endeavor a success.

First, I would like to express my gratitude to my academic supervisor **Hlel Jihene** for her guidance and support throughout this project.

I would also like to thank the team at **Accretio** for offering me the opportunity to undertake this internship, which has taught me so much. Special thanks to **Mr. Ghomri Ilyes**, my supervisor at Accretio, for all the assistance he provided.

Finally, my heartfelt thanks goes to my family and friends who have helped me tremendously during this project, and a big thank you to all my professors at **ESPRIT**. You have my deep gratitude.

# Contents

# List of Figures

# Abbreviations List

- **AI** : Artificial Intelligence

- **ASR** : Automatic Speech Recognition

- **HR** : Human Resources

- **RAG** : Retrieval-Augmented Generation

- **RTX** : Ray Tracing Texel eXtreme

- **vLLM** : Virtual Large Language Model

- **API** : Application Programming Interface

- **UI** : User Interface

- **GPU** : Graphics Processing Unit

- **VRAM** : Video Random Access Memory

- **PDF** : Portable Document Format

- **NoSQL** : Not Only Structured Query Language

- **OVH** : OVHcloud (Hosting Provider)

- **JIRA** : Project Management Tool by Atlassian

- **NLU** : Natural Language Understanding

- **H2O GPT** : AI Framework for GPT-Based Tasks

- **LLaMA** ：Large Language Model Meta AI

- **JSON** ：JavaScript Object Notation

# General Introduction

### Purpose

In today's fast-paced work environment, artificial intelligence has become an essential tool for streamlining operations and enhancing productivity. AI technologies now enable companies to save time and resources by automating processes that once required significant human effort. The ability of AI to analyze, interpret, and provide relevant information quickly has brought about a new level of efficiency, especially in fields where documentation and communication are critical.

The internship project I embarked on focused on addressing key challenges in the areas of **Human Resources (HR) information access** and **meeting management**. New users interacting with complex HR management solutions often struggle to find the information they need, as it's embedded within lengthy user guides and documentation. Facilitating easier access to HR resources can help users get familiar with the system faster, improving their overall experience and productivity. Additionally, meetings play a crucial role in communication within any organization, but they often suffer from issues like interruptions, overlapping conversations, and extended durations. This project aimed to solve these problems with AI-based solutions that provide quick responses and organize meeting information efficiently for later reference.

### Problem Statement

**User guides and other HR documents**: These documents can be lengthy, dense, and challenging to navigate, which makes it difficult for employees to find the specific information they need in a timely manner. This can lead to frustration and a steeper learning curve, especially for new users of the HR system. An AI-powered chatbot like peopleASK could serve as an efficient, user-friendly interface for accessing information without needing to read through extensive documentation.

**Meetings**: Meetings often involve interruptions, overlapping conversations, or lengthy discussions that can hinder clarity. Attendees may find it difficult to follow the conversation or to recall key points after the meeting ends, particularly when discussions are complex or multifaceted. By introducing a live transcription service, these meetings can be summarized accurately, providing participants with an easy-to-review record that captures the main points and decisions made.

### Objectives

To address these challenges, this project established clear goals for each system:

- **peopleASK**: The primary objective of peopleASK is to improve access to HR information by enabling users to quickly find answers to their questions about the company's HR solution. By offering instant, contextually relevant responses, peopleASK helps users navigate the HR system more efficiently, potentially reducing the time and frustration involved in searching through traditional documentation.
- **Transcription Service**: The goal of the transcription service is to provide a reliable, real-time summary of meeting discussions. This service aims to capture spoken content accurately, organize it in a structured format, and allow users to review it post-meeting. This can save time for participants and improve communication efficiency by minimizing misunderstandings or missed information during meetings.

# Chapter 1

# General Presentation

**Plan**

# Introduction

This chapter provides foundational context by presenting an overview of the organization where this work took place, along with its core operational domains. Additionally, we outline the central problems in which we work on, and detail our strategic approach to addressing them.

## 1.1 Host organization

### 1.1.1 Company overview

Accretio is a forward-thinking technology company focused on streamlining Human Resources (HR) processes through digital transformation. Known for its comprehensive, integrated HR solutions, Accretio is dedicated to enhancing how organizations manage employee data, performance tracking, and administrative processes. Positioned as an enabler of HR innovation, the company leverages its expertise to provide tools that help companies create more efficient and dynamic work environments.

Figure 1.1: Company LOGO

### 1.1.2 Provided Services

Accretio offers a suite of HR management tools aimed at simplifying and automating essential HR functions. Their platform covers areas such as employee information management, performance monitoring, payroll, and benefits administration. Through its customizable and user-centric approach, Accretio enables organizations to optimize resource allocation and improve the overall employee experience, which is critical in today's competitive talent market.

## 1.2 Study of the Existing

### 1.2.1 Onboarding and Documentation Access

Accretio supports user onboarding for its HR management platform, *peopleYou*, through a set of resources designed to help users navigate the platform's features and understand core HR processes. New users receive access to comprehensive user guides, training manuals, and supporting documentation that cover a range of HR functionalities, such as performance tracking, employee information management, and policy guidelines.

These materials are structured to give users a thorough understanding of the platform, and they are tailored to address common HR workflows managed within *peopleYou*.

### 1.2.2 Meeting Management and Reporting

For team coordination and meeting management, Accretio employs *peopleMap*, a solution designed to streamline the organization of meetings and support collaborative decision-making. Within *peopleMap*, users can schedule meetings, set agendas, assign roles, and track attendance, making it a versatile tool for structured team interactions.

During meetings, designated attendees often take on the role of recording discussion points and noting action items, creating a record for later review. After the meeting, one participant typically compiles these notes into a summary and shares them with attendees, allowing for follow-up on key tasks and decisions. By using *peopleMap*, Accretio facilitates a consistent and organized approach to meetings, helping teams stay aligned on objectives and responsibilities.

## 1.3 Critic of the Existing

### 1.3.1 Onboarding and Documentation Access

The reliance on lengthy user manuals and HR-led training for understanding Accretio's HR solution can be time-consuming and inefficient. New users may find the information overwhelming, and valuable time is spent locating specific sections relevant to their roles or immediate needs. This process could be simplified with more intuitive access to information.

### 1.3.2 Meeting Management and Reporting

The manual approach to note-taking and summarization is prone to errors and omissions, particularly in meetings with multiple participants and frequent interruptions. This can result in incomplete records, leading to misunderstandings or missed actions. Without automated recording and transcription, the current process does not provide attendees with immediate or easily accessible recaps of discussions, diminishing the efficiency and effectiveness of meetings.

## 1.4 Proposed Solution

To address Accretio's challenges in HR information access and meeting management, two solutions were developed: peopleASK, an AI-powered chatbot for streamlined HR support, and a live transcription service for real-time meeting summaries in peopleMap.

### 1.4.1 peopleASK: AI Chatbot for HR Support

peopleASK uses an advanced language model and Rasa intent recognition to make HR information from peopleYou more accessible. By allowing users to ask questions directly instead of browsing extensive documentation, peopleASK simplifies onboarding and supports quicker, more personalized responses to user inquiries.

### 1.4.2 Live Transcription Service: Real-Time transcription and Summaries

The live transcription service in peopleMap captures and displays meeting content in real time, improving focus by reducing the need for manual note-taking. This service provides attendees with an accurate, accessible meeting record, facilitating follow-up and ensuring critical information is readily available post-meeting.

## 1.5 Working Methodology

The success of any software development project heavily relies on the implementation of an appropriate project management methodology. This section details our approach to project management, the reasoning behind our methodological choices, and the tools we employed to ensure efficient workflow management throughout the development cycle.

