KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade
Anand Nagar, Krishnankoil - 626126. Srivilliputtur (Via), Virudhunagar (Dt), Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

SCHOOL OF COMPUTING
DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

A Project Report On

CODE MEETS CONTACTLESS: A PYTHONIC APPROACH TO NFC APPLICATIONS

Submitted for partial fulfilment of the requirements for the award of the degree of

MASTER OF SCIENCE IN DATA SCIENCE

BY

Mr. Om Prakash.E (9923146021)
Ms. Namrutha.S.R (9923146018)

Under The Guidance Of

Mrs. G. VIJAYALALITHA

Assistant Professor
Department of Computer Applications
Kalasalingam Academy of Research and Education

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
Kalasalingam Academy of Research and Education
Deemed to be University
(ACCREDITED BY NAAC WITH"A"GRADE)
Anand Nagar- Krishnankoil- 626126
January– 2024

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade
Anand Nagar, Krishnankoil - 626126. Srivilliputtur (Via), Virudhunagar (Dt), Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

School of Computing

Department of Computer Science & Information Technology

CERTIFICATE

This is to certify that the project work entitled **"CODE MEETS CONTACTLESS: A PYTHONIC APPROACH TO NFC APPLICATIONS"** is a bonafide work carried out by Mr. Om Prakash.E (9923146021) and Ms. Namrutha.S.R (9923146018), in partial fulfillment of the requirements for the award of the degree of MASTER OF DATA SCIENCE by KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION, Krishnankoil, under our guidance and supervision during the academic year 2023-2023.

PROJECT GUIDE

Mrs. G. VIJAYALALITHA

Assistant Professor,
Department of Computer Applications
KARE

HEAD OF THE DEPARTMENT

Dr. K. BALASUBRAMANIAM

Professor & Head
Department of Computer Applications
KARE

Submitted for the project Viva-Voce examination held on ________________

Internal Examiner

External Examiner

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade
Anand Nagar, Krishnankoil - 626126. Srivilliputtur (Via), Virudhunagar (Dt), Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

School of Computing

Department of Computer Science & Information Technology

DECLARATION

This is to certify that the work reported in the present project entitled **"CODE MEETS CONTACTLESS: A PYTHONIC APPROACH TO NFC APPLICATIONS"** is a record of work done by us in the Department of Computer Science and Information Technology, Kalasalingam Academy of Research and Education. The report is based on the project work done entirely by us and not copied from any other source.

Place: Krishnankoil

Date: **Signature of the Students**

OM PRAKASH.E (9923146021)

NAMRUTHA.S.R (9923146018)

.

ACKNOWLEDGEMENT

First of all we thank the almighty God, who showered blessing on us for succeeding on the project work. Our sincere and heartily thanks go to our parents who are the heart and soul of our life.

We would like to express our wholehearted respects and thanks to our Chancellor "Ilayavallal" **Dr.K.Sridharan** and our honorable Vice President **Dr.S.Shasi Anand**, Kalasalingam Academy of Research and Education, Krishnankoil who gave me the golden opportunity to do this project, which also helped me in doing a lot of Research and I came to know about so many new things.

We would like to thanks to our Vice-Chancellor **Dr.S.Narayanan,** Kalasalingam Academy of Research and Education, Krishnankoil for having offered the facilities required to complete this project.

I am also thankful to Head of the department **Dr. K. Balasubramanian** for providing excellent infrastructure and a nice atmosphere for completing this project successfully

I would like to express my sincere gratitude and indebtedness to my project supervisor **Mrs. G. Vijayalalitha** for her valuable suggestions and interest throughout the course of this project

Finally, I would like to take this opportunity to thank my family for their support through the work. I sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

Imagine a world where walls whisper directions, doorways announce their purpose, and knowledge flows effortlessly through a simple tap. This is the vision behind our project, where near field communication (NFC) technology mediates the physical and the perceived, empowering visually impaired individuals to rent with newfound confidence of the strong points.

We offer a Python-orchestrated symphony of accessibility, with strategically placed NFC tags that act as silent sentinels, collecting sensitive information such as place names, directional signs, and even room information in their digital vaults with users equipped with NFC readers. They are information drivers, opening each tag with a gentle touch. In response, mobile applications, their own master, convert the scanned data into a composite language, creating clear audio visualizations of their surroundings and not destinations only reported; It tells the story of the journey, navigates roads and corners with confidence, and inspires independence and confidence every step of the way.

Our work is not limited to technology; It’s about empowerment. It’s about reclaiming independence, the joy of exploration, and the dignity of navigating the world on your own terms. It’s about brightly colored public spaces that are inclusive, where everyone can join the big orchestra of life.

