A Project Report on

NLP: Speech2Code

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY,
PUNE IN THE PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

Marathwada Mitra Mandal's College of Engineering,
Karve Nagar, Pune

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

Bhavika Mauli Karale	**B150454262**
Sameeksha Avinash Joshi	**B150454258**
Siddharaj Sudhakar Gangavane	**B150454241**
Rutuja Rajendra Shewale	**B150454341**

DEPARTMENT OF COMPUTER ENGINEERING
MARATHWADA MITRA MANDAL'S COLLEGE OF ENGINEERING,
KARVENAGAR, PUNE-411052

SAVITRIBAI PHULE PUNE UNIVERSITY
2021-22

CERTIFICATE

This is to certify that the project report entitles

"NLP: Speech2Code"

Submitted by

Bhavika Mauli Karale
Sameeksha Avinash Joshi
Siddharaj Sudhakar Gangavane
Rutuja Rajendra Shewale

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Dr. Aarti Agarkar** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Dr. Aarti Agarkar
Internal Guide
Department of Computer Engineering

Dr. Kalpana Thakre
Head of Department
Department of Computer Engineering

Dr. V. N Gohokar
Principal,
Marathwada Mitra Mandal's College of Engineering, Karvenagar, Pune – 411052

Place: Pune
Date:

ACKNOWLEDGEMENT

We take this opportunity to express my deep sense of gratitude towards our esteemed guide Dr. Aarti Agarkar for giving us this splendid opportunity to select and present this seminar and also providing facilities for successful completion.

We thank Dr. Kalpana Thakre, Head, Department of Computer Engineering, for opening the doors of the department towards the realization of the seminar, all the staff members for their indispensible support, priceless suggestions and for most valuable time lent to as and when required. With all the respect and gratitude, We would like to thank all the people who have helped me directly or indirectly.

Name of the Students

Bhavika Mauli Karale

Sameeksha Avinash Joshi

Siddharaj Sudhakar Gangavane

Rutuja Rajendra Shewale

Abstract

Repetitive strain injury (RSI) is a syndrome that affects thousands of professionals around the world every year, among these professionals are programmers, professionals who tend to make extensive use of their hands in carrying out their tasks and that is why more susceptible to RSI.

Programmers affected by this syndrome have great difficulty in using their hands and consequently great difficulty in carrying out their main activity: programming, a historically text-based activity that requires extensive use of the hands. That said, we come to the central theme of this work: the creation of an assistive technology tool capable of helping programmers to program in the JavaScript language using their voice. The tool aims to enable programmers to program without using their hands, using only voice commands, parse sentences belonging to a given grammar using stack automata, how to connect and control code editors using UI automation tools, the process of creating a code editor extension Visual Studio Code and the creation of a cross-platform application using ReactJS and the Electron framework. The tool aims to enable programmers to program without the use of hands, using only voice commands.

Keywords

Speech Recognition, Stack Automaton, Assistive Technology, User Interface Automation,Speech-to-Code Conversion.

Contents

List of Figures

List of Tables

1 TECHNICAL KEYWORDS

1.1 Domain Name

Artificial Intelligence

1.2 Technical Keywords

Speech Recognition, Stack Automaton, Assistive Technology, User Interface Automation,Speech-to-Code Conversion.

2 INTRODUCTION

Much has been said about the so-called accessibility software, software that aims to facilitate the use of computers and equipment by people with some kind of disability. These types of software fall within a larger area called Assistive Technology (AT). Assistive technologies are resources and services that aim to facilitate the development of activities of daily living for people with disabilities, promoting greater independence by allowing these people to perform tasks that they were previously unable or had great difficulty in performing. Assistive technologies help people who have difficulty speaking, typing, writing, seeing, hearing, learning, walking and many others. Different types of disabilities require different types of specialized assistive technologies. Some of the best known types of assistive technology include: the wheelchair, a device that facilitates locomotion for people who are unable or have difficulty walking; the hearing aid, a device that amplifies the sound waves of the environment in order to correct any hearing loss in the user; and so-called screen magnification software, software that allows you to magnify certain portions of the screen to help people with low vision.

In the context of diseases that affect the use of the hands, one of the best known is repetitive strain injury (RSI), a syndrome characterized by a group of diseases - tendonitis, tenosynovitis, carpal tunnel syndrome, trigger finger, etc. - that affects muscles, nerves and tendons of the upper limbs. In addition to causing pain and inflammation, this disorder can also compromise the affected region. RSI is usually caused by repetitive movements such as typing, poor posture, playing the piano, driving, crochet and so on.

According to data from the Social Security Statistical Yearbook, in 2019, LER was among the 50 most common causes of work accidents in Brazil. Among these causes are tenosynovitis (inflammation or infection in the tissue covering the tendon) and mononeuropathy of the upper limbs (injury to the peripheral nerve), both conditions that compromise the use of the hands.

Individuals unable or with difficulty to interact with computers due to RSI can count on the help of assistive technologies based on voice recognition, in this type of technology the user uses their voice, instead hands, to interact directly with the computer. Some examples of assistive technologies based on voice recognition include voice assistants: Siri, Alexa and Cortana. Using Siri voice assistant, for example, it is possible to write and send an email without the use of your hands, using only your voice.

Among the countless professionals affected by RSI are programmers, professionals whose work demands the constant use of their hands, and therefore, more susceptible to RSI. This is due to the fact that, historically, the software development process has taken place through repeated interactions between mouse, keyboard and hands. An example of this are code editors, where the main form of interaction is through the keyboard (BEGEL, 2005).

That said, we come to the central theme of this work: the creation of an assistive technology tool based on voice recognition capable of helping programmers with RSI to program.

2.1 Motivation

The objective of this work is to create an application capable of helping pro-

grammers to program in JavaScript without using their hands. This application will be able to listen and analyze voice commands and then manipulate any code editor in order to execute this command.

Using this application instead of using your hands to write in the code editor, for example, the instructions needed to declare a variable, the programmer can simply express aloud that he wants to declare a variable which then the application will automatically connect to the code editor with the intention of writing the necessary instructions to declare a variable in the JavaScript language.

For this, the application must be able to: capture voice commands through from the computer's microphone, transform the voice command into text, validate the command and, finally, connect to the code editor in order to execute the command.

2.2 Domain Description

Software that aims to facilitate the use of computers and equipment by people with some kind of disability. These types of software fall within a larger area called Assistive Technology (AT). Assistive technologies are resources and services that aim to facilitate the development of activities of daily living for people with disabilities, promoting greater independence by allowing these people to perform tasks that they were previously unable or had great difficulty in performing. Assistive technologies help people who have difficulty speaking, typing, writing, seeing, hearing, learning, walking and many others.

2.3 Problem Definition

To create an application capable of helping programmers to program in JavaScript without using their hands. This application will be able to listen and analyze voice commands and then manipulate any code editor in order to execute this command.

1. Implement a functionality capable of real-time voice recognition. This functionality will turn voice commands into text.
2. Define a language and develop a mechanism to recognize that language using finite automata. The intent of this language is to represent all available voice commands in textual form.
3. Develop a tool capable of interacting with code editors. This tool will be able to manipulate a code editor to, for example, remove a line.

3 LITERATURE SURVEY

Following are the papers and articles that were studied regarding the topic which helped us in understanding the subject deeper.

The table shows the literature survey by comparing techniques propose in various references:

Table 1: Literature survey

Sr No.	Title of paper	Author	Advantages	Limitations
1	Repetitive Strain Injury among Computer Users: A case study in telecommunication company. [1]	Nurul Huda Baba, Dian Daruis	Paper defines what Repetitive Strain Injury is and causes of RSI. It helps us understand the root of RSI and the people who are most likely to get affected by it.	The measures to be taken by the people to avoid RSI are not given clearly. Along with this, the survey is conducted on a small group of people due to which there are chances the results may vary more.

Sr No.	Title of paper	Author	Advantages	Limitations
2	A Study on Automatic Speech Recognition [3]	Saliha Benkerzaz, Abdeslem Dennai, Youssef Elmir	The paper gives an overview of the main definitions of Automatic Speech Recognition. The authors have discussed about the speaker dependence systems, architecture and working of ASR.	An in depth study of Language model, acoustic model and pronunciation model is missing.

