Software Requirements Specification Document for Computer-aided Simultaneous Interpreting Software

Abdelrahman Ashraf
Ahmed Sameh
Mostafa Mahran
Wessam Ahmed

Supervised by

Dr. Ashraf Abdelraouf
Eng. Nada Shorim

November 2018

1 Introduction

Language interpreters rely on a number of methods for the easy look-up and translation of words, from glossary search engines to speech-to-text transcription software. The increasing complications that come with the use of these software makes their adoption harder to achieve on a wider scope, so their use is limited to diminishing groups within the language services industry. This document will explore the use of an artificially intelligent computer system for the transcription, semantic (natural) understanding, and translation of live speech using proven methods.

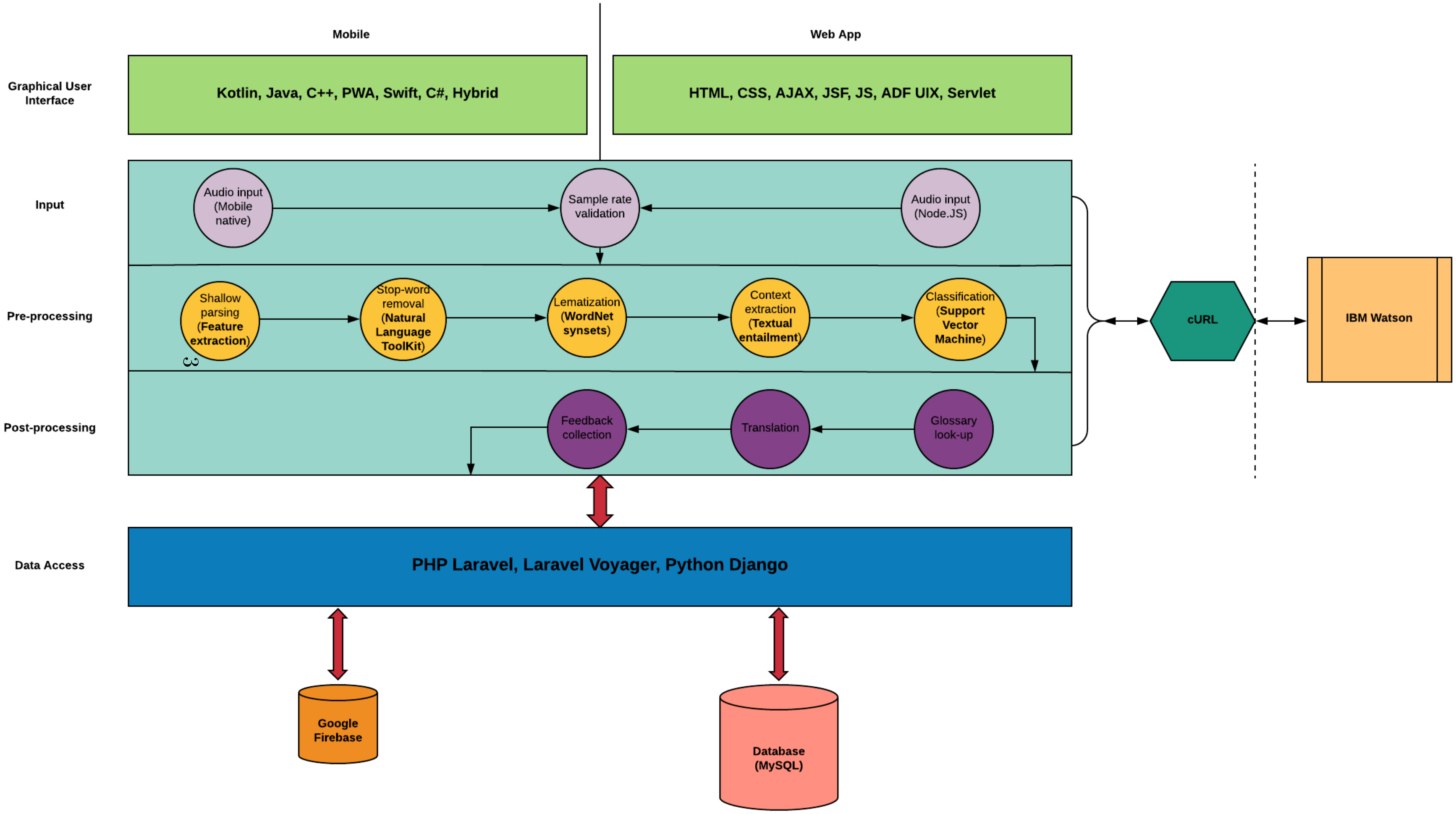
1.1 Purpose of this document

This document specifies a simultaneous linguistic interpretation system that relies on audio input (speech) for outputting translated text between Arabic and English in a bidirectional manner. The system is a simultaneous interpretation software designed to replace the role of language interpreters in transcribing and translating text during conferences, such as in a political and diplomatic situations, or other media units. It includes the added functionality of the semantic understanding of text through the use of artificial intelligence. The document specifies preliminary requirements set by the client (Al Alsun) to be used as a reference throughout the development process.

1.2 Scope of this document

The system described in this document builds on existing standards set by other language translation and interpretation software available on the market. It provides benefits for users in the Middle East and North Africa region through the use of advanced methods for context/domain specification in Arabic (and English) text. The accuracy of translation depends on the semantic understanding of words in context and not vocabulary translation alone through the use of a dictionary. Optimization benefits are gained by minimizing the number of terms the system has to look through in order to arrive to an accurate translation, resulting in cost savings. The system can be added to an existing network of components that perform tasks it is dependant on or exist independently, the limitations that drive this choice are based on implementation and network configuration, as it can exist solely to provide users with a display for contextual extraction or translation as well as domain specification.