### 1.5.1 Agile Methodology

Agile methodology represents a set of principles and practices that promote adaptive planning, evolutionary development, and continuous improvement. Unlike traditional waterfall approaches, Agile embraces change and encourages frequent reassessment of direction throughout the project lifecycle. This iterative approach allows teams to:

- Deliver value incrementally and frequently

- Respond quickly to changing requirements
- Emphasize working solutions over comprehensive documentation

### 1.5.2 Chosen Methodology

When selecting our project management approach, we carefully evaluated two popular Agile frameworks: Scrum and Kanban. Below is a comparative analysis of both methodologies:

Table 1.1: Comparison of Agile methodologies

| **Aspect** | **Kanban** | **Scrum** |
|---|---|---|
| Time Constraints | Fixed-length sprints (typically 2-4 weeks) | Continuous flow, no fixed time boxes |
| Roles | Defined roles (Scrum Master, Product Owner, Development Team) | No prescribed roles |
| Work Units | Sprint backlog with fixed commitments | Continuous flow of work items |
| Changes | Changes wait for next sprint | Changes can be made at any time |
| Planning | Regular sprint planning meetings | Planning on demand |
| Metrics | Velocity, burndown charts | Lead time, cycle time, WIP limits |
| Board Reset | Reset after each sprint | Persistent, continuous flow |
| Work in Progress | Limited by sprint capacity | Limited by explicit WIP limits |

After careful consideration, we opted for Kanban as our primary project management methodology. This choice was influenced by several factors:

1. **Flexibility**: Our project requirements were expected to evolve significantly throughout development, making Kanban's fluid nature more suitable than Scrum's fixed sprint structure.
2. **Continuous Delivery**: The project's nature demanded continuous delivery of features, which aligned perfectly with Kanban's continuous flow model.
3. **Visual Management**: Kanban's emphasis on visual workflow management provided better visibility into bottlenecks and process improvements opportunities.

### 1.5.3 Kanban Board

The Kanban board serves as the central visualization tool for our workflow management. Our implementation consists of columns representing different stages of work:

- **Backlog**: Contains all upcoming tasks
- **To Do**: Tasks selected for immediate attention
- **In Progress**: Currently active tasks
- **Review**: Tasks undergoing testing or peer review
- **Done**: Completed tasks

Each column has specific Work in Progress (WIP) limits to prevent bottlenecks and maintain efficient flow. Tasks move from left to right across the board, providing a clear visual representation of work status and progress.

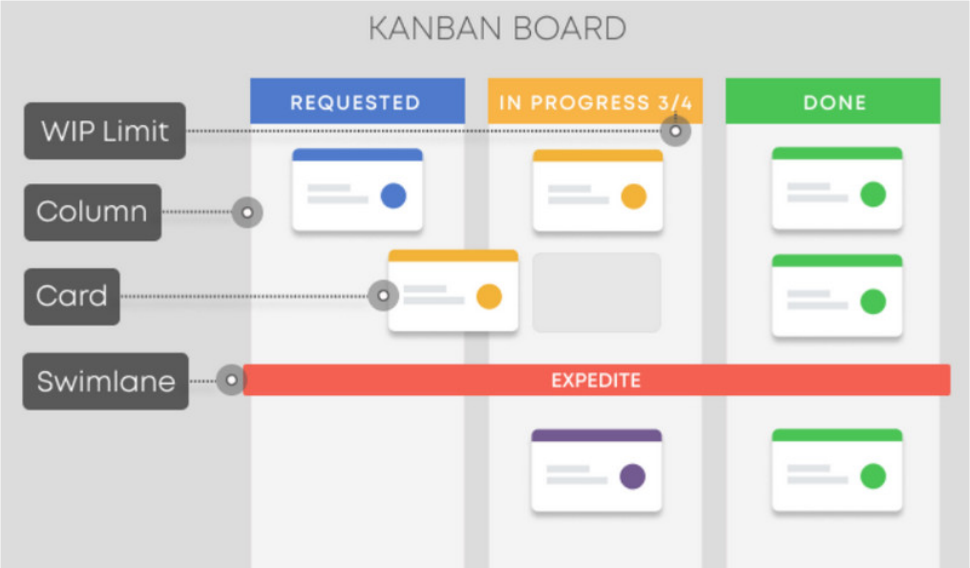


Figure 1.2: Kanban Board

### 1.5.4 Jira

For implementing our Kanban methodology, we utilized Jira, Atlassian's industry-standard project management tool. Jira's robust features particularly suited our needs in the following areas:

### Ticket Management

- **Detailed Issue Tracking**: Each task is created as a ticket with comprehensive information including description, acceptance criteria, and attachments
- **Custom Fields**: Ability to add specific fields relevant to our project needs
- **Priority Levels**: Clear indication of task urgency and importance
- **Time Tracking**: Built-in functionality to monitor time spent on tasks

### Kanban Board Implementation

Jira's Kanban board functionality provided several key advantages:

- **Customizable Workflows**: Ability to tailor columns and states to match our specific process
- **WIP Limits**: Visual indicators when work in progress exceeds defined limits
- **Swimlanes**: Organization of tasks by categories or team members
- **Automation Rules**: Automatic ticket transitions based on defined triggers
- **Real-time Updates**: Immediate visibility of task status changes across the team

The combination of Jira's powerful features with our Kanban methodology created an efficient and transparent project management environment that significantly contributed to the project's success.

# Conclusion

In this chapter, we have described the general context of the project and organized the issues faced by the company, the proposed solutions, and the adopted methodology.

# Chapter 2

# Description of Proposed Work

**Plan**

# Introduction

This chapter outlines the issues encountered with Accretio’s current HR and meeting management systems, describes the functional and non-functional requirements for *peopleASK* and the transcription service, and explains the reasoning behind each technology choice.

## 2.1 Problem Definition

Accretio's existing systems rely heavily on comprehensive user guides and manual meeting note-taking. However, accessing specific HR information from extensive documentation is time-consuming, particularly for new employees. Likewise, manually recorded meeting notes are often inconsistent and do not provide real-time support. This project aims to enhance the user experience by providing an intelligent chatbot (*peopleASK*) and a real-time transcription service to streamline information access and improve meeting comprehension.

## 2.2 Functional and Non-Functional Requirements

### 2.2.1 Functional Requirements

**peopleASK**

- **Question Answering:**

  Provides HR and workplace-related answers from company documentation.

- **Memory Retention:**

  Retains conversational context to maintain coherence.

- **Document Injection:**

  Accept and process documents in various formats like PDF, Word, and Excel.

- **Natural Language Understanding :**

  Accurately identify user needs/intentions based on input.

- **User Interface:**

  Provides an accessible, intuitive UI within Accretio's peopleYou solution.

**Transcription Service**

- **Real-Time Transcription:**

  Transcribes audio in real time during meetings.

- **Meeting Summarization:**

  Summarizes discussion points to highlight key information.

- **Multilingual Support:**

  Enables transcription in multiple languages as needed.

- **Speaker Identification:**

  Differentiates between speakers for clarity in transcriptions.

### 2.2.2 Non-Functional Requirements

- **Scalability:**

  Allows for system growth as user demand increases.

- **Performance:**

  Ensures responsive and timely service delivery.

- **Security:**

  Protects sensitive data in line with data privacy standards.

## 2.3 Technology Choices

This section explains the technology choices made for each solution, highlighting their advantages over alternative options and aligning with project needs.