Sr No.	Title of paper	Author	Advantages	Limitations
3	Automatic Identification of Stop Words. [3]	W. John Wilbur, Karl Sirotkin	This paper presents an automated theory for identification of stop words which are irrelevant to the useful content of a particular document. This paper replace the conventional nearest neighbor method to find such words by a more efficient way of applying cosine similarity of the document-document pair to the vector retrieval model.	This approach appears limited due to only document-document similarity. So cannot be predicated to be useful for general textual queried documents. More thought needs to be given on which terms are indexed. The application of this theory on general texted queries remains untested.

Sr No.	Title of paper	Author	Advantages	Limitations
4	Deep Variational Filter Learning Models. [4]	Purvi Agarwal, Sriram Ganapathy	The proposed method uses an unsupervised data-driven modulation filter learning approach that preserves the key modulations of speech signal. This is achieved by a deep generative modeling framework to learn modulation filters using convolutional variational autoencoder. .	Using Convolutional Variational Autoencoder can give blurry outputs. .

Sr No.	Title of paper	Author	Advantages	Limitations
5	Unifying Speech Recognition and Generation with Machine Speech Chain. [5]	Andros Tjandra, Sakriana Sakti, Satoshi Nakamura	Paper proposes a closed-loop between ASR and TTS model to construct a machine speech chain mechanism. By using similar sequence-to-sequence architecture and interchangeable source-target domain, it allows to train both labeled and unlabeled data in a single loop. . .	The TTS system is very time consuming as it requires huge databases and hard-coding of combination to form these words. As a result speech synthesis consumes more processing power

4 SOFTWARE REQUIREMENTS SPECIFICATIONS

The following section gives information regarding the software specifications, scope of the project and various requirements. **NOTE: High speed internet is required for accurate functioning of the application**

4.1 Introduction

4.1.1 Project Scope

Speech recognition systems have applications in several areas. In fact, any activity that involves human-machine interaction can potentially utilize these systems. Currently, several applications are already being designed with a built-in speech recognition system.

This assistive technology application is capable of enabling programmers to program in JavaScript without the use of their hands, using only voice commands. The purpose of the application is to enable programmers who are unable or with great difficulty to use their hands, whether due to injury or disability, to continue exercising their profession.

This same methodology can be used in the creation of other applications with a similar purpose, such as a tool that, in addition to JavaScript, also supports Python.

4.1.2 User Classes and Characteristics

Voice recognition software can be used when the user needs to generate a large volume of textual content without the need of writing manually. It can also be used by users who are unable or have great difficulty using a keyboard. It can be used by individuals who are not well-versed with the syntax of a particular language but still wish to code. It can be used by programmers suffering from Repetitive Strain Injury(RSI).

4.1.3 Assumptions and Dependencies

Errors in Speech Identification can take place. Inability to understand spoken words can lead to severe errors. While developing this system, learning curves may lead to delays. Users could inaccurate expectations from the systems.

4.2 Functional Requirements

Admin is considered the person with ultimate authorities in the system with many functionalities.
1. Admin can select whether he wants to use voice for coding or whether he wants to convert commands written in simple English to his code.
2. Admin can change the program's main settings like selecting a preferred language for giving voice commands, selecting the editor of choice for coding and selecting the debug option.
3. Furthermore, admin select between whether he wants Azure speech to text service or simple Chrome speech recognition system for his voice coding. Admin has full control of when to start and stop the mic recording.

4.3 Non-functional Requirements

1. With ideal conditions, system response should be fast and error free.
2. System performance should not decrease with time or by usage.
3. System should recognize any voice without any fault.
4. Performance and speed should not depend on newer or older versions of VSCode editor or Chromium based search engine.
5. The system is easy to use and learn, a help page is also provided in the menu for all new users.
6. Meaningful notification messages will be displayed when some error happens

4.4 System Requirements

The minimum requirements for the application to work fully are as follows:

1. Network: Stable and fast internet connection
2. Operating System: Tested on Windows 10 only
3. Memory: At least 3GB of ram and 270MB of available disk space
4. Visual Studio Code: Tested with version 1.55.2 of VS Code
5. Language: Javascript, HTML, CSS
6. FrameWork: ReactJS, NodeJS

5 SYSTEM DESIGN

5.1 System Architecture

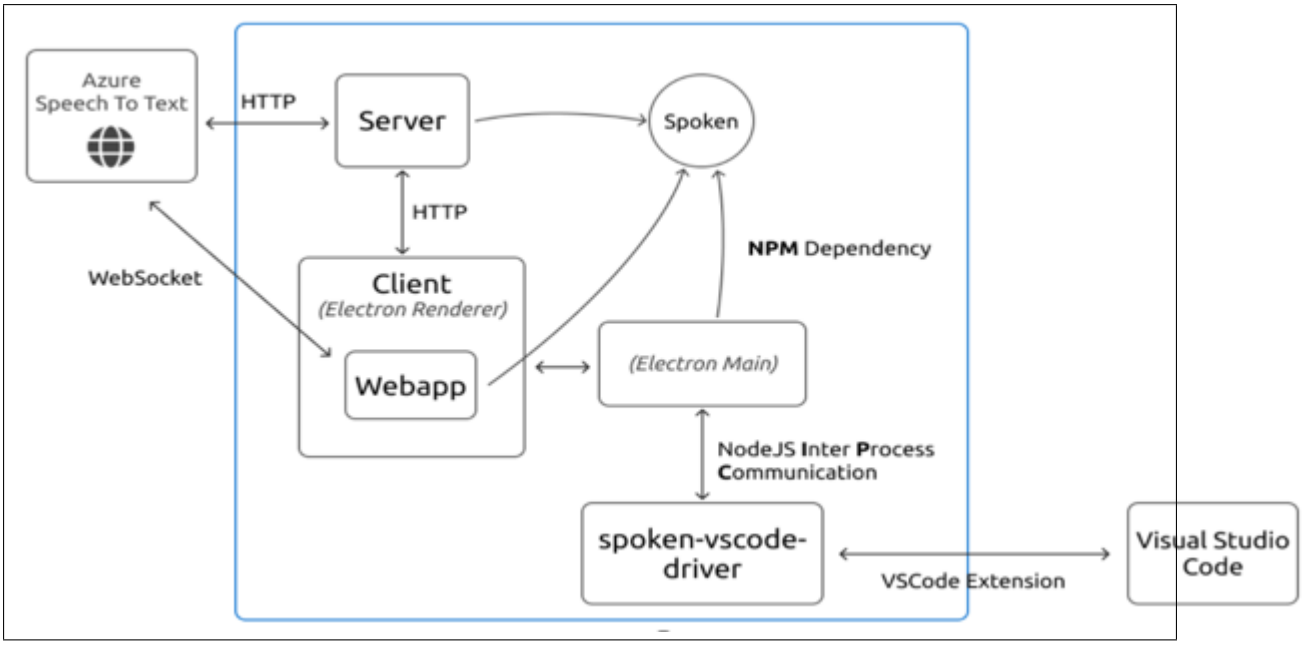


Figure 1: System Architecture

The above diagram shows the system architecture. It helps in understanding the working and the workflow of the project in a precise manner.