1.3 Overview


Mobile
Web App
Graphical User Interface
Kotlin, Java, C++, PWA, Swift, C#, Hybrid
HTML, CSS, AJAX, JSF, JS, ADF UIX, Servlet
Input
Audio input (Mobile native)
Sample rate validation
Audio input (Node.JS)
Pre-processing
Shallow parsing (Feature extraction)
Stop-word removal (Natural Language ToolKit)
Lematization (WordNet synsets)
Context extraction (Textual entailment)
Classification (Support Vector Machine)
Post-processing
Feedback collection
Translation
Glossary look-up
cURL
IBM Watson
Data Access
PHP Laravel, Laravel Voyager, Python Django
Google Firebase
Database (MySQL)

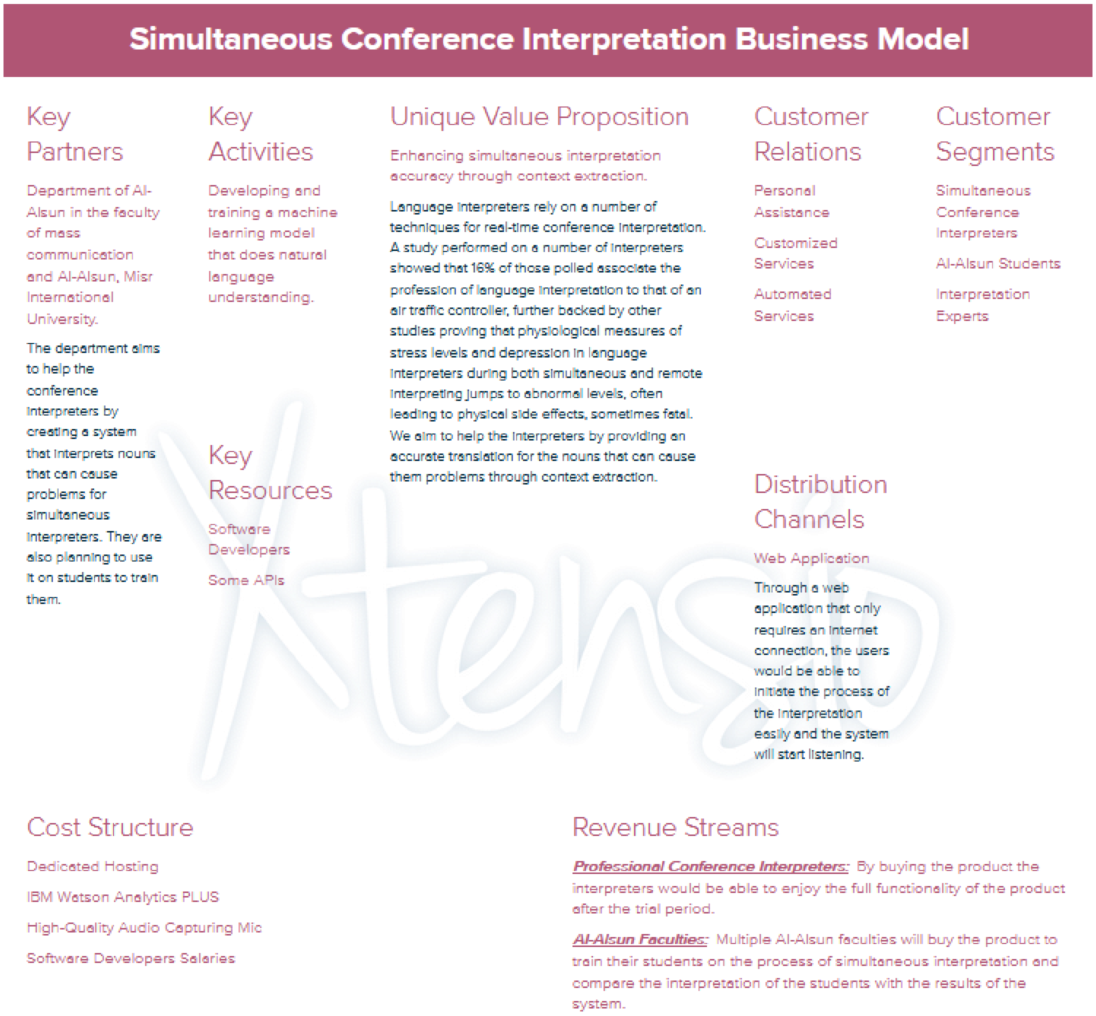
The system relies on third-party software for the task of translation as well as speech transcription in the form of IBM's Watson question-answering computer system, and will perform the task of context extraction (glossary) through the use of deep learning methods that enable it to have a natural understanding of both English and Arabic text. The reader should be able to deduce a number of key points through this document, which are outlined below.

- Preliminary requirements set at the time of writing this document
- All forms of supporting software used for the creation of the system
- Material required for total project completion

The system will see usage within the Alsun department at the Misr International University, as such, its completion is subject to the the work of both the Computer Science and Alsun departments. The Alsun department will be responsible for gathering a glossary of terms in both English and Arabic that will be used by the system for translation as well as natural understanding. As such, this document will provide a road map for fulfilling the requirements laid out by the Alsun department at the Misr International University.

1.4 Business Context

Glossary search engines are available for language interpreters to use in order to quickly look up terms they may not understand during a simultaneous interpretation session. This process is time-consuming and provides an additional challenge for interpreters who are usually in a hurry to provide a translation for part of a speech and move on to the next. The system in question is designed to replace the interpreter's task in two main aspects: transcription (writing down the audible speech), and translation. It performs these task in succession using shallow deep learning methods for natural language understanding. Such systems are available on the market in the form of individual components that do not perform these tasks together, clearing a vacancy in the language services industry for a piece of software that does away with interpreters' difficulties and provides them with an automated alternative. The project is sponsored by the department of Al-Alsun in the faculty of Al-Alsun at the Misr International University.



Simultaneous Conference Interpretation Business Model

Key Partners

Department of Al-Alsun in the faculty of mass communication and Al-Alsun, Misr International University.

The department aims to help the conference interpreters by creating a system that interprets nouns that can cause problems for simultaneous interpreters. They are also planning to use it on students to train them.

Key Activities

Developing and training a machine learning model that does natural language understanding.

Key Resources

Software Developers

Some APIs

Unique Value Proposition

Enhancing simultaneous interpretation accuracy through context extraction.

Language interpreters rely on a number of techniques for real-time conference interpretation. A study performed on a number of interpreters showed that 16% of those polled associate the profession of language interpretation to that of an air traffic controller, further backed by other studies proving that physiological measures of stress levels and depression in language interpreters during both simultaneous and remote interpreting jumps to abnormal levels, often leading to physical side effects, sometimes fatal. We aim to help the interpreters by providing an accurate translation for the nouns that can cause them problems through context extraction.

Customer Relations

Personal Assistance

Customized Services

Automated Services

Distribution Channels

Web Application

Through a web application that only requires an internet connection, the users would be able to initiate the process of the interpretation easily and the system will start listening.

Customer Segments

Simultaneous Conference Interpreters

Al-Alsun Students

Interpretation Experts

Cost Structure

Dedicated Hosting

IBM Watson Analytics PLUS

High-Quality Audio Capturing Mic

Software Developers Salaries

Revenue Streams

Professional Conference Interpreters: By buying the product the interpreters would be able to enjoy the full functionality of the product after the trial period.