### 2.3.1 peopleASK Technologies

- **H2O GPT Framework :**
  Used primarily as a structured, reliable framework for document handling and prompt management, *H2O GPT* was chosen over more flexible options like *LangChain* due to its simplicity and robustness. While *LangChain* offers greater flexibility, *H2O GPT* met our needs with a straightforward setup that easy integration.
- **Rasa for Intent Recognition :**
  Rasa, an open-source intent recognition platform, was selected for its customization and easy integration into *peopleASK*. Compared to alternatives like Dialogflow or Microsoft LUIS, Rasa provided better community support and was more adaptable, allowing quick tuning to match Accretio's specific HR needs without incurring additional licensing fees.
- **Zephyr-7B Quantized LLM :**
  The Zephyr-7B model was chosen for its light, quantized design, which allowed it to run on Accretio's available GTX 1660 GPU. Larger models such as *LLaMA-2* or *Falcon-40B* offered improved accuracy but would have required more powerful hardware. Zephyr provided acceptable performance within our budgetary and hardware constraints.

### 2.3.2 Transcription Service Technologies

- **Whisper ASR :**
  OpenAI's *Whisper ASR* was selected for its open-source accessibility and robust multilingual transcription capabilities, making it ideal for *peopleMap*'s diverse user base. While solutions like Google Speech-to-Text and Amazon Transcribe provided quality alternatives, Whisper's flexibility and privacy controls better aligned with Accretio's deployment needs.

### 2.3.3 Shared Technologies

- **MongoDB for NoSQL Storage :**

  MongoDB's flexible, schema-less structure made it a suitable choice for handling *peopleASK* query data and transcription metadata. Other NoSQL options like Firebase or DynamoDB were considered but lacked MongoDB's extensive customizability, which was necessary for HR and meeting data needs.

- **Docker for Containerization :**

  Docker was chosen to containerize both solutions, enabling seamless deployment and integration across Accretio's infrastructure. While Kubernetes could offer advanced orchestration, Docker's simplicity made it ideal for this project's moderate scaling requirements and the timeline constraints of the internship.

# Conclusion

This chapter has outlined the existing challenges with Accretio's HR information access and meeting management and the proposed work to address these issues through peopleASK and the live transcription service. It has also detailed the functional and non-functional requirements and explained our rationale for the technology choices.

The following chapter will explore the implementation details, elaborating on the technical decisions, development challenges, and adjustments made to bring these solutions into practical operation.

# Chapter 3

# Developing the peopleASK Chatbot

**Plan**

# Introduction

In this chapter, we delve into the development journey of peopleASK, a chatbot designed to provide HR-related assistance. The chapter outlines the technical challenges faced during different phases of the project, from the initial setup of the conversational AI to the final deployment of a performant, user-friendly solution. The following sections explore the iterative process of model selection, infrastructure upgrades, architectural innovations, and feature implementations, highlighting key solutions that shaped peopleASK into a functional tool.

## 3.1 Leveraging h2oGPT's Document Management for RAG

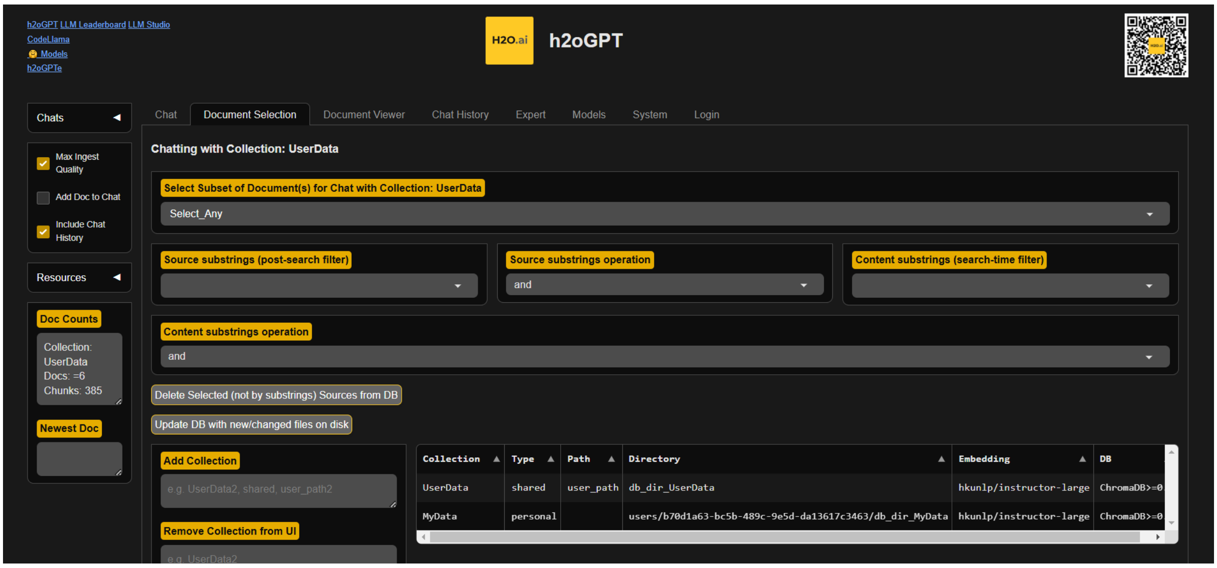
One of the core strengths of the **peopleASK chatbot** lies in its ability to provide precise and contextually relevant responses by leveraging **h2oGPT's integrated document management functionality**. This feature forms the backbone of the **Retrieval-Augmented Generation (RAG)** approach, enabling the system to extract information from uploaded documents and tailor responses to user queries.

### 3.1.1 Document Upload and Processing

The process begins with document ingestion through **h2oGPT's intuitive web interface**. Users upload relevant documents, such as user manuals, HR policies, or training guides, directly into the system. These documents are then preprocessed to ensure compatibility with the underlying **text embeddings model**. The preprocessing includes tasks like:

- Extracting text content from various document formats (e.g., PDFs, Word documents).
- Removing non-textual elements, such as images or unsupported characters.
- Splitting the text into manageable chunks to facilitate efficient embedding.

This preprocessing step ensures that the content is properly structured for downstream embedding and retrieval tasks.


h2oGPT LLM Leaderboard LLM Studio
CodeLlama
Models
h2oGPTe
H2O.ai
h2oGPT
Chats
Resources
Chat
Document Selection
Document Viewer
Chat History
Expert
Models
System
Login
Delete Selected (not by substrings) Sources from DB
Update DB with new/changed files on disk
Add Collection
Remove Collection from UI

☑ Max Ingest Quality

□ Add Doc to Chat

☑ Include Chat History

Doc Counts: Collection: UserData Docs: =6 Chunks: 385

Newest Doc:

Chatting with Collection: UserData

Select Subset of Document(s) for Chat with Collection: UserData: Select_Any

Source substrings (post-search filter):

Source substrings operation: and

Content substrings (search-time filter):

Content substrings operation: and

Add Collection: e.g. UserData2, shared, user_path2

Remove Collection from UI: e.g. UserData2

| Collection | Type | Path | Directory | Embedding | DB |
|---|---|---|---|---|---|
| UserData | shared | user_path | db_dir_UserData | hkunlp/instructor-large | ChromaDB>=0 |
| MyData | personal | | users/b70d1a63-bc5b-489c-9e5d-da13617c3463/db_dir_MyData | hkunlp/instructor-large | ChromaDB>=0 |