CHAPTER1

INTRODUCTION

OVERVIEW

This project attempts to address the specific challenges faced by visually impaired individuals in navigating unfamiliar environments, such as educational institutions. The focus is on developing an intuitive and accessible navigation system utilizing Near Field Communication (NFC) technology.The project envisions a scenario where a visually impaired individual carries an NFC tag, uniquely programmed with information about the destination classroom. The NFC tag is scanned by an NFC shield connected to an Arduino, initiating a sequence of interactions aimed at providing clear and audible guidance to the user.

This GUI is not just a static interface; it is a dynamic space that reads and writes data on the NFC tag, creating a two-way communication channel. The GUI becomes a personalized guide, offering multilingual audio instructions tailored to the individual's preferred language. In essence, the visually impaired user experiences not just navigation but a dialogue with technology, enhancing their sense of autonomy and engagement with the surrounding environment.

Throughout this project, our focus extends beyond functionality to user-centric design, ensuring that the GUI is not only a conduit for information but also a tool that individuals with visual impairments can navigate with ease. The GUI's ability to both read and write on the NFC tag further expands the scope of interaction, opening avenues for personalized data storage and retrieval.

DOMAIN

This project resides in the domain of Assistive Technologies, focusing on enhancing the navigation experience for individuals with visual impairments in educational settings. Leveraging Near Field Communication (NFC) technology, an Arduino-based system facilitates communication between personalized NFC tags and a computer. The Python script on the computer employs text-to-speech conversion, adhering to principles of Human-Computer Interaction (HCI) for a user-friendly experience. Additionally, the project addresses considerations of security and privacy in handling personal information stored on NFC tags, contributing to the broader fields of accessibility, wireless communication, and educational technology.

SCOPE OF WORK IN THE PROJECT

This project's scope is vast and involves a multitude of tasks and activities, including hardware, software, and user interaction. Hardware Integration: NFC tags are programmed and associated with specific classroom information. The NFC shield is connected and configured with Arduino for communication with NFC tags. Arduino scripts are developed to read and process data from NFC tags. Communication Protocols: A reliable serial communication link is established between the Arduino and the computer for data transmission. A consistent format is defined for the data sent from Arduino to the computer.

Software Development: A Graphical User Interface (GUI) using Python is created to process and display information received from the Arduino.Text-to-speech conversion is implemented within the python GUI to generate audio instructions. A feature is implemented allowing users to select their preferred language for audio guidance.. User Interface and Interaction: It is ensured that the Python GUI generates clear and concise audio instructions for the user. Robust error handling is implemented to gracefully manage unexpected situations during NFC scanningor data transmission.

WORKING PRINCIPLE

1. **Initialization**: When your Arduino system starts, the first step is to initialize the NFC shield and set up the UART communication. This involves configuring the necessary parameters such as baud rate, data bits, stop bits, and parity.
2. **UART Communication**: The NFC shield and the Arduino communicate with each other using UART, which is a serial communication protocol. UART involves two pins: TX (Transmit) and RX (Receive). The Arduino's TX pin is connected to the NFC shield's RX pin, and vice versa. This allows them to send and receive data in a synchronized manner.
3. **Connecting NFC Tags**: NFC tags, which are passive devices that can store data, are brought into proximity with the NFC shield. The shield generates a magnetic field, and when an NFC tag enters this field, it is powered and able to communicate with the shield.
4. **Reading NFC Tags**: The NFC shield, in response to the presence of an NFC tag, reads the data stored on the tag. This data could include unique identifiers, authentication information, or any other relevant information programmed onto the tag.
5. **Data Transfer**: The NFC shield, upon successfully reading the data from the NFC tag, transmits this information to the Arduino through the UART communication. The Arduino's RX pin receives this data, and the Arduino sketch can then process it.
6. **Arduino Processing**: In the Arduino sketch, you can define specific actions based on the data received from the NFC shield. This may include tasks such as displaying information on an LCD screen, activating a specific output (like an LED or a motor), or even triggering communication with other devices or systems.
7. **Feedback to NFC Shield**: Depending on the project requirements, the Arduino may send feedback or data back to the NFC shield through UART communication. This could involve updating information on the NFC tag, writing data to a different NFC tag, or even initiating further communication with other devices.
8. **User Interaction**: In some applications, the Arduino might be connected to user interfaces such as buttons, displays, or sensors. The Arduino can use the

information obtained from the NFC tag to customize the user experience or provide specific feedback.