Al-Alsun Faculties: Multiple Al-Alsun faculties will buy the product to train their students on the process of simultaneous interpretation and compare the interpretation of the students with the results of the system.

Figure 2: Business Model

2 General Description

2.1 Product Functions

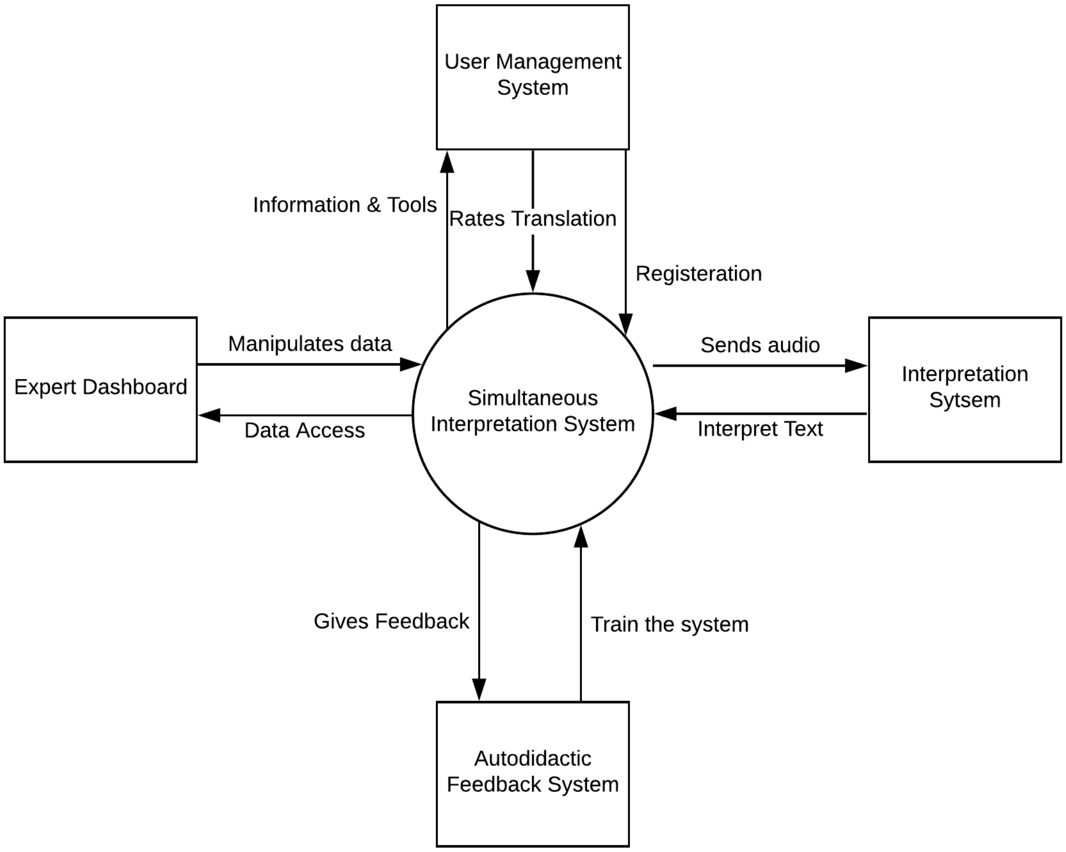


Figure 3: Context Diagram

-Interpretation System :

1. Receive the audio stream
2. Audio pre-processing
3. Speech-to-text transcription
4. Context extraction
5. Extract important nouns

6. Translate into target language

-Expert Dashboard System:

1. data manipulation
2. Review Interpreter Feedback
3. Update new words

-User management system :

1. Enrollment
2. User authentication
3. Editing existing users
4. Removing an existing user
5. Adding a new user
6. Assigning user role(s)

-Auto didactic Feedback system:

1. Receive interpreter feedback
2. Train the system through deep learning methods

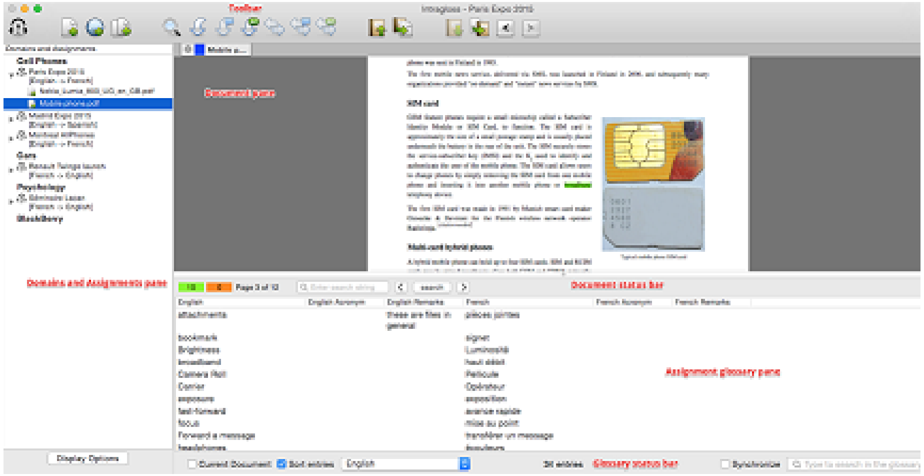
2.2 Similar System Information

2.2.1 TAKMI

The Text Analysis and Knowledge Mining [4] toolkit was created by IBM's Tokyo Research Laboratory in 1997 as part of the quest for more structured data within corporations. Advances in natural language processing (NLP) at the time didn't allow for meaningful analysis of text, instead treating any English text as a bag of words. This meant that data analysts often dealt with the English language by tokenizing words based on white space. Considering Japanese does not have any whitespace, TAKMI was created in order to mine text in different languages and extract meaningful information through the grammatical structure of each sentence as well as identifying relationships between different words[4].

2.2.2 Intragloss

Intragloss [5] is a collection of tools that enable interpreters to compare documents, create multilingual glossaries, as well as look up and search for equivalent terms. Intragloss is the work of a conference interpreter, Dan Kreig, and a software developer, Daniel Pohoryles. It exists alongside other options that perform much of the same task, Interpreter's Help, Interpret Go, and other glossary collection suites. The purpose of Intragloss is to cut preparation time for interpreters before live/simultaneous/conference interpretation sessions in the booth, and provide them with a quick toolkit through their PDA, tablet, or smartphone.