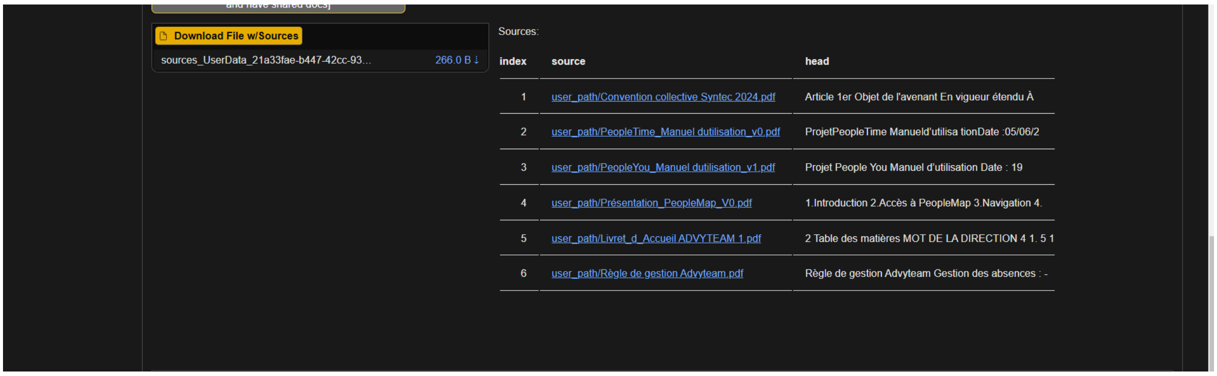
Figure 3.1: h2oGPT graphical interface

### 3.1.2 Text Embeddings and Storage in ChromaDB

Once the documents are uploaded, the content is passed through a **text embeddings model** integrated into h2oGPT. This model generates high-dimensional vector representations for each chunk of text, capturing its semantic meaning. These embeddings are then stored in **ChromaDB**, a high-performance vector database optimized for managing and querying large volumes of embedding data.

Key advantages of using ChromaDB include:

- **Efficient Search**: The database supports fast, approximate nearest-neighbor searches, which is critical for real-time query responses.
- **Scalability**: It can handle a growing volume of documents without a significant drop in performance.
- **Custom Metadata Storage**: Alongside embeddings, ChromaDB stores metadata (e.g., document source, section titles), enhancing the contextual richness of search results.



Sources:

| index | source | head |
|---|---|---|
| 1 | user_path/Convention collective Syntec 2024.pdf | Article 1er Objet de l'avenant En vigueur étendu À |
| 2 | user_path/PeopleTime_Manuel dutilisation_v0.pdf | ProjetPeopleTime Manueld'utilisa tionDate :05/06/2 |
| 3 | user_path/PeopleYou_Manuel dutilisation_v1.pdf | Projet People You Manuel d'utilisation Date : 19 |
| 4 | user_path/Présentation_PeopleMap_V0.pdf | 1.Introduction 2.Accès à PeopleMap 3.Navigation 4. |
| 5 | user_path/Livret_d_Accueil ADVYTEAM 1.pdf | 2 Table des matières MOT DE LA DIRECTION 4 1. 5 1 |
| 6 | user_path/Règle de gestion Advyteam.pdf | Règle de gestion Advyteam Gestion des absences : - |

Figure 3.2: ChromaDB document content

### 3.1.3 Query Processing and Information Retrieval

When a user submits a query to the chatbot, h2oGPT performs the following steps to generate an accurate response:

1. **Query Embedding**: The user's input is embedded using the same text embeddings model used during document ingestion.
2. **Vector Search in ChromaDB**: The query embedding is matched against the stored embeddings in ChromaDB to identify the most relevant text chunks.
3. **Contextual Retrieval**: The retrieved text chunks are forwarded to the language model as part of the prompt, providing the necessary context for generating an informed response.

This architecture ensures that peopleASK delivers answers grounded in the uploaded documents, minimizing the risk of hallucinations and maximizing relevance. Furthermore, the modular nature of the system allows for continuous updates to the document database, ensuring that users always have access to the latest information.

## 3.2 Initial Model Selection and Configuration

The starting point for peopleASK was selecting a foundational model suitable for its requirements. Given hardware limitations, the zephyr-7b-gguhf model from the huggingface ecosystem was chosen for its ability to offload some processing to the CPU. The initial setup was as follows:

- **CPU:** Intel Xeon E5-2430
- **GPU:** GTX 1660 12GB

To address these issues, the focus shifted to quantized models like zephyr-7b awq. This approach reduced memory demands, enabling the model to run on the available GTX 1660. However, early tests revealed significant drawbacks, including low response quality marked by irrelevant or repetitive outputs.

This phase highlighted the urgent need for better infrastructure. The company provided an alternative machine equipped with an AMD Ryzen 5 8600G CPU. This allowed inference on the zephyr-7b GGUHF model, leading to moderate improvements in quality at the cost of performance.

## 3.3 The Challenge of Streaming Responses

One of the earliest hurdles in developing peopleASK was dealing with the limitations in how responses were handled. The original architecture depended on Rasa's NLU application and the Rasa action server, which interacted through REST APIs. While this setup worked for static responses or actions triggered by intent recognition, it posed a significant issue for streaming responses. REST, being inherently stateless and response-driven, was not designed to facilitate a continuous flow of data between a server and client. This limitation became apparent when attempting to stream responses from h2oGPT, a feature critical to enhancing user experience. [old architecture diagram goes here] This rigid system failed to support the dynamic nature of streaming responses, making it clear that a different approach was necessary.

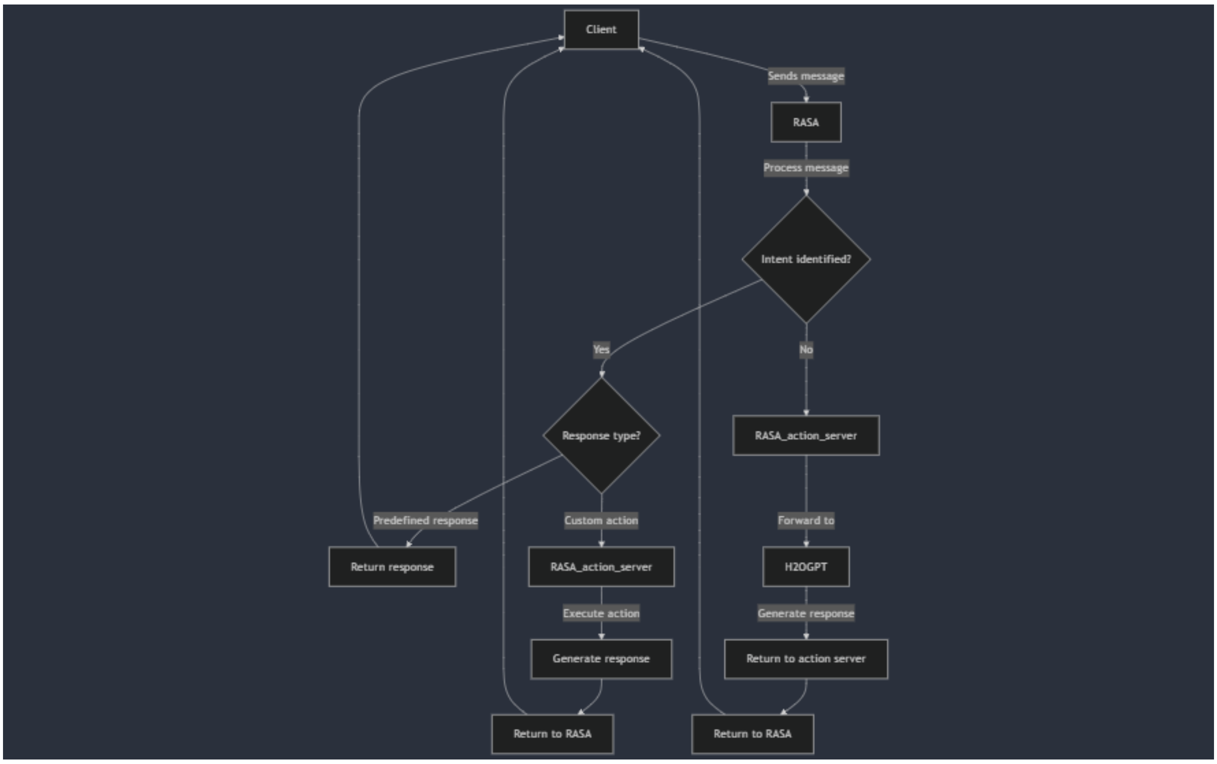


Figure 3.3: Initial peopleAsk architecture flowchart

### 3.3.1 Rethinking the Architecture

To overcome this obstacle, I proposed and implemented a revised architecture. The key idea was to introduce a WebSocket server as an intermediary between the user interface and h2oGPT. This architectural shift allowed responses to be delivered in real time while preserving compatibility with Rasa for intent recognition. [new architecture diagram goes here] This modification decoupled Rasa and h2oGPT, enabling smooth streaming responses while maintaining intent recognition capabilities.