5.2 UML Diagrams

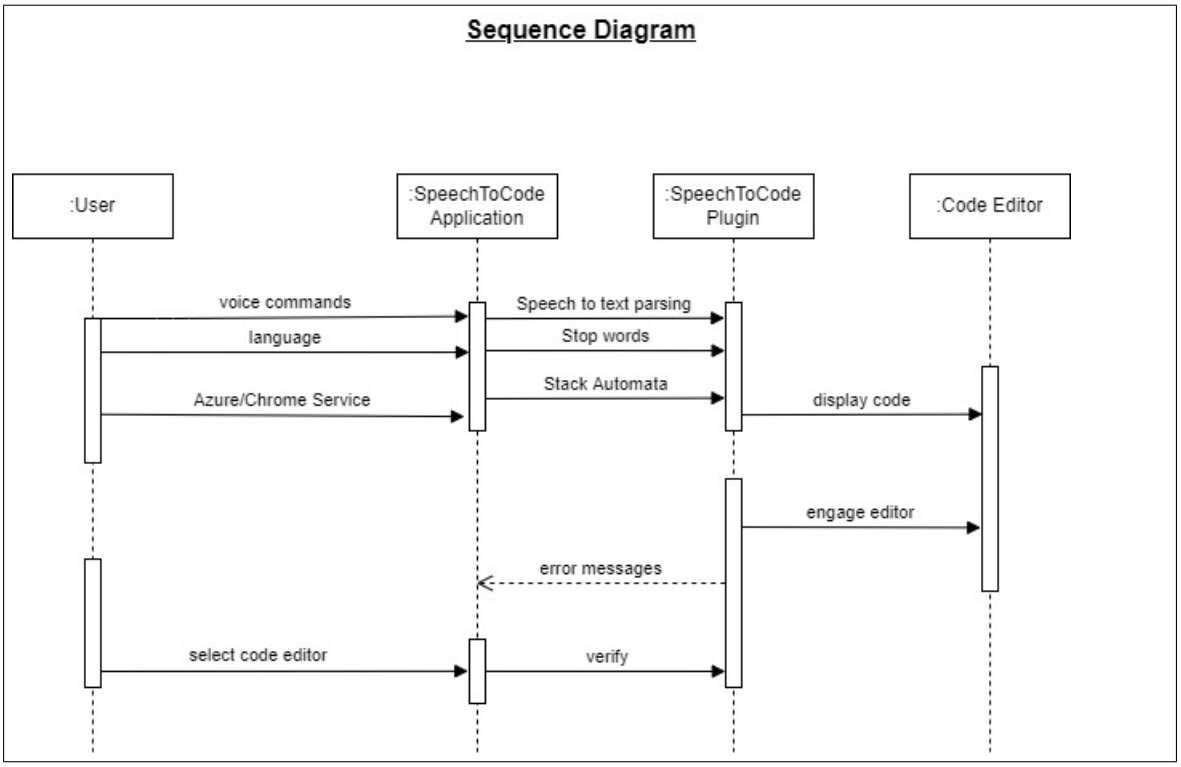


Figure 2: Sequence Diagram

It portrays the communication between two lifelines as a time-ordered sequence of events such that these lifelines took part at the run time. The lifeline is represented by vertical bars whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. This sequence diagram shows that the code editor is the least active performer in the whole process.

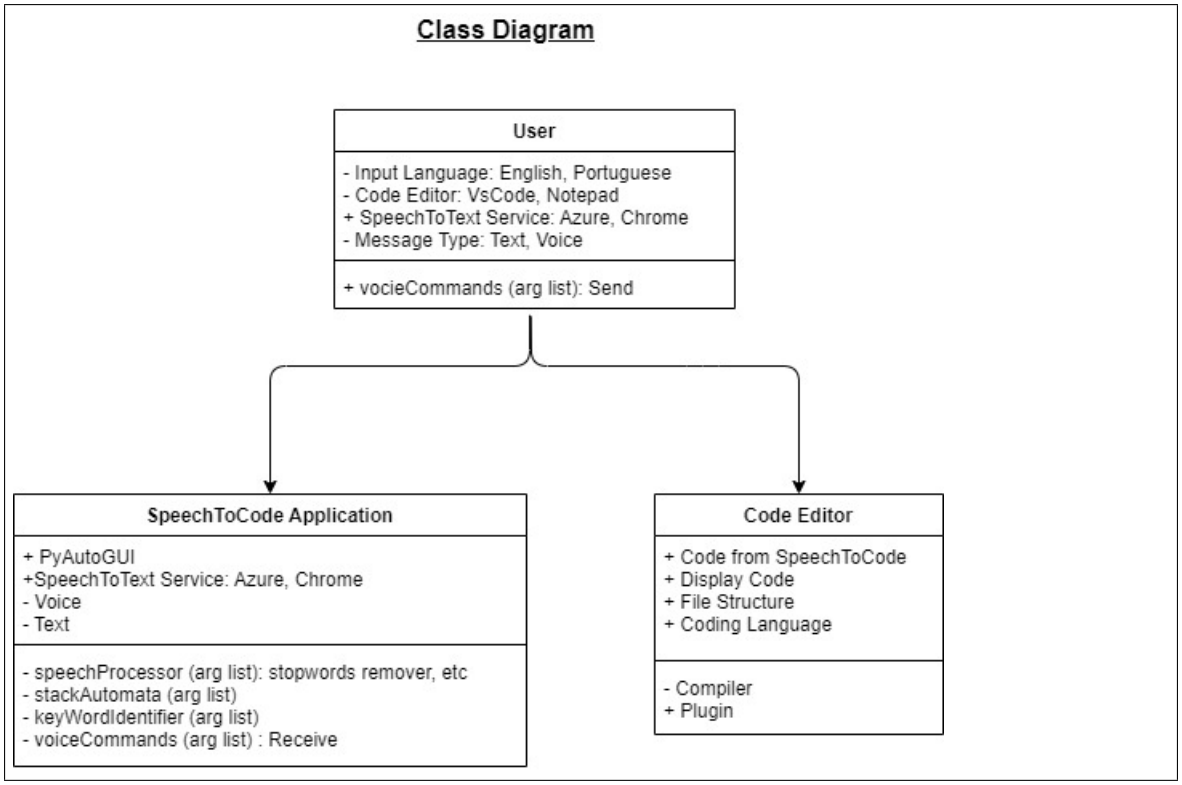


Figure 3: Class Diagram

Here, we can see three classes in our project. This class diagram is drawn with the specific perspective. The + symbol shows that the attribute is public and - shows the attribute is private. The user class has input language , code editor selection, speech to text service selection and message type selection services.

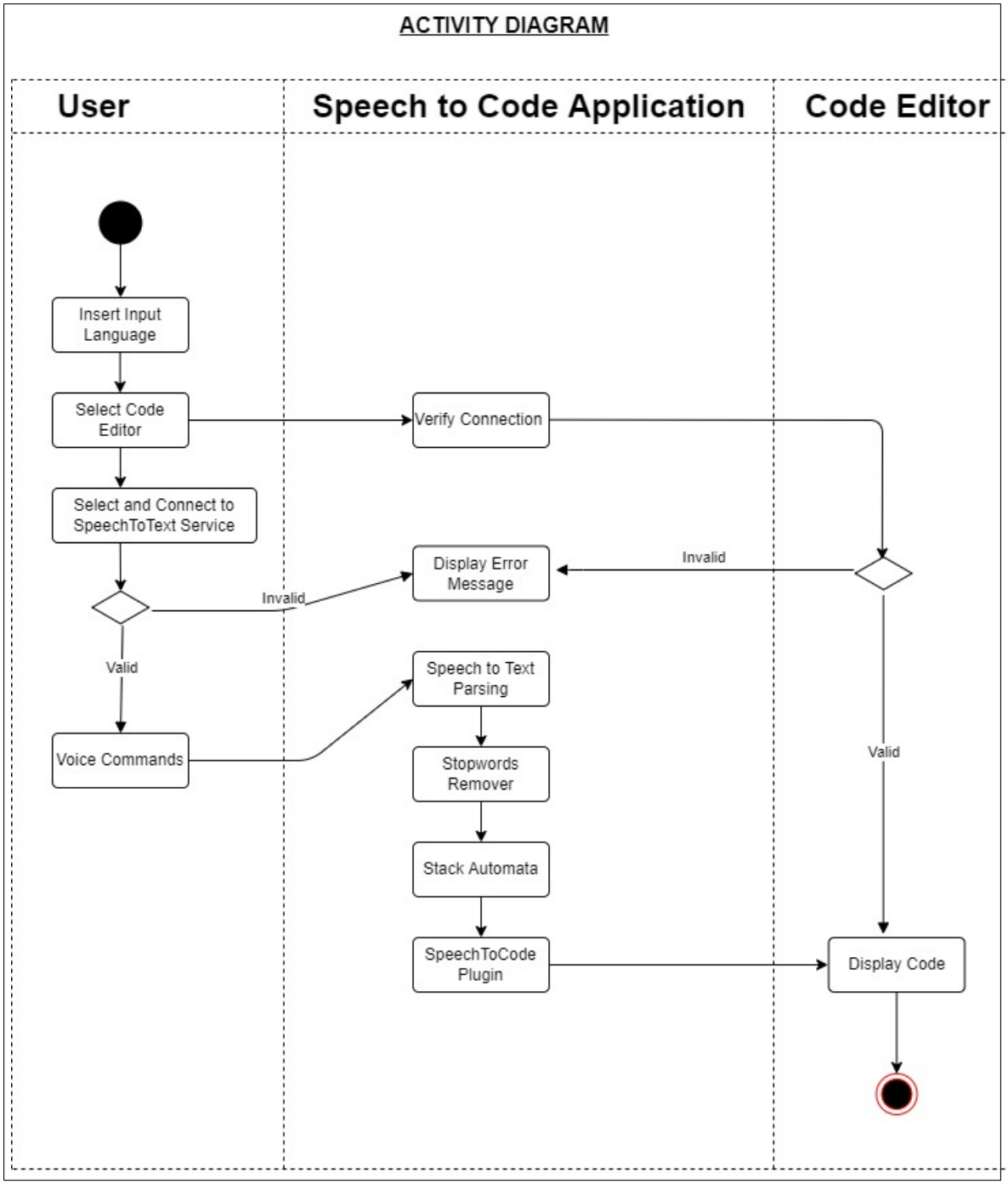


Figure 4: Activity Diagram

The activity diagram helps envisioning the workflows from one activity to another. It puts emphasis on the condition of flow and the order in which it occurs. The dotted lines represent the swimlanes, activities performed by the same actor are grouped in the same swimlane. The end node is present in the code editor swimlane with as bordered solid circle.