English	English Acronym	English Remarks	French	French Acronym	French Remarks
attachments		these are files in general	pièces jointes		
bookmark			signet		
Brightness			Luminosité		
broadband			haut débit		
Camera Roll			Pellicule		
Carrier			Opérateur		
exposure			exposition		
fast-forward			avance rapide		
focus			mise au point		
Forward a message			transférer un message		
headphones			écouteurs		

Figure 4: Interagloss user interface

This tool offering comes at a significant cost for interpreters, with subscription plans ranging from $49 per month to $269 per year.

2.3 User Characteristics

Users of the system can be separated into two main classes by the time of its deployment.

Academics will use the system for objectives relevant to their subdomain(s). For example, college instructors will use the system for conference interpretation classes that see students interacting with video or audio feeds from real conferences, and translate them accordingly. These academics will have no technical background or knowledge of how the system operates but will expect it to perform according to domain requirements and specifications for optimal results. Students may also be placed under this category.

Professional Conference Interpreters that will use the system as their main tool during their work, the system will help them through the interpretation process and extract for them the translation of the relevant nouns that they might get clumsy with.

Experts will use the system for live/simultaneous interpretation sessions, where they will require it to perform a number of tasks for them. They will have no knowledge of how the different system modules operate but will expect it to perform in a manner that does not interfere with external domain specifications irrelevant to the implementation of the system itself.

2.4 User Problem Statement

Existing systems do not solve the problem of semantic interpretation of text for domain specification in that the the accuracy of obtaining textual entailment for the English language is not sufficient for efficient semantic interpretation and needs to be matched in the case of the Arabic language.

2.5 User Objectives

The successful semantic interpretation and translation of text is expected of the system. Accuracy and speed are important considerations in the overall design of the system, and are more delved into within the rest of this document.

3 Functional Requirements

3.1 Recognize speech

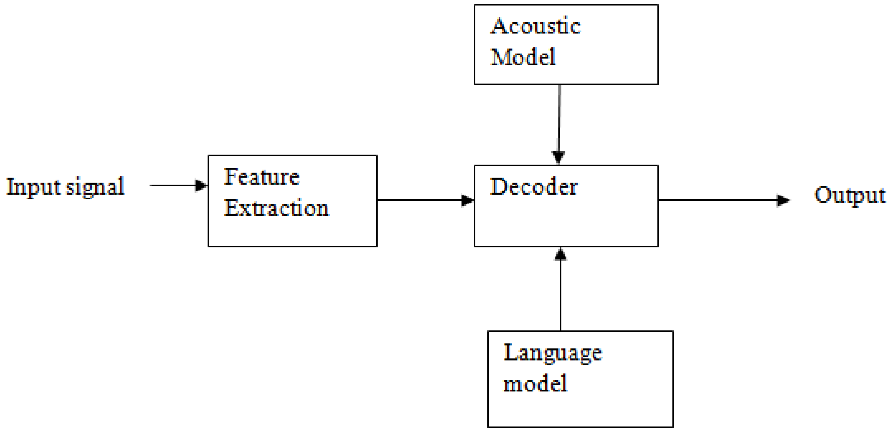


Figure 5: Speech to Text Operational Sequence

Functional Requirement	Recognize speech
Input	Audio file/audio stream (.wav, .mp3 etc.)
Description	This functionality of the system is used to capture audio input in the form of an audio file and transform it into ASCII characters˜ ˜
Priority	Must have
Risk(s)	Incompatible audio format Unusual/abnormal sample rate (44.1 kHz) Web socket communication error Unavailable UDP port Possible remedies: Operational Internet connection
Constraint(s)	Audio capture stream must not be interrupted
Dependency	N/A
Output	String of text (ASCII characters)

3.2 Transcribe text

Functional Requirement	Transcribe text
Input	String
Description	This functionality of the system is used to transform a string of ASCII characters into a structured sentence that can be read by a human
Priority	Must have
Risk(s)	String does not contain ”white space” or other tokenizeable traits
Constraint(s)	String must contain text that is tokenizeable
Dependency	Dependent on˜**FR3.1**˜and˜**FR3.3**
Output	A tuple of tokenized strings

3.3 Parse text

Functional Requirement	Parse text
Description	This functionality of the system is used to parse input text and remove any unnecessary characters if the source of the input text is not an audio stream (i.e textual/string input, file, etc) using the following regular expression for the English language: [[̂**a-zA-Z0-9**]*$] and the following regular expression for the Arabic language: [[̂**\u0621-\u064A**]+$]
Priority	Should have
Risk(s)	Unavailable input and/or unrecognizable text format
Constraint(s)	An available parsing tool must be available for use, i.e an HTML5 parser
Dependency	N/A
Output	Parsed string of text free of unnecessary characters that do not belong to the English language alphabet

3.4 Extract context

Functional Requirement	Extract context
Input	String (paragraph or sentence) that is the result of transcribing either an audio file or raw text input
Description	This functionality of the system is used to segment a string of text into three components: nouns, verbs, and focus. The frequency of each component is estimated and stored for further processing. The formula for estimating the frequency of these terms is [log(N / D)] where N is the number of documents in the corpus (length of the input text) and D is the number of documents (times) it appears within. An external ”WordNet” API call is then made to compare the highest-weighted nouns and confirm they carry similar meanings in order to define the context of the input string.
Priority	Must have
Risk(s)	Unavailable/invalid language model
Constraint(s)	The language model used must be either ”en-ar” or ”en-us” (refer to ”IBM Watson” documentation for details)
Dependency	Dependent on **FR3.1** and **FR3.2** and/or **FR3.3**
Output	Tuple containing the context/subject of the input string of text

3.5 Structure sentence

Functional Requirement	Structure sentence
Input	String
Description	This functionality of the system is used to highlight parts of a string of text that are present in a database using a binary value field. If this part of the string is present in the database, this value field is switched to 1. The default value is 0.
Priority	Should have
Risk(s)	Database-query-related communication errors Possible remedies: Validate database connection
Constraint(s)	A list of words (terms) and a set of associated attributes must be present in the database. The attributes are: definition, Arabic translation, English translation
Dependency	Dependent on˜ **R3.2** and/orc**FR3.3**
Output	Visual representation of highlighted terms

3.6 Classify data

Functional Requirement	Classify data
Input	SVM-compatible format for classification containing terms to be classified
Description	The system needs to be trained on a dataset containing 10,000 terms for English-Arabic bidirectional translation
Priority	Must have
Risk(s)	N/A
Constraint(s)	The data file will contain the features of the input collection of words for˜training, in a dedicated format suitable for SVM classification
Dependency	N/A
Output	Model file