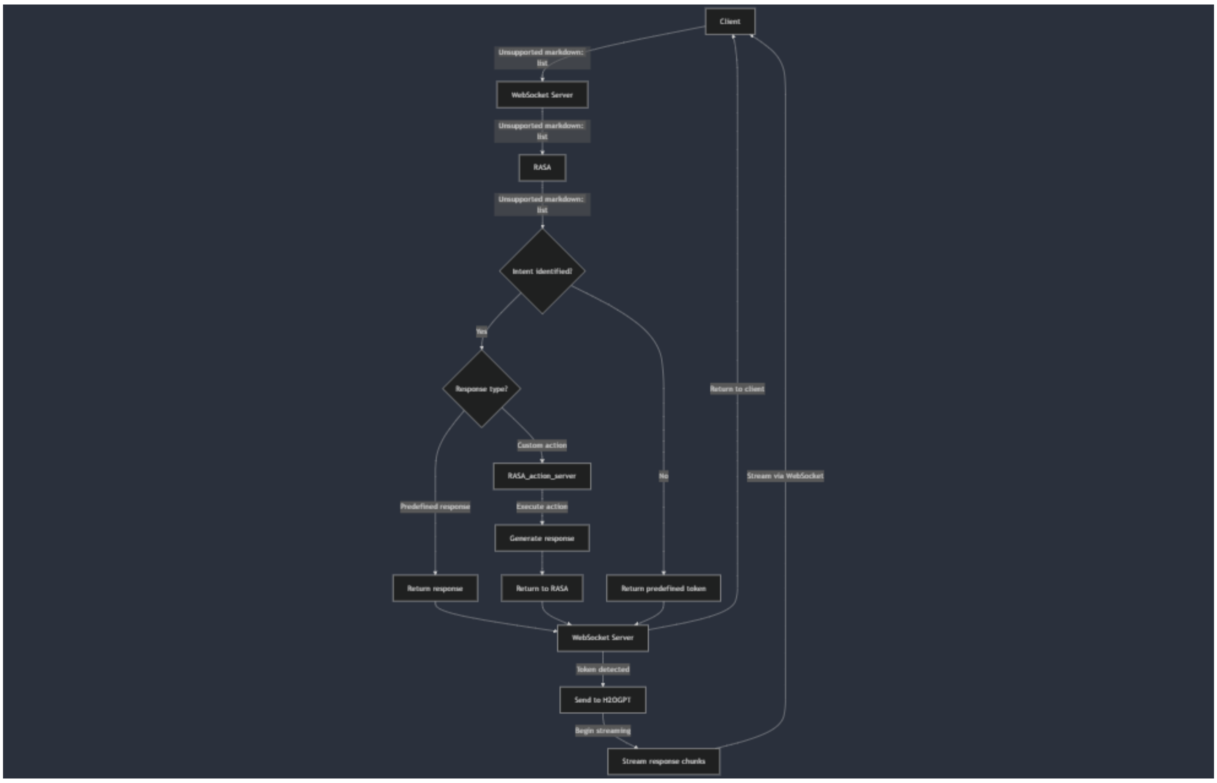


Figure 3.4: Revised peopleAsk architecture flowchart

### 3.3.2 Building a Custom Angular Interface

The existing client interface was designed exclusively to interact with Rasa, and it lacked support for the WebSocket-based communication required by the new architecture. To address this, I developed a custom Angular-based web component, packaged for easy integration into existing online platforms. **Key Features of the Angular Component**:

- **WebSocket Integration**: Seamless interaction with the new server for streaming responses.
- **Authentication**: Leveraged Keycloak for user verification. Guest users were supported with limited functionality.
- **Customizable UI**: Allowed easy incorporation of user details, such as profile pictures and names, using provided props.
- **Portability**: Packaged as a JavaScript file, making it simple to integrate into any webpage using a `<script>` tag.

Implementation required minimal effort for the end-user:

```
<script src="company.provided.domain/elements.js"></script>
```

```
<ai-chat-bot user="userDetails" kcToken="keycloakToken"/>
```

This modular approach ensured scalability and usability, aligning with the needs of **peopleASK's** diverse deployment scenarios.

### 3.3.3 Advanced Logging and User Management

In parallel with the architectural redesign, I added robust logging and user management features to the WebSocket server. Logs were maintained daily, with automatic deletion of older logs beyond a 30-day threshold. MongoDB was used for storing chat history, ensuring users could retain conversations even after closing the web page.

By addressing these architectural challenges, the project transitioned from a rigid and constrained system to a flexible and scalable one. This transformation not only improved performance but also laid the groundwork for future enhancements.

## 3.4 Enhancing Performance and Scalability with vLLM

As the peopleASK project evolved, improving performance and scalability has been a priority. Early implementations of the system struggled to meet requirements, with slow response times and an inability to handle multiple requests simultaneously. These challenges highlighted the need for a more efficient inference method, leading to the adoption of the vLLM framework. This transition marked a turning point, significantly enhancing both the speed and capacity of the system.

### 3.4.1 Advantages of vLLM

1. **Enhanced Concurrency**

   Traditional inference methods processed user queries sequentially, limiting scalability and causing delays during peak usage. vLLM's architecture enabled **parallel processing of multiple user requests**, allowing the system to serve concurrent users seamlessly. This feature was critical for scaling the solution to accommodate the company's growing user base.

2. **Increased Token Generation Speed**

   A significant improvement was observed in token generation speed, a core metric for response time. vLLM optimized memory management and GPU utilization, reducing latency during inference. These optimizations allowed the system to deliver responses in near real-time, greatly enhancing user experience.

### 3.4.2 Quantitative Impact

The benefits of transitioning to vLLM were evident in the performance metrics gathered during testing. Token generation speed saw a dramatic increase, significantly reducing response times and making interactions feel seamless. Before vLLM, the system struggled to manage more than one user without severe delays. Post-implementation, the solution handled ten or more simultaneous users with ease, demonstrating its newfound scalability. Additionally, system throughput improved substantially, enabling the system to process a higher volume of queries in less time, which was critical for meeting user demands during peak periods.

## 3.5 Model Evaluation and Testing

The performance and quality of the language models used in peopleASK were critical to ensuring a satisfactory user experience. Early iterations with quantized models, such as Zephyr-7b-AWQ, revealed significant shortcomings, including irrelevant, nonsensical, or repetitive responses. Addressing these issues required systematically evaluating alternative models under controlled conditions.

To achieve this, I utilized an OVH server equipped with a 48 GB VRAM GPU, capable of running more powerful models. A structured testing methodology was devised to assess each model's quality and performance, using metrics designed to capture both user satisfaction and system efficiency.