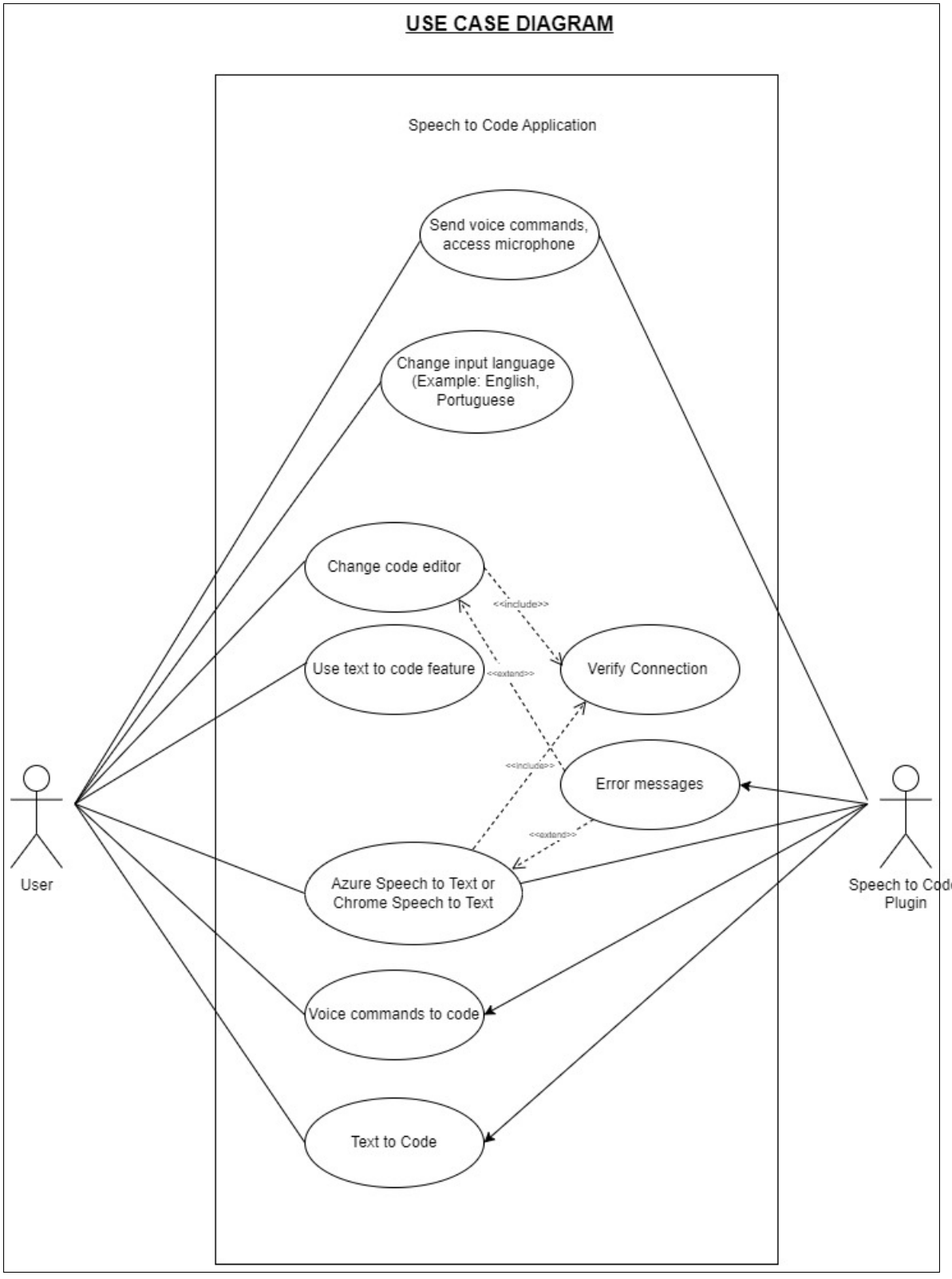


Figure 5: Use Case Diagram

The main purposse of a use case diagram is to portray the dynamic aspect of a system. As shown, this diagram encapsulates the system's functionality by incorporating use cases, actors and their relationships.

6 SOFTWARE DEVELOPMENT LIFE CYCLE

This section gives an overview of the Software Development Lifecycle

6.1 Introduction to Agile SDLC Model

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

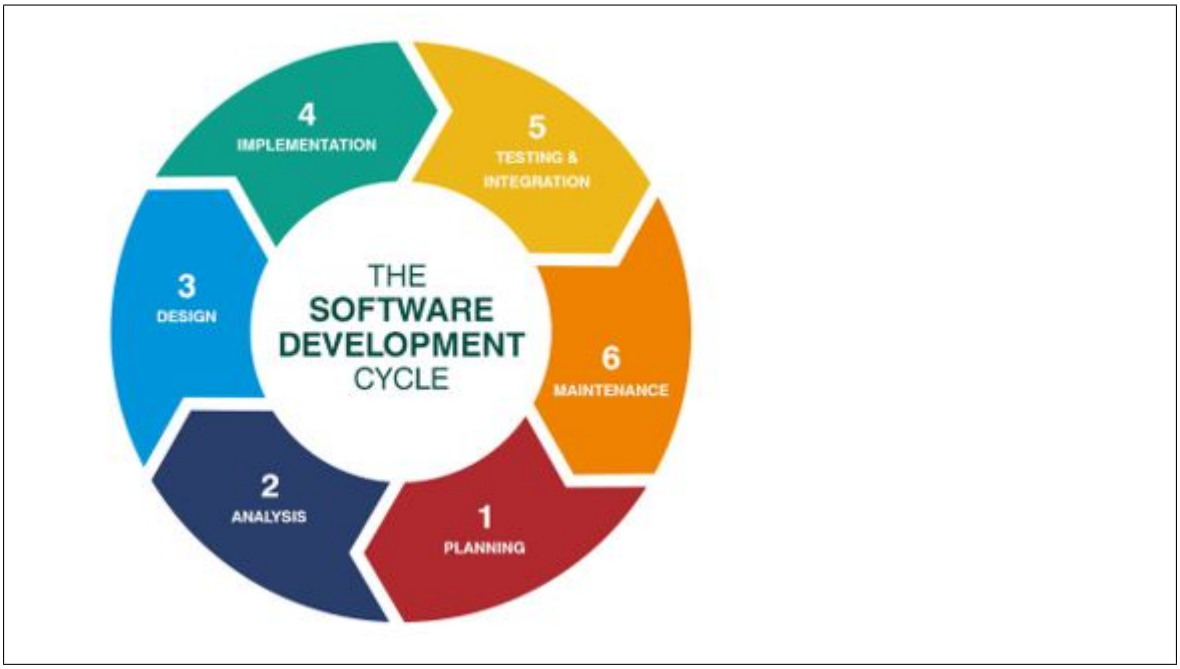


Figure 6: SDLC Diagram

6.2 Advantages of Agile Model

1. Customer satisfaction by rapid, continuous delivery of useful software.
2. People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
3. Working software is delivered frequently (weeks rather than months).
4. Face-to-face conversation is the best form of communication.
5. Close, daily cooperation between business people and developers.
6. Continuous attention to technical excellence and good design.
7. Regular adaptation to changing circumstances.
8. Even late changes in requirements are welcomed.