3.7 Manipulate data

Functional Requirement	Manipulate data
Input	Database entries in the form of strings
Description	Users with the role of ëxpert” can create, update, and delete entries in the database related to ”Words” and their corresponding ”Context”
Priority	Should have
Risk(s)	N/A
Constraint(s)	N/A
Dependency	Dependent on **FR3.5** and **FR3.3**
Output	Corresponding database rows

3.8 Get feedback

Functional Requirement	Get feedback
Input	Integer with a value between 1-5
Description	The user will provide feedback in the form an integer with a value between [1-5]
Priority	Should have
Risk(s)	N/A
Constraint(s)	N/A
Dependency	Dependent on **FR3.4**
Output	Corresponding row in the database with ˜the value of the rating

3.9 Review feedback

Functional Requirement	Review feedback
Input	Database entries in the form of integers
Description	Users with the role of expert can review user-provided feedback for word interpretation and make necessary changes
Priority	Should have
Risk(s)	N/A
Constraint(s)	N/A
Dependency	Dependent on **FR3.8**
Output	Updated row in the database with value of the rating

3.10 Tokenize sentence

Functional Requirement	Tokenize sentence
Input	Paragraphs in the form of transcribed strings
Description	A sentence tokenizer will be responsible for splitting large paragraphs into sentences below 255 characters and placing them in a JSON array
Priority	Should have
Risk(s)	N/A
Constraint(s)	N/A
Dependency	N/A
Output	JSON array

3.11 REST API

Functional Requirement	REST API
Input	JSON array
Description	A REST API endpoint will make put, patch, and delete˜ requests that send and receive JSON responses
Priority	Should have
Risk(s)	HTTP error Possible remedies: Restart network connectivity dependencies
Constraint(s)	N/A
Dependency	N/A
Output	JSON array/HTTP response code

3.12 Upload audio file

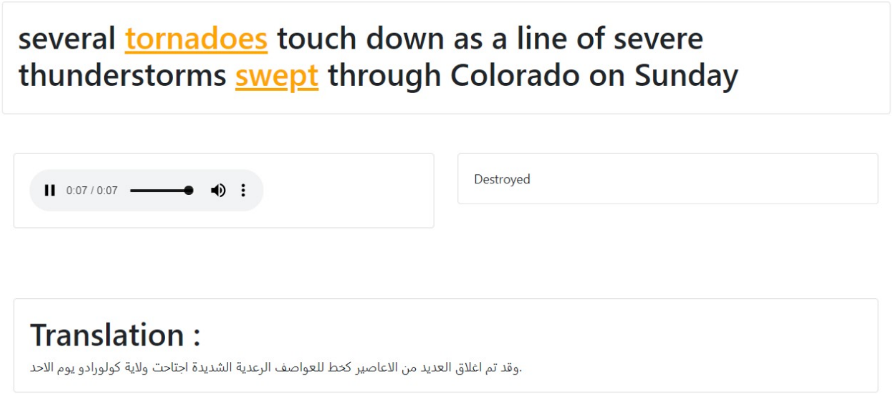
Functional Requirement	Input audio file
Input	Audio file (.wav, .mp3 etc.)
Description	Users will be able to upload an audio file for transcription and translation
Priority	Should have
Risk(s)	HTTP error Possible remedies: Restart network connectivity dependencies
Constraint(s)	N/A
Dependency	Dependent on **FR3.11**
Output	Transcribed text

3.13 Input text

Functional Requirement	Input text
Input	Audio file (.wav, .mp3 etc.)
Description	Users will be able to upload a text file for transcription and translation
Priority	Should have
Risk(s)	HTTP error Possible remedies: Restart network connectivity dependencies
Constraint(s)	N/A
Dependency	Dependent on **FR3.11**
Output	Transcribed text

4 Interface Requirements

4.1 User Interfaces



several tornadoes touch down as a line of severe thunderstorms swept through Colorado on Sunday

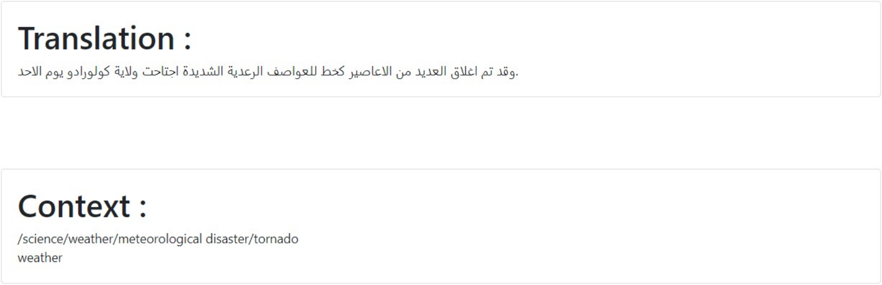
Destroyed

Translation :

.وقد تم اغلاق العديد من الاعاصير كخط للعواصف الرعدية الشديدة اجتاحت ولاية كولورادو يوم الاحد

Figure 6: System GUI

The system is designed as a web application that can be easily accessed through a web browser such as Google Chrome or Mozilla Firefox. UI screens are easy to traverse and allow for minimal interaction with the system during operation. The user is only required to either input a string of text for the context extraction process to begin, or an audio stream (audio file). In the case of Arabic text, the system automatically switches to right-to-left transcription, and vice versa. A display field is used to output the transcribed text with highlighted terms that are present in a database holding a glossary of terms that are relevant to the detected real-life domain through context extraction.



Translation :

وقد تم اغلاق العديد من الاعاصير كخط للعواصف الرعدية الشديدة اجتاحت ولاية كولورادو يوم الاحد.

Context :

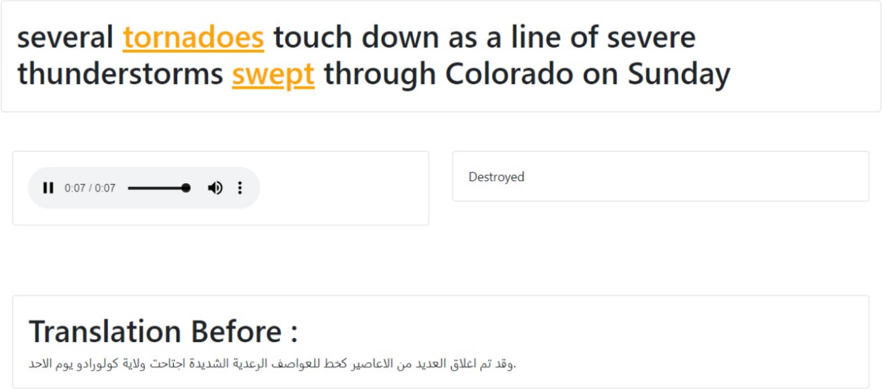
/science/weather/meteorological disaster/tornado
weather

Figure 7: Example of Context Extraction

The highlighted terms are also underlined as shown in fig. 4 so the user can hover over them, triggering another display field that shows the user a set of alternative meanings for that term. This sequence of operation continues until the user halts the live interpretation session.