### 3.5.1 Testing Tool Development

To streamline the evaluation process, I developed a **custom testing tool** that automated the comparison of language models. The tool significantly reduced manual effort and ensured consistent testing procedures.

**Features of the Testing Tool**:

- **Automated Query Submission**:
  - Sent thousands of predefined queries to the models.
  - Saved responses in a **MongoDB database** for easy retrieval and analysis.
- **Response Analysis**:
  - Stored response times and token usage for performance metrics.
  - Compared answers across models to the same query, allowing for side-by-side quality assessment.
- **Visual Performance Metrics**:
  - Generated graphs to visualize token processing rates (tokens/second) under heavy workloads.
  - Identified the fastest and slowest models in terms of throughput.
- **Customizable Metrics**: Allowed fine-tuning of evaluation criteria to suit the unique requirements of **peopleASK**.

### 3.5.2 Evaluation Metrics

The evaluation process focused on both qualitative and quantitative aspects of model behavior. Each model was rated based on the following metrics:

- **Cohesion**:
    - Assessed the logical flow and structure of the response.
    - Measured how smoothly ideas connected within the output.
- **Precision**:
    - Evaluated the factual accuracy of the response.
    - Ensured information matched the source documents provided.
- **Relevance**:
    - Determined how well the response addressed the user query.
    - Penalized tangential or off-topic information.
- **Answer Completeness**:
    - Checked if the response fully answered the query.
    - Highlighted cases where critical details were missing.

Each model was rated on a 10-point scale for each metric, with higher scores indicating better performance.

### 3.5.3 Results and Observations

#### Results

After testing was concluded, the following results were obtained:

| Model | Cohesion | Precision | Relevance | Completeness | Average Score |
|---|---|---|---|---|---|
| **HuggingFaceH4/zephyr-7b-beta** | 7.5 | 7.0 | 7.8 | 7.7 | **7.5** |
| **01-ai/Yi-1.5-9B-Chat** | 3.0 | 2.5 | 3.5 | 3.0 | **3.0** |
| **lmsys/vicuna-13b-v1.5** | 9.0 | 8.8 | 9.0 | 9.2 | **9.0** |
| **lmsys/vicuna-7b-v1.5** | 5.0 | 4.8 | 5.2 | 5.1 | **5.0** |
| **Qwen/Qwen2-7B-Instruct** | 8.5 | 8.2 | 8.8 | 8.7 | **8.5** |

#### Observations

In addition to quality, performance metrics such as **token processing speed** were analyzed. A Python script visualized token throughput during high workloads, yielding the following ranking from fastest to slowest:

1. **Qwen/Qwen2-7B-Instruct**
2. **HuggingFaceH4/zephyr-7b-beta**
3. **lmsys/vicuna-7b-v1.5**
4. **01-ai/Yi-1.5-9B-Chat**
5. **lmsys/vicuna-13b-v1.5**

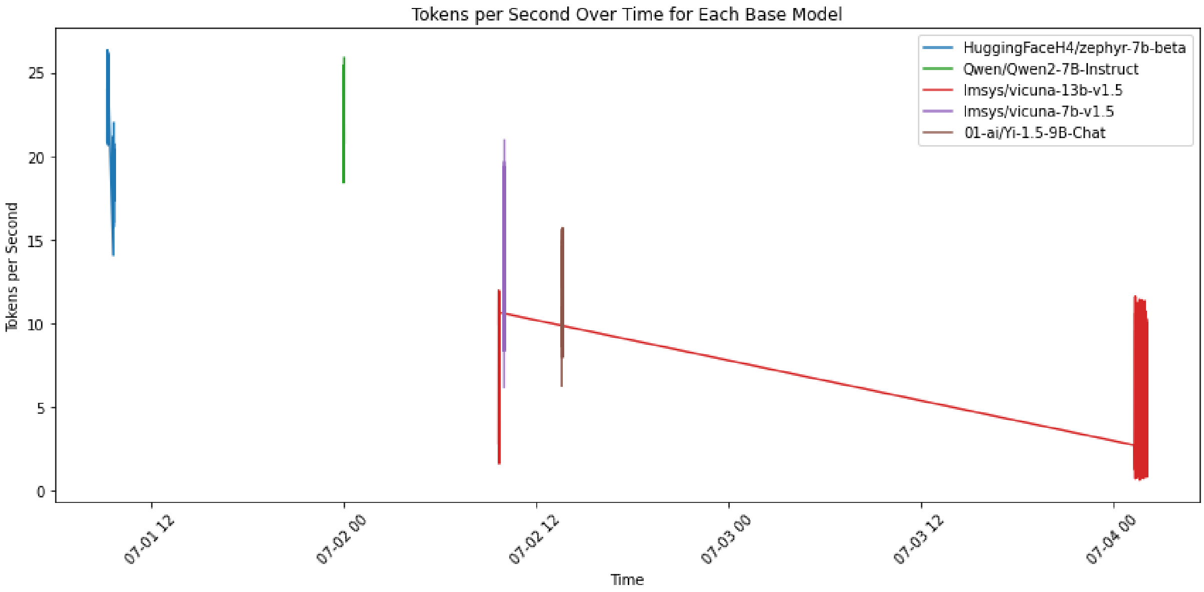


Figure 3.5: LLM performance test results

### 3.5.4 Selection of the Optimal Model

Based on these results, the **Qwen/Qwen2-7B-Instruct** model was chosen as the best fit for **peopleASK**:

- **Quality**: It delivered responses that were cohesive, accurate, relevant, and complete, with an average score of 8.5.
- **Performance**: Outperformed other models in token throughput, ensuring responsiveness even under heavy loads.

### 3.5.5 Deployment on an RTX 3090

The upgrade from the GTX 1660 (12GB VRAM) to the RTX 3090 (24GB VRAM) was a significant milestone in the development of peopleASK. The GTX 1660, while serviceable for initial testing with smaller or quantized models, posed severe limitations in terms of memory and processing power. These constraints not only restricted the size and quality of models that could be deployed but also hindered performance, leading to slower response times and an inability to fully exploit advanced inference techniques.

When the RTX 3090 became available, it presented an opportunity to overcome these barriers. Its vastly superior VRAM and processing capabilities allowed for the deployment of larger, more sophisticated models, including the Qwen2-7B Instruct, which had previously been tested on external hardware. This upgrade marked a turning point, enabling significant enhancements in both model quality and system responsiveness.

### 3.5.6 Performance Improvements

The transition to the RTX 3090 yielded immediate and measurable benefits:

- **Model Size Capability**: The increased memory capacity supported larger, more complex models, such as the Qwen2-7B Instruct, without resorting to aggressive quantization. This allowed the system to provide more accurate and contextually relevant responses.
- **Token Generation Speed**: With the RTX 3090, token generation was accelerated dramatically. Testing showed a significant improvement in speed compared to the GTX 1660, resulting in faster response times and smoother interactions.

- **Concurrent User Support**: The powerful GPU enabled the system to handle more simultaneous users with ease, further improving scalability and reducing latency during peak usage periods.

## 3.6 Integration of an External API for Vacation Credit Queries

One of the final enhancements to **peopleASK** was the integration of an external API to provide personalized answers to user queries regarding their vacation credit balance. This feature required **Rasa** to interact seamlessly with the company's HR database, enabling dynamic responses based on individual user data.

The implementation process involved configuring the **Rasa action server** to authenticate API requests using a token derived from the user's **Keycloak** credentials. If the user was logged in, the system fetched their email from the token and used it to query the vacation credit API securely. For users who were not authenticated, the server returned a predefined response indicating restricted access to this feature.