9. **Continuous Monitoring**: The system can continuously monitor for the presence of NFC tags and repeat the process as needed. This enables real-time interaction with NFC-enabled devices or objects.

By understanding these working principles, you can design and implement projects that leverage the capabilities of Arduino and NFC technology for various applications, such as access control systems, interactive displays, or personalized device configurations.

CIRCUIT DIAGRAM

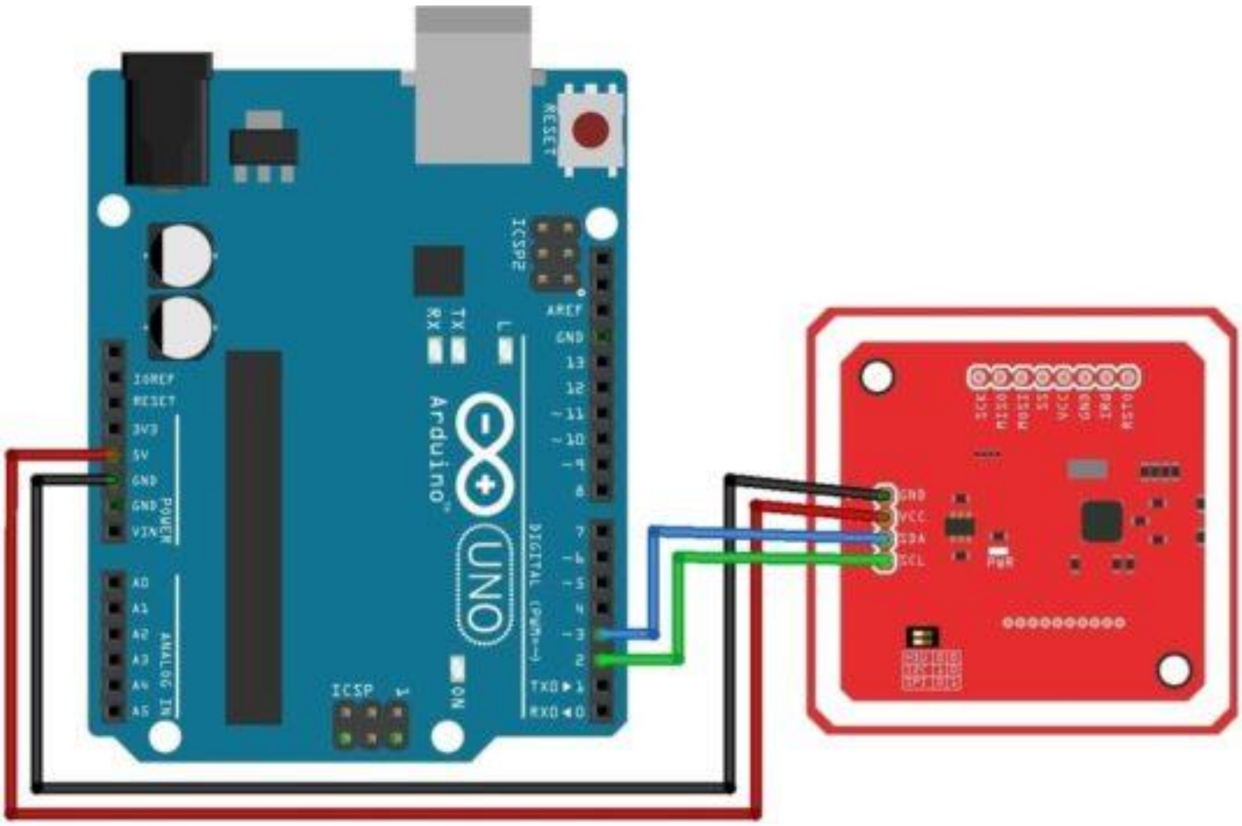


Figure: 1.1

Interfacing PN532 with Arduino in UART Mode

- GND (Ground) <-> GND
- Vcc (Power supply) <-> 5V
- Tx (Clock) <-> D3
- Rx (Data) <-> D2

FLOWCHART

Start

User carries the NFC scanning device

User approaches NFC placed area

NFC tag is scanned by NFC shield connected to Arduino

Arduino processes scanned data

Arduino sends data to Python GUI through serial communication

Python GUI reads data on the NFC tag

Python GUI converts information to multilingual audio

Audio instructions are heard by the user

End

Explanation of the flowchart:

- Start: The process begins when the user approaches a classroom.
- NFC Tag Interaction: The NFC tag reader scans the tag attached to the classroom. The classroom name is read from the tag's memory.
- Arduino Processing: The Arduino receives the classroom name data from the NFC reader. It sends the data to the Python program through serial communication.
- Python Interpretation: The Python program receives the classroom name from the Arduino. It converts the text into an audio file using a text-to-speech library. The audio announcement is played through the speakers or headphones.
- End: The user hears the classroom name, informing them of their location

CHAPTER 2

SYSTEMANALYSIS

EXISTINGSYSTEM

As of my last knowledge update in January 2023, there are various existing technologies and systems designed to assist individuals with visual impairments in navigation. It's essential to note that technology evolves rapidly, and there may be new developments or systems introduced since then. Here are some existing systems and technologies for navigation assistance for the visually impaired:

1. White Canes: Traditional white canes remain a widely used tool for individuals with visual impairments. They provide tactile feedback and detect obstacles on the ground.
2. GPS Navigation Apps: Apps like BlindSquare, Lazarillo, and Soundscape leverage GPS technology to provide real-time navigation assistance, location descriptions, and nearby points of interest.
3. Indoor Navigation Systems: Systems like BlindSquare Event and RightHear focus on indoor navigation, utilizing Bluetooth beacons or Wi-Fi networks to guide users within buildings or complexes.
4. Smart Canes: Smart canes equipped with sensors and technologies like ultrasonic sensors or infrared sensors can detect obstacles and provide haptic or auditory feedback.
5. Wearable Devices: Wearable devices, such as smart glasses or vests, equipped with cameras and sensors, can detect obstacles and provide auditory feedback to the user.
6. Voice Assistants: Voice-controlled virtual assistants like Amazon Alexa, Google Assistant, or Apple's Siri can provide information, answer queries, and assist with various tasks.
7. Accessible Pedestrian Signals (APS):APS are equipped with audible signals at pedestrian crossings, providing information about the status of traffic lights to assist in safe street crossing.

8. Electronic Travel Aids: Devices like the Binaural Echo Location Development (BELD) system use echolocation principles to help individuals navigate by interpreting echoes of sounds.

PROPOSEDSYSTEM

The proposed system for the NFC-based blind navigation project involves integrating NFC technology, Arduino-based hardware, and a Python-based Graphical User Interface (GUI) to provide a comprehensive and user-friendly navigation solution for visually impaired individuals.

User Interaction: Visually impaired user approaches the NFC scanning area.

NFC Tag Scanning: NFC tag is scanned by an NFC shield connected to an Arduino.

Arduino Processing: Arduino processes data from the NFC tag and transmits it to the computer.

Python GUI Initialization: Python GUI is initiated on the computer.

Read/Write Operations on NFC Tag: Python GUI reads data from the NFC tag.

Language Preference Selection: User selects preferred language through the GUI.

Text-to-Speech Conversion: Python GUI converts classroom information to audio based on the selected language.

Audio Output: Multilingual audio instructions are played through speakers or headphones for the user.

User Interaction Loop: The system can loop back to NFC scanning for navigation to multiple destinations.

FEATURES OF THE PROPOSED SYSTEM

1. Bi-Directional Interaction: The GUI has the ability to both read and write data on the NFC tag, allowing for personalized data storage.
2. Multilingual Support: Users can choose their preferred language for audio instructions, enhancing accessibility.
3. Real-time Interaction: Users receive real-time audio guidance, creating an interactive and responsive navigation experience.
4. User-Friendly Interface: The Python GUI is designed with a focus on simplicity and accessibility for individuals with visual impairments.
5. Security Measures: Data encryption is implemented to ensure secure communication between the Arduino and the computer.

CHAPTER 3

FUNCTIONALITY

Arduino uno

The Arduino Uno is a widely-used open-source microcontroller board renowned for its versatility and accessibility in the realm of electronics and programming. Powered by the Atmega328P microcontroller, it boasts 14 digital input/output pins, including 6 PWM-capable outputs, and 6 analog input pins, making it an ideal platform for connecting a variety of sensors, actuators, and other electronic components. It features a microcontroller, digital and analog input/output pins, USB connectivity, and compatibility with various expansion boards, or "shields," that can extend its capabilities.Its USB interface facilitates easy communication and programming, enabling seamless connectivity to computers. Arduino Uno supports expansion through shields—additional boards that extend its capabilities, making it adaptable to diverse applications such as home automation, robotics, interactive art, and educational projects.

Figure: 4.1

NFC SHEILD

NFC Shield is a Near Field Communication interface for Arduino build around the popular NXP PN532 integrated circuit. NFC is a short-distance radio technology that enables communication between devices that are held close together.NFC traces its roots in RFID technology and is an open platform technology standardized in ECMA-340 and ISO/IEC 18092.