6.3 Disadvantages of Agile Model

1. In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
2. There is lack of emphasis on necessary designing and documentation.
3. The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
4. Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

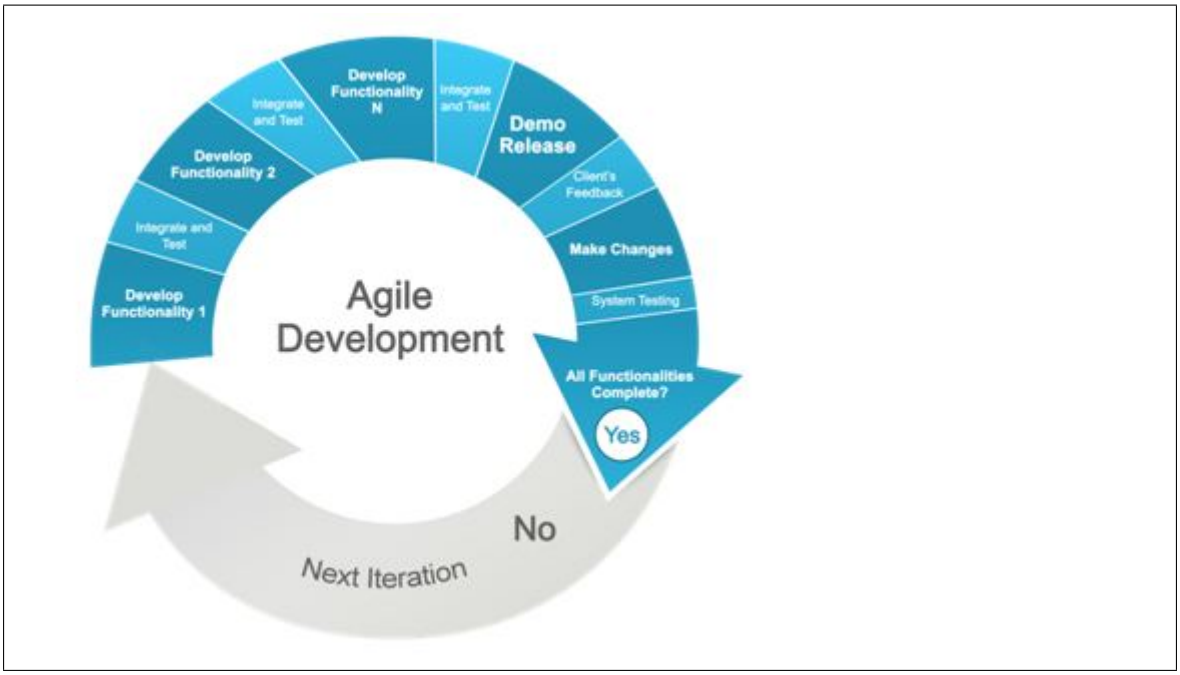


Figure 7: Agile Model

6.4 When to use Agile model

1. When new changes need to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
2. To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
3. Unlike the waterfall model in agile model very limited planning is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.

4. Both system developers and stakeholders alike find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

7 Project Plan

This section contains details information regarding the project resources, estimates and risks associated with the project.

It also includes the project schedule and a timeline chart.

7.1 Project Estimate

7.1.1 Breakdown of tasks

Semester 1

1. Decide the topic of interest
2. Design the look of frontend
3. Front end development with ReactJS, HTML, CSS

Semester 2

1. Map the working of the application
2. Back end development with NodeJS
3. Risk Analysis and Software testing

7.1.2 Resource billing

1. Azure Speech to Text service has different billing rates are around 76 rupees (1 USD)
2. Chrome does not charge any fees

7.2 Project Resources

1. Stable and strong internet connectivity
2. OS - Windows 10 or above
3. Knowledge of JavaScript, HTML, CSS, NodeJS, ReactJS
4. Chrome/Azure speech to text service
5. Web browser - Google Chrome/ Microsoft Edge

7.3 Risk Management

7.3.1 Risk Identification

The following risks are identified in the application:

1. Input not recognized
2. Input not recorded
3. Output not received
4. Misunderstanding voice commands of the user

7.3.2 Risk Analysis

The major reason for the above mentioned risks are -

1. Unstable and weak internet connection
2. Extreme noise disturbance

7.3.3 Risk Monitoring

1. Have a stable and strong internet connection at all times while working with the application
2. Have no to negligible noise disturbance while giving voice commands
3. Speak as clearly as possible while giving voice commands to avoid miscalculations

7.4 Project Schedule

7.4.1 Project Task Set

1. The application must run flawlessly given high internet speed connectivity
2. The voice commands must be recognized accurately
3. The output calculation should be accurate

7.4.2 Task Network

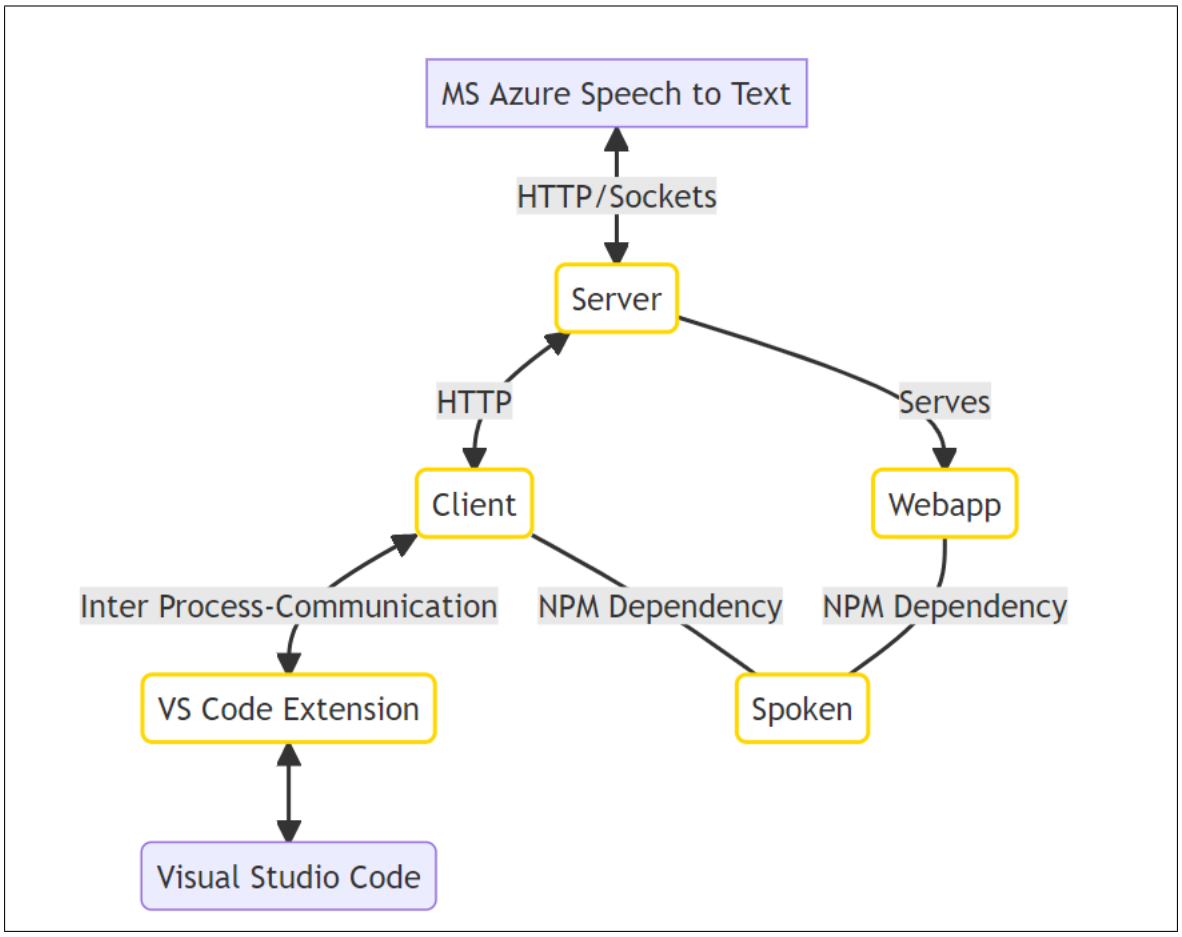


Figure 8: Project Workflow

7.4.3 Timeline Chart

Table 2: Timeline Report

Sr No.	Date	Time	Topic Discussed
1	9/10/21	5:00 pm	Project Review 1
2	16/10/21	5:00 pm	Project Review 2
3	8/11/21	5:00 pm	Project Review 3
4	23/11/21	10:00 am	Semester 1: Project Examination
5	5/4/22	3:00 pm	Project Review 1 : Semester 2
6	14/4/22	2:00 pm	Project Review 2
7	2/5/22	3:00 pm	Project Review 3
8	2/5/22	11:00 am	Project report review
9	7/5/22	10:00 am	Project report and black-book submission

8 Project Implementation

The Project Implementation section gives the details about the modules, tools used and the working of the project. It explains the workflow of the project.

