MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

FACULTY OF AUTOMATION AND COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTMENT

Safer museum guide interaction during a pandemic and further
Using NLP in human interactive museum visits
Museum guide Chatbot

LICENSE THESIS

Graduate: **Laura CEUCA**

Supervisor: **Assoc. Prof. Dr. Eng. Emil Ștefan CHIFU**

2021

MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY OF CLUJ-NAPOCA, ROMANIA

FACULTY OF AUTOMATION AND COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTMENT

DEAN,
Prof. dr. eng. Liviu MICLEA

HEAD OF DEPARTMENT,
Prof. dr. eng. Rodica POTOLEA

Graduate: **Laura CEUCA**

Safer museum guide interaction during a pandemic and further
Using NLP in human interactive museum visits
Museum guide Chatbot

1. **Project proposal:** *The purpose of this project is to create an interactive chatbot that can engage in conversations with museum visitors and can answer basic questions regarding the museum exhibits, artists, and culture, therefore serving as a non-human museum guide.*
2. **Project contents:** *Presentation page, Introduction, Project Objectives, Bibliographic Research, Analysis and Theoretical Foundation, Detailed Design and Implementation, Testing and Validation, User's manual, Conclusions, Bibliography, Appendices.*
3. **Place of documentation**: Technical University of Cluj-Napoca, Computer Science Department
4. **Consultants**: Assoc. Prof. Dr. Eng. Emil Ștefan CHIFU
   Eng. Ana REDNIC
5. **Date of issue of the proposal:** November 1, 2020
6. **Date of delivery:** July 7, 2021

Graduate: Laura ____________________

Supervisor: ____________________

MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

FACULTY OF AUTOMATION AND COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTMENT

Declaraţie pe proprie răspundere privind autenticitatea lucrării de licenţă

Subsemnatul(a) **Ceuca Laura**, legitimat(ă) cu CI seria AX nr. 783199 CNP 2980709011153, autorul lucrării "Safer Museum guide interaction during a pandemic and further Using NLP in human interactive museum visits Museum guide Chatbot" elaborată în vederea susţinerii examenului de finalizare a studiilor de licență la Facultatea de Automatică și Calculatoare, Specializarea Calculatoare din cadrul Universităţii Tehnice din Cluj-Napoca, sesiunea iulie 2021 a anului universitar 2020-2021, declar pe proprie răspundere, că această lucrare este rezultatul propriei activităţi intelectuale, pe baza cercetărilor mele şi pe baza informaţiilor obţinute din surse care au fost citate, în textul lucrării, şi în bibliografie.

Declar, că această lucrare nu conţine porţiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislaţiei române şi a convenţiilor internaţionale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în faţa unei alte comisii de examen de licenţă.

In cazul constatării ulterioare a unor declaraţii false, voi suporta sancţiunile administrative, respectiv, *anularea examenului de licenţă*.

Data	Nume, Prenume
06.07.2021	Ceuca Laura

Semnătura

Table of Contents

Chapter 1. Introduction

The following chapter will offer the reader an overview of the current impact that the ongoing pandemic has over the museum industry and what solutions could possibly improve the current state of crisis.

1.1. Project Context

The current epidemiological context has imposed strict human distancing rules that make it difficult for industries based on interaction to follow. Examples of such industries are the museums and art galleries, that rely on trained guides to address curiosities of visitors.

Since the beginning of the pandemic, restrictions applied worldwide to prevent the spread of the virus caused serious difficulties for museums. Since most of them have had their gates closed to visitors, they also encountered several other problems, such as having to keep their staff members working from home, or in serious cases even having to reduce the number of their employees.

Another, and probably the most serious problem museums and galleries have been facing is the income loss resulting from tickets, shops, or other services that cannot be provided in the current context. It has been found that out of a series of museums participating in a study done by the Network of European Museum Organizations, 44% lose up to 1.000 Euro per week, 31% lose up to 5.000 Euro per week, 18% lose up to 30.000 Euro per week, and 8% lose more than 50.000 Euro per week [1].

The most important struggles of museums are depicted in a circular causal relationship, in the diagram below:

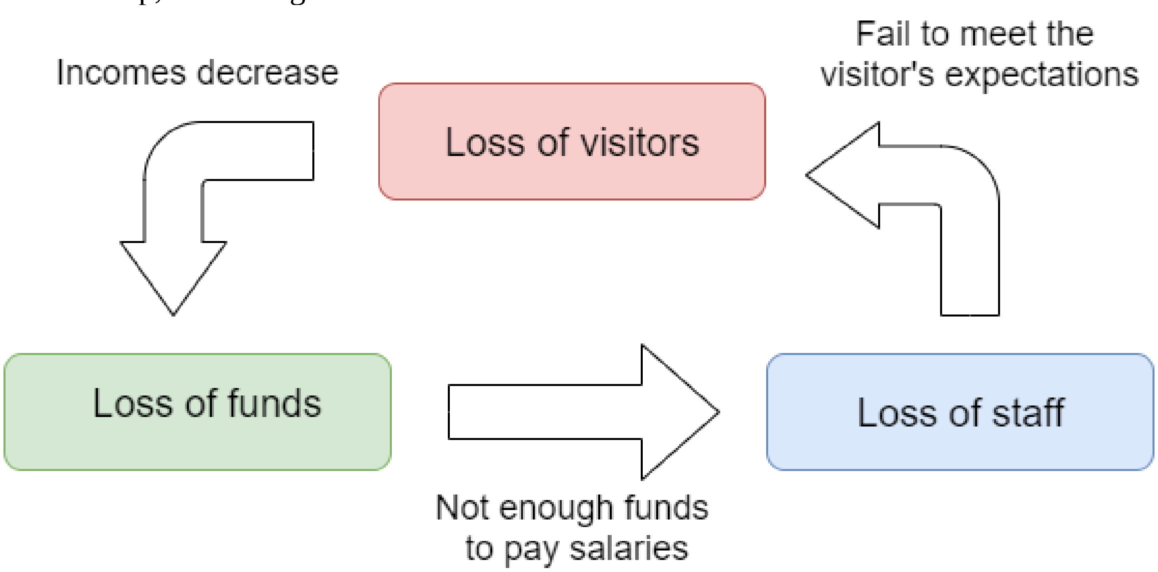

Figure 1.1 Effects of the pandemic over the museum industry

These problems are expected to have long-term effects in the future, a scenario for which museums are already preparing and trying to find alternative ways of engaging the public, making their presence known in the on-line medium or reducing their overall costs of functioning. The current study also found some significant changes that museums expect in the future, and they are presented in the diagram below:

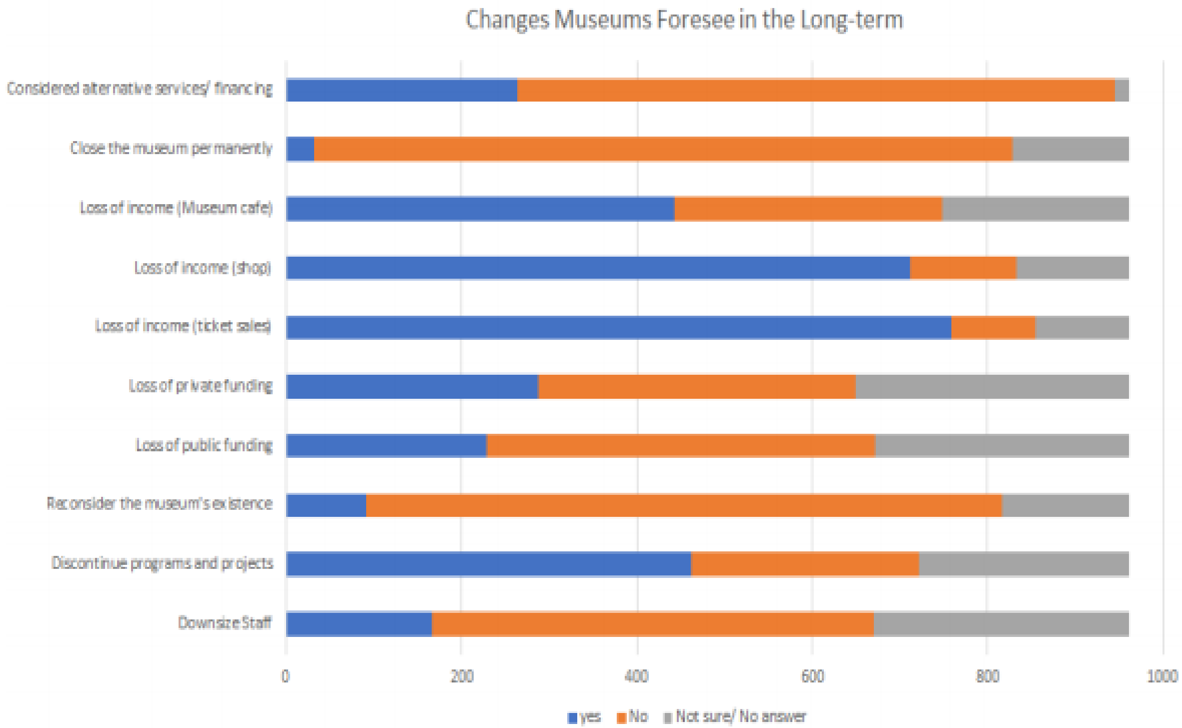

Figure 1.2 Changes museums foresee in the long-term [1]

One of the solutions that has been very popular along museums and galleries has been growing an Online presence, through offering different services, such as learning platforms, Online exhibitions, virtual tours, podcasts, newsletters, or listing objects in Online collections. The Online services offered by museums are analyzed in the chart below:

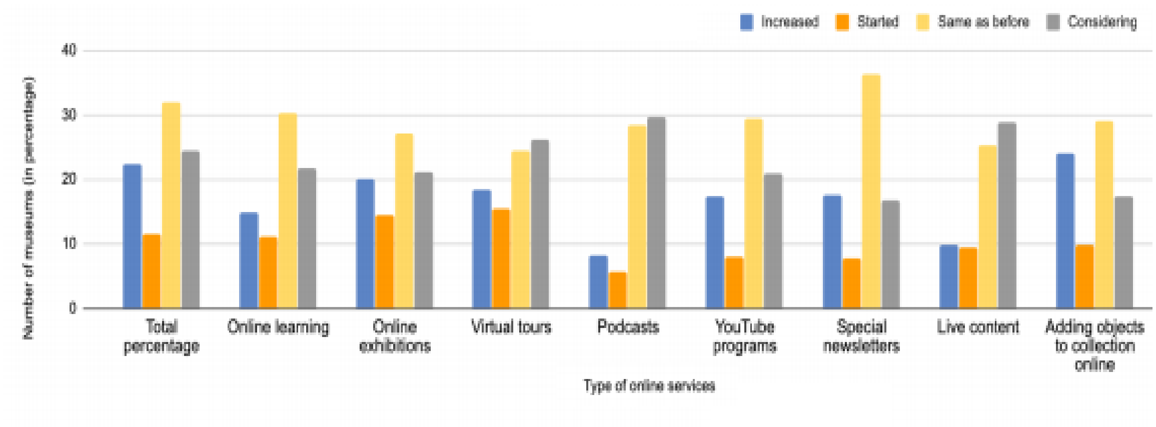

Figure 1.3 Online service activity [1]

Other solutions have been adopted by museums all over the world, for example, The Museum of Digital Art in Zurich offered live streamed creative classes grouped by the target audience, while The Museo Nacional Thyssen-Bornemisza in Madrid adapted their new exhibition “Rembrandt and Amsterdam portraiture, 1590-1690” to a digital format [1].

The presented solution can compensate for the lack of guided tours, while respecting sanitary measures and protecting the staff.

In particular, the Art Museum in Baia Mare has faced a drastic decrease in number of visitors during the pandemic. Strict rules of hygiene were imposed for physical visits. To protect the staff, only a single exposition (10% of the museum) remained open, and guided tours are not available. Only one visitor is allowed in the museum every half an hour and the visit must be scheduled beforehand. The museum came up with online solutions to keep visitors interested. They organized virtual tours, 3D expositions using augmented reality, live broadcast of events, and permanent updates regarding their activities.

1.2. Problem statement

The current project addresses the presented issue by creating a software entity that holds knowledge about the exhibits of a certain art museum, the artists, and the artistic currents that the pieces belong to. This knowledge base is used in a natural language processing model that can answer basic questions of the visitors, offering an alternative to the human guide. The following table (Table 1.1) summarizes the issue that is addressed in this paper and the identified solutions.

Table 1.1 Problem statement

The problem of	Lack of guided tours offered to visitors because of the pandemic
Affects	All the visitors and employees of the museum
The impact of which is	Offering a safer alternative to human interaction
A successful solution would be	Able to offer relevant information to the visitors Able to communicate properly with visitors Easy to use Easy to enrich the knowledge base with new information Fast response time Friendly user interface

Chapter 2. Project Objectives

This chapter offers an overview of the purpose and objectives of the project presented. In this chapter, the users and stakeholders of the project are also identified along with the main features of the chatbot and other requirements of the project.

2.1. Project Purpose

The purpose of this project is to create an interactive chatbot that can engage in conversations with museum visitors and can answer basic questions regarding the museum exhibits, artists, and culture, therefore serving as a non-human museum guide. The main objectives are recognizing human speech and converting it into text (speech-to-text), giving a correct answer to the addressed question based on a knowledge base and converting the text answer into speech (text-to-speech).

2.2. Product Position Statement

The current epidemiological context has made it more difficult for humans to interact with each other in places like museums or art galleries. For example, the Art Museum in Baia Mare decided to keep their doors closed as in not to expose their employees to the danger of contracting the deadly virus.

In this case, a software system that is capable of partially taking on the duties of museum guides can facilitate the safe reopening of the museum and offer visitors a pleasant experience. The purpose of the product proposed to fulfil the needs presented is detailed in the table below (Table 2.1).

Table 2.1 Product position statement

For	The Art Museum in Baia-Mare
Who	Needs a safer alternative to human museum guides in the current context
The museum chatbot	is a virtual museum guide that uses NLP processes to interact with visitors and offer information about the exhibits
That	Help visitors and museum employees have a safer experience regarding human interaction
Unlike	Human museum guides
Our product	Will be interactive and will offer relevant information

2.3. Project objectives

The objectives of this project are the following:

1. The application offers a customized tour to the visitors, in which all visitors can ask their own curiosities and receive an answer about the exhibits of the museum.
2. The interaction between the application and the users is done in a familiar manner, using spoken language or, if chosen, written text.

3. The application offers the administrator the possibility of adding new information in the knowledge base.
4. Existing information can be modified, or if it is no longer relevant, removed.
5. Along with the answer to their questions, the application can also display images to the visitors.
6. The user interface of the application is simple and intuitive, suitable for people with different levels of familiarity with technology.

The objectives of the project are SMART (Specific, Measurable, Attainable, Relevant and Time-bound). For example, the first objective is SMART because the specific task of the application is giving an answer to the questions of the visitors, the result of which can be measurable by the users. It is attainable since all the information about the museum exhibits are stored in a knowledge base and relevant since it offers information to the visitors instead of a human guide. Finally, it is time-bound since this feature has a finite time for implementation and collecting information, that of a month.

2.4. Stakeholder and User Descriptions

The stakeholders of this project are presented in table 2.2. A brief description of each stakeholder has been provided, and also a summary of their responsibilities regarding the development and further usage of this application.

Table 2.2 Stakeholder summary

Name	Description	Responsibilities
Museum administrator	Is interested in offering a safe experience to employees and visitors	Main stakeholder He is responsible for keeping the museum open and respecting the health guides and regulations
Museum employees	Are interested in minimizing their interaction with visitors due to current context	They are responsible for offering information and an overall pleasant experience to visitors of the museum
Museum visitors	Are interested in having a pleasant experience and gaining information about the exhibits	They are responsible of interacting with the application and possibly offer feedback for further improvements

The users of the application are summarized in Table 2.3. A brief description of each user and their main responsibilities has been provided in the table.

Table 2.3 User summary

Name	Description	Responsibilities	Stakeholder
Museum employee	Holds knowledge about the museum exhibits to be added in the knowledge base of the system	Helps develop the knowledge base and uses the system to ensure that given answers are the desired ones and the information offered is relevant	Museum employees
Museum administrator	Is interested in having a user-friendly application that offers a pleasant experience to visitors	Responsible for museum management and visitors experience	Museum administrator
System administrator	Has technical knowledge and understands the system functioning	Is responsible for maintaining the system and adding information to enrich the knowledge base as museum exhibits increase in number	Museum administrator
Museum visitors	The main end-users of the application	Interact with the system in order to gain information	Museum visitors

2.5. User environment

2.5.1. Users

The system will be operated by an average of 50 concurrent users.

2.5.2. Time limits

The application should have a real-time response rate, so processing and building the answers should be done with minimum overhead.

2.5.3. Collaboration

This application is intended for a single user at a time and there are no known use cases that require more than one user at a time.

2.5.4. Infrastructure

The application will run on all browsers that support JavaScript. On the server side, the module should run on any platform supporting Node.js.

2.5.5. Summary of user needs

Table 2.3 Summary of user needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Offer information	High	Visitors, administrator, employees	Human face-to-face interaction	Human-computer interaction
Facilitate interaction	High	Visitors	Human face-to-face interaction	Friendly interface and text-to-speech capabilities
Clear input parsing	Medium	Administrator, visitors	Several libraries and services available	Speech-to-text recognition
Friendly interface	Medium	Visitors	No other software solution	User interface and voice interaction
Usability	Medium	System administrator	No other software solution	Easy to manage and enhance knowledge base
Respect health guidelines	High	Administrator, employees, visitors	Not respected in the case of direct human interaction	Respected if common surfaces are properly sanitized

2.6. Product perspective

The Museum Chatbot should provide a friendly interface, user interaction both via typed text and voice interaction, and offer relevant information to users, answering their questions about the museum, artists, and cultural currents of the exhibits. It should be designed as a standalone web application.

The product will offer new features such as voice interaction, comparing to its competitors that are mainly designed for text interaction. Also, the knowledge base will be specifically designed to accommodate the exhibits of the museum.

2.7. Product features

FEAT 01: Information storage

The product should contain a knowledge base with information about the museum exhibits, artists, and currents, that it will use to construct answers to the questions. All data will be stored in a NoSQL database.

FEAT 02: Knowledge base management

The information stored in the knowledge base can be modified, deleted, or new information can be included, thus the system offers CRUD operations on the data stored in the database.

FEAT 03: Knowledge base interrogation including pictures

An answer stored in the database can contain text as well as images, that are stored in the base64 format and can be displayed to the users as part of the answer to their questions.

FEAT 04: NLP

The system incorporates an NLP model that is trained on the data available and is able to answer questions even if they are only similar or just contain common words with the questions in the knowledge base.

FEAT 05: Text communication

The application offers a friendly chat interface that allows users to type a question and get a text answer to their question (possibly including a picture).

FEAT 06: Speech-to-text

The product uses a library for parsing spoken user input and transposing it into written text. So, instead of typing a question, the user can just press the "Speak" button and ask it directly.

FEAT 07: Text-to-speech

The application also features the possibility of transforming written text into speech using a dedicated library. If this option is enabled, the application will be able to read the answers to the user, engaging in verbal communication.

FEAT 08: Settings and preferences

The product will also contain a "Settings" option that will allow users to enable or disable the text-to-speech and speech-to-text options.

2.8. Product requirements

Visual application and intuitive icons

The application UI should be very intuitive to use for any visitor of the museum. For this purpose, each action will contain an intuitive icon to guide the user in the utilization of the product.

Usability

The user interface should be very easy to use. For this purpose, using extra menus is avoided and user actions are kept to a minimum when it comes to performing any activity in the application.

Open to extension and improvements

The architecture of the module should be open and easy to extend with additional features in the future. Also, the application is designed such that it can be used for different museums, art galleries and not only. The only modification would be the knowledge base, that is easily manipulated through the CRUD operations offered by the administrator page.