The shield includes an NFC controller that manages the communication between the Arduino and NFC-enabled devices or tags. An integrated antenna allows the shield to send and receive NFC signals, facilitating close-range communication with other NFC devices. NFC shields are designed to be compatible with Arduino boards, allowing for easy integration into Arduino projects.

NFC is a short-range communication technology, typically operating within a range of a few centimeters. The shield's specifications will define the effective communication range.NFC operates at specific frequencies, and the shield should support the relevant NFC frequency bands, such as 13.56 MHz.NFC shields may draw power directly from the Arduino board or require an external power source, depending on the shield's power requirements.

The shield will have a specified data transfer speed for communication with NFC-enabled devices or tags. Arduino libraries are often available for NFC shields, providing pre-written functions and methods that simplify the programming and interaction with the shield.NFC shields may support both reading and writing operations, allowing you to interact with and modify data on NFC-enabled tags.

Figure: 4.2

NFC TAG

An NFC (Near Field Communication) tag is a small electronic device or sticker that contains a microchip and an antenna. These tags are designed to store and transmit data when in close proximity to an NFC reader or another NFC-enabled device. NFC tags are passive devices, meaning they do not require a power source of their own. Instead, they draw power from the electromagnetic field generated by the NFC reader.

The microchip embedded in an NFC tag holds the data that can be read by an NFC reader. This data can range from a simple identification number to more complex information, depending on the type and capacity of the NFC tag.NFC tags have an antenna that allows them to receive power from the NFC reader and transmit data back to the reader.

NFC tags are passive devices, relying on the energy supplied by the NFC reader to operate. They become active only when in the presence of an NFC field generated by a compatible reader.NFC tags come in various types with different memory capacities. Some tags only store a small amount of data, while others may have larger memory for storing more extensive information. NFC tags can store a variety of data types, including text, URLs, contact information, and commands. This versatility makes them suitable for a wide range of applications.

Figure: 4.3

CHAPTER 5

Software implementation

ARDUINO IDE

Arduino IDE (Integrated Development Environment) is a software application that is used to program and develop code for Arduino microcontrollers. Arduino is an open-source platform that consists of both hardware and software components, and it is designed to make it easy for beginners and hobbyists to build electronic projects.

The Arduino IDE is a cross-platform application that runs on Windows, Mac, and Linux operating systems. It provides a simple and user-friendly interface for writing and uploading code to Arduino boards. The IDE includes a code editor with features such as syntax highlighting, auto-completion, and debugging tools that help developers write efficient and error-free code.

The Arduino IDE also includes a library manager that makes it easy to install and manage libraries of pre-written code. These libraries can be used to add additional functionality to Arduino projects, such as communicating with sensors or controlling motors.

One of the key advantages of the Arduino IDE is that it is open-source, meaning that it can be modifiedandextendedbytheusercommunity.Thishasledtothedevelopmentofmanypluginsand extensions that add additional features and functionality to the IDE.

Overall, the Arduino IDE is an essential tool for anyone who wants to program Arduino boards, whether they are beginners or experienced developers. It provides a powerful and flexible environment for developing electronic projects of all kinds.

ROLE OF ARDUINO IDE

- Sketch Development:The Arduino IDE is the primary environment for developing the Arduino sketch, which is the program that runs on the Arduino microcontroller.
- Hardware Configuration:The Arduino IDE assists in selecting the appropriate Arduino board model (e.g., Arduino Uno) and the corresponding communication port to which the Arduino is connected.
- Library Integration:The IDE facilitates the inclusion of libraries that provide pre-written code for specific functionalities, such as NFC communication.
- Code Compilation:The Arduino IDE compiles the written sketch into machine code that the Arduino microcontroller can execute.
- Serial Communication:The Arduino IDE provides a Serial Monitor tool for monitoring and debugging the communication between the Arduino and the computer.
- Code Upload:The IDE simplifies the process of uploading the compiled code (firmware) to the Arduino board.
- Firmware Updates:The Arduino IDE is used for uploading firmware updates when modifications or improvements are made to the code.

PYTHON GUI

The Python GUI or Graphical User Interface acts as an interface between users and Python applications, presenting information visually and allowing users to interact with the application through graphical objects The most widely used GUI library in Python one of which is Tkinter, which includes the standard Python distribution. Tkinter provides modules and classes for creating windows, buttons, fonts, and other GUI components. Developers can use Tkinter to create intuitive and visually appealing interfaces through a combination of code and design tools. Another popular option is PyQt, which is a set of Python bindings for the Qt application framework. PyQt is known for its robust features, offering a wide range of customizable widgets and supporting graphics through Qt Designer and programming through Python code GUI programming in Python typically follows an event-driven process background, with actions such as button clicks or text input triggering default actions. This approach enables the creation of interactive applications that enhance the user experience, making GUI development an integral part of software design in various industries.