8.1 Overview of Project Modules

8.1.1 Main modules

1. Webapp, Server : are responsible for the application UI, capture audio and transform audio into text.
2. Spoken: is responsible for testing if a given phrase is a valid voice command and to extract important information out of it.
3. Spoken VSCode Extension is a Visual Studio Code extension able to receive commands to manipulate VSCode. Is through this extension that Speech2Code is able to control the Visual Studio Code.

8.1.2 Voice commands

Voice commands are transformed into text using the Chrome Speech to Text service and later parsed by Spoken, which makes use of several pushdown automaton to extract information of the text.

Currently, Speech2Code only supports voice commands for the JavaScript language. All commands can be said in English.

8.2 Tools and Technology used

Requirement: Stable and high speed internet connection is required for working of the project.

1. OS: Tested on windows 10
2. IDE: Visual Studio Code
3. Programming Languages: NodeJS, HTML, CSS, ReactJS
4. Memory: At least 3GB of ram and 270MB of available disk space

8.3 Working of the Project

Speech2Code is an application that enables you to code using just voice commands, with Speech2Code instead of using the keyboard to write code in the code editor, you can just express in natural language what you wish to do and that will be automatically written, as code, in the code editor.

Using Speech2Code instead of using the mouse and keyboard to navigate to line 42 of a file, you can just say: "line 42", "go to line 42" or even "please go to line 42". It's possible to say stuff like: new variable answer equals the string Hello, World string

```
let answer = Hello, World
```

Call function max with arguments variable answer and expression gap plus number 42 on namespace Math

```
Math.max(answer, gap + 42)
// gap can later be replaced later by an actual value
```

8.3.1 Workflow

1. The user gives input to the application (in the webapp) in the form of voice commands.
2. The input given in the form of voice commands is passed forward to the Azure Speech to text or Chrome services. Here, the voice commands are converted from speech to text.
3. The output from the above mentioned services is passed to spoken through the server. Spoken extracts the meaning from the text and converts it in the Javascript code.
4. The required output is displayed on the screen.

9 Software Testing

9.1 Type of testing

Manual Testing

9.2 Test cases and test results

Note: The test cases hold true when the device is connected to a stable, high speed internet.

Table 3: Test cases

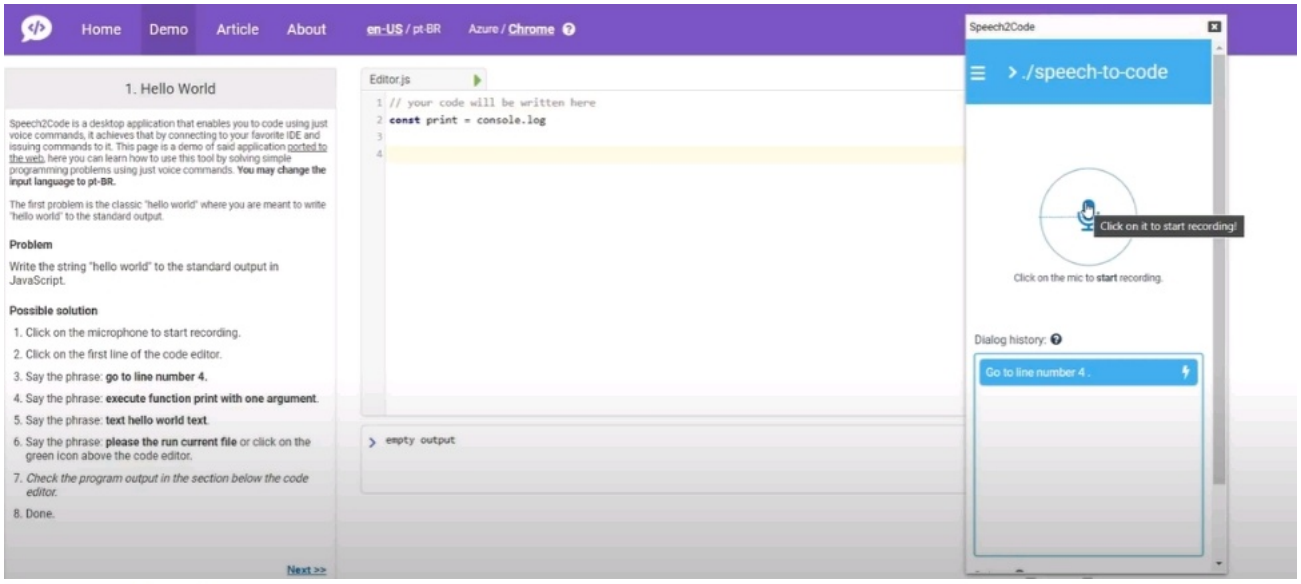
Sr No.	Expected output	Output	Pass/Fail
1	Given a voice input, go to line number 4, the output matches the input	Cursor moves to line number 4	Pass
2	Given a voice input, run the current file, the output matches the input	Executes the file	Pass
3	Given a voice input, function print with two arguments as gap, the output matches the input	Creates a function with two arguments	Pass
4	Given a voice input, select word gap, the output matches the input	Selects the word gap	Pass
5	Given a voice input, text hello world text, the output matches the input	Writes hello world as a string	Pass