PASSME Requirements

1. Performance

The application performance should be measured in:

- Response time (desired: real time communication)
- Revenue from increased interest in visiting the museum
- Less exposure of employees to face-to-face interaction

2. Availability

The application should be available during the museum visiting hours, but even more, should be available 24h/day to be accessed over the internet in order to raise the public curiosity about visiting the museum.

3. Security

Data integrity and confidentiality is ensured by using an external service that handles the authentication and stores the sensitive data such as user accounts and passwords. The rest of the data that is stored in our database is not considered sensitive data.

4. Scalability

The application should be able to accommodate an increasing number of users and a larger amount of information without major changes to the application.

5. Maintainability

The application offers a system administrator feature that will be able to oversee the data and the overall functionality of the application, so that maintainability is ensured.

6. Extendibility

Additional features may be developed in the future if the overall response of the public is positive towards the application. It should be extended easily, without modifying the current functionality.

2.9. Expected results

It is expected from the application to be able to parse user input and give relevant answers to user questions with a high accuracy. The chatbot should be able to answer all questions of the visitor regarding the exhibits of the museum, such as:

- The details of a painting (the materials used, the time period it was realized, distinctive marks)
- Details about the authors (When did they live, what are their artworks)
- What artwork of a certain author you can find presented at the museum

Also, the communication between the user and the application must be done in real time, with no considerable processing and response time. The communication is done both in written text, if the user chooses so, or through speech, using speech-to-text and text-to-speech integrated features. For the text interaction there is a designated chat room interface.

The desired outcome is a good parsing and understanding of the input questions, as well as giving the right answer with a high degree of accuracy. The intent of the user should be perceived in any particular formulation based on certain keywords in the input and on the stem root of those words. The output should be clearly formulated, addressing the information need of the user.

The application is intended to have a friendly and interactive user interface that makes it easy to interact with the chatbot either through speech or text. This kind of interaction should be natural and a pleasant experience to the visitors, replacing the human interaction with the guides of the museum that is made impossible by the current pandemic context.

Chapter 3. Bibliographic Research

This chapter explores different approaches to the design of a Natural Language Processing agent, as well as the approaches for the speech interaction with visitors. Also, in this chapter, there is a presentation of similar work, along with their most significant features and a comparison between the projects.

3.1. Text parsing and interpretation

3.1.1. Named entity recognition (NER)

Named entity recognition is a technique of locating and classifying named entities in a text segment and placing them into predefined categories, such as names of people, locations, organizations, quantities, monetary value, etc. It represents the starting point to information extraction and is used extensively in the fields of Natural Language Processing [2], answering questions like:

- "Who is this text referring to?"
- "What company is this article about?"
- "What is the location this action is taking place at?"

3.1.2. Sentiment analysis (SA)

Sentiment analysis is a research field of text mining that studies the opinion, attitude, and emotions of a person towards an entity. It is a computational study of the overall sentiment expressed in a text fragment which is then classified as shown in the following picture [3]:

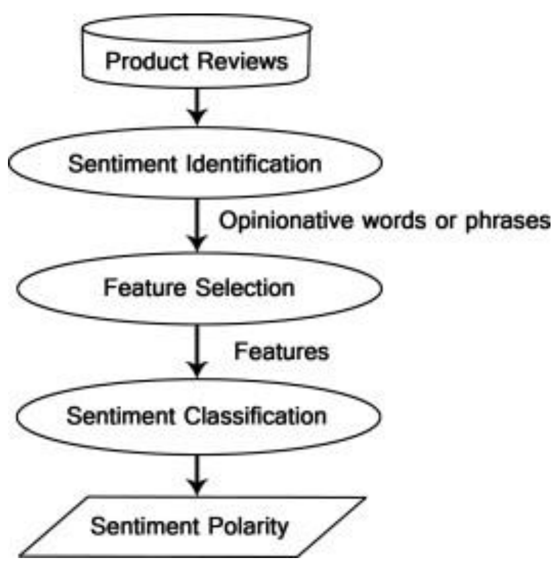

Figure 3.1 Sentiment analysis process [3]

3.1.3. Stemming and Lemmatization

Stemming is the process of reducing an inflected or derived word to their word stem, by removing all its suffixes.

Lemmatization uses a morphological analysis of words, aiming to remove inflectional endings in order to obtain the dictionary form of the word, known as lemma [4].

Both techniques are used in order to broaden the understanding of sentences by recognizing certain words in any form they might be encountered in text. In this way, a question that is formulated differently from the ones that are stored in a knowledge base can be recognized by matching the stem or lemma of the words it contains.

3.1.4. Question Answering (QA)

Questions answering is a research field concerned with automatic machines being able to answer questions formulated in natural language [5]. A question answering system is able to provide the user with specific information regarding a topic of interest, in a real time interaction, relieving the user of the need to traverse long writings in order to find the answer. This kind of system is particularly useful in the context of a museum or gallery since it can provide real time targeted information about the exhibits of the museum that is of interest to the users.

The chatbot presented in this paper is mainly a question answering system that uses all NLP techniques and processes mentioned above, in order to parse the questions of the visitors and offer helpful information to them.

3.2. User interaction using speech techniques

3.2.1. Speech-to-text

Speech recognition is the capability of a computer to convert spoken language into written text. Its main goal is to establish a more natural communication between humans and machines [6].

An automatic speech recognition system works in two different phases. The first one is a training phase in which the system learns different reference patterns corresponding to spoken sounds. Each reference sound is learned from examples and then stored as a model that contains the statistical characteristics of the pattern. In the second phase, the system recognizes an unknown input sound based on its stored set of references [7]. The principle of automatic speech recognition can be seen in Figure 3.3. presented below:

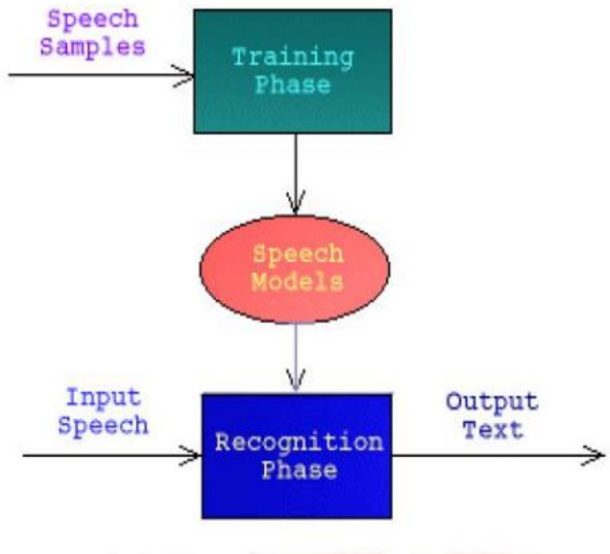

Figure 3.3 Principle of automatic speech recognition [7]

3.2.2. Text-to-speech

"Text-to-speech technology is the ability of a machine/computer to convert given text into speech" [6]. It facilitates the human computer interaction by offering the users a more natural way of communicating with a machine. In this project, the answers of the chatbot are articulated into speech, to simulate the interaction with a human guide.

Text-to-speech is achieved through several techniques such as articulatory speech synthesis, formant speech synthesis and concatenative speech synthesis [6].

3.3. Similar projects

3.3.1. Conversational agent Max

Max is an Embodied conversational agent (ECA) developed by a group of A.I. researchers from Germany. It started as a platform for studying the generation of natural multimodal behavior and has then been extended to a virtual assistant. In January 2004, Max was welcomed in the Heinz Nixdorf MuseumsForum (HNF) in Paderborn, Germany. Its task is to engage visitors of the museum in conversation, providing them with interesting information about the museum, the exhibitions, and many others. Max receives input from the visitor via keyboard and replies verbally, using a synthetic voice. The agent also communicates through gestures, facial expressions, and locomotion in order to create a more natural interaction [8].

Figure 3.4 Max interacting with visitors in the Heinz-Nixdorf-MuseumsForum [8]

3.3.2. The Leonardo Project

In 2002, in a project to rebuild the whole web site of the Science Museum of Milan, a group of researchers came up with the idea of integrating an on-line chatbot to engage in conversation with visitors after a successful experiment of using real museum staff to chat with pupils. The chatbot was initially designed to answer questions about Leonardo da Vinci, since the museum hosted a big exhibition about his work. It was built using the Lingubot technology from Kiwi logic and designed to accommodate the Italian language. Since the chatbot was mainly intended for the younger generation, its language and concepts were kept simple to understand [9].

Speech and listening features were also intended for this project, but they were never implemented since the project was terminated at the end of 2002, so in its final form the chatbot was able to hold a conversation only via written text.

3.3.3. The House Museums of Milan

The House Museums of Milan is composed of four historical homes, belonging to different styles and time periods ranging between the 19th and 20th centuries. They contain collections of items with interesting stories, but with little to no space for curatorial placards. For this matter, in 2016 they came with an initiative to attract younger generations: a chatbot game.

The chatbot is designed to interact with the visitors over Facebook Messenger and take them on a “treasure hunt” game regarding the exhibits of the museums. The chatbot asks users different questions about the exhibits, that are meant to engage the users in a vigilant observation of the objects. The gamification of this chatbot was a result of analyzing the needs of visitors and the target population of the application [10]. A screenshot of this project and its functioning is presented in Figure 3.5.

Figure 3.5 Question from the House-Museum Chatbot game [10]

3.3.4. Anne Frank House bot for Messenger

The Anne Frank House bot for Messenger was launched on the International Day for the Elimination of Racial Discrimination by Prince Constantijn of the Netherlands. This chatbot is powered by a deep learning AI and its role is to answer questions of visitors, customizing the experience of visiting the House of Ann Frank to each individual, but even more, has the mission of spreading information about the story of Ann Frank and warn the users of the risks and effects of racism and discrimination [11]. The chatbot can be seen giving answers over a Facebook Messenger conversation in the figure below (Fig 3.6).

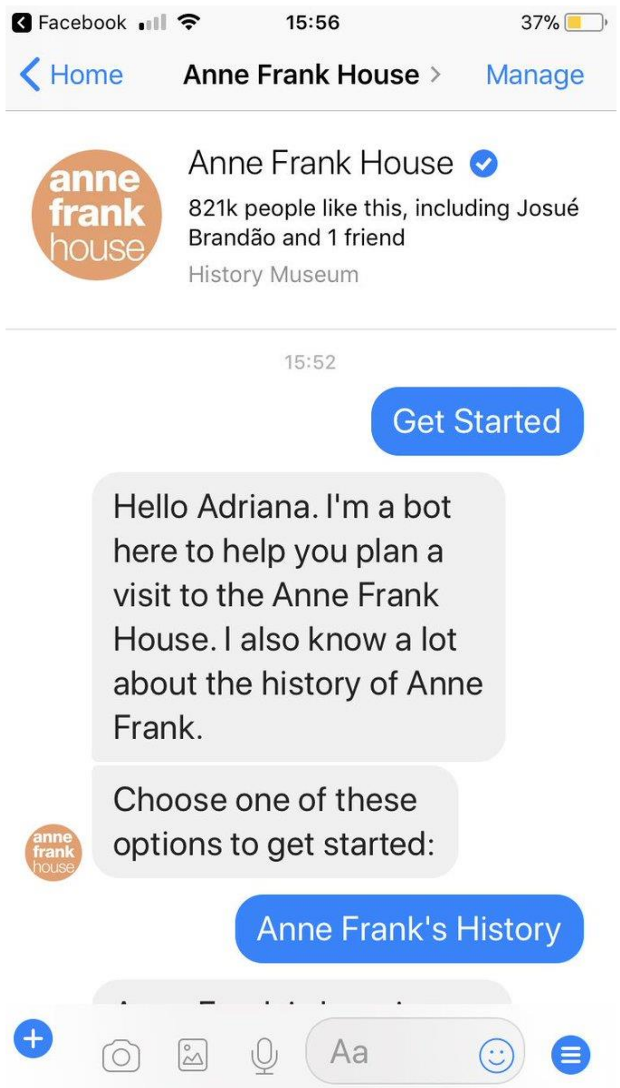

Figure 3.6 The Ann Frank House bot for Messenger [12]

3.4. Comparison between similar projects

The projects mentioned above were thoroughly analyzed based on complexity and interaction features that they possess. These features determine aspects such as the number of people that can interact simultaneously with an instance of the application, its conversational capabilities, its learning capabilities, and other features offered by the applications.

The Table 3.1 summarizes some of the aspects of the chatbots presented:

- Can it handle multiple users at a time?
- Can it hold a conversation or just answer to questions?
- Can it hold a conversation in spoken language?
- Has support for multiple languages?
- Can it use learning techniques to enhance its answers over time?
- Can it offer guided tours of the museum?

Table 3.1 Comparison of the features of the presented chatbot projects

Agent	Can handle groups or individuals?	Holds conversations or just answers questions?	Can conversate in speech or text?	Can address in multiple languages?	Uses learning to improve its capabilities?	Also guides tours besides holding conversations?
Max	One individual at a time	Holds conversation	Speech	No	Yes	No
Leonardo project	One individual at a time	Answers questions	Text	No	No	No
House Museums of Milan	One individual at a time	Asks and answers questions	Text	No	No	No
Anne Frank House bot	One individual at a time	Answer questions	Text	Not stated	No	No

After the comparison of the presented projects, it was concluded that the preferred way of interaction is question answering. From an implementation point of view, it is the most accessible natural language processing paradigm and it serves its purpose of delivering information requested by visitors. Text interaction is preferred as a safe option because the background noise can make it difficult for the input to be parsed properly. Also, speech should not be the only input option available, as in a museum setting the noise level should be kept as low as possible. The conversion of spoken input into text is a source of overhead, considerably increasing the response time of the application. Although, given the reduced dimension of the exhibition in the Art Museum of Baia Mare, the background noise is a controllable variable, especially in the case of a pandemic, when visitors enter the exhibition one by one, based on a certain schedule.

So, the project presented in this paper is a system based on question answering interaction, that offers support for both vocal and written input, to meet all expectations of users on this matter. It is able to interact with one individual at a time. The other features observed in the presented projects will be a subject of further development.

Chapter 4. Analysis and Theoretical Foundation

This chapters presents the block diagram of the system, the main use cases of the application, as well as the theoretical foundation of the algorithms and natural language processing techniques used in this project.

4.1. Block diagram of the system

Figure 4.1 represents the block diagram of the system. As it is a web application, it is composed of 2 significant parts, the client side, which handles all the interaction with the users and the server side, which is responsible for data storage and routing.

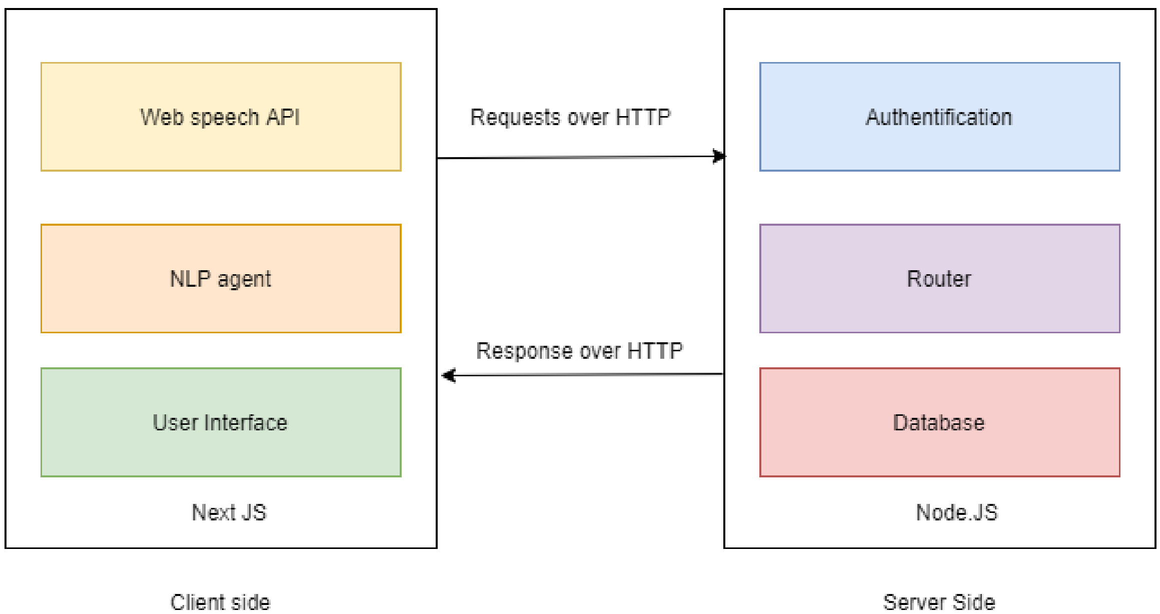

Figure 4.1 The block diagram of the system

The client side of the application contains the user interface, that ensures a friendly interaction with the chatbot, the NLP agent which processes the questions and finds the right answer to return, and the web speech API, which does the conversion of speech-to-text and text-to-speech to facilitate communication in spoken language. The server side contains the authentication service to secure the parts of the application that should be only accessible to certain users, the routing of the application and the database that stores all the information required by the application.

4.2. Use cases

The use cases of the application are presented below:

1. Start tour
2. Add information in knowledge base
3. Edit knowledge base
4. Show images

The first two use cases represent the main functionality of the application, so they will be presented in detail in this chapter. The rest of the use cases represent an extension of the functionality of the main use cases, and they will be presented briefly.

Use case 1: Start tour

This is the main use case of the application, capturing the flow of interaction of main users of the application, the museum visitors, in order to obtain desired information about the exhibits and have a safe and insightful experience all together. This use case follows the action of asking the chatbot questions and receiving answers in return, establishing the communication of visitors with the application.

Use case 2: Add information in the knowledge base

This use case regards the other users of the application, the administrator and museum employees, permitting them to add new information about the collections and exhibits of the museum, should new items be added to the museum. The application offers a very simple way to introduce knowledge in the application, making it user-friendly and easy to use.

Use case 3: Edit knowledge base

This use case has the purpose of allowing the administrator or the museum employees to edit the existing information in the application, by altering existing items or removing them from the knowledge base. This is especially useful in case some information has been wrongfully added or in cases exhibits are no longer part of the museum collection, making the information stored about them no longer necessary.

Use case 4: Show images

This use case is an extension of the primary use case of the application, in which the chatbot offers answers to the questions of the visitors. This extended use case allows the application to accompany the given answers with pictures, in order to enrich the experience of the visitors, by showing them a portrait of the artists behind the art or the actual paintings on display.

The first two use cases are described in detail with the help of use case diagrams and a detailed description of the steps of each use case. For each use case, the main flow of events, along with all the alternative flows are described in the following section.