ROLE OF PYTHON GUI

Python GUI (Graphical User Interface) plays an important role in facilitating user interaction, processing data from Arduino, and audio output for visually impaired user Here are the basic Python GUI functions of your project:

- Data Processing and Interpretation:The Python GUI receives data transmitted by the Arduino through the serial port. It interprets this data, which typically contains information about the destination classroom extracted from the scanned NFC tag.
- Text-to-Speech Integration:The GUI incorporates text-to-speech functionality, utilizing a Python library or external tool to convert the interpreted data (classroom information) into audible instructions.
- Multilingual Support:The GUI allows users to select their preferred language for receiving audio instructions. It manages the integration of multilingual capabilities to cater to diverse user preferences.
- User Interface Design:The GUI design focuses on creating an accessible and user-friendly interface for visually impaired individuals. It includes components such as buttons, labels, and audio output sections.
- Error Handling and Feedback:The GUI incorporates mechanisms for handling errors or unexpected situations that may arise during data processing or communication with the Arduino.
- Real-time Interaction:The GUI establishes a real-time interaction with the Arduino, continuously receiving and processing data. It dynamically updates the interface based on the information received.
- User Preferences and Settings:The GUI allows users to configure preferences, such as language settings or audio volume, providing a customizable experience.

CHAPTER 5

OBSERVATION AND RESULTS

OBSERVATION

The NFC-based blind navigation system stands out as the most innovative user-assisted technology solution designed to empower visually impaired individuals to navigate educational environments with a Python GUI for information processing Includes Arduino microcontroller and NFC gradient Seamless integration of hardware components reflects a holistic approach to address the specific needs of targeted user groups Real-time communication between hardware and software components, with faults along with robust management strategies demonstrates a commitment to providing a reliable and quality user experience

The outstanding strength of the project is its user-centered design, as reflected in features such as multilingual support, customizable preferences, text-to-speech integration, and more incorporated These features acknowledge the needs and preferences of visually impaired individuals, helping to move towards a personalized and inclusive approach to use. The project's emphasis on real-world testing and user feedback reflects an iterative development process, ensuring that the system is designed with practical user experience

Ethical considerations, including obtaining user consent for real-world testing and compliance with privacy laws, underscore the project's commitment to responsible technological development Selection of choice applying the policy to the educational context is consistent with the broader societal goal of promoting inclusion and equal access to education.

RESULT

The NFC-based blind navigation system has proven to be a successful combination of hardware and software components to ensure real-time communication and precise data processing The user-centered design of the Python GUI, which has multilingual support and customizable preferences, enhance the overall user experience Continuous improvement and adaptation to user needs through an iterative development process that includes user feedback and world-class testing own Ethical considerations including user consent and compliance with privacy contribute to system reliability. Initial positive user feedback demonstrates the effectiveness of the system in providing reliable transportation assistance to individuals with visual impairments in educational settings. This project shows promise to foster an inclusive learning environment, increasing user confidence and autonomy. Overall, NFC-based navigation systems for the blind offer practical and effective solutions to address the unique challenges faced by visually impaired individuals.

CHAPTER 6

CONCLUSIONANDFUTUREWORK

The NFC-based blind navigation system represents a significant advance in assistive technology, featuring improved hardware and software integration, user-centric design Successful real-time processing of data from NFC tags, including multilingual support and customizable preferences , which contribute to system reliability and user confidence. Initial positive feedback from users in educational settings highlights the potential impact on autonomy and strategic development.

The project lays the foundation for future growth and development that will further enhance its capacity and impact. Some possible future strategies include:

- Enhanced sensor integration: Consider features such as obstacle detection and environmental awareness and investigate the integration of sensors or other technologies to provide improved navigation capabilities.
- Mobile Application Integration: Create a mobile application companion for the system, allowing users to access navigation instructions on their smartphone. This creates a portable and personalized experience.
- Machine Learning for User Adaptation: Machine learning algorithms were used to allow the system to adapt to individual user preferences and adjust its guidance based on historical data.
- Extended Accessibility Features: Integrate features such as voice recognition and proximity alerts for hands-free communication to increase overall system availability
- Community Collaboration and Open Source Development: Consider the system's codebase as open source, fostering collaboration and allowing developers around the world to contribute to its development.
- User Education and Training Programs: A comprehensive training program and educational materials were developed to provide users with the necessary skills to maximize the benefits of the system.
- Expansion to Different Environments: Explore adapting the system for use in a variety of settings beyond educational settings, such as public transportation hubs or shopping centers.