4.2 GUI

The system user interface is designed to be simple enough and allow minimal interaction.



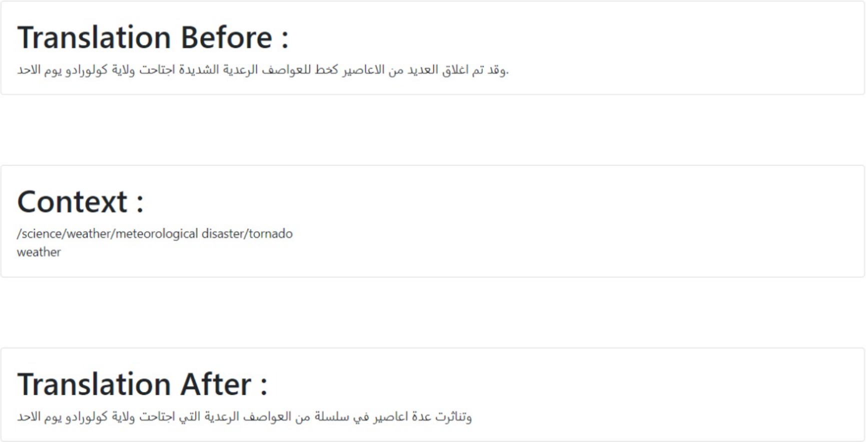
several tornadoes touch down as a line of severe thunderstorms swept through Colorado on Sunday

Destroyed

Translation Before :

وقد تم اغلاق العديد من الاعاصير كخط للعواصف الرعدية الشديدة اجتاحت ولاية كولورادو يوم الاحد.

Figure 8: Example of Text Before Processing and Translation



Translation Before :

وقد تم اغلاق العديد من الاعاصير كخط للعواصف الرعدية الشديدة اجتاحت ولاية كولورادو يوم الاحد.

Context :

/science/weather/meteorological disaster/tornado
weather

Translation After :

وتناثرت عدة اعاصير في سلسلة من العواصف الرعدية التي اجتاحت ولاية كولورادو يوم الاحد

Figure 9: Example of Text After Processing and Translation

4.3 CLI

- **cURL**: Client URL (v7.62.0) is a command-line tool used for transferring data over HTTP and other protocols. cURL is used to make external API calls coupled with PHP (v7.2).

4.4 API

- **IBM Watson**: IBM Watson [1] is a question-answering computer system that offers a suite of tools for natural language processing. The speech-to-text and language translator API modules are used within this system.

- **Stanford NLP**: The Stanford NLP [2] library offers language understanding tools. This library is used within the system for part-of-speech tagging.

- **Natural Language ToolKit**: NLTK [3] is a Python library with a suite of NLP modules. The remove-stop-words and HTML5 parsing modules are used within this system.

5 Performance Requirements

The language interpretation and domain specification modules within the system shall meet the following requirements laid out in Fantinuoli, Claudio, "Computer-assisted interpreting: challenges and future perspectives."

5.1 Language translation general specification

It's imperative that the process of translating text does not result in the loss of the meaning of the original text. The result of translation must meet a set of requirements before it can be labelled as "true" instead of "close." Any mistakes done in the sequence of this operation can have severe consequences on the overall dependability of the system. As such, the following balance must be satisfied without the prioritization of one over the other.

Speed Some domains are more demanding than others in a business context. For example, a document dealing with lawful matters require timely execution as well as due diligence in translation operations. In other domains such as pharmaco-vigilance changes may need to be made quickly for the successive compilation of reports and other documentation. It is important for these considerations to be made when translating text between English and Arabic during the operation of the system.

Accuracy The meaning of words when translated between different languages may be completely lost if due considerations are not made before translation is done. The accurate contextual understanding of text is important in this regard because it provides the end user with a true reflection of what the speaker meant to say when the audio input was made to the system. It is vital that this operation is done without risking "loss in translation.

5.2 Accuracy of context extraction

The system shall perform the task of specifying the real-life domains discussed within any given context, with support for transcribed text obtained from audio input. During this operation, artificial intelligence techniques will be used with deep learning methods that enable the system to learn from its mistakes when translating text between Arabic and English in a bidirectional manner.

5.3 Visual understanding of system operations

All system visual interfaces must act to provide the user with information including but not restricted to the following:

- Feedback for successful operation
- Textual representation of any inputs to the system and corresponding output
- Uninterrupted updates on system back workings related to different partial modules (translation, audio capture etc.)

In addition to the above required display operations, all system interfaces must abide by the following rules:

- Visual interfaces are constrained to feedback output and must not require user interaction
- Subtle interaction with the system will be allowed, e.g. a mouse hovering over a piece of text to display additional information
- System components must not serve to detract from user roles external to the system's existential state

All visual components of the system will serve to provide the user with an understanding of what the system is doing, instead of how it is doing it.

6 Design Constraints

6.1 Interactivity

The system must not force the user to interact with it during a live interpretation session. Display interfaces are meant to provide the user with a visual understanding and must under no circumstances prompt the user for further interaction within a 5-second time span.

Example: The system is fed speech input in the form of audio files and transforms it to transcribed text. The system must not prompt the user for confirmation to perform this task, instead performing it successively without requiring any further interaction on the user's part.

6.2 Compliance with Domain Standards

Existing domain standards must be met by the system in a language services context. The system exists besides the user's role in operating the system and not despite it.

6.3 Network Dependency

The system relies on a number of components external to the implementation of the system, and must therefore exist in co-dependence with these components and not replace them. The network of services performing the system's overall functionality may not infringe on each other's perspectives, roles, and/or privileges.

6.4 Web Accessibility

The system is designed as a web application accessible through an Internet browser, as such, an Internet connection is required for the various modules that make up the overall system to interact and be fully functional.