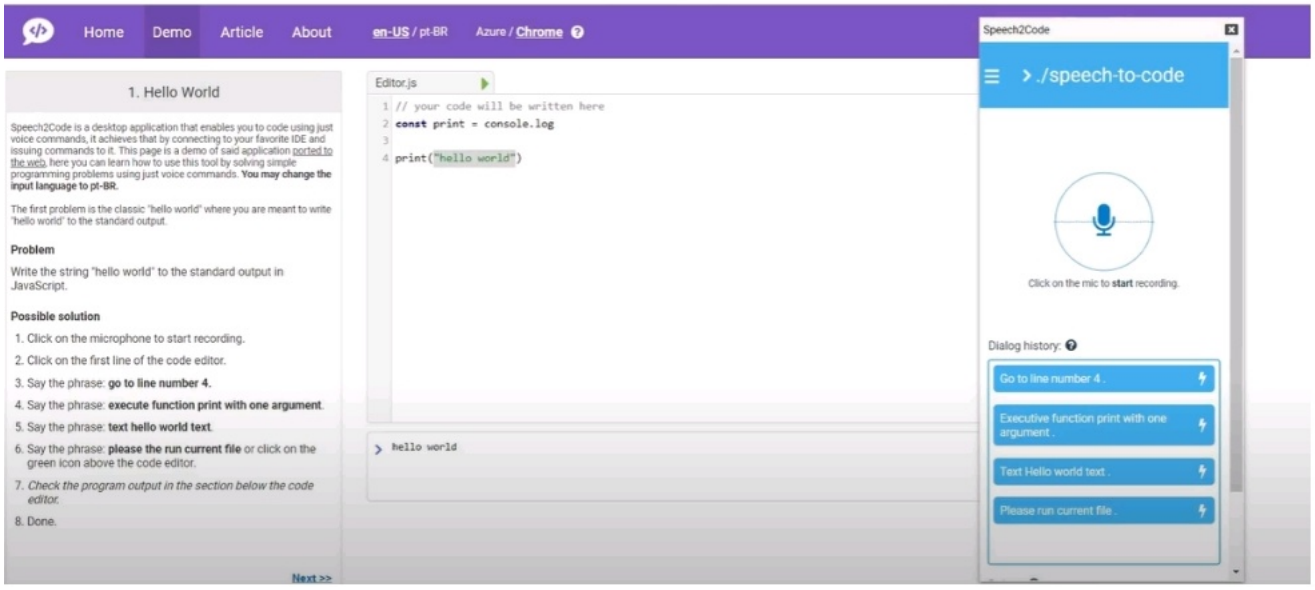
10 Results

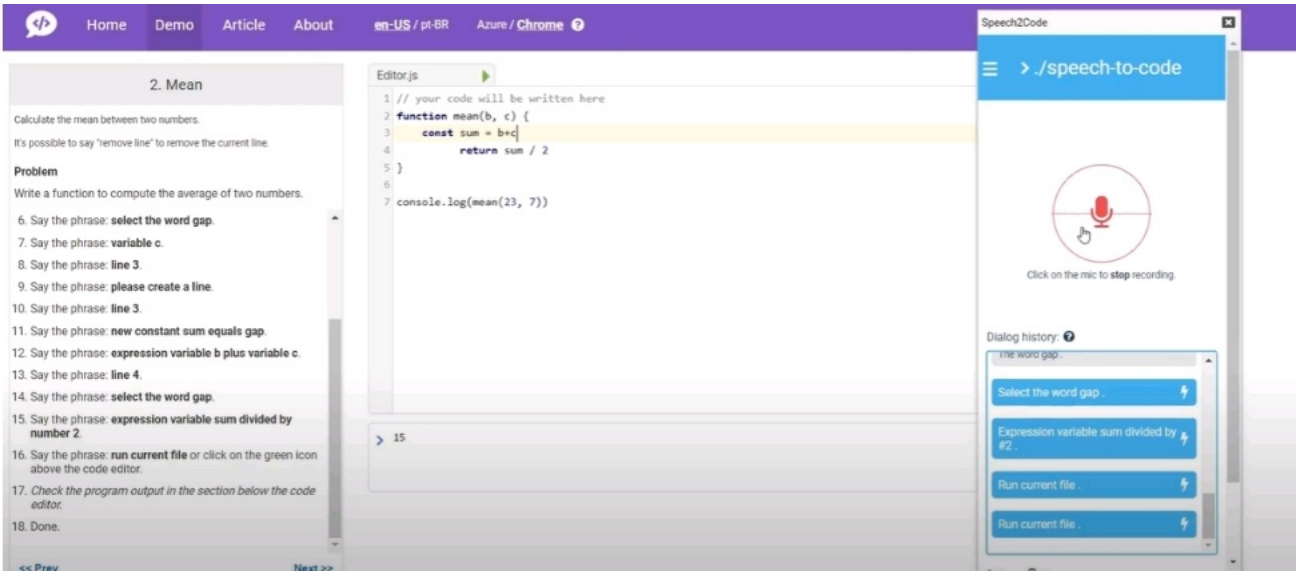
10.1 Outcome

We were successfully able to build an assistive technology tool to help programmers code in JavaScript using voice commands.

10.2 Screenshots






Home
Demo
Article
About
en-US / pt-BR
Azure / Chrome
2. Mean
Calculate the mean between two numbers.
It's possible to say "remove line" to remove the current line.
Problem
Write a function to compute the average of two numbers.
6. Say the phrase: select the word gap.
7. Say the phrase: variable c.
8. Say the phrase: line 3.
9. Say the phrase: please create a line.
10. Say the phrase: line 3.
11. Say the phrase: new constant sum equals gap.
12. Say the phrase: expression variable b plus variable c.
13. Say the phrase: line 4.
14. Say the phrase: select the word gap.
15. Say the phrase: expression variable sum divided by number 2.
16. Say the phrase: run current file or click on the green icon above the code editor.
17. Check the program output in the section below the code editor.
18. Done.
<< Prev
Next >>
Editor.js
// your code will be written here
function mean(b, c) {
const sum = b+c
return sum / 2
}
console.log(mean(23, 7))
> 15
Speech2Code
./speech-to-code
Click on the mic to stop recording.
Dialog history:
Select the word gap .
Expression variable sum divided by #2 .
Run current file .
Run current file .

11 OTHER SPECIFICATIONS

11.1 Advantages

This assistive technology application is capable of enabling programmers to program in JavaScript without the use of their hands, using only voice commands. The purpose of the application is to enable programmers who are unable or with great difficulty to use their hands, whether due to injury or disability, to continue exercising their profession.

11.2 Limitations

1. Currently the application is qualified only for Visual Studio Code Editor and Notepad.
2. Information about supported languages needs to be added manually, an better approach can be taken to add languages automatically.
3. acceptable voice commands needs to be added manually.
4. Requires paid services like Azure to perform natural language processing tasks,

11.3 Applications

Speech recognition systems have applications in several areas. In fact, any activity that involves human-machine interaction can potentially utilize these systems. Currently, several applications are already being designed with a built-in speech recognition system. This same methodology can be used in the creation of other applications with a similar purpose, such as a tool that, in addition to JavaScript, also supports Python.

12 CONCLUSION AND FUTURE WORK

In this work, an assistive technology application capable of enabling programmers to program in JavaScript without the use of their hands, using only voice commands was proposed and developed. The purpose of the application is to enable programmers who are unable or with great difficulty to use their hands, whether due to injury or disability, to continue exercising their profession. Its development was based on the concepts of automata theory, speech recognition and user interface automation. This same methodology can be used in the creation of other applications with a similar purpose, such as a tool that, in addition to JavaScript, also supports Python or is capable of manipulating the linux bash terminal.

In addition to increasing the number of JavaScript language functionality supported, effectively completing Tables 3 and 4, future work includes some improvements in voice command recognition and parsing:

1. Train and create a custom speech recognition template in azure speech to text only with the phrases that represent commands. This will make the speech recognizer prefer these phrases. An example of this is the phrase used to write something on the screen: “write something”. By saying “ write test” out loud is often recognized as “right test”, because the words “ write” and ”right” have similar pronunciation. This can be resolved by creating a custom template or adding more context to the sentence.

2. Evaluate replacing the mechanism for recognizing and analyzing commands, with more robust machine learning-based solutions such as Microsoft Azure LUIS or the Google Cloud Natural Language AI.

3. Include other programming languages in the application which will in turn increase the scope of the project. With Javascript, add HTML and CSS so the application will be ready to use for the front-end web development.

4. Create and deploy client which will work on Visual Studio Code.

13 Appendix A

13.1 Feasibility Study

1. A reasonable priced voice to code extension does not exist at the time study was conducted.

2. Introduction of multiple languages for giving voice commands was done. This increased usefulness and flexibility of the system.

3. The extension worked properly across all platforms like Linux, Mac and windows.

4. The system used both the paid service like Azure Speech to Text service and free Chrome's inbuilt Speech to Text as per users choice and showed no variations in results.

5. The findings of this feasibility study indicated that an effective voice based training delivery system could be developed by integrating an IBM clone personal computer with a graphics board and supporting software, signal processing board and supporting software for audio output and input, and instructional authoring software.

14 Appendix B

14.1 Literature Review

14.1.1 Repetitive Strain Injury among Computer Users: A case study in telecommunication company. [1]

Repetitive Strain Injury (RSI) is the most common occupational injury faced by computer users. Computer users faced higher possibility of getting RSI due to their prolonged working time and static posture. The three main objectives of this study are, first: to identify the prevalence of RSI among computer users; second, to investigate and determine the RSI risk factors; and third to analyze the association between RSI risk factors and the prevalence of RSI among computer users at a Telecommunication Company X. A total of 100 respondents were selected based on their daily exposure to computer usage of more than 4 hours.

Nordic modified questionnaire was used to gather respondents' socio-demographic data, job's information, physical risk exposure, physical symptoms and their awareness level towards RSI. Body Parts Symptoms Survey (BPSS) form were also used to identify the body parts exposed to the RSI risk among computer users. Study results showed that the 41 percent of computer users in Telecommunication Company X felt tired at the upper-back near the neck and 38 percent at the shoulder region. There is a significant association between risk factors of RSI and the prevalence of RSI among the computer users at Telecommunication Company X.

14.1.2 A Study on Automatic Speech Recognition [2]

Automatic speech recognition is one of the most automatic speech processing areas, allowing the machine to understand the user's speech and convert it into a series of words through a computer program, thus creating a kind of natural communication between man and machine. Automatic speech recognition is also called speech recognition; it can be defined as graphical representations of frequencies emitted as a function of time. All speech processing techniques (speech synthesis and processing, speaker identification, Speaker verification make it possible to create voice interfaces (Human Machine Interface) or perform voice interaction. Voice recognition can be applied in several applications.