Description of use case 1: Start tour

1. **Primary Actor**

Museum visitor

2. Use case diagram

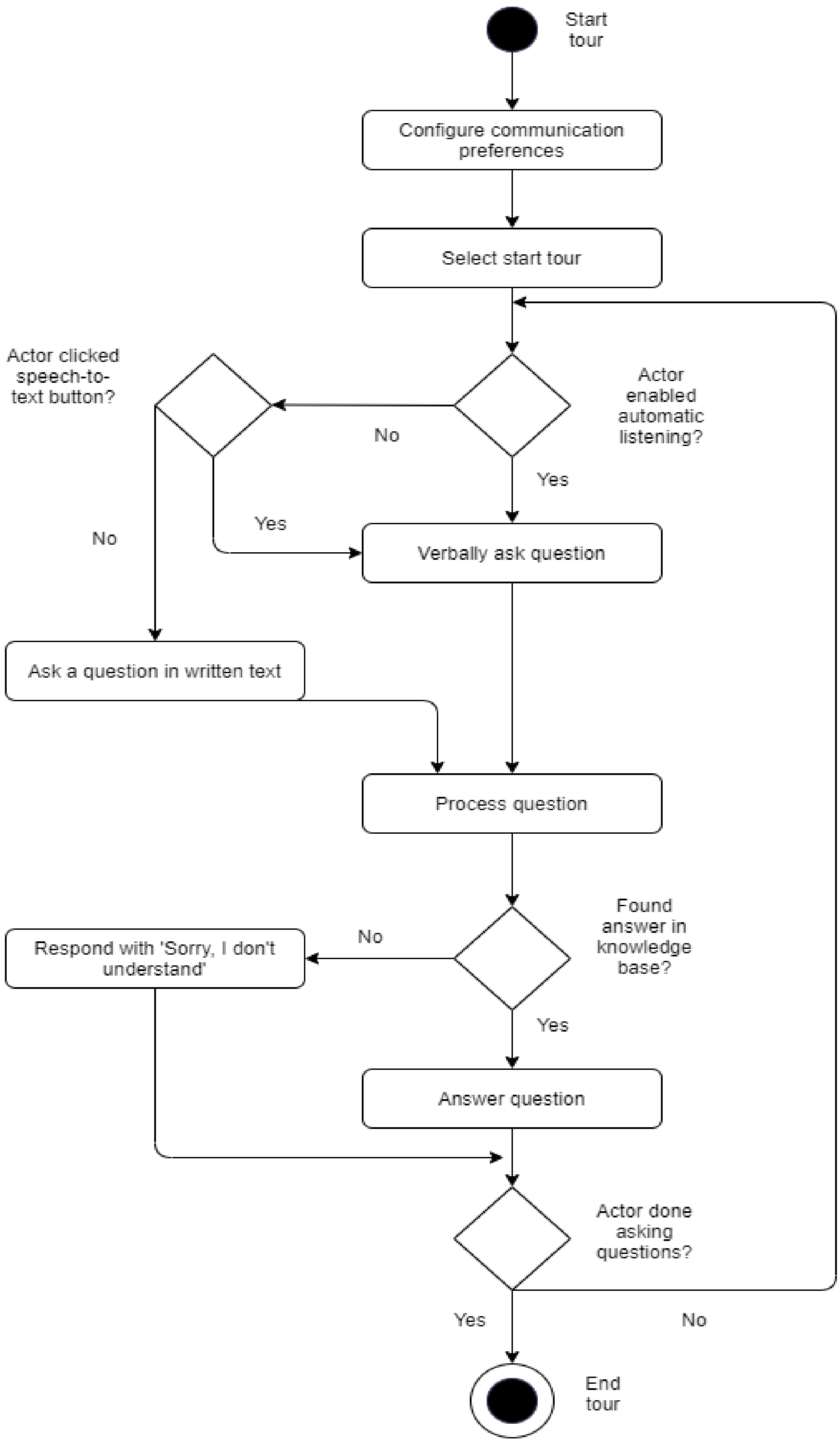

Figure 4.2 The use case diagram – Start tour

3. Basic flow

Use-Case Start

This use case starts when the actor wants to start a new tour and start asking questions.

3.1 The actor configures the communication preferences with the application, that is asking questions via spoken language or via written text.

3.2 The system saves the preferred configuration for the current user.

3.3 The actor selects the start tour option.

3.4 The system displays the char room for the questions.

3.5 The actor verbally asks a question (see alternative flow 4.1)

3.6 The system processes the question.

3.7 The system returns the answer to the question (see alternative flow 4.2)

3.8 The actor ends the interaction with the application (see alternative flow 4.3)

Use-Case End

The actor stops asking questions and ends the communication.

4. Alternative flows

4.1 Automatic listening is not enabled

This flow can occur in step 3.5. The actor can either:

4.1.1 Select the text-to-speech option and ask the question verbally

4.1.2 Type the question using the keyboard

4.1.3 Go to step 3.6

4.2 Answer to the question is not found in knowledge base

This flow can occur in step 3.7

4.2.1 The system responds with ‘Sorry, I don’t understand’.

4.2.2 Go to step 3.8

4.3 Actor wants to ask another question

This flow can occur in step 3.8

4.3.1 Go to step 3.5

5. Preconditions

Application is open on the homepage. There is no other precondition since this use case does not require the user to be authenticated in the application.

6. Post-conditions

6.1 The knowledge base in not changed following this interaction.

6.2 The application displays the answer to the question.

6.3 The conversation between the actor and the system remains until restarting the tour or reopening the application.

Description of use case 2: Add information in the knowledge base

1. Use case diagram

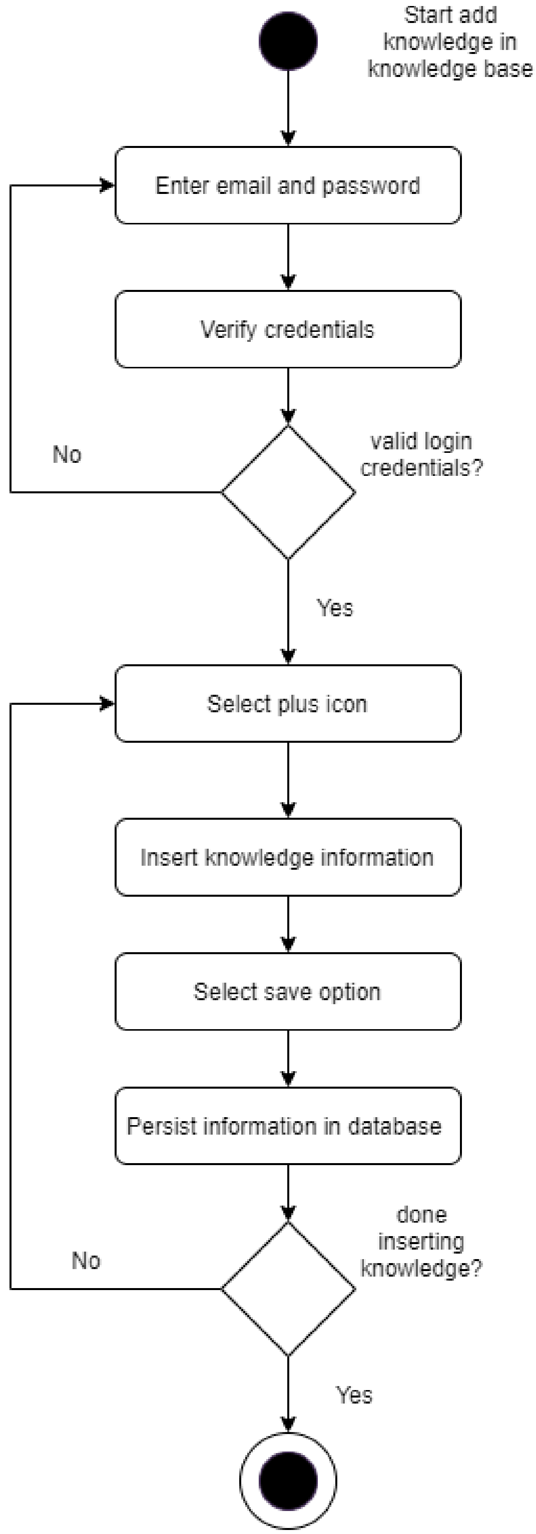

Figure 4.3 The use case diagram – Add information in the knowledge base

2. Basic flow

Use-Case Start

This use case starts when the actor wants to add new information into the knowledge base.

2.1 The actor enters the email and password to authenticate in the application.

2.2 The system verifies if the credentials correspond to a valid user account.

2.3 The actor selects the plus icon. (see alternative flow 3.1)

2.4 The actor introduces the information they want to add to the knowledge base into the dedicated fields.

2.5 The actor selects the save option.

2.6 The system persists the information into permanent storage.

2.7 The actor ends the interaction with the application (see alternative flow 3.2)

Use-Case End

The actor stops introducing new information and ends the interaction.

3. Alternative flows

3.1 The credentials are not valid

This flow can occur in step 2.3.

3.1.1 The system displays an error message

3.1.2 Go to step 2.1

3.2 Actor wants to insert more information into the knowledge base

This flow can occur in step 2.7.

3.2.1 Go to step 2.3

4. Preconditions

4.1 Application is open on the administrator page.

4.2 Authentication is required for this use case.

5. Post-conditions

5.1 The knowledge base is changed following this interaction.

5.2 The newly inserted information is displayed in the knowledge base and remains in persistent storage.

System Sequence Diagram: Start tour

The main use-case of the application, Start Tour, is also represented from the point of view of the interaction between the user and the application in the following picture (Figure 4.4).

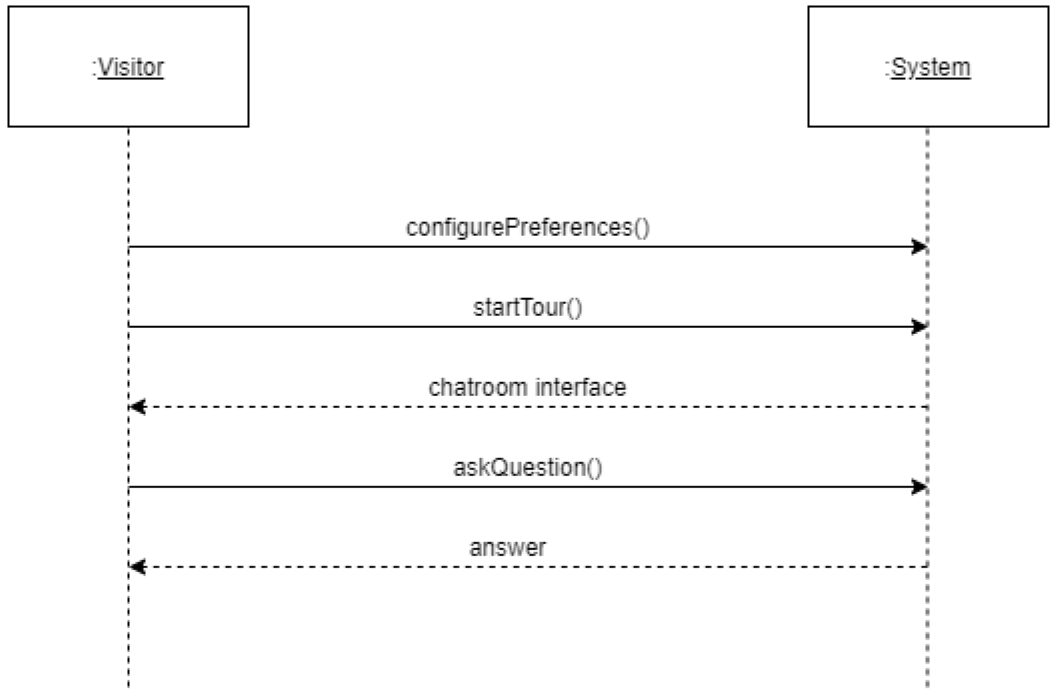

Figure 4.4 System Sequence Diagram – Start Tour

4.3. Spoken communication between users and computer

An innovative feature that differentiates the chatbot presented in this paper from the ones that are already in use in different museums all over the world is the possibility of interaction in spoken words. This form of communication is already familiar to the users since we use this type of communication in human-to-human interaction. For example, if a visitor would request information from a human guide in a museum, they would probably do this by verbally asking a question. That is the interaction that this application is trying to mimic.

In the following paragraphs, there will be a more in depth look at what are the processes that allow this form of communication between the users and the computer, mainly speech recognition and speech synthesis.

4.3.1. Speech recognition

Speech recognition has already been a topic of interest for humankind since it offers the possibility of controlling devices by voice. It is already integrated in a large number of devices, varying from different domains and has been widely researched in recent years.

Speech recognition is described as the process of converting acoustic speech into text [13]. This is a challenging field of research because it has to consider many outside factors, such as noise, speaker variability and language variability [13]. The process of speech recognition can be divided into two parts: front-end and back-end. The high-level representation of a speech recognition system is presented in the picture below (Figure 4.5).

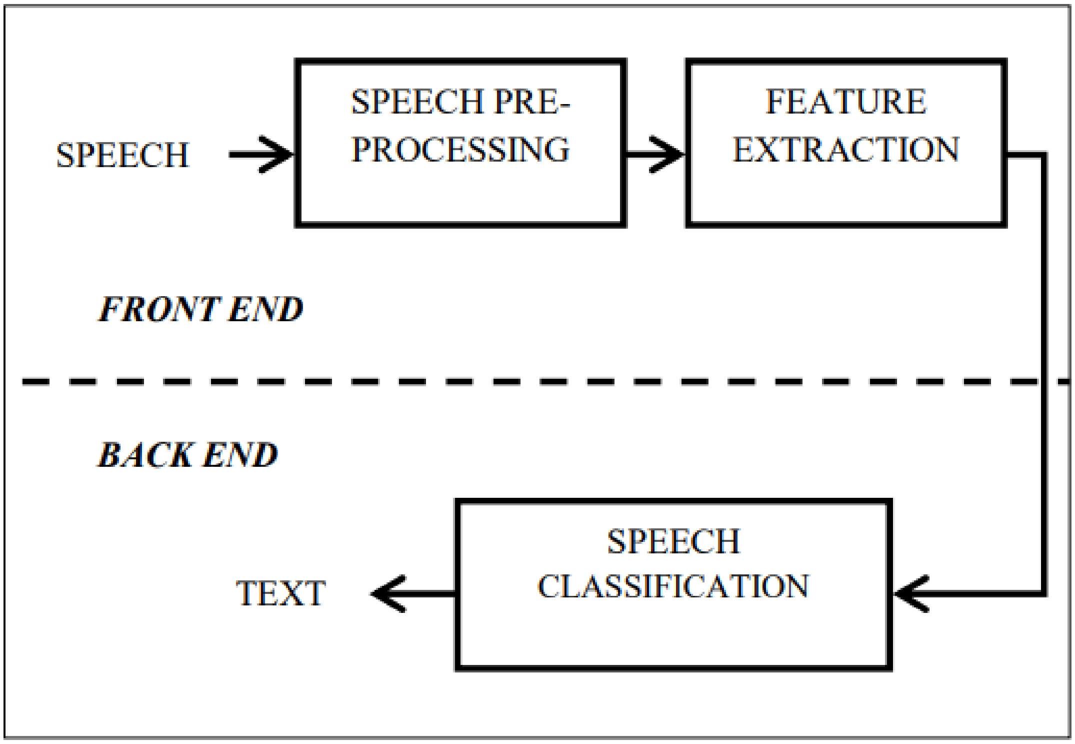

Figure 4.5 Speech recognition system [13]

Signal preprocessing deals with noise, differences in amplitude and timing variations between samples [13]. The process of pre-processing is an essential prerequisite of data analysis since it removes detected errors and prepares the signals for processing. Feature extraction is the process of extracting certain features of the signals being processed [13], while speech classification is the process of classifying the extracted features by matching them to the closest sound that is already existent in a database.

The output of this process is a representation of the input signal, in the case of our application, the desired output is a written text representation of the spoken input.

4.3.2. Speech synthesis

Speech synthesis, or text-to-speech (TTS) is a field comprised of many disciplines, such as acoustics, linguistics, digital signal processing and statistics [14]. The purpose of speech synthesis is converting written text into speech.

There are many known technologies used along the years for achieving speech synthesis, from formant based parametric synthesis, to waveform concatenation-based methods and more recently statistical parametric speech synthesis [14]. Along with the change of technology there has been a significant increase in how well the synthesized speech is understood by humans and how natural it sounds. However, the expressivity of human speech could not have been captured in artificial language.

The most promising technology for synthesizing speech in recent years has been deep learning (DL) that is able to capture the hidden internal structures of data and has a more powerful data modelling capability [14]. Using deep learning for speech synthesis is currently a heavily researched topic.

4.4. Algorithms used

4.4.1. Train NLP agent algorithm

The NLP agent used by the application is trained on the available data in order to match questions that are similar (or contain only keywords) to the ones existing in the knowledge base. In this way, a broader spectrum of questions is recognized by the application, without the need to introduce many different versions of the same question in order to cover all the possible formulations of the users. The training is done when the application is loaded, on available data or data received from the server. The algorithm for training the NLP agent and preparing it for answering questions is presented below (Figure 4.6).

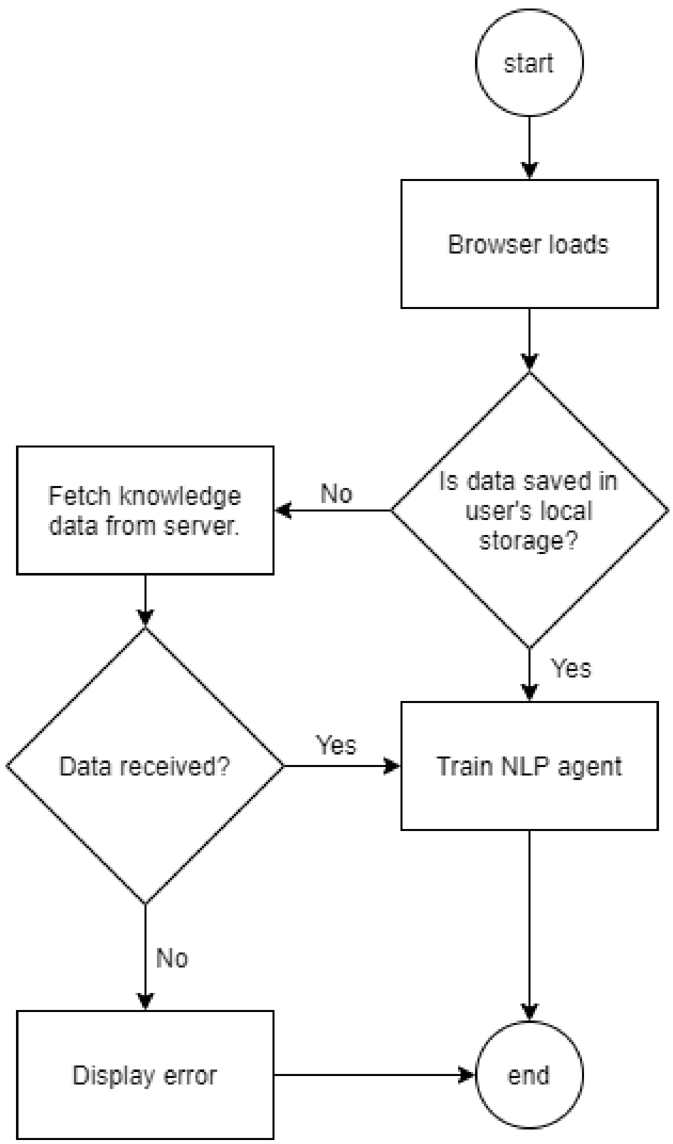

Figure 4.6 Activity diagram for the NLP training algorithm

This algorithm takes into consideration two main scenarios, the end goal of both being to train the agent on the available data. If the application was previously used by the user, then the data is already available in local storage and can be used for training. For performance reasons, data is not requested from the server if it already exists. If however the data is not found locally, then it is requested from the server and then the NLP agent is trained on the received data. This scenario happens when the application is used for the first time in a certain browser.

The structuring of the knowledge base groups multiple questions under a unique intent. The intent is identified by the chatbot using certain keywords or phrases that belong to it, and the answer is then selected based on the identified intent. The training of the agent is based on this feature, the goal being the classification of input phrases under a certain intent, in order to determine the right answer. So, by training the agent, action that is done transparently by the NLP module, the agent receives the data in the knowledge base and the intents associated with it; and then, after the training, the agent will try to place new input data (i.e., new user questions in natural language) under a certain already existent intent to determine its answer.

After the agent is trained, the application is ready to proceed to the next algorithm, that consists of interacting with users and answering questions.

4.4.2. Algorithm for questions answering

This algorithm describes the main flow of the application, in which the chatbot interacts with the users, by listening for their question and tries to give them an answer. The steps of this algorithm repeat in a cyclic manner for as long as the user want to keep asking questions. Each question asked is processed individually, an answer is returned and then the application is ready for the next question automatically. The graphical illustration of this process is presented in the next figure (Figure 4.7).