By leveraging this API, **peopleASK** could deliver highly specific and accurate responses, enhancing its utility for employees. The feature was well-received, demonstrating the potential for integrating additional APIs in the future to expand the system's functionality.

# Conclusion

This chapter has detailed the evolution of the peopleASK project, from early architectural challenges and hardware limitations to model testing, optimization, and feature enhancements. Key milestones, such as the deployment of a custom WebSocket server for streaming responses, adopting vLLM for better concurrency, and upgrading to an RTX 3090 GPU, have significantly improved the system's performance and user experience. The integration of an external API for vacation credit queries showcased the project's versatility and ability to address real-world use cases.

In the next chapter, we will explore the development of a Real-Time Transcription Service, an ambitious component aimed at transcribing and summarizing live conversations within video conferences. This extension builds on the foundations laid by peopleASK, expanding its scope to meet new challenges and opportunities.

# Chapter 4

# Building a Real-Time Transcription Service

**Plan**

# Introduction

This chapter focuses on the development of a real-time transcription service as part of the peopleMap video conferencing solution. The goal was to create an accurate and scalable transcription system capable of handling multiple users simultaneously, integrating summarization capabilities, and providing a polished meeting summary in PDF format. The following sections outline the project's evolution, from leveraging existing solutions to tackling technical challenges and enhancing functionality.

## 4.1 Leveraging Existing Open-Source Solutions

The project began with an exploration of existing tools and frameworks. After some research on GitHub, the *Whisper_Streaming* repository emerged as a promising candidate. This Python-based app simulated the streaming of audio chunks to a Whisper model and included robust output management. Even more valuable was its connection to a research paper detailing filtering and contextual management of Whisper outputs, aligning perfectly with the requirements for live transcription.

A fork of *Whisper_Streaming* offering live transcription via WebSocket was particularly appealing. The system provided an HTML test page, which delivered accurate results. However, a critical shortcoming emerged: it lacked multi-user support, a necessity for integration with the peopleMap solution.

The initial implementation of a processing queue allowed the system to handle multiple users but revealed a new issue—shared context among users. This led to inaccurate and nonsensical outputs. Recognizing the need for a more complex solution, I designed a system architecture based on the following entities:

- **Meeting**: Representing a single transcription session.
- **Room**: Assigned to a single meeting and serving as the container for users.
- **User**: Associated with a room, with one designated as the meeting's *owner*.

This schema allowed each user to maintain an independent context, resolving the shared context issue. Data for ongoing sessions remained in live memory for optimal performance, while meeting details were stored in MongoDB only after the session concluded.

## 4.2 Integration with peopleMap's Front-End

To ensure a seamless user experience, the transcription backend was integrated into the React-based front end of peopleMap. This integration aimed to provide an intuitive interface that enabled users to easily interact with the transcription system while maintaining a natural flow during meetings.

Several key functionalities were added to the front-end interface:

1. **Start and Stop Meeting Buttons**:

   These buttons allowed users to initiate and conclude transcription sessions with minimal effort. The *Start Meeting* button established a WebSocket connection with the backend and initialized the transcription process. The *Stop Meeting* button terminated the session and saved relevant data. This straightforward interaction reduced user friction, ensuring that even non-technical users could access the feature effortlessly.

2. **Custom React Hook for Backend Communication**:

   A custom React hook was developed to handle the WebSocket interaction between the front end and backend. This hook ensured the smooth transmission of audio data and real-time updates from the backend, handling reconnections and errors gracefully. The hook abstracted complex logic, keeping the front-end code clean and maintainable.

3. **Custom Audio Worklet for Microphone Input**:

   The audio worklet processed raw microphone input and divided it into manageable chunks suitable for real-time transcription. This ensured low latency and high fidelity, preserving the natural flow of speech during conversations.

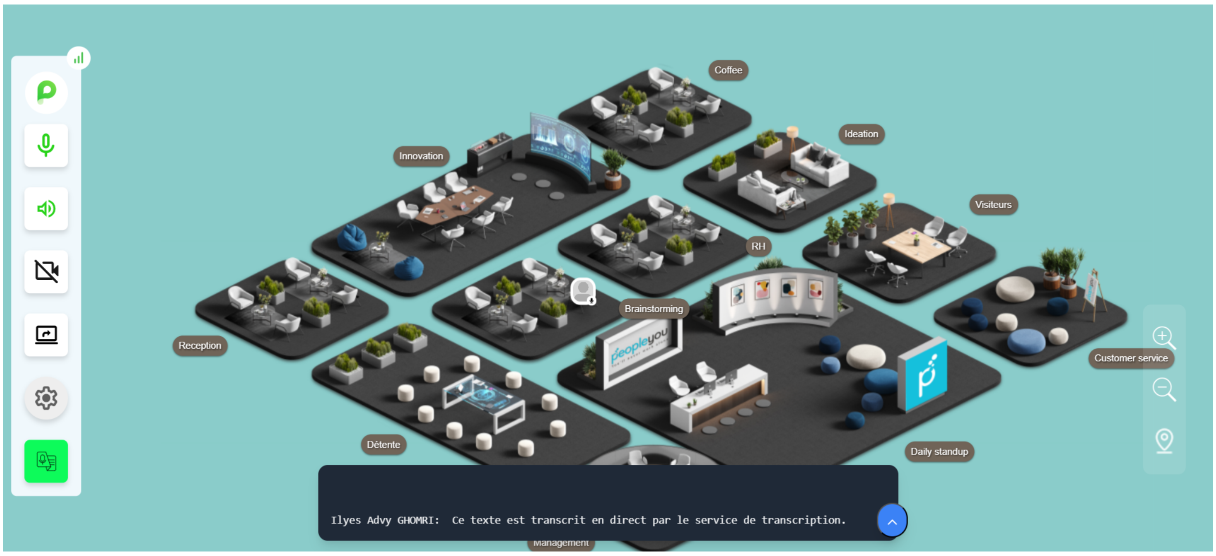
4. **Participant Output Display**:

   A small box was added at the bottom of the screen, displaying each participant's transcribed output. This feature offered several advantages:

   - **Real-Time Feedback**: Users could see their spoken words being transcribed in real time, which provided confidence in the system's functionality.

- **Transparency**: Displaying individual outputs reduced ambiguity about what the system heard and transcribed for each participant.
- **Collaboration**: The feature encouraged accountability and active participation, as all users could verify their contributions to the meeting.

By including this display, the interface catered to both the technical requirements of the transcription service and the practical needs of users during meetings.



Ilyes Advy GHOMRI: Ce texte est transcrit en direct par le service de transcription.

Figure 4.1: Transcription service front end integration

## 4.3 Summarization via Sliding Window Prompting

Once the transcription system was functional, I added a summarization module to the backend, leveraging the vLLM server from the peopleAsk project. The sliding window prompting method was used to ensure accurate summaries of long conversations.

Sliding window prompting works by dividing the transcribed text into overlapping chunks, ensuring continuity between successive prompts. Each chunk receives a summary, and the results are concatenated or further refined into a final output.

Here's an example workflow:

1. Divide text into overlapping chunks (e.g., each chunk contains 75% new content and 25% overlap from the previous chunk).
2. Pass each chunk through the model for summarization.
3. Combine the summaries into a cohesive final result.

This approach mitigates the loss of context common in long text inputs and provides accurate, concise outputs.