7 Other non-functional attributes

7.1 Reliability

The software being developed has to be reliable because it is dealing with a very sensitive subject which is simultaneous interpretation. It has to imitate the process of simultaneous interpretation done by the human being at the same rate to be an actual help and not a tool that is hindering their work. Also its results has to be as accurate as possible to make it more reliable and trustworthy for the interpreters during their work.

7.2 Maintainability

We are using the MVC design pattern so that helps a lot in maintaining the system. This design pattern divides the system into three modules which are Model, View and controller, it simply separates handling of the data from the how the interface appears to the user and the intermediate communicator between both of them.

7.3 Portability

The software being developed has to be portable because it will be used by different interpreters in a multitude of locations, so it cannot be developed in a static or fixed state that it cannot be used outside of this place. Being portable will make it more reliable for the interpreters as it would be a very handy tool that they can use wherever they go.

7.4 Scalability

The software being developed has to be scalable that if the users or the developers would like to add more domains or more glossaries to enhance the process of interpretation and context extraction, minimum work would be needed to perform this task. So at any time the interpreters if they create a new glossary for a new or an existing domain they can do that with ease.

8 Object-oriented Domain Analysis

8.1 Class diagram

8.2 Class description

8.2.1 Model

1. Class name: Model
2. Superclasses: N/A
3. Subclasses: User, Role, Permission, Context, Word

4. Purpose: This class holds all ORM functionality for model classes.

5. Collaborations: N/A

6. Attributes: N/A

7. Operations: Object Relational Model interface.

8. Constraints: N/A.

8.2.2 User

1. Class name: User

2. Superclasses: Voyager

3. Subclasses: N/A

4. Purpose: This class is instantiated to hold the state of a user throughout their login session.

5. Collaborations: Voyager, Notifiable, Role.

6. Attributes: Name: string(64), e-mail(255), password(255), remember token(255).

7. Operations: useModel(): Name: Object, Action: Object. Invokes the model name as a parameter and the action specified (e.g, ORM access). Returns an instance of the model.

8. Constraints: N/A.

8.2.3 Role

1. Class name: Role

2. Superclasses: Model

3. Subclasses: N/A

4. Purpose: This class is instantiated to assign each user role.

5. Collaborations: Model, HasRelationships

6. Attributes: guarded: Array. (Holds guarded/non-nullable attributes).

7. Operations: users(): Returns the (1-M) relationship between the User class and the Role class.
   permissions(): Returns the (M-1) relationship between the Permission class and the Role class.

8. Constraints: N/A.

8.2.4 Permission

1. Class name: Permission
2. Superclasses: Model
3. Subclasses: N/A
4. Purpose: This class holds all permissions for users.
5. Operations: roles(): Returns the relationship (M-1) between the Permission class and the Role class.
   generateFor(): Returns a generated permission for each user.
   removeFrom(): Soft deletes a permission from the database.
6. Constraints: N/A.

8.2.5 Context

1. Class name: Context
2. Superclasses: Model
3. Subclasses: N/A
4. Purpose: This class is instantiated to hold the state of a context within the database.
5. Collaborations: Model
6. Attributes: id(11), Context(255).
7. Operations: getWords(): Context: String. Returns all words within the invoked context in the database.
8. Constraints: N/A.

8.2.6 Word

1. Class name: Word
2. Superclasses: Model
3. Subclasses: N/A
4. Purpose: This class is instantiated to hold the state of a word within the database.
5. Collaborations: Model
6. Attributes: id(11), Word(255), Definition(255).
7. Operations: getWord(): Word: String. Returns the parameterized word and its definition.
8. Constraints: N/A.

9 Operational Scenarios

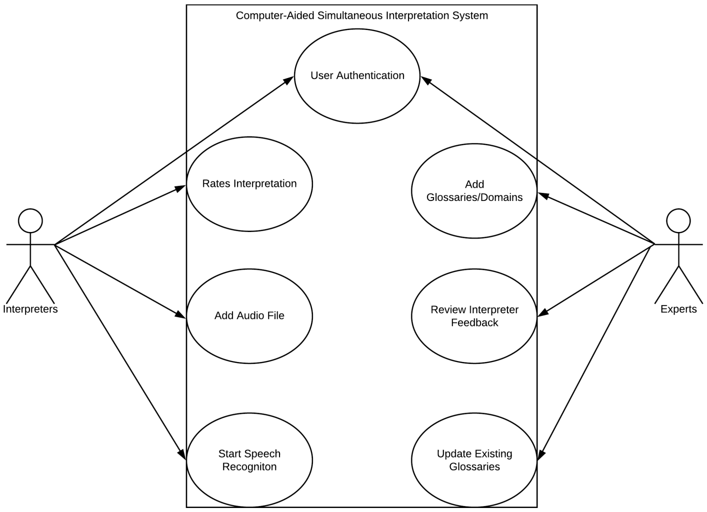


Figure 10: System UseCase

9.1 Speech Recognition

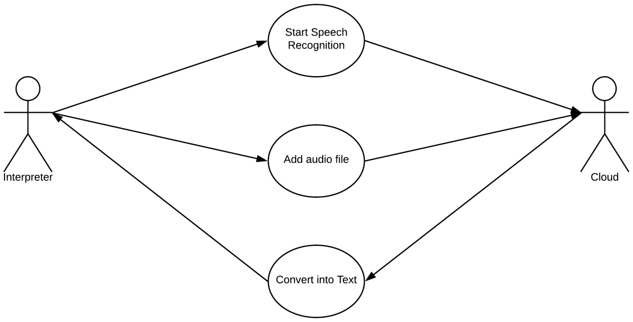


Figure 11: Speech Recognition UseCase

Goal in Context: Goal in Context: Recognizing the speech being said and converting it into text.
Description: The system will capture the sound and recognize the speech that is being said, then convert the audio file into a readable text that can be processed.

Preconditions: The system would need to be connected to a mic that is always listening and capturing sound.
Success End Condition: The system will be able to recognize the speech successfully.
Failure End Condition: The system fails to recognize the speech.
Trigger: The mic starts to detect sound.
Main Success Scenario:
1-) The system captures sound successfully.
2-) The system converts the speech successfully into a text.
3-) The system creates some text that can be processed.