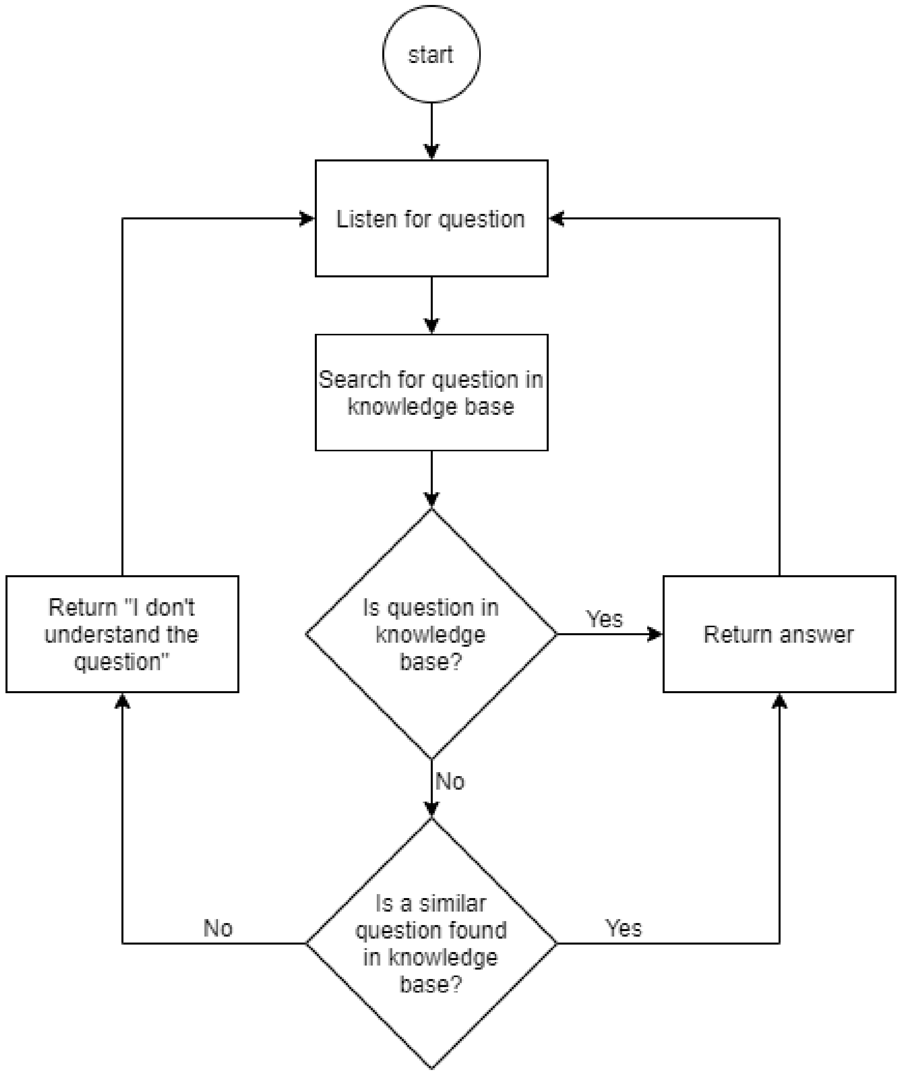

Figure 4.7 Activity diagram for the questions answering algorithm

The first of the two main scenarios of this algorithm is the one in which a question is identified in the knowledge base, in which case the answer is simply returned and the application is waiting for the next question. The second scenario is the one in which a question is not found in the knowledge base. In this case, the NLP agent tries to match the question with the most similar one that exists. If the question cannot be matched with anything that is already existent in the knowledge base, the application will simply return a default answer of not understanding the question.

4.4.3. Algorithm for adding data in the knowledge base

The following algorithm is intended to offer a simple and intuitive way to the museum employees to add new information in the application storage. The application displays dedicated fields for the information to be added and then does the necessary formatting and processing before persisting the data in storage, as depicted in the figure below (Figure 4.8).

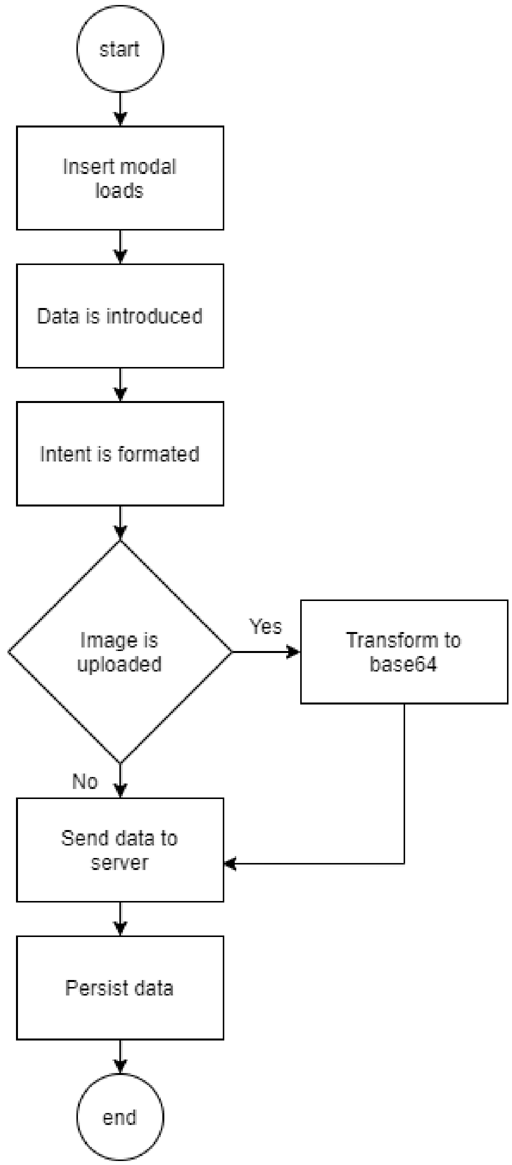

Figure 4.8 Activity diagram for the inserting data algorithm

The intent introduced is formatted before saving to ensure that it does not contain spaces or other characters that are not supported by the NLP agent. In case an image is also uploaded along with the answer, the image must be converted into base64 format before being sent to the server. After all the processing is performed, the data is sent to the server and stored in the database.

Chapter 5. Detailed Design and Implementation

This chapters presents the detailed structure and implementation of the application. The application is structured at the highest level into two parts: the server side and the client side.

The server side of the application is slim, containing the routers, modules and schema components and facilitating all communication with the database. The client side of the application contains most of the application logic, including the NLP agent, the text-to-speech and speech-to-text capabilities, and all the user interface of the application. The main components of the client side are pages, layouts, context, views, components, constants, types, and helpers.

The structure of the application is presented at package level in the diagram below:

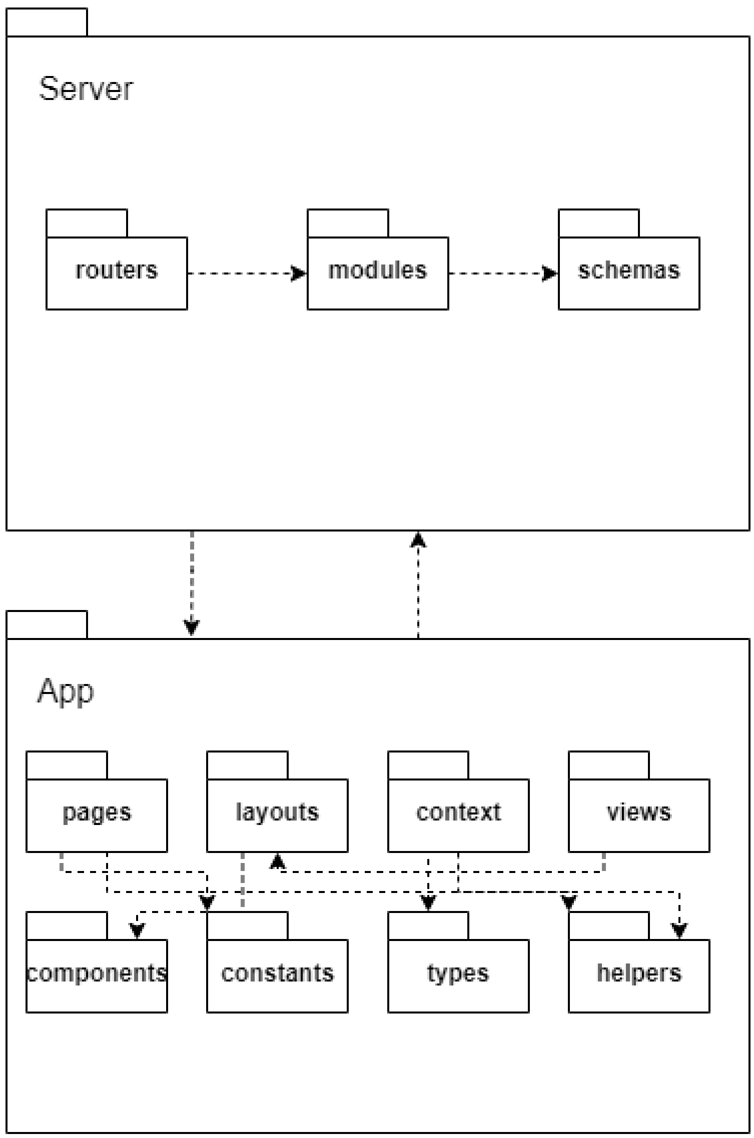

Figure 5.1 Package diagram of the application

Both sides of the application are written in TypeScript, an application-scale JavaScript. The code will compile to standard JavaScript. From a development point of view, TypeScript is more rigorous when it comes to variable and method typing. In this way, a lot of common mistype errors from JavaScript are avoided at compile time.

5.1. Server side

The application contains a slim server side that facilitates the authentication, the routing, and the communication with the database. This part of the application makes use of Node.js capabilities and integrates and Express.js server.

Node.js is a back-end JavaScript runtime environment that allows the creation of Web Servers using the JavaScript language, but also supports different compile-to-Js options, such as TypeScript, alongside a collection of modules that can perform different core functionalities. The use of Node.js provides this application with several advantages:

- Slim back end of the application and ease of creating a Web server
- Unified language of both sides of the application, without the need of using another language for the server side
- Any dependency provided by NPM can be added with a single command line

Express.js is a minimalistic web framework that uses the capabilities provided by Node to create a powerful web server in three lines of code (Figure 5.2). Express, as any other web frameworks, allows the developer to create routers, middlewares and parse the content of a request (or response). In this specific implementation, the framework is used to serve the Next.JS app, create routes and assure that the user is authenticated in order to perform administrative actions.

```
import express from "express";
const app = express();
app.listen(ENV.PORT);
```

Figure 5.2 Creating a Web server using Express

5.1.1. Authentication

The management or administrator part of the application is designated for museum employees and not to all the users of the application, therefore it needs authentication. For security and usability reasons, this application integrates Auth0 for the authentication process.

Auth0 is a free (up to 7000 users) authentication service that follows the current industry protocols to provide secure, scalable and a fully customizable authentication pipeline. It uses secure HTTP calls, and the service is compatible and manageable from any device type. It offers many advantages, some of the most relevant for this project being the following:

- It is extensible and customizable, being suitable for any type of application or sign in method
- It uses standard protocols such as SAML, OpenID Connect, JWT, OAuth 2.0 and others [15] to provide security
- It is easy to use and integrate with any language or technology stack

- It allows users to authenticate with the form of credentials they are most familiar with [15].

In the case of this application the chosen form of authentication is by using the email address. The authentication page using Auth0 is presented in the picture below:

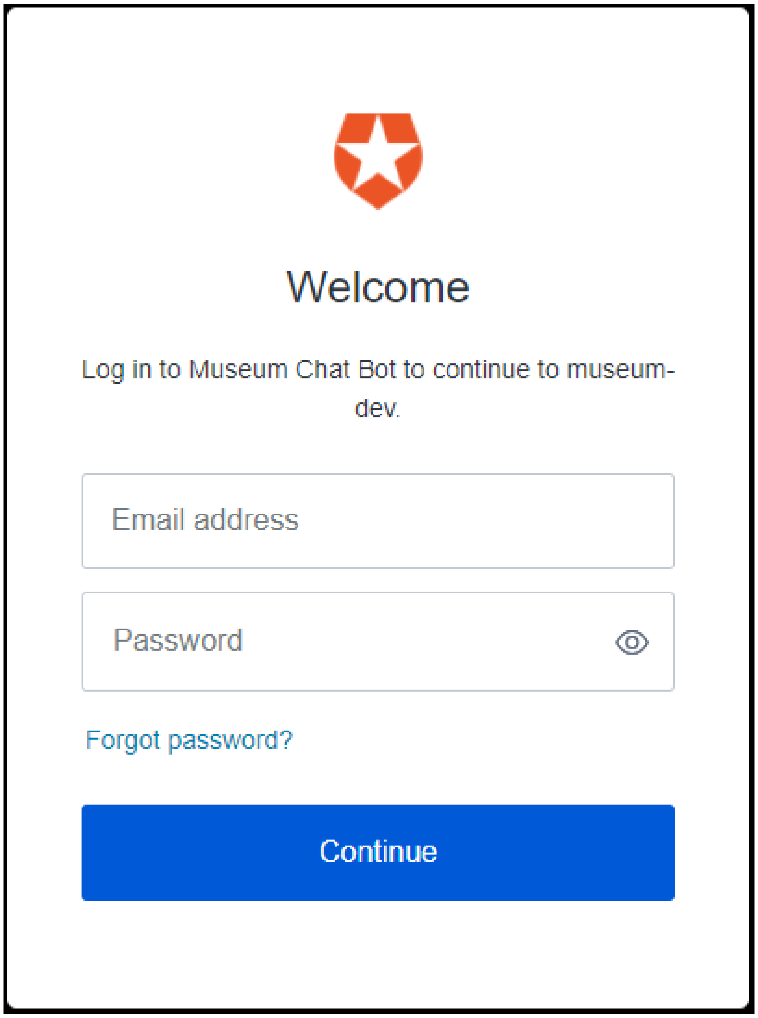

Figure 5.3 Application log in page

5.1.2. Router

Navigating from one page to another in an application is done using a router in the server-side of the application. For this application there are two different routers.

The first one is called apiRouter and its purpose is very similar to the REST controller of a Spring Boot application. It handles REST api requests to control the operations performed on the data in the database(CRUD create-read-update-delete operations). A request is performed on a specific route that corresponds to the desired operation. For example, an authorized GET request to the route /api/knowledge/all will return an array of all the knowledge objects that are stored in the database.

The second router is called nextRouter because is handles the mapping of pages in the NEXT.js application. An example of mapping the home page to the path '/' in the nextRouter is presented below (Figure 5.4). The *name* is given a constant assigned to make the redirects in the front-end easier from a coding perspective, *pattern* is the URL pattern, and the *page* is the NextPage component that will be rendered when the URL is hit.

```
NextRouter.add({
  name: ROUTES.HOME,
  page: "Home",
  pattern: "/",
});
```

Figure 5.4 Adding the route for the home page in nextRouter

5.1.3. Database

The chatbot answers questions of visitors based on the information it has available about the museum exhibits. All the information should be stored for the application to use and there should be a possibility of modifying, deleting, or inserting new data into the application. These requirements are easily solved by using a database to store the information the application needs. The chosen database for this application is MongoDB.

MongoDB is a document database, storing data in a format similar to JSON, allowing arrays and nested objects [16]. Given the structure of data stored, that of an array of objects containing questions, answers and intent, this format is the most suitable. The data storing format is also compatible with the JavaScript objects used by the application, so there is no need for extensive formatting of data before storing it into the database.

The structure of data is presented in the code snippet below (Figure 5.5). An object of type KnowledgeData represents a set of questions and their corresponding answers along with the intent associated to that question or set of questions. The intent is used by NLP to uniquely identify a question or set of questions that basically have the same meaning. The id uniquely identifies an object of this type in the database, and it is used for operations such as updating or removing an item.

```
export interface KnowledgeData {
  _id: string;
  answers: string[];
  intent: string;
  utterances: string[];
}
```

Figure 5.5 Structure of data that is stored in the database

The similarity in structure between the objects used in the application and the format stored in the database allows for simple CRUD (Create, Read, Update, Delete) operations on the data. The function for inserting a new structure of type KnowledgeData into the database is presented in the following code snippet (Figure 5.6). The name parameter identifies the database for the chosen museum (only one in this case) since the application offers the option of storing multiple datasets for different museums.

For example, two or more different museums can use the same application, having different datasets stored in the database and different administrative accounts (each username belongs to a certain tenant, in our case museum or gallery).

The function is asynchronous since this is the required type of communicating with the database. A database transaction or operation occurs independently in time from the execution of the rest of the code. The function then calls the updateOne method from mongoose (the ORM used for MongoDB).

```
async pushData(
  data: KnowledgeData, name: string
): Promise<void> {
  await this.Model.updateOne({ name }, { $push: { data } });
}
```

Figure 5.6 Function for inserting data into the database

5.2. Client side

The client side is the most complex one, containing almost all the logic of the application, the NLP agent, the integration of speech-to-text and text-to-speech functionalities and the user interface. The front-end is written in NextJS with TypeScript.

Next.JS is a framework built on top of React, which offers several advantages, such as the following:

- Page-based routing system, that simplifies the process of creating routes and pages when compared to vanilla React.
- Static-generation and server-side-rendering [17] which enhances the overall user experience. When a page is requested from a browser, the server will send a pre-rendered web page almost immediately (based on the user's connection to the internet and the platform they use).
- Automatically splits the code and creates bundles [17] which increases the performance of the application.
- In most of the use cases there is no config needed to create a Next.JS application.

5.2.1. Web Speech API

Web Speech API is an experimental feature that is found in the most popular browsers (for example Google Chrome and Mozilla Firefox). This API enables web applications to implement voice processing ability. It provides multiple APIs where the ones relevant to our application are SpeechSynthesis and SpeechRecognition.

5.2.1.1. Text-to-speech using SpeechSynthesis

SpeechSynthesis API was used to implement the custom useTextToSpeech react hook, that will read-out-loud any text it is provided with, using voices that the browser supports. In the figure below (Figure 5.7) there is the function readText that uses SpeechSynthesis to read the string given as parameter.

The prepVoice function defines the voice used for reading the text to be the first voice available, in this way avoiding the overhead of loading a new voice into the browser. The language, pitch, rate, and volume are adjustable characteristics of the voice used to synthesize the text. The stopReading function is used before the speak function to abort any reading from a previous call to the function. In this way, the voices produced by two successive calls to the function will not overlap. Finally, the speak function is called to start the actual synthesis of the text.

```
const readText = (text: string): void => {
  const speech = new window.SpeechSynthesisUtterance();
  speech.voice = prepVoice();
  speech.text = text;
  speech.lang = language || "en-GB";
  speech.pitch = pitch || 0.8;
  speech.rate = rate || 1;
  speech.volume = volume || 1;
  speech.text = text;

  stopReading();
  window.speechSynthesis.speak(speech);
};
```

Figure 5.7 SpeechSynthesis used to read text in useTextToSpeech react hook

5.2.1.2. Speech-to-text

Speech-to-text feature is created using a NPM package called react-speech-recognition that, at its core, is using Web Speech API. The package provides a react hook that can take as parameters an object. In the figure below (Figure 5.8), there is an example of using the useSpeechRecognition hook in the case of continuous listening. In this case a keyword at the beginning of each phrase (or command) is required in order to avoid the application listening to itself. The user has to start each question using the keyword specified, in our case 'ok' since it is a distinguishable sound. If the application recognizes the keyword, it will give anything that follows the keyword as the value parameter and run the callback. The value is then verified to be not empty and to be of type string. The transcript is reset to ensure new commands will be processed correctly and the current value is sent to be processed by the NLP agent.

```
const commands = [{
    callback: async (
      value: string,
      { resetTranscript }: { resetTranscript: () => void }
    ): Promise<void> => {
      if (value && isString(value)) {
        resetTranscript();
        await processMessage(value);
      }
    },
    command: `${NAMES.LISTEN_COMMAND} *`,
}];
const { transcript } = useSpeechRecognition({ commands });
```

Figure 5.8 Example of using the useSpeechRecognition react hook

5.2.2. NLP.js module

NLP.js is a powerful Node.js natural language processing package that implements many NLP agent capabilities and utilities, making it a great option for building natural language processing applications such as chatbots, sentient analysis agents, text summarization applications and many others. It is a very general tool with support for many different features, some of the most relevant for this project being:

- Stemming and tokenization of words [18]; this feature helps the chatbot understand certain words in any form they are found, using the stem of the word
- Named entity recognition (NER) [18]; this is the ability to recognize certain named entities in the text, such as names of people, places or paintings
- Natural language processing classifier [18], that aids grouping different utterances into a common intent
- Natural language generation manager, giving it the ability to generate an answer based on intent and certain conditions [18]

In the following figure (Figure 5.9) the function for loading the data into the NLP agent and training the model is presented. The function takes as argument the NLP instance that is unique and global for the application. Since in JavaScript any object is passed by reference, we start by cloning the NLP instance to avoid any overwriting of previous entrances. The function first attempts to retrieve the data from local storage, to avoid the overhead of reloading the data if it has previously been loaded. In case the application is loading data for the first time, it will not be available in local storage, and it will be retrieved from the database. After the data is loaded in the model, the train function is called to train the model with the loaded dataset.

```
export const loadKnowledgeAndTrain =
async (nlp: any): Promise<any> => {
  const nlpClone = cloneDeep(nlp);
  const knowledge = getKnowledgeFromStorage();

  if (knowledge) {
    nlpClone.addLanguage(knowledge.locale);
    nlpClone.addData(knowledge.data, knowledge.locale.slice(0
, 2));
    await nlpClone.train();
  }
  return nlpClone;
};
```

Figure 5.9 Function for loading the data into NLP and training the model

Figure 5.10 presents another key function of the application, the process function. While the previously presented function was just preparing the data and setting up the model, the process function of the NLP.js does the actual processing of the message. Its purpose is to find the best fitting answer to the given question (the message parameter of the function). The process function does this by searching the knowledge base to find the question given to processing and taking its corresponding answer. If the question is not present in the knowledge base, it will attempt to match it to the closest question, by using intent, keywords, NER, word stems and other NLP techniques.

```
async _process({language = "en”, message,}:
 {language?: string; message: string;}): Promise<any>
{
    const resp = await this.nlp.process(language, message);
    return resp;
}
```

Figure 5.10 Function for processing a message

5.2.3. User interface

The application is composed of two separate parts, one part for the museum employees or administrator to facilitate easy maintenance of the information stored in the knowledge base and the other one for the visitors of the museum that interact with the application in order to gain knowledge about the exhibits. Since the application is designed for people of different levels of familiarity with technology it must be intuitive and simple to use. The user interface has a minimalistic design, and all actions are marked with intuitive icons or extra information in case it is needed (for example, the setting on the user side).