- Continuous User Feedback and Iterative Development: Establish a framework for ongoing user feedback and engage in iterative continuous improvement to meet emerging needs and challenges.

Beyond these future projects, NFC-based navigation systems for the blind could turn out to be even more sophisticated and revolutionary solutions, making a lasting impact on the lives of individuals with low vision and contributing to technology supporting has improved.

APPENDICES

A.SAMPLECODE

```
import tkinter as tk

from tkinter import ttk

from ttkthemes import ThemedTk

import serial

from googletrans import Translator, LANGUAGES

import pyttsx3

translator = Translator()

engine = pyttsx3.init()

root = ThemedTk(theme="arc")  # Use a ttk themed Tkinter window

root.title("NFC Tag Reader")

icon_path = "C:\\Users\\DELL\\Downloads\\logoforGUI.ico"

root.iconbitmap(icon_path)

serial_port = "COM16"  # Replace with your actual port

baud_rate = 115200

ser = serial.Serial(serial_port, baud_rate, timeout=1)

bg_color = "#303030"  # Dark grey 5

button_color = "#D2B48C"  # Light brown 3

def play_original_script():

    tag_info = message_label["text"].split("\n")[1].split(":")[-1].strip()  # Extracting the original
message content

    engine.say(tag_info)
```

```
    engine.runAndWait()
def play_translated_script():
    try:
        if engine.isBusy():
            engine.stop()
        tag_info = message_label["text"].split("\n")[-1].split(":")[-1].strip()
        engine.say(tag_info)
        engine.runAndWait()
    except RuntimeError as e:
        print(f"Error: {e}")
def update_dropdown(event):
    entered_text = language_var.get().lower()
    matching_items = [lang for lang in language_options if lang.lower().startswith(entered_text)]
    language_menu['values'] = matching_items
    language_menu.focus_set()
    language_menu.event_generate('<Down>')
def set_speed(speed):
    engine.setProperty('rate', speed)
def set_speed_and_play():
    speed = float(speed_var.get())
    set_speed(speed)
    play_original_script()
def exit_gui():
```

```
    root.destroy()

def change_speed(speed_ratio):

    speed = float(speed_ratio)

    set_speed(speed)

    play_original_script()

style = ttk.Style()

style.configure("TLabel", background=bg_color, font=("Arial", 14), foreground="white")

style.configure("TButton", font=("Arial", 12), foreground="black", background=button_color)

message_label = ttk.Label(root, text="", padding=(10, 10))

message_label.pack(fill="both", expand=True)

language_options = list(LANGUAGES.values())

language_var = tk.StringVar()

language_menu = ttk.Combobox(root, values=language_options, textvariable=language_var,
style="TCombobox")

language_menu.set(language_options[0])

language_menu.pack(pady=10)

language_menu.bind("<KeyRelease>", update_dropdown)

play_original_button = ttk.Button(root, text="Play Original Script",
command=play_original_script, style="TButton")

play_translated_button = ttk.Button(root, text="Play Translated Script",
command=play_translated_script, style="TButton")

exit_button = ttk.Button(root, text="Exit", command=exit_gui, style="TButton")

speed_buttons = [

    ttk.Button(root, text="0.75x", command=lambda: change_speed(0.75), style="TButton"),
```