Speech is a useful expression and has a particular meaning and it is composed of several words, which in turn contain several letters accompanied by voices. This voice can spread in objects of air and empty and inanimate in the form of waves; a wave that overlaps between them or begins in the form of small circles of the sound source. This situation is characterized by force and then widens these circles little by little until they disappear completely when they spread over long distances. Logical speech is done in speakers who talk the same language, meaning that the sender and recipient have the same keys that help them decipher the meaning. The researchers applied this phenomenon and developed it to become a key branch in human communication with the machine where the sound has helped to facilitate the use of the machine with the user and to make a natural communication between them. Automatic speech recognition has greatly contributed to the development of artificial intelligence, which seeks to create very flexible methods of handling the machine, this allows the user to communicate and exchange information without using known input/output modules

such as the keyboard. Voice-based input/output techniques are very useful in several areas, such as the care of disabled people, the use of cars, in particular when driving, distress calls, etc.

14.1.3 Automatic Identification of Stop Words. [3]

A stop word may be identified as a word that has the same likelihood of occurring in those documents not relevant to a query as in those documents relevant to the query. In this paper they have shown how the concept of relevance may be replaced by the condition of being highly rated by a similarity measure. Thus it becomes possible to identify the stop words in a collection by automated statistical testing. They describe the nature of the statistical test as it is realized with a vector retrieval methodology based on the cosine coefficient of document-document similarity.

14.1.4 Deep Variational Filter Learning Models [4]

The performance degradation of speech applications such as voice search or conversational-bots in noisy and reverberant environment demands the need for improved robustness in automatic speech recognition (ASR) systems. While several advancements have been made in the acoustic modeling for ASR, the presence of extrinsic noise sources and reverberations continue to pose challenge to the ASR system deployment. The noise robustness can be partly addressed by multi-condition training (utilizing noisy training data from multiple environments). In spite of this training, the performance difference between multi-condition train-test and the clean train-test of ASR is pronounced, which warrants the need for attaining noise robustness either at speech representation stage or the training stage. This work focuses on the robust representation learning using unsupervised generative modeling method. Modulation filtering is an approach to robust feature extraction that is based on enhancing the key dynamics of the speech signal in the spectro-temporal domain, while suppressing speech modulations that are susceptible to noise/reverberation.

In this work, they propose a new approach to learn temporal and spectral modulation filters from a variational modeling perspective. In particular, convolutional variational autoencoder (CVAE) is used to learn multiple modulation filters from the input spectrogram. To learn multiple irredundant modulation filters, they propose a skip connection based network architecture.

The 'skip' architecture in the encoder of CVAE is employed to serve two purposes: to find residual of the input with some filtered representation (which has already been learnt), and to combine the two different filtered representations to be fed to the rest of the network for joint minimization of the loss function.

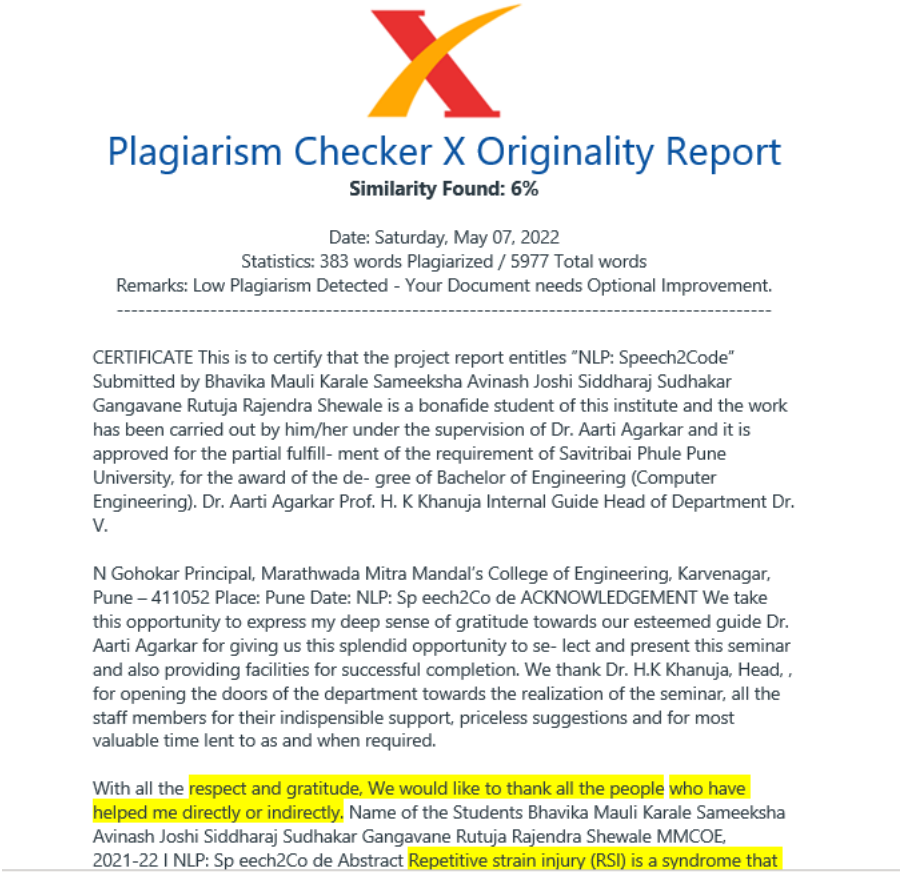
14.1.5 Unifying Speech Recognition and Generation with Machine Speech Chain. [5]

In this paper, they have proposed a closed-loop between ASR and TTS model to construct a machine speech chain mecha- nism. By using similar sequence-to-sequence archi- tecture and interchangeable source-target domain, it allows to train both labeled and unlabeled data in a single loop. The ASR model transcribes the unla- beled speech utterances, then TTS reconstructs the speech waveform based on the predicted text from ASR. In the opposite case, the TTS reconstructs the speech features, then ASR reconstructs the text based on predicted speech from TTS. Based on the knowledge, this is the deep learning model that integrates human speech perception and production behaviours.

It consists of a sequence-to- sequence ASR, a sequence-to-sequence TTS, and a loop connection from ASR to TTS and from TTS to ASR. The key idea is to jointly train both the ASR and TTS models.

15 Appendix C

15.1 Plagiarism Report



Plagiarism Checker X Originality Report

Similarity Found: 6%

Date: Saturday, May 07, 2022
Statistics: 383 words Plagiarized / 5977 Total words
Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

CERTIFICATE This is to certify that the project report entitles "NLP: Speech2Code" Submitted by Bhavika Mauli Karale Sameeksha Avinash Joshi Siddharaj Sudhakar Gangavane Rutuja Rajendra Shewale is a bonafide student of this institute and the work has been carried out by him/her under the supervision of Dr. Aarti Agarkar and it is approved for the partial fulfill- ment of the requirement of Savitribai Phule Pune University, for the award of the de- gree of Bachelor of Engineering (Computer Engineering). Dr. Aarti Agarkar Prof. H. K Khanuja Internal Guide Head of Department Dr. V.

N Gohokar Principal, Marathwada Mitra Mandal's College of Engineering, Karvenagar, Pune – 411052 Place: Pune Date: NLP: Sp eech2Co de ACKNOWLEDGEMENT We take this opportunity to express my deep sense of gratitude towards our esteemed guide Dr. Aarti Agarkar for giving us this splendid opportunity to se- lect and present this seminar and also providing facilities for successful completion. We thank Dr. H.K Khanuja, Head, , for opening the doors of the department towards the realization of the seminar, all the staff members for their indispensible support, priceless suggestions and for most valuable time lent to as and when required.

With all the respect and gratitude, We would like to thank all the people who have helped me directly or indirectly. Name of the Students Bhavika Mauli Karale Sameeksha Avinash Joshi Siddharaj Sudhakar Gangavane Rutuja Rajendra Shewale MMCOE, 2021-22 I NLP: Sp eech2Co de Abstract Repetitive strain injury (RSI) is a syndrome that

References

[1] Nurul Huda Baba, Dian Darina Indah Daruis, "Repetitive Strain Injury (RSI) among computer users: A case study in telecommunication company ",*Article in Malaysian Journal of Public Health Medicine, January 2016.*

[2] Sahila Benkerzaz,Youssef Elmir, Abdeslem Dennai, "A Study on Automatic Speech Recognition", *August 2019.*

[3] W John Wilbur, Karl Sirotkin, "Automatic Identification of Stop Words ", *National Center of Biotechnology Information, USA.*

[4] SPurvi Agrawal, Sriram Ganapathy, "Deep Variational Filter Learning Models for Speech Recognition", *Indian Institute of Science, Bangalore.*

[5] Andros Tjandra,, Sakriani Sakti, Satoshi Nakamura, "Unifying Speech Recognition and Generation with Machine Speech Chain", *Nara Institute of Science and Technology, Japan.*