5.2.3.1. Administrator interface

The administrator part of the application is restricted to only some authorized people, so in order to access this part of the application, the user must first log in with valid credentials. The administrator part opens with a simple page with just a ’Log in’ button. No other actions have been added to this page since the ’Log in’ action is the only one permitted for an unauthorized user trying to access this part of the application.

After pressing the ’Log In’ button, the user is then redirected to the authentication page offered by Auth0, the authentication platform used in the application (for reasons presented earlier in this chapter). The authentication method chosen was using e-mail, since it is the most commonly used by people of all ages and backgrounds.

After a successful authentication, the user is redirected to the main administrator page (Figure 5.11), that displays all the information currently stored in the knowledge base of the application, along with possible actions for these entries, such as delete and edit, and the option to add new entries and persist them into storage.

Each entry presented is displayed in a separate rectangle, that contains the utterances or questions, the unique intent, and the answer. If the answer optionally contains a picture, then that is also displayed in the circle corresponding to each entry. Each of the entries displayed have two intuitive icons corresponding to the ’Edit’ and ’Remove’ actions. The plus icon at the bottom right of the page corresponds to the ’Add new entry’ operation.

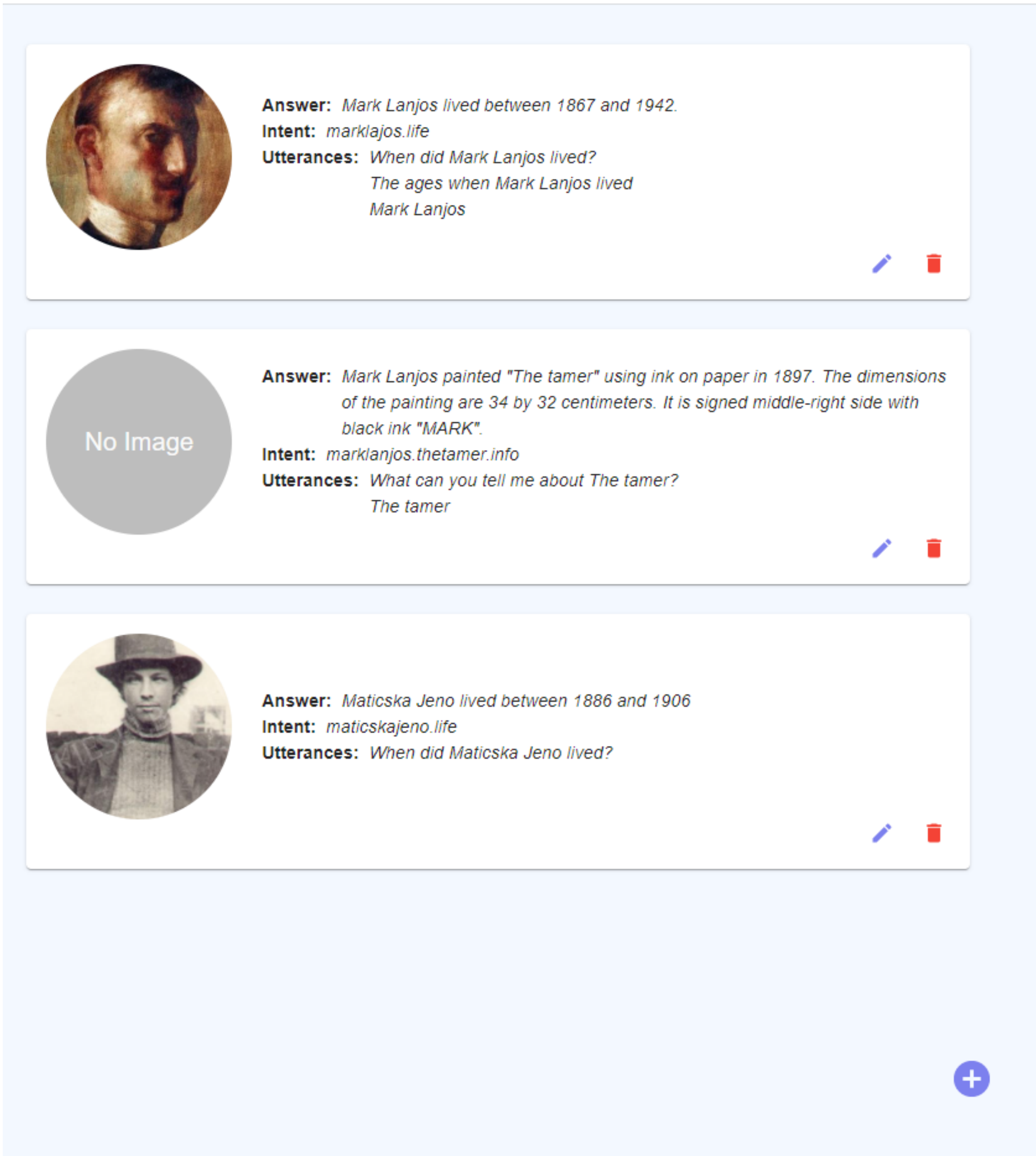

Figure 5.11 Section of the administrator view of entries in the knowledge base

If the user presses the ’Add new entry’ icon, a modal is displayed (Figure 5.12), containing the fields necessary for a new entry. The answer field is a mandatory field in which the user fills in the information that is desired to be returned by the chatbot in a conversation with the visitors, should the questions correspond. The picture field is an optional field in which the user can simply select a file from the device and upload it in the application. The image is converted to base64 format to be further used in the application. The intent is mandatory and is the unique identifier of a group of utterances for the NLP agent. The intent must have a certain format, that is all letters must be lowercase and the white spaces are replaced by ’.’,but this is not validated in the field, instead it is automatically formatted before being stored in the database, to further

simplify the actions of the user. The last field of the modal is the ’Utterances’ field. This is not represented by a single field but can be extended to any number of necessary fields, in order to accommodate as many formulations and questions is desired by the user.
Except the first field, which is mandatory, the rest of the fields representing utterances can be added or removed as desired. When the user has finished introducing the data, the ’Save’ button confirms the action and proceeds to store the data into the database. In case any mandatory field is left empty, the save action will not work and the field that has validation errors will be highlighted in red.

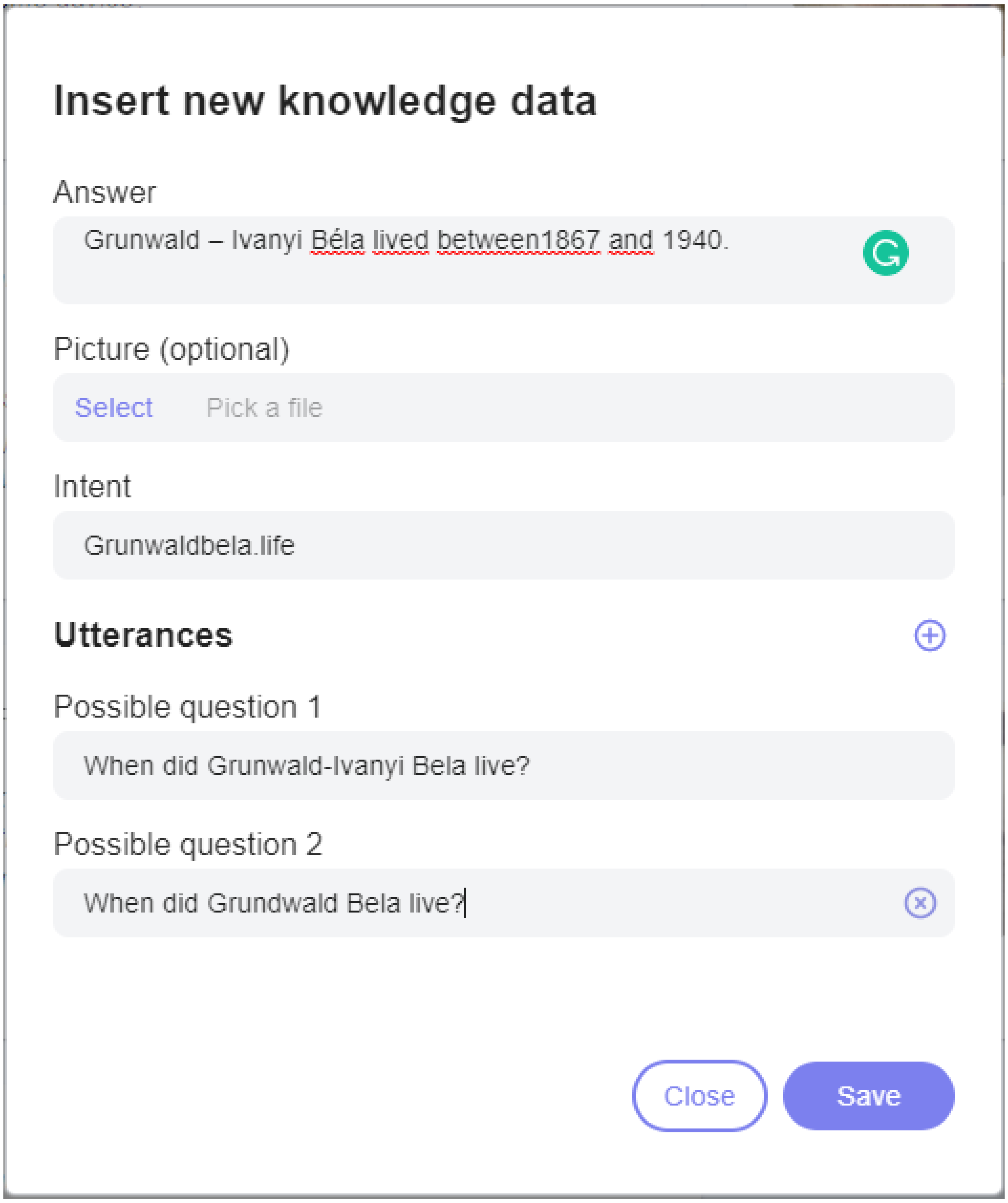

Figure 5.12 Add new entry modal page

A similar modal to the one previously presented is the modal for editing an entry (Figure 5.13) that is opened in case the user pressed the 'Edit' button on the data they want to modify. The difference between this form and the one for adding data is that the edit one will open having all the fields already filled in with the data already stored in the knowledge base. The user can choose to modify any of the fields, including the picture one (by uploading another picture). When all the data has been modified according to the user's desire, the user presses the 'Save' button and all the data in the database is replaced with the new values.

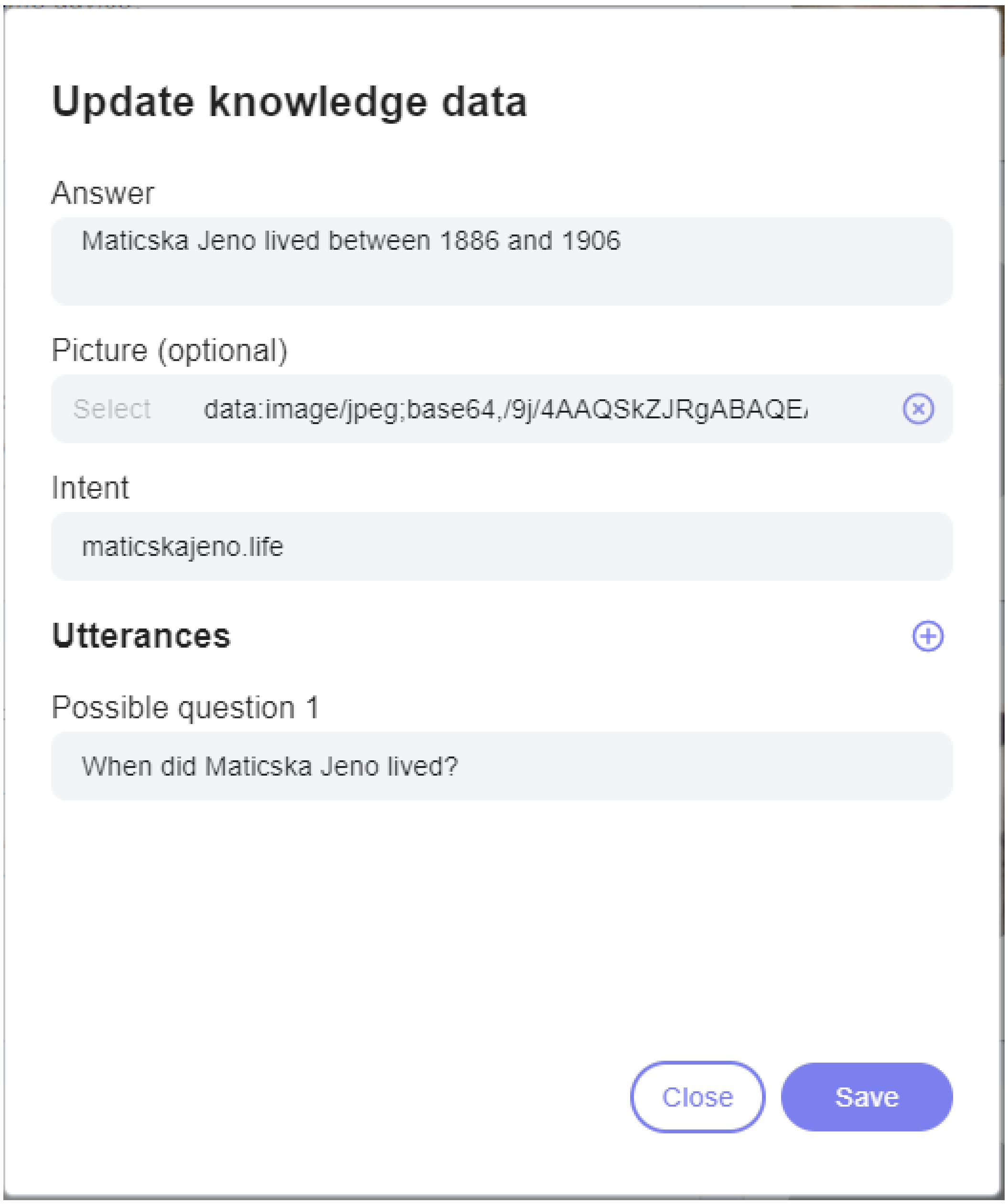

Figure 5.13 Edit entry modal

Another action the user can perform on the existent data is to delete an entry, by simply clicking the ’Delete’ button corresponding to that entry. A confirmation modal (Figure 5.14) is opened, since the scenario of the user deleting entries by accident should be avoided. If the deletion action is indeed intended, the user should click ’Proceed’ and the item will be permanently removed from the database.

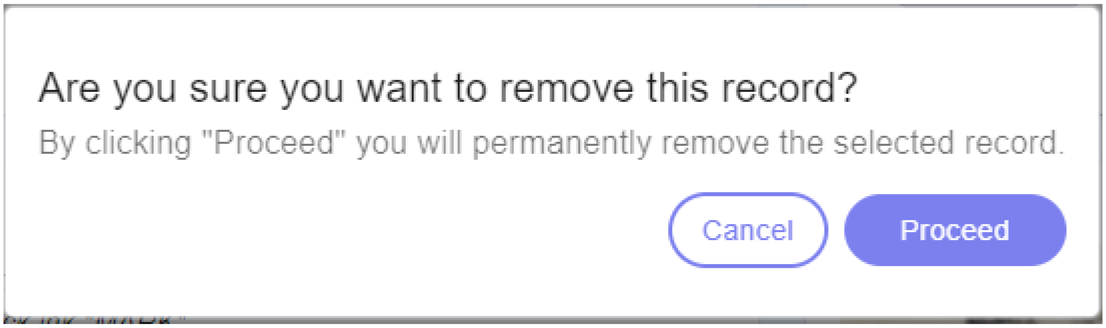

Figure 5.14 Deletion confirmation modal

5.2.3.2. Visitor interface

The part of the application that is intended for the visitors of the museum requires no authentication, making it easier to use for anyone. The design is simple and intuitive, but at the same time modern-looking to make it more engaging for the users. The first page is the welcome page of this part of the application (Figure 5.15).

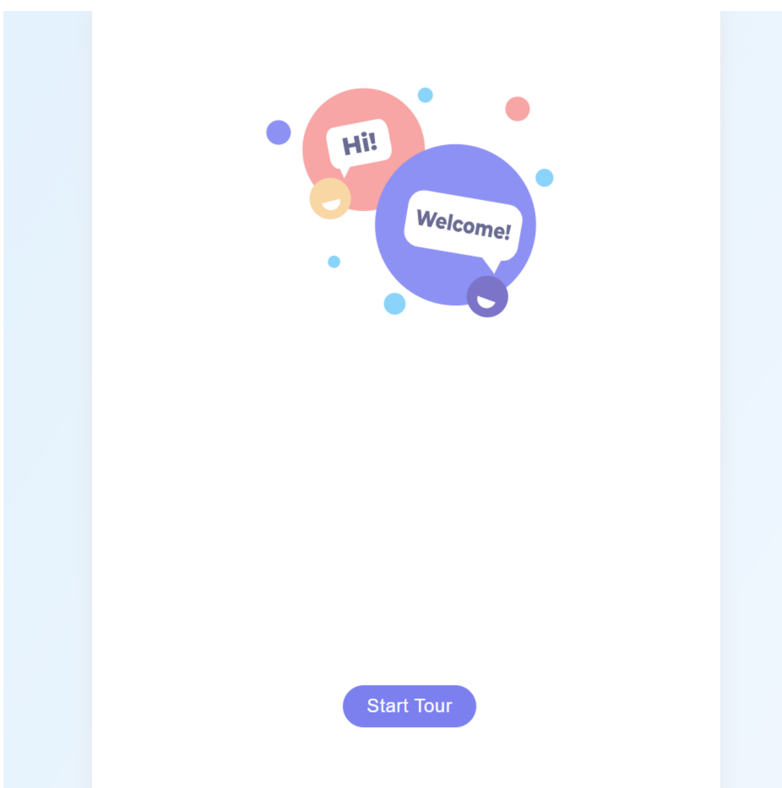

Figure 5.15 The welcome page of the visitor application

The welcome page contains an illustration that is meant to suggest a conversation. The illustration is minimalistic and colorful to match the theme of the application and to urge the visitors to try to engage in conversation with the chatbot. The page also contains the ’Start tour’ button that redirects the user to the main page of the application, the chatroom.