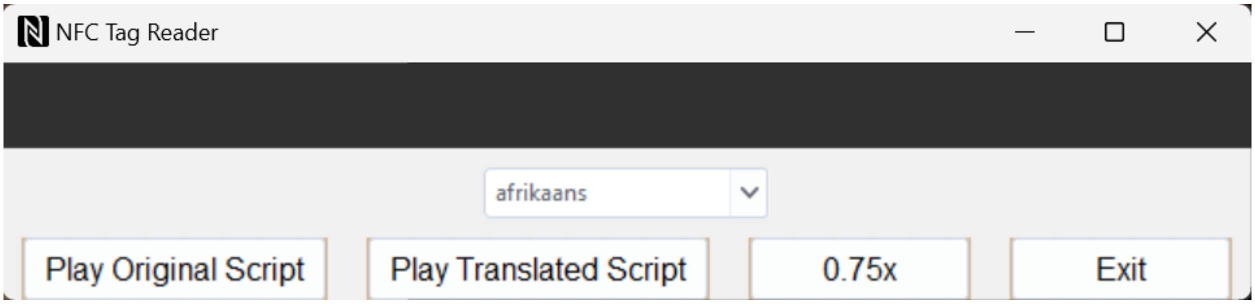
```
]
play_original_button.pack(side="left", padx=10)
play_translated_button.pack(side="left", padx=10)
exit_button.pack(side="right", padx=10)
for button in speed_buttons:
    button.pack(side="left", padx=10)
tag_info_dict = {
    "99.46.182.18": "To your right you have Admin block",
    "195.233.59.146": "Go straight for Krishna Auditorium",
    "4.133.125.58": "Turn left for Block nine"
}
def read_tag_data():
    while True:
        line = ser.readline().decode("utf-8").strip()
        if line.startswith("tagId is :"):
            tag_id = line.split(":")[1].strip()
            translate_and_display_text(tag_id)
            play_original_script()  # Play the original script every time a card is read
def translate_and_display_text(tag_id=None):
    if tag_id:
        tag_info = tag_info_dict.get(tag_id, "Tag information not found")  # Handle missing tag IDs
        message_label["text"] = f"Tag ID: {tag_id}\nOriginal Message: {tag_info}"
    else:
```

```
        tag_info = message_label["text"].split("\n")[1].strip()
    translated_text = translator.translate(tag_info, dest=language_menu.get()).text
    message_label["text"] += f"\nTranslated Message: {translated_text}"
import threading
thread = threading.Thread(target=read_tag_data)
thread.start()
root.mainloop()
```

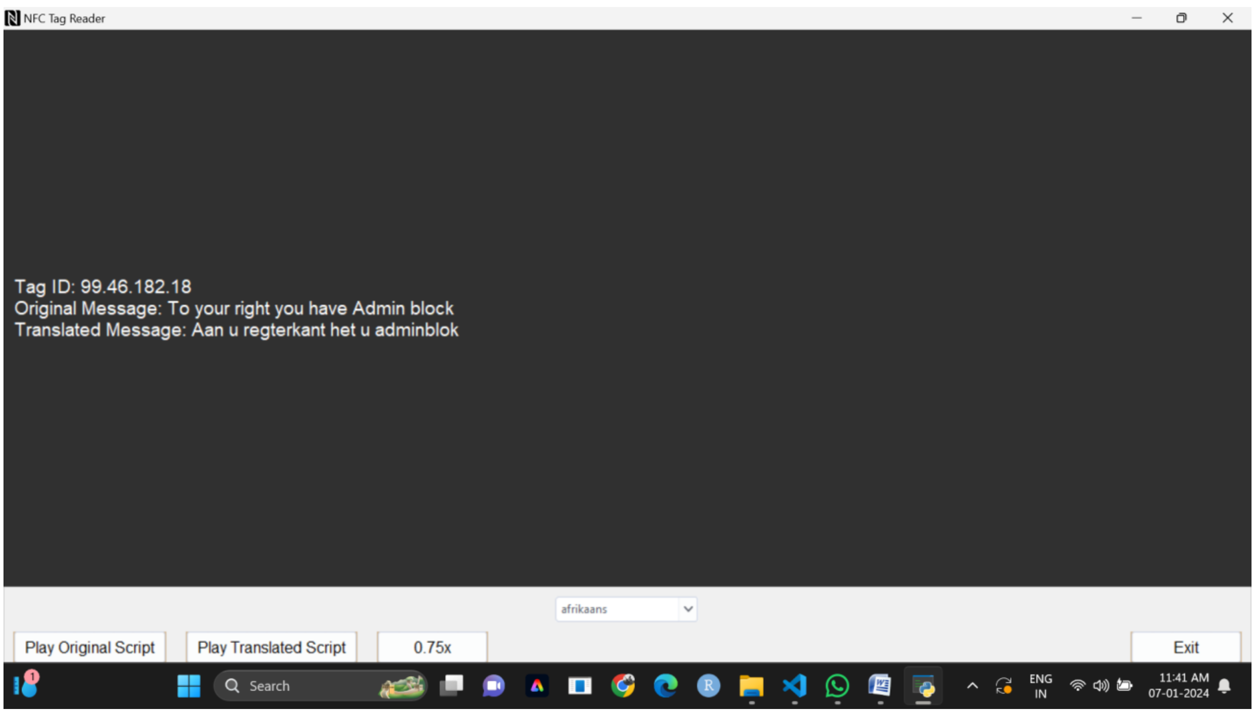
APPENDICES

B.SCREENSHOTS