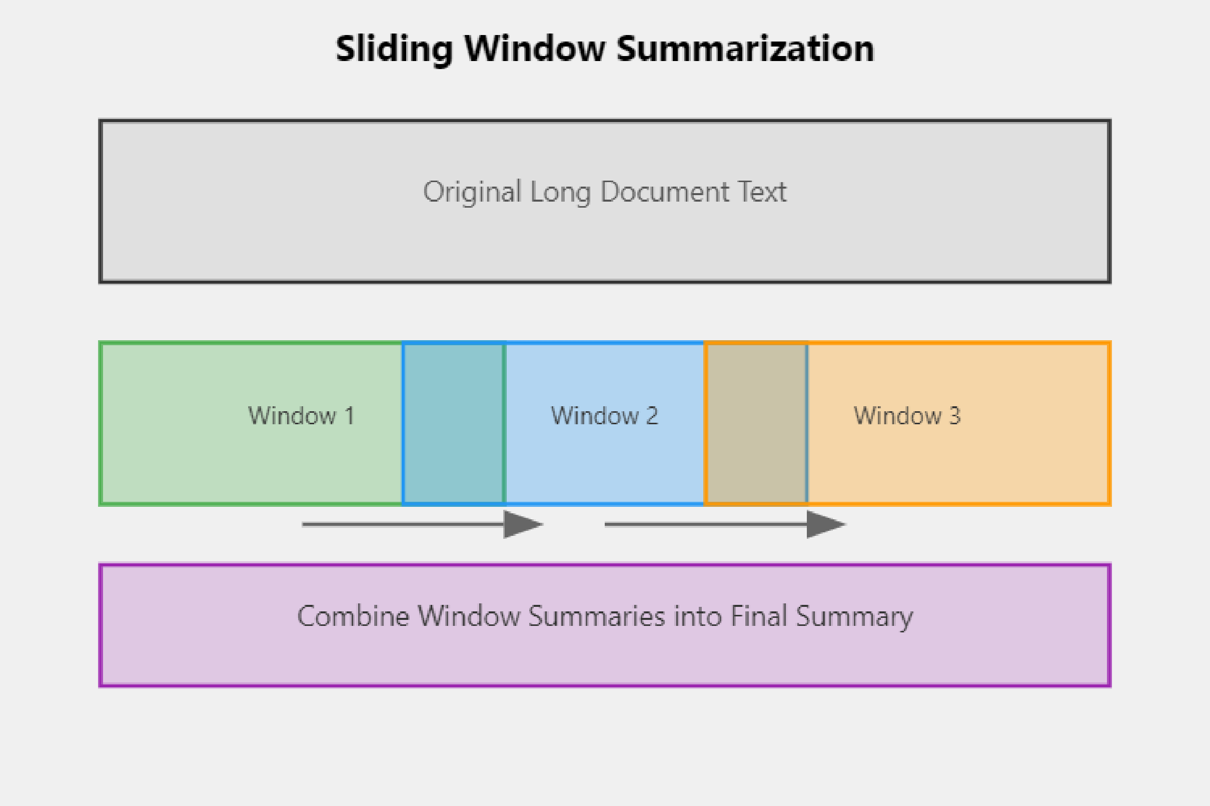

Sliding Window Summarization
Original Long Document Text
Window 1
Window 2
Window 3
Combine Window Summaries into Final Summary

Figure 4.2: Sliding window summarization

## 4.4 Generating and Delivering Summaries

To enhance usability, I developed a PDF emailer module. This system formatted meeting summaries into a professional PDF, including details such as participants, date, time, and duration. The PDF was sent to the meeting *owner* alongside a custom HTML email template.

The backend combined summarization, PDF generation, and email delivery, ensuring users received a polished record of their meetings.

## 4.5 Final Results and Feedback

The real-time transcription service received overwhelmingly positive feedback from colleagues and my supervisor. They expressed enthusiasm and were impressed by its accuracy, multi-user support, and summarization features. The transition from a single-user proof of concept to a robust, multi-user, real-time transcription solution demonstrated the project's success and potential for broader adoption.

## Conclusion

This chapter outlined the development and integration of a real-time transcription service, from its inception using *Whisper_Streaming* to its seamless integration into peopleMap. The journey included solving architectural challenges, implementing advanced summarization techniques, and ensuring a user-friendly experience.

# General Conclusion

In this report, I've shared the journey of developing two key projects during my final year internship: the peopleASK chatbot and the real-time transcription service. Both were designed to make tasks easier and faster for users in a workplace setting.

The **first chapter** introduced the context of the internship, explaining the problems we aimed to solve and the goals we set for these projects.

The **second chapter** outlined the planning phase, focusing on the requirements and technology choices.

The **third chapter** detailed the creation of the peopleASK chatbot, which combined language models and Rasa to answer user queries. It covered the improvements and testing that helped make the chatbot both accurate and reliable.

The **fourth chapter** focused on the real-time transcription service, which transformed audio into live text summaries. It showed how the system was adapted to work with the company's video call platform and handle multiple users effectively.

Each project faced challenges, like optimizing model performance or managing multiple users at once. Solving these issues helped improve the final solutions and taught me valuable lessons along the way. Feedback from my supervisor and colleagues was very positive, with many impressed by the functionality and usefulness of the tools.

While the results are promising, there's always room for improvement. The chatbot could benefit from a better text embedding model and even faster response times, while the transcription service could be made smarter by experimenting with different models and more advanced summarization methods.

This internship was a rewarding experience, helping me grow both technically and professionally. It has given me the skills and confidence to take on future challenges and continue building tools that make a difference.

[1] [2] [3]

# Bibliography

[1] Qwen. Qwen2 technical report. `https://huggingface.co/Qwen/Qwen2-7B-Instruct`, 2024.

[2] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022.

[3] Dominik Macháček. whisper_streaming. `https://github.com/ufal/whisper_streaming`, April 2023. Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

# Résumé

Ce projet a été mené dans le cadre d'un stage chez Accretio et s'est concentré sur la création de solutions innovantes pour améliorer la productivité sur le lieu de travail grâce à des outils pilotés par l'IA. Le travail a porté sur le développement de deux systèmes clés : **peopleASK**, un chatbot tirant parti de Retrieval-Augmented Generation (RAG) et de la reconnaissance d'intention via Rasa, pour simplifier l'accès à la documentation RH ; et un **Service de transcription en temps réel** pour la plateforme de vidéoconférence de l'entreprise, offrant des transcriptions et des résumés de réunions en direct. Ces outils ont été conçus pour relever des défis tels que l'amélioration de l'accès des utilisateurs à l'information et la rationalisation des processus de documentation des réunions. Des technologies de pointe, notamment vLLM, MongoDB, React et Whisper ASR, ont été utilisées pour mettre en œuvre des solutions efficaces, évolutives et conviviales.

**Mots clés** : IA, RAG, Chatbot, Transcription en temps réel, vLLM, Whisper ASR, Accretio, Automatisation

# Abstract

This project was conducted as part of an internship at Accretio and focused on creating innovative solutions to enhance workplace productivity through AI-driven tools. The work involved the development of two key systems: **peopleASK**, a chatbot leveraging Retrieval-Augmented Generation (RAG) and intention recognition via Rasa, to simplify access to HR documentation; and a **Real-Time Transcription Service** for the company's video conferencing platform, offering live meeting transcriptions and summaries. These tools were designed to address challenges such as improving user access to information and streamlining meeting documentation processes. Cutting-edge technologies, including vLLM, MongoDB, React, and Whisper ASR, were used to implement efficient, scalable, and user-friendly solutions.

**Keywords**: AI, RAG, Chatbot, Real-Time Transcription, vLLM, Whisper ASR, Accretio, Automation

ESPRIT SCHOOL OF ENGINEERING

**www.esprit.tn -  E-mail : contact@esprit.tn**
**Siége Social** : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889
**Annexe** : Z.I. Chotrana II - B.P. 160 - 2083 - Pôle Technologique - El Ghazala - Tél. : +216 70 685 685 - Fax. : +216 70 685 454