Another feature of this welcome page is the ’Settings’ icon, which opens a modal (Figure 5.16) that offers the user the possibility to customize the experience to their own liking. The possible setting is ’Start listening on Start Tour’ which enables the continuous listening option of the chatbot. If this option is enabled the user can interact with the chatbot by saying the command ’Ok’ followed by the question they want to ask. If this option is disabled, a ’Listen’ icon will appear at the bottom of the chatroom, that when pressed will enable the listening of questions.

The second option is ’Read messages on receive’, that when enable will read the answer messages in the synthesized voice. If this option is disabled, the application will only reply in written text and not in speech.

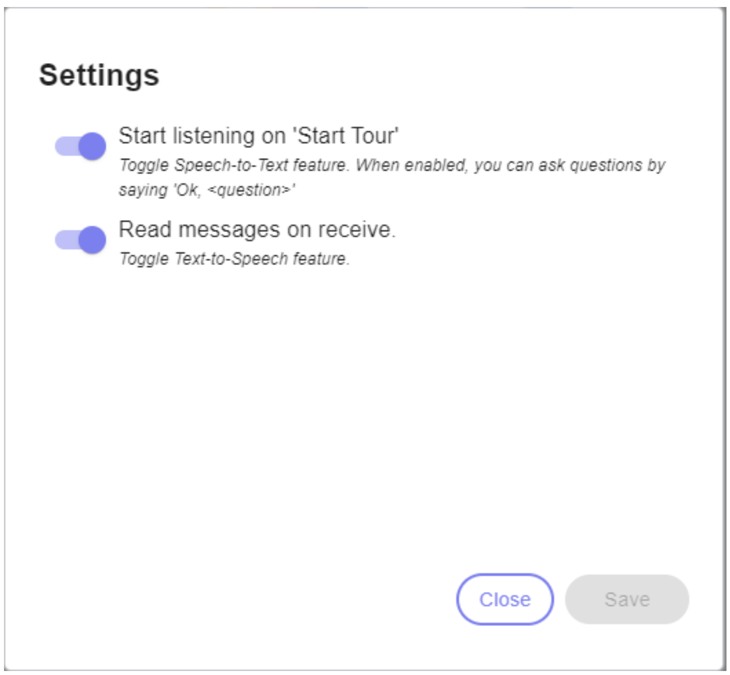

Figure 5.16 Settings modal

The main page of the application is the chatroom (Figure 5.17), where the actual interaction of the visitors with the chatbot happens.

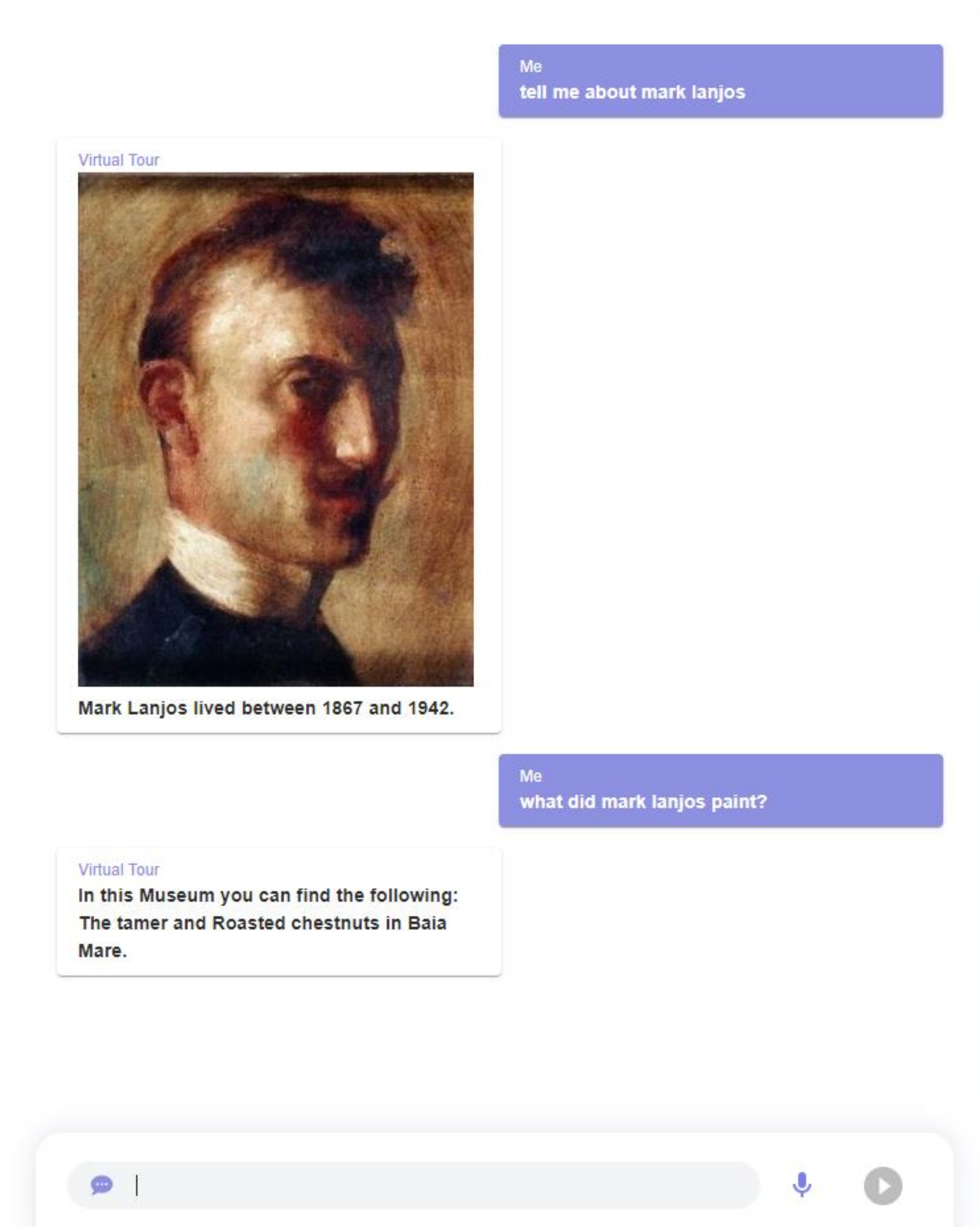

Figure 5.17 Main page of the visitor application

The main page of the visitor application contains a chatroom that displays both the questions asked to the chatbot and the answers it has given in written text. The messages are split in two parts: the part on the right, in purple rectangles are the questions of the user that are inputted either in spoken language (by pressing the ’Listen’ icon and asking the question verbally) and then transcribed into text, either written directly into the textbox at the bottom of the page. The messages on the left part, displayed in white rectangles represent the answers given by the chatbot. They appear in text on the screen and can also be read out loud in synthesized voice if this option is enabled. The answers may be accompanied by pictures, which will be displayed on the screen for the users. In the above picture (Figure 5.17) there are examples of both answers containing only text, as well as an answer containing a picture of the painter Mark Lanjos.

The 'Settings' icon that was presented previously is present at all times in the main page of the application as well in case the user want to try different settings preferences throughout the interaction with the chatbot. If the preferences are modified, the page is reloaded, and the new settings are applied.

5.3. Communication between server and client

5.3.1. Requests over HTTP

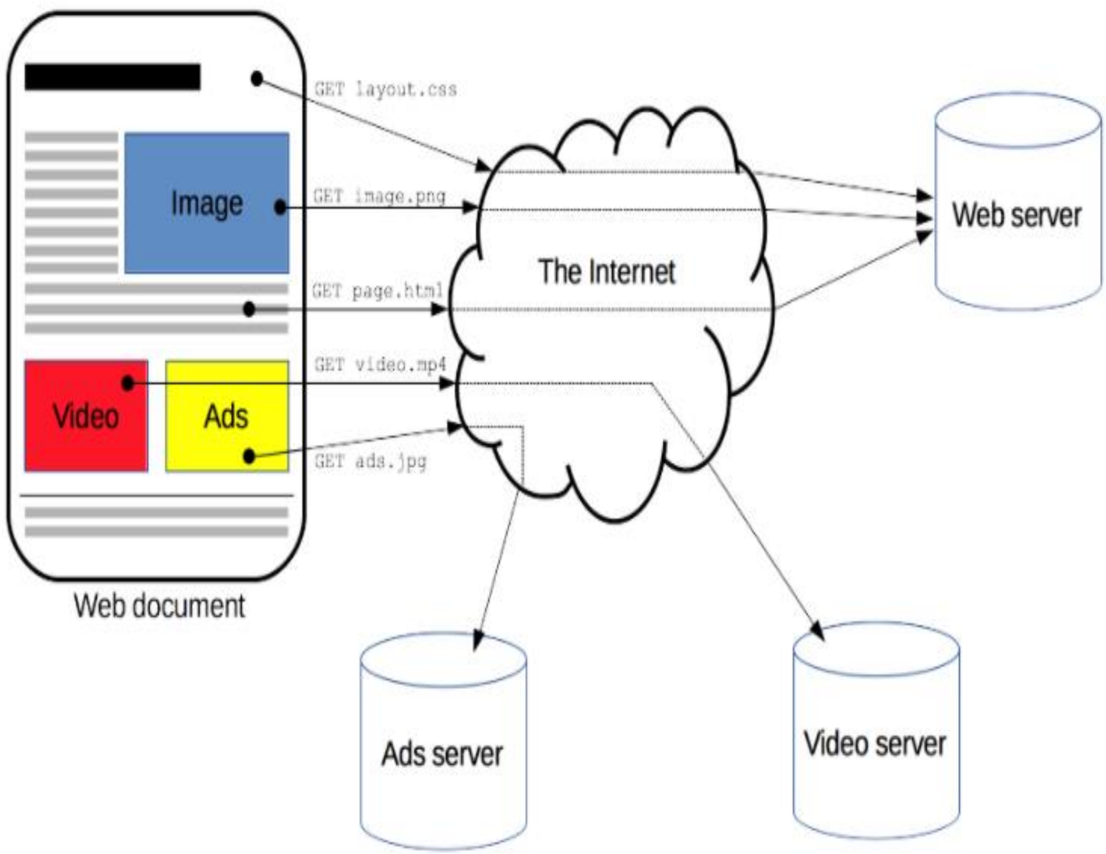

Figure 5.18 HTTP protocol communication [19]

The communication between the client side and the server side of the application is done using the HTTP protocol. Hypertext transfer protocol (HTTP) is a protocol that allows the transfer of resources, such as HTML pages over the internet. The protocol is of type client-server, the communication being usually initiated by the client (the web browser) [19]. The above figure (Figure 5.18) presents an overview of how the transfer of data over HTTP is usually done.

The communication is done using certain methods such as GET and POST. The GET methods is usually used for requesting information from the server, such as an HTML document, while the POST method is used for sending data to the server, such as authentication credentials.

In this application the HTTP requests are done using the Axios package, which is a Node.js module that simplifies the use of HTTP GET and POST methods (these are the only types of requests used in this application). Using Axios the application can execute HTTP calls in a synchronous manner by returning the result of the call as a Promise. The simplicity of making a request of type POST in the application is presented in the figure below (Figure 5.19).

```
axios.post(`${URLS.KNOWLEDGE}${name}/pushData`, data, {
      headers: {
        Authorization: `Bearer ${accessToken}`,
      },
});
```

Figure 5.19 POST request using Axios

The above method is a request of type POST used for sending data to the server to be inserted in the database. The URL for this request is specified as parameter to the post() function, the next parameter is the data to be sent to the server and the last one is the header of the request containing the authorization token, since this is a request that requires the user to be authenticated.

Chapter 6. Testing and Validation

This chapter presents some metrics of the application response time and accuracy of response obtained from experimental results and interaction with users that are using the application for the first time.

6.1. Response time

The response time of the application is represented by the time the application needs to load until it is ready to use. The response time is measured on the user part of the application, since this is the part of the application that takes the longest to load all the additional modules such as NLP. This metric is also important to test if the users will not have to wait a long time to interact with the application.

The response time has been measured 25 times and the results of the measurement are presented in a histogram form in the following figure (Figure 6.1). The values on the horizontal axis represent the response, in milliseconds, while the values on the vertical axis represent the probability of obtaining that response time.

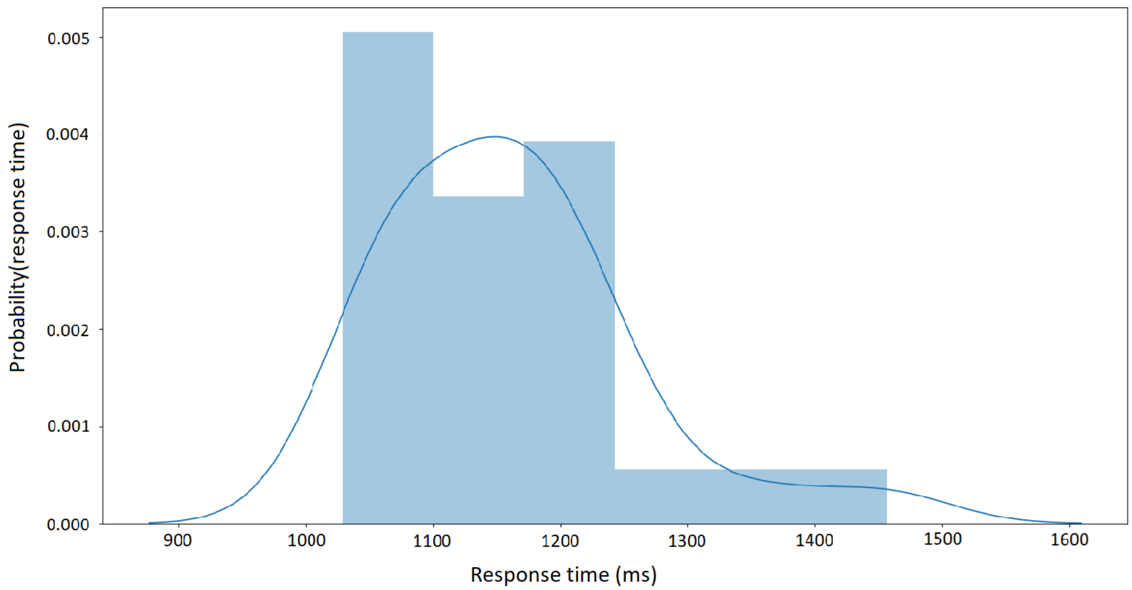

Figure 6.1 Response time of application (ms)

The response time is measured in milliseconds, and from the presented measurements, it has been concluded that the application loads within an average time of 1160 milliseconds (1.16 seconds). The standard deviation of the response time based on the above histogram is 98.45 milliseconds.

6.2. Accuracy of answers

A very important metric in the case of a chatbot is the accuracy of the answers it returns to the users. This is influenced also by its accuracy in understanding the user input in case the communication is done using speech. The following table (Table 6.1) will present the experimental result of the chatbot accuracy in the following situations:

- The input is formulated exactly as it is stored in the knowledge base
- The input is formulated similarly to what is stored in the knowledge base, with a few differences
- The input question is completely rephrased in comparison to what is stored in the knowledge base
- Only a keyword is provided as input

The measurements were performed in two scenarios, first in written communication (the input is typed in the textbox) and secondly in spoken communication (the input is given verbally). For each entry in the table, there will be 10 different measurements and the result will be given as a percentage of correct answers returned by the application.

Table 6.1 Comparison of answer accuracy

	Exact question formulation	Similar question formulation	Rephrased formulation	Keyword
Accuracy (text)	100%	90%	80%	80%
Accuracy (speech)	80%	90%	80%	70%

The next figure (Figure 6.2) is a graphical representation of the results illustrated in the previous table. The chart presents the difference in accuracy of response between different variations of input formulation for written communication. An exact formulation of the input will give a correct answer in 100% of time, but the formulations with varying differences also obtained good results. The rephrased input and keyword input ranked similarly, since the rephrased version of the input also contains some keyword to referentiate the answer that is expected from the chatbot.

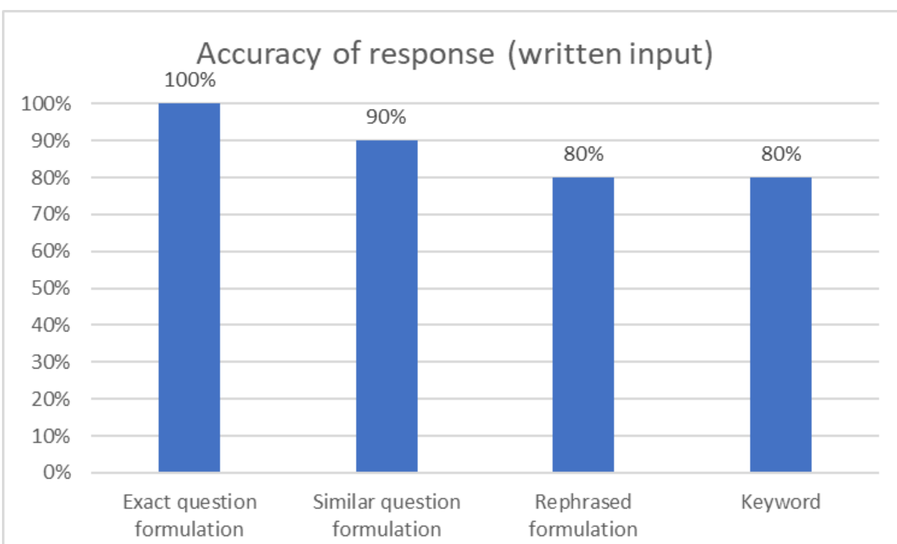

Figure 6.2 Accuracy of answer for written input

The following figure (Fig 6.3) presents the results of the experiments performed for spoken input. In this case, the accuracy of the answer in not only given by the NLP agent, but is also correlated with the accuracy of the speech-to-text component of the application. In order for the chatbot to return the desired answer, it needs to first understand the input and convert it into text so it can be further processed by the NLP module. The result are satisfying, small differences between exact formulation and similar formulation being given by the difference in understanding and converting to text of the input.

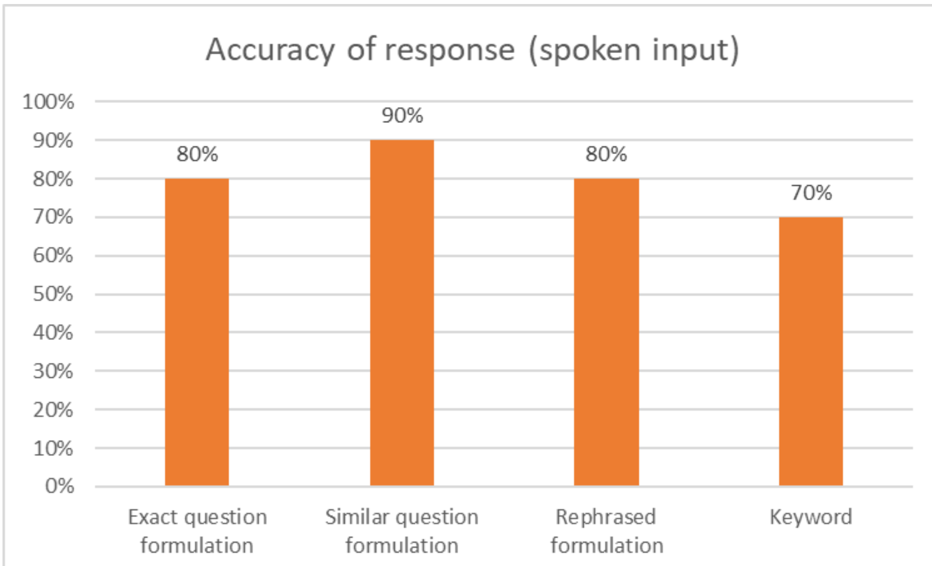

Figure 6.3 Accuracy of answer for spoken input

An in-depth analysis of different formulations of input and how they are understood by the NLP agent is presented in Table 6.2 and Table 6.3. The words that are a part of the formulation stored in knowledge base and can identify a certain intent are perceived as keywords by the agent. For example, “Mark Lanjos” is a keyword, that is enough to identify that the answer should be about Mark Lanjos’ life. However, “When” is not a keyword since it does not contain enough information to identify an intent.