9.2 Context Extraction

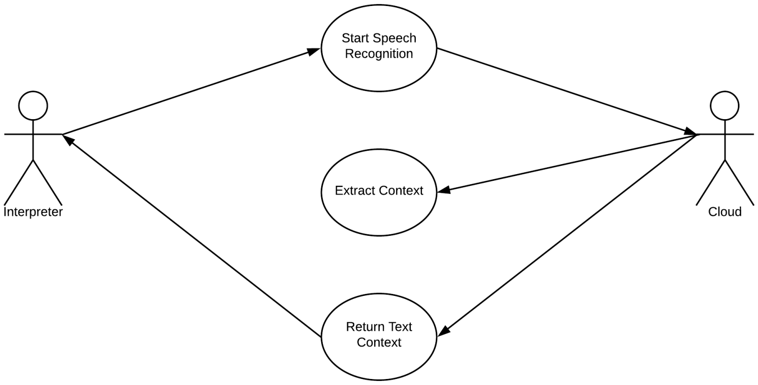


Figure 12: Context Extraction UseCase

Goal in Context: Goal in Context: The system extracts context from the text being converted from the audio file
Description: After the system convert the speech into text, the system process this text by doing tasks like sentence structure and segmentation to extract the appropriate context or domain to achieve the most accurate translation.
Preconditions: The system would need to recognize the speech first to be able to process the text and extract its context.
Success End Condition: The system will be able to extract the context successfully.
Failure End Condition: The system fails to extract the context of the text.
Trigger: The audio file is converted into text.
Main Success Scenario:
1-) The system translates manages to extract the accurate context of the text.
2-) The context being extracted from the text helps in translating the text to achieve the most accurate translation.

9.3 Translating the text

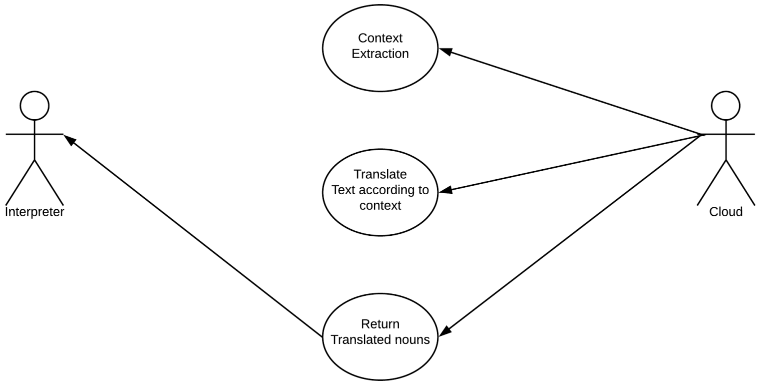


Figure 13: Translating the text UseCase

Goal in Context: Goal in Context: The system translates the text according the context.
Description: The system translate the text output to the other language. As we are working with two languages only (English / Arabic) the system will translate from Arabic to English or vice versa, this happens after extracting the context of the text to achieve an accurate translation.
Preconditions: The system would need to recognize the speech first and then extract its context to be able to translate the text.
Success End Condition: The system will be able to translate the speech successfully.
Failure End Condition: The system fails to translate the speech.
Trigger: The audio file is converted into text, and then its context is extracted successfully.
Main Success Scenario:
1-) The system translates manages to translate the text accurately.
2-) The system output the appropriate nouns and its translation to the user to help with the interpretation.

9.4 User Authentication

Goal in Context: Goal in Context: The user would be able to access the system.
Description: The user will access the system using functionalities like Sign Up and Login, start the speech recognition task to start the whole interpretation process.
Preconditions: None
Success End Condition: The user has their own account with their history of previous translated speeches.
Failure End Condition: The user cannot access his account nor his data.
Trigger: The user creates their account.
Main Success Scenario:
1-) The user will login before they use the system.
2-) Their data would be accessible from the cloud.
3-) The user would be able to create an account.
4-) The user would be able to start the process of interpretation from the system

10 Preliminary Schedule Adjusted

Table 1: Time plan for Fall 2018/Spring 2019

Task	From	To
Information gathering	25/7/2018	12/9/2018
Prototype delivery	12/9/2018	26/9/2018
Proposal evaluation	12/9/2018	26/9/2018
Survey paper	25/7/2018	20/10/2018
Dataset collection (Alsun)	27/9/2018	25/12/2018
Database design	14/9/2018	26/9/2018
Writing SRS document	30/10/2018	28/11/2018
SRS evaluation	28/11/2018	28/11/2018
Writing SDD document	1/1/2019	12/2/2019
SDD evaluation	12/2/2019	12/2/2019
System implementation	TBD	TBD
Implementation evaluation	TBD	TBD
Validation & testing	TBD	TBD
Preliminary preparations	TBD	TBD
Final presentation	5/5/2019	5/5/2019

The above table showcases our time table for the 2018/2018 academic year. Any dates that haven't yet been decided by the faculty board have a placeholder value.

11 Preliminary Budget Adjusted

Table 2: Resources required for project completion

Tool	Budget	Necessity
IBM Watson Analytics PLUS	$0.02 USD/1000 characters	High
High-quality audio capturing mic	$50-100 USD	Medium
Dedicated hosting	$25-75 USD	Low

The use of IBM's Watson engine for speech-to-text transcription is subject to a paid premium plan amounting to $0.02 USD per thousand characters.

High-quality audio-capturing hardware such as a cardioid dynamic microphone will be required for experimentation with the software for noise isolation while recording.

12 Appendices

12.1 Definitions, Acronyms, Abbreviations

-NLP :Natural language processing is a subfield of computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human languages

-NLU:Natural-language understanding or natural-language interpretation is a subtopic of natural-language processing in artificial intelligence that deals with machine reading comprehension

-API:An application programming interface (API) is a set of routines, protocols, and tools for building software applications

-PDA:Personal Digital Assistant

-GUI:Graphical user interface

-NLTK:Natural Language Toolkit [3] (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP)

-CLI:command line interface

-cURL:Client URL is a command-line tool used for transferring data over HTTP and other protocols. cURL is used to make API calls in this system.

-ASCII:American Standard Code for Information Interchange) is the most common format for text files in computers and on the Internet

13 References

References

[1] https://www.ibm.com/watson/

[2] https://nlp.stanford.edu/software/tagger.shtml

[3] https://www.nltk.org/

[4] https://www.ibm.com/ibm/history/ibm100/us/en/icons/takmi/

[5] http://intragloss.com/Product.php

[6] Gile, Daniel. ”Testing the Effort Models’ tightrope hypothesis in simultaneous interpreting-A contribution.” HERMES-Journal of Language and Communication in Business 12.23 (1999): 153-172.

[7] Fantinuoli, Claudio. ”Computer-assisted interpreting: challenges and future perspectives.” Trends in E-Tools and Resources for Translators and Interpreters (2018): 153.