Table 6.2 Asking about the age of a painter

	User question	**Keywords identified**	**Answer**
Exact formulation	*When did Mark Lanjos live?*	*When did Mark Lanjos live?*	*Mark Lanjos lived between 1867 and 1942.*
Similar formulation	*What is the period when Mark Lanjos lived?*	*What is the period when Mark Lanjos lived?*	*Mark Lanjos lived between 1867 and 1942.*
Rephrased formulation	*Give me the year in which Mark Lanjos was born.*	*Give me the year in which Mark Lanjos was born.*	*Mark Lanjos lived between 1867 and 1942.*

	When was the painter of The Tamer born?	*When was the painter of The Tamer born?*	*Mark Lanjos painted "The tamer" using ink on paper in 1897. The dimensions of the painting are 34 by 32 centimeters. It is signed middle-right side with black ink "MARK".*
	When was he born?	*When was he born?*	*Sorry, I don't understand.*
Keyword	*Mark Lanjos*	*Mark Lanjos*	*Mark Lanjos lived between 1867 and 1942.*

Table 6.3 Asking details about a painting

	User question	**Keywords identified**	**Answer**
Exact formulation	*Tell me about Roasted Chestnuts in Baia Mare.*	*Tell me about Roasted Chestnuts in Baia Mare.*	*Mark Lanjos painted "Roasted chestnuts in Baia Mare" using ink on paper in 1897. The dimensions of the painting are 32 by 42 centimeters. It is signed bottom-right with black ink "MARK".*
Similar formulation	*Give me information about Roasted Chestnuts*	*Give me information about Roasted Chestnuts*	*Mark Lanjos painted "Roasted chestnuts in Baia Mare" using ink on paper in 1897. The dimensions of the painting are 32 by 42 centimeters. It is signed bottom-right with black ink "MARK".*
Rephrased formulation	*Did Mark Lanjos paint chestnuts?*	*Did Mark Lanjos paint chestnuts?*	*In this Museum you can find the following: The tamer and Roasted chestnuts in Baia Mare.*
	Artistic work from 1897	*Artistic work from 1897*	*Sorry, I don't understand.*
Keyword	*Mark Lanjos*	*Mark Lanjos*	*Mark Lanjos lived between 1867 and 1942.*
	Roasted chestnuts	*Roasted chestnuts*	*Mark Lanjos painted "Roasted chestnuts in Baia Mare" using ink on paper in 1897. The dimensions of the painting are 32 by 42 centimeters. It is signed bottom-right with black ink "MARK".*

6.3. Interaction with users

A very important aspect of this application is the way it responds in interaction with the users. The users are supposedly people who have not previously interacted with the application and will formulate the questions in a way that is familiar with them, but not necessarily familiar for the chatbot. Since the chatbot cannot be tested in the museum to interact with the visitors due to the museum being closed during the current epidemiological context, the number of users that tested the application is restricted.

The following table (Table 6.4) presents the statistical results of the interaction between first time users and the application, the answers of the application being split in four categories:

- Useful answers, in which the users received exactly the information they were looking for
- Related answer, in which the users received answers on the topic of their question, but not exactly the information they were looking for
- Unrelated answer, in which the users received answers which were not related to the topic of their question
- No answer, meaning that the chatbot did not understand the question and the answer returned was about not understanding the input

In the table, the duration of the interaction of the users with the application (measured in minutes and seconds) as well as the number of questions asked by the user are also documented.

Table 6.4 Statistics of answers given by the chatbot in interaction with users

User	Useful answer	Related answer	Unrelated answer	No answer	Time of interaction	Number of questions
Person 1	66%	0%	17%	17%	1:28	6
Person 2	70%	10%	0%	20%	1:40	10
Person 3	40%	20%	10%	30%	0:59	10
Person 4	80%	0%	0%	20%	0:42	6

From the data presented in the table above (Table 6.4) the accuracy of answers of the chatbot engaging real users seems to vary from as low as 40% correct answers to as high as 80% correct answers. These results are highly dependent on the correctness of pronunciations of the users as well as the other factors mentioned above, such as the accuracy of the speech-to-text module.

The current context made it difficult for the chatbot to be tested in the museum scene. However, it was tested in interaction with different people to simulate an actual museum visit. The only thing left to measure was the interest in such a system being placed in a museum environment and if visitors would be inclined to engage with it. This matter was subjected to test using an on-line survey to which the results are shown in Figure 6.4.

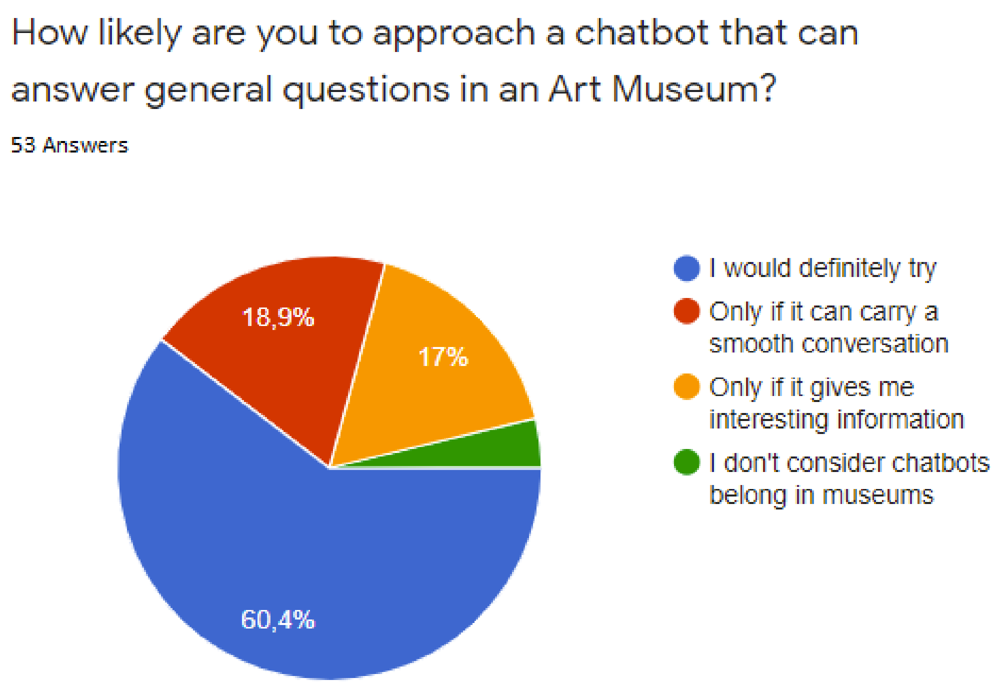

Figure 6.4 Survey regarding interest of people in museum chatbots

Chapter 7. User's manual

The following chapter presents the steps necessary for the installation of the application as well as the steps the user must take in order to use it.

7.1. Installation

This application will run on any Operating System and there are no minimum hardware requirements other than a stable connection to the internet. In order for a machine to run the application, it should have installed the following applications:

- NodeJS version 10.16.3 or later (this will also help install any additional modules needed by the application)
- Git (for downloading the application and later versions if necessary)
- Latest version of MongoDB
- Optionally a NoSQL viewer for manipulating data if necessary (e.g., Robo3T)

The installation of this application is made easier by the fact that all the node modules used in this project are installed automatically when the command 'npm install' is ran. In this way, the user does not have to download and install each dependency individually.

Once all the prerequisite software is installed, to start the application the user will open a *command prompt* (*or terminal*) window in the project location and will run the following commands:

- >*npm install* this line will install of the project's additional dependencies (Figure 7.1)

```
PS C:\Users\ceuca\Desktop\Licenta NLP\licenta> npm install
```

Figure 7.1 Command to install node dependencies

- >*npm run build* to create a production build of the application.

```
PS C:\Users\ceuca\Desktop\Licenta NLP\licenta> npm run build
```

Figure 7.2 Command to build the application

- >*npm run start* to start the production build.

```
PS C:\Users\ceuca\Desktop\Licenta NLP\licenta> npm run start
```

Figure 7.3 Command to run the application

If the application needs to be opened in a development environment, instead of the last two command (*npm run build* and *npm run start*) the user should runt *npm run dev*. This command will start a development version of the application and all the changes to the code will be automatically reflected in the application.

7.2. User manual

To use this application, it is recommended to use the latest version of the Chrome web browser to have full access to all the voice recognition and text to speech functionalities.

The general users of the application should access the URL (localhost:300 in this case since the application is not deployed) using the Web Browser. On the first interaction with the website, and on later ones if desired, the user should access the settings menu located at the bottom right corner (the cog icon button). In the settings menu(Figure 7.4) the user can set the preferred interaction method with the application.

- Start listening on 'Start Tour' controls when the application will listen for voice input. When enabled, the application will listen for the input command and when disabled, the application will listen only on the talk button press.
- Read messages on receive controls if the application will read the answers or not.

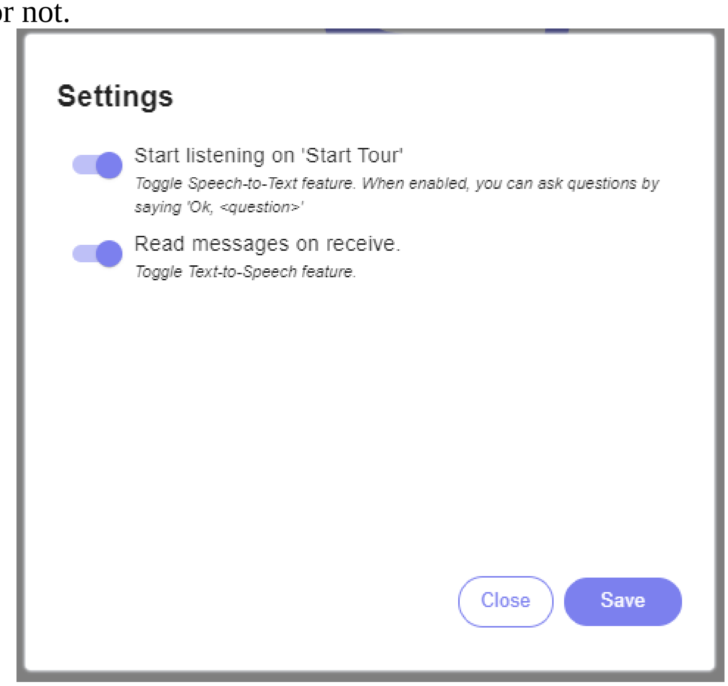

Figure 7.4 Settings menu of the application

After the preferences are set, the user will click the Start Tour button which will initialize the application, and will either listen for voice input, or the user can manually type questions in the lower input bar.

The administrative users will access the URL of the application, but will append to it /mgmt/admin. This URL will open the login page where the provided email and password combination is inserted. If the email and password combination are correct, then the user can add, edit, and remove entries into the database.

- To insert data into the database the user will click the Plus icon button, located in the bottom right corner of the screen, will fill in the information that is requested and will press the Save button to complete the action.

- To edit data, the user will click on the pencil icon button located in the card of the desired entry. This will open a screen similar to the insert one, edit the information there and then complete the action by clicking save.
- To remove data, the user will click the red trash icon button located in the same card as the desired entry. A confirmation screen will pop up and to complete the action the user clicks Proceed.

Chapter 8. Conclusions

The purpose of this project was to develop a safe and engaging way to entertain visitors of the art museum in the current epidemiological context. The need of information, and conversation about exhibits of the people interested in artistic pieces was at the core of developing this chatbot, that serves as an assistant in human interaction during the pandemic, and an entertaining addition to the classical museum setting after it.

The intertwining of technology with the museum setting is not a new idea, this practice being used already in museums all over the world. With this project implemented in the Art Museum of Baia-Mare, this museum will add its name to the list of places people can visit to admire art, learn new information and interact with an experimental piece of technology that is meant to fascinate generations young and old.

The scope of this project consisted in developing an interactive and easy to use application, in which the visitors of the art museum can ask questions and obtain information about the exhibits of the museum, the artists that painted them, the artistic currents they belong to and even see what the painters looked like. The chatbot interacts with the users in a manner that is natural to them, that is in spoken language, using text-to-speech and speech-to-text conversion, but also via text.

The interest of people in such a project was subjected to a survey in which people of all ages and categories were asked if they would be willing to interact with a chatbot in the museum and the feedback obtained from the survey concluded that this will be a good option for engaging visitors and increasing interest in the museum. The accuracy of the application in engaging with real users was subjected to test, but unfortunately due to most of places being closed during the pandemic, the chatbot could not be tested in the museum with a substantial number of visitors.

Several tests were however performed using a limited number of people to demonstrate the accuracy of answers and the validity of information that the chatbot can offer. Other aspects of the application have been subjected to test, such as the response time which is an important performance factor. The results of the tests were promising, and the application is ready to engage a greater audience as soon as it will be possible in the epidemiologic situation.

Future development of the application aims to include a significant increase in the quantity of information the chatbot is able to offer. Besides all the exhibits and collections on display at the museum, the future versions will focus on also giving information about art pieces that are not currently part of the exhibitions and are held in storage. Another important feature that will be the subject of further research, is the ability of the chatbot to entertain more complex conversations with the visitors, to give recommendations based on their preferences, as well as guide them to other rooms of the museum where more paintings of the same artist they were interested in are located.

Another important part of the application is the one designed for the administrator or museum employees, and the process of enriching the knowledgebase will also be the subject of further improvements. To make the process easier, a feature that allows employees to enter information using only speech should become available. In this way, the chatbot should retain the information it hears, store it, and return that information to visitors in a useful context.

Also, the communication with the visitors should take place in multiple languages, to engage a large number of people coming from different countries to visit the museum and its art pieces.

Finally, the current version of the application is fully functional and contains all the features and capabilities that were contained in these specifications. The chatbot awaits its introduction to the museum and interaction with crowds of visitors of different ages. Following its performance in the field further conclusions and improvements ideas will be documented.

Bibliography

[1] Network of European Museum Organisations, “Survey on the impact of the COVID-19 situation on museums in Europe”, 12.05.2020 [Online]. Available: www.ne-mo.org

[2] S. Li, "Named Entity Recognition with NLTK and SpaCy", Towards data science, 2018 [Online]. Available: https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da

[3] Walaa Medhat, Ahmed Hassan, Hoda Korashy, “Sentiment analysis algorithms and applications: A survey”, Ain Shams Engineering Journal, Volume 5, Issue 4, 2014, pp. 1093-1113.

[4] Dinesh Kumawat, "7 Natural Language Processing Techniques for Extracting Information", 2021 [Online]. Available: https://www.analyticssteps.com/blogs/7-natural-language-processing-techniques-extracting-information

[5] D. Molla, J. Vicedo, "Question Answering in Restricted Domains: An Overview" Association for Computational Linguistics, 2006.

[6] L. Mohasi, D. Mashao, “Text-to-Speech Technology in Human-Computer Interaction”, 5th Conference on Human Computer Interaction in Southern Africa, 2006 , pp. 79-84.

[7] P. Khilari, V. P. Bhope, "A Review on Speech to Text Conversion Methods" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), July 2015, Volume 4 Issue 7.

[8] S. Kopp, L. Gesellensetter, N. C. Krämer, I. Wachsmuth, "A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application", Springer Berlin Heidelberg, 2005, pp. 329-343.

[9] S. Boiano, G. Gaia, M. Caldarini, "Make Your Museum Talk: Natural Language Interfaces for Cultural Institutions.", Museums and the Web 2003: Selected Papers from an International Conference, 2003.

[10] G. Gaia, S. Boiano, A. Borda, “Engaging Museum Visitors with AI: The Case of Chatbots”, 2019.

[11] Official website of the Ann Frank House, [Online]. Available: https://www.annefrank.org/en/about-us/news-and-press/news/2017/3/21/anne-frank-house-launches-bot-messenger/

[12] Capture of interaction with a chatbot, 2017 [Online], Available: https://twitter.com/AdrianaRouanet/status/926478566619041792/photo/1

[13] N. Washani, "Speech Recognition System: A Review", International Journal of Computer Applications, Volume 115, April 2015.

[14] N. Yishuang, H.Sheng, W. Zhiyong, X. Chunxiao, Z. Liang-Jie “A Review of Deep Learning Based Speech Synthesis”, 2019.

[15] Auth0 website, [Online]. Available: https://auth0.com/why-auth0

[16] MongoDB website, [Online]. Available: https://www.mongodb.com/

[17] Next.JS website, [Online]. Available: https://nextjs.org/learn/basics/create-nextjs-app

[18] NPM website, [Online]. Available: https://www.npmjs.com/package/node-nlp

[19] MDN Web Docs, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

Safer museum guide interaction during a pandemic and further Using NLP in human interactive museum visits

Museum guide Chatbot

Laura Ceuca
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
ceucalaura@gmail.com

Ana Rednic
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
anarednic@gmail.com

Emil Ștefan Chifu
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
emil.chifu@cs.utcluj.ro

***Abstract*—This paper introduces an NLP based solution for conversational agents in art exhibits. The conversational techniques (text-to-speech and speech-to-text) applied on chatbots are an important issue according to the user needs in different environments, art domains, and regions. The NLP agent is trained on relevant data by matching questions that are partially similar or contain only keywords from the knowledge base, assuring an understanding of a broader spectrum of questions. This paper proposes the theoretical background, the solution implementation, running scenarios, and experimental results as well. The application can be implemented in any kind of exhibition, as long as the knowledge base is adapted to the exhibits in focus.**

Keywords—NLP, chatbot, museum technologies.

I. Introduction

The current epidemiological context has imposed strict human distancing rules that make it difficult for industries based on interaction to follow. An example of such industry is the museums and art galleries, that rely on trained guides to address the curiosities of visitors.

The Art Museum in Baia-Mare has faced a drastic decrease in number of visitors during the pandemic. Strict rules of hygiene were imposed for physical visits. To protect the staff, only a single exposition (10% of the museum) remained open, and guided tours are not available. Only one visitor is allowed in the museum every half an hour and the visit must be scheduled beforehand. The museum came up with online solutions to keep visitors interested. They organized virtual tours, 3D expositions using augmented reality, live broadcast of events, and permanent updates regarding their activities.

This project addresses the issue by creating a software entity that holds knowledge about the exhibits of a certain art museum, the artists, and the artistic currents. This knowledge base is used in a natural language processing model that can answer basic questions of visitors, offering an alternative to the human guide.

II. Related work

A. *Text parsing and interpretation*

There are various techniques that stand at the core of Natural Language Processing and are found in most applications for chatbots. Named entity recognition is a technique of locating and classifying named entities in a text segment and placing them into predefined categories, such as names of people, locations, organizations, quantities, monetary value, etc. [1] Sentiment analysis is a research field of text mining that studies the opinion, attitude, and emotions of a person towards an entity. It is a computational study of the overall sentiment expressed in a text fragment which is then classified. [2] Stemming is the process of reducing an inflected or derived word to their word stem, by removing all its suffixes. Lemmatization uses a morphological analysis of words, aiming to remove inflectional endings in order to obtain the dictionary form of the word, known as lemma. [3] Question answering is a research field concerned with automatic machines being able to answer questions formulated in natural language. [4] A question answering system is able to provide the user specific information regarding a topic of interest, in a real time interaction, relieving the user of the need to traverse long writings in order to find the answer.

The chatbot presented in this paper is mainly a question answering system that uses all NLP techniques and processes mentioned above, in order to parse the questions of visitors and offer them helpful information.

B. *Chatbots based on question answering*

In 2002, in a project to rebuild the whole web site of the Science Museum of Milan, a group of researchers came up with the idea of integrating an on-line chatbot to engage in conversation with visitors after a successful experiment of using real museum staff to chat with pupils. The chatbot was initially designed to answer questions about Leonardo da Vinci, since the museum hosted a big exhibition about his work. Since the chatbot was mainly intended for the younger generation, its language and concepts were kept simple to understand. [5] Speech and listening features were also intended for this project, but they were never implemented since the project was terminated at the end of 2002.

The Anne Frank House bot for Messenger was launched on the International Day for the Elimination of Racial Discrimination by Prince Constantijn of the Netherlands. This chatbot is powered by a deep learning AI and its role is to answer questions, customizing the experience of visiting the House of Ann Frank to each individual, but even more, has the mission of spreading information about the story of Ann Frank and warn the users of the risks and effects of racism and discrimination. [6]

The House Museums of Milan is composed of four historical homes, belonging to different styles and time periods ranging between the 19th and 20th centuries. Because of the lack of space for curatorial placards, in 2016 they came with an initiative to attract younger generations: a chatbot game. The chatbot is design to interact with the visitors over Facebook Messenger and take them on a "treasure hunt" game regarding the exhibits of the museums. The chatbot asks users different questions about the exhibits, that are meant to engage them in a vigilant observation of the objects. [7]

C. *Chatbots based on machine learning*

Max is an Embodied conversational agent (ECA) developed by a group of A.I. researchers from Germany. It started as a platform for studying the generation of natural multimodal behavior and has then been extended to a virtual assistant. Since January 2004, its task is to engage visitors of the Heinz Nixdorf Museums Forum (HNF) in conversation, providing them with interesting information about the museum, the exhibitions, and many others. Max receives input from the visitor via keyboard and replies using a synthetic voice. The agent also communicates through gestures, facial expressions, and locomotion to create a more natural interaction. [8]

Table I presents a comparison on speech techniques between the above-mentioned projects.

TABLE I. COMPARISON BETWEEN PRESENTED CHATBOTS

Agent	Conversational or question answering?	Communicates through speech or text?	Uses learning to improve its capabilities?
Max	Holds conversation	Speech	Yes
Leonardo project	Answers questions	Text	No
House Museums of Milan	Asks and answers questions	Text	No
Anne Frank House bot	Answer questions	Text	No

III. SYSTEM ARCHITECTURE

The proposed solution is composed of the following modules, depicted in the block diagram from Fig. 1:

The client side of the application contains:

- the user interface, that ensures a friendly interaction with the chatbot;
- the NLP agent which processes questions and finds the right answer to return;
- the web speech API, which does the conversion of speech-to-text and text-to-speech to facilitate communication in spoken language.

The server side contains:

- the authentication service to secure parts of the application that should be only accessible to certain users;
- the application routing;
- the database that stores all information required by the application.

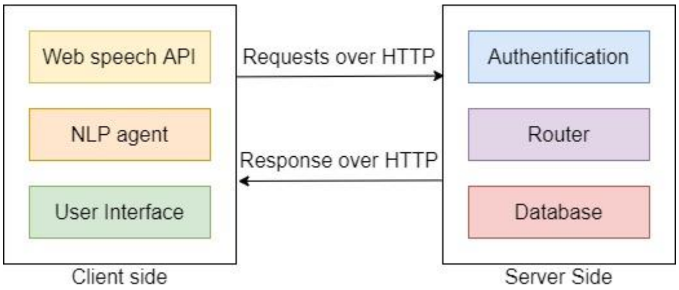

Fig. 1. System architecture

A. *NLP agent training*

The NLP agent used by the application is trained on the available data in order to match questions that are similar (or contain only keywords) to the ones existing in the knowledge base. The structuring of the knowledge base groups multiple questions under a unique intent. The intent is identified by the chatbot using certain keywords or phrases that belong to it, and the answer is then selected based on the identified intent. In this way, a broader spectrum of questions is recognized by the application, without the need to introduce many different versions of the same question in order to cover all possible formulations of users. The training is done when the application is loaded, on available data or data received from the server. The algorithm for training the NLP agent and preparing it for answering questions is presented in Fig. 2.

For the implementation of the NLP agent, the library used was NLP.js. Some of the features used for this project that were offered by the NLP library are: stemming and tokenization of words, which helps the chatbot understand certain words in any form they are found using the word stem, named entity recognition (NER), which finds certain named entities in the text, such as names of people, places or paintings, natural language processing classifier, that aids grouping different utterances into a common intent, and natural language generation manager, giving it the ability to generate an answer based on intent and certain conditions. [9] NLP.js library supports 40 different languages, but this project only makes use of the English language capabilities.

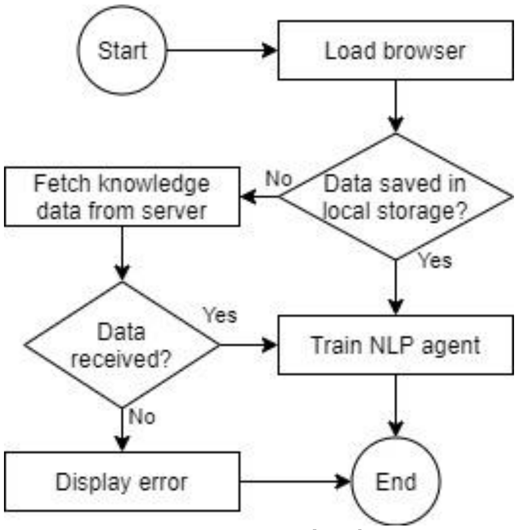

Fig. 2 NLP training algorithm

B. Speech recognition and synthesis

The ability to converse in spoken words is a novel feature that distinguishes the chatbot discussed in this research from others that are already in use in museums throughout the world. Users are already accustomed with this sort of communication because it is commonly used in human-to-human interactions. For example, if a visitor in a museum seeked information from a human guide, they would most likely ask a question verbally. This is the interaction that the proposed application is attempting to replicate.

1) Speech recognition

Speech recognition has already been a topic of interest for humankind since it offers the possibility of controlling devices by voice. It is already integrated in a large number of devices, varying from different domains and has been widely researched in recent years.

Speech recognition is described as the process of converting acoustic speech into text. This is a challenging field of research because it must consider many outside factors, such as noise, speaker variability and language variability [10]. The high-level representation of a speech recognition system is presented in Fig. 3.

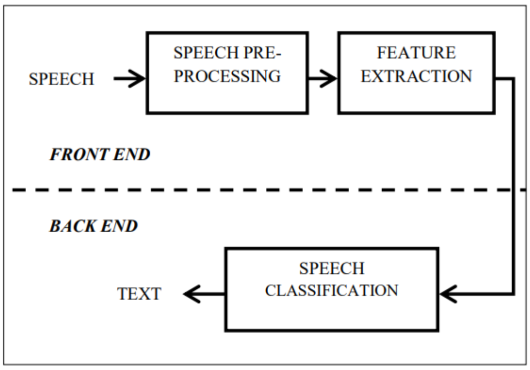

Fig. 3 Speech recognition system [10]

Signal preprocessing deals with noise, differences in amplitude and timing variations between samples [10]. The process of pre-processing is an essential prerequisite of data analysis since it removes detected errors and prepares the signals for processing. Feature extraction is the process of extracting certain features of the signals being processed [10], while speech classification is the process of classifying the extracted features by matching them to the closest sound that is already existent in a database.

2) Speech synthesis

Speech synthesis, or text-to-speech (TTS) is a field comprised of many disciplines, such as acoustics, linguistics, digital signal processing and statistics. [11] The purpose of speech synthesis is to convert written text into speech.

There are many known technologies used along the years for achieving speech synthesis, from formant based parametric synthesis, to waveform concatenation-based methods and more recently statistical parametric speech synthesis. [11] Along with the change of technology there has been a significant increase in how well the synthesized speech is understood by humans and how natural it sounds. However, the expressivity of human speech could not have been captured in artificial language.

C. User interface

The application is divided into two parts: one for museum staff or administrators, which allows for easy maintenance of the information stored in the knowledge base, and the other for museum visitors who interact with the program to learn more about the exhibits. Since the application is designed for people with different levels of technological experience, it must be straightforward and simple to use. The user interface has a minimalistic design, and all actions are marked with intuitive icons or extra information in case it is needed.

1) Administrator interface

After a successful authentication, the user is redirected to the main administrator page, that displays all information currently stored in the knowledge base of the application, along with possible actions for these entries, such as delete, edit, and the option to add new entries and to persist them into storage.

2) Visitor interface

The part of the application that is intended for museum visitors requires no authentication. The design is simple and intuitive, but at the same time modern-looking to make it more engaging for users. The welcome page for the visitors' interface can be seen in Fig. 4.

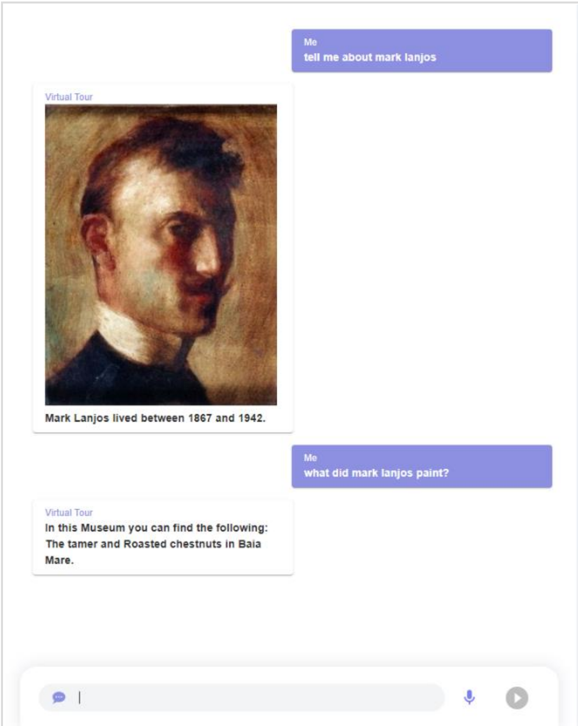

Fig. 4 Main page of the visitor application

The visitor application's main page contains a chatroom that displays both the questions posed to the chatbot and the written responses it has provided. The messages are split in two parts: the part on the right, in purple rectangles are the questions of the user, and the messages on the left part, displayed in white rectangles represent the answers given by the chatbot. They appear in text on the screen and can also be read out loud in synthesized voice if this option is enabled. The answers may be accompanied by pictures, which will be displayed on the screen for a visually enriched experience. In Fig. 4 there are examples of written answers, as well as a picture of the painter Mark Lajos.

IV. EXPERIMENTAL RESULTS

A. Response time

The application response time is represented by the time it needs to load until the user can start the interaction. The response time is measured on the user side, since this is the part of the application that takes the longest to load all additional modules such as NLP. The measurement results are presented in a histogram form in Fig. 5. The response time is measured in milliseconds, and from the presented measurements, it has been concluded that the application loads within an average time of 1160 milliseconds (1.16 seconds). The standard deviation of the response time based on the above histogram is 98.45 milliseconds.

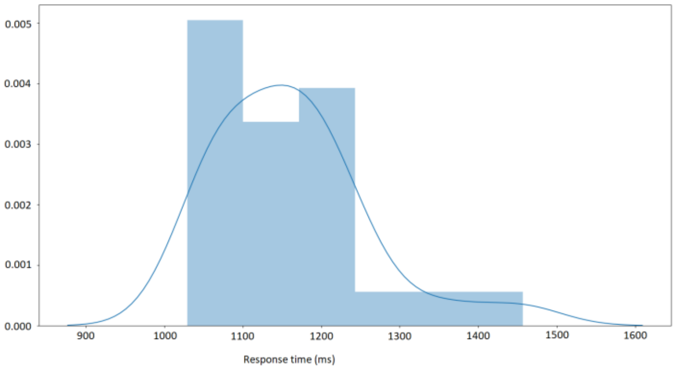

Fig. 5 Application response time

B. Accuracy of answers

A very important metric in the case of a chatbot is the accuracy of the answers it returns to the users. This is influenced also by its accuracy in understanding the user input in case the communication is done via speech. Table II presents the chatbot accuracy in the following situations:

- The input is formulated exactly as it is stored in the knowledge base.
- The input is formulated similarly to what is stored in the knowledge base, with a few differences.
- The input question is completely rephrased in comparison to what is stored in the knowledge base.
- Only a keyword is provided as input.

The measurements will be performed in two scenarios, first in written communication (input is typed in the textbox) and secondly in spoken communication (input is given verbally). For each entry in the table, there will be 10 different measurements and the result will be given as a percentage of correct answers returned by the application.

TABLE II. COMPARISON OF ANSWER ACCURACY

Accuracy	Exact question formulation	Similar question formulation	Rephrased formulation	Keyword
Text	100%	90%	80%	80%
Speech	80%	90%	80%	70%

Fig. 6 is a graphical representation of the results illustrated in the previous table. The chart presents the difference in response accuracy between different variations of input formulation for both text and written communication. The rephrased input and keyword input ranked similarly, since the rephrased version of the input also contains some keyword to refer the answer that is expected from the chatbot. The results are satisfying, small differences between exact phrasing and similar formulation of questions being given by the difference in understanding and converting the input to text.

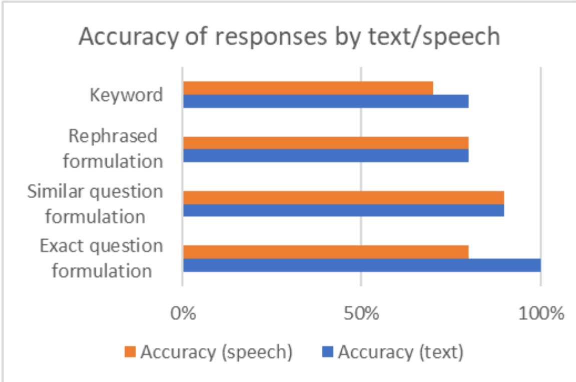

Fig. 6 Accuracy of answers for different types of input

C. Interaction with users

A very important aspect of this application is the way it reacts in interaction with users. The users are supposedly people who have not previously interacted with the application and will formulate questions in a way that is familiar to them, but not necessarily familiar to the chatbot.

Table III presents the statistical results of the interaction between first time users and the application. The answers are split in four categories:

- useful answers, in which users received exactly the information they were looking for;
- related answers, in which users received answers on the topic of their question, but not exactly the information they were looking for;
- unrelated answers, in which users received responses which were not related to the topic of their question;
- no answer, meaning that the chatbot did not understand the question and the answer returned was related to its inability of understanding the input.

Table III documents the duration of users' interaction with the application (measured in minutes and seconds) and the accuracy of answers split in the four categories mentioned above.

TABLE III. STATISTICS OF ANSWERS IN INTERACTION WITH USERS

Test	Useful answer	Related answer	Unrelated answer	No answer	Time spent
1	66%	0%	17%	17%	1:28
2	70%	10%	0%	20%	1:40
3	40%	20%	10%	30%	0:59
4	80%	0%	0%	20%	0:42

From the data presented in Table III, the accuracy of answers of the chatbot engaging real users seems to vary from as low as 40% correct answers to as high as 80% correct answers. These results are highly dependent on the correctness of users' pronunciations as well as the other factors mentioned above, such as the accuracy of the speech-to-text module.

The current context made it difficult for the chatbot to be tested in the museum scene. However, it was tested in interaction with various people to simulate an actual museum visit. The only thing left to measure was the interest of people having different social and technological backgrounds for such a system being placed in a museum environment, and if visitors would be inclined to engage with it. This matter was subjected to test using an on-line survey to which the results are shown in Fig. 7.

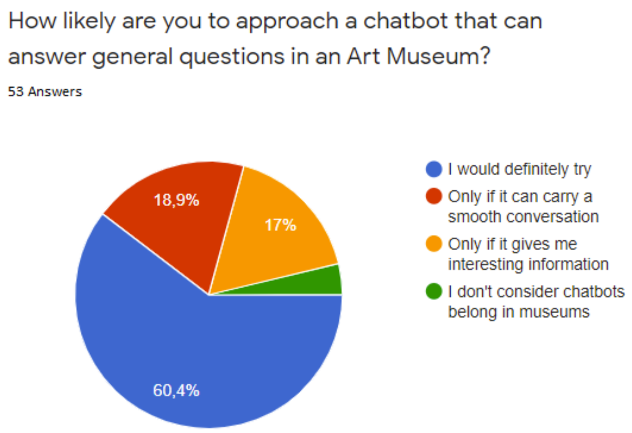

Fig. 7 Survey regarding interest of people in museum chatbots

V. CONCLUSIONS

The purpose of this project was to develop a safe and engaging way to entertain visitors of an art museum in the current epidemiological context. The need of information and conversation about exhibits among individuals interested in artistic pieces was at the core of developing this chatbot, which serves as an assistant in human interaction during the pandemic, and an entertaining addition to the classical museum setting afterward.

The intertwining of technology with the museum setting is not a new idea, this practice being used already in museums all over the world. With this project implemented in the Art Museum of Baia-Mare, this museum will add its name to the list of places people can visit to admire art, learn new things, and interact with an experimental piece of technology that is meant to fascinate generations young and old.

The scope of this project consisted in developing an interactive and easy to use application, in which visitors of the art museum can ask questions and obtain information about the exhibits of the museum, the artists that painted them, the artistic currents they belong to and even see what the painters looked like. The chatbot interacts with users in a manner that is natural to them, i.e. spoken language, using text-to-speech and speech-to-text conversion, but also via text.

The interest of people in such a project was subjected to a survey in which people of all ages and categories were asked if they would be willing to interact with a chatbot in a museum and the feedback obtained from the survey concluded that this will be a good option for engaging visitors and increasing interest in the museum. The accuracy of the application in engaging with users was tested, but unfortunately due to most places being closed during the pandemic, the chatbot could not be tested in the museum with a substantial number of visitors.

Several tests were however performed using a limited number of people to demonstrate the accuracy of answers and the validity of information the chatbot can offer. Other aspects of the application have been subjected to test, such as the response time, which is an important performance factor. Tests results were promising, and the application is ready to engage with a greater audience as soon as it will be possible in the epidemiologic situation.

REFERENCES

[1] S. Li, "Named Entity Recognition with NLTK and SpaCy," Towards Data Science, 17 August 2018. [Online]. Available: https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da. [Accessed 13 March 2021].

[2] A. H. H. K. Walaa Medhat, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal,* vol. 5, no. 4, pp. 1093-1113, 2014.

[3] D. Kumawat, "7 Natural Language Processing Techniques for Extracting Information," Analytics Steps, 18 November 2021. [Online]. Available: https://www.analyticssteps.com/blogs/7-natural-language-processing-techniques-extracting-information. [Accessed 21 March 2021].

[4] J. L. V. Diego Mollá, "Question Answering in Restricted Domains: An Overview," *Computational Linguistics,* vol. 33, no. 1, pp. 41-61, 2007.

[5] G. G. M. C. Stefanie Boiano, "Make Your Museum Talk: Natural Language Interfaces for Cultural Institutions," in *Archives & Museum Informatics*, Charlotte, North Carolina, USA, 2003.

[6] A. F. House, "Anne Frank House launches bot for Messenger," Anne Frank House, 21 March 2017. [Online]. Available: https://www.annefrank.org/en/about-us/news-and-press/news/2017/3/21/anne-frank-house-launches-bot-messenger/. [Accessed 24 February 2021].

[7] S. B. A. B. Giuliano Gaia, "Engaging Museum Visitors with AI: The Case of Chatbots," in Museums and Digital Culture, Springer Series on Cultural Computing, 2019, pp. 309-329.

[8] L. G. N. C. K. I. W. Stefan Kopp, "A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application," in Intelligent Virtual Agents, 5th International Working Conference, Kos, Greece, 2005.

[9] NPM, "node-nlp," AXA Group Operations Spain S.A., February 2021. [Online]. Available: https://www.npmjs.com/package/node-nlp. [Accessed 4 April 2021].

[10] N. Washani, "Speech Recognition System: A Review," *International Journal of Computer Applications,* vol. 115, no. 18, 2015.

[11] S. H. Z. W. C. X. L.-J. Z. Yishuang Ning, "A Review of Deep Learning Based Speech Synthesis," *Applied Sciences,* vol. 9, no. 19, 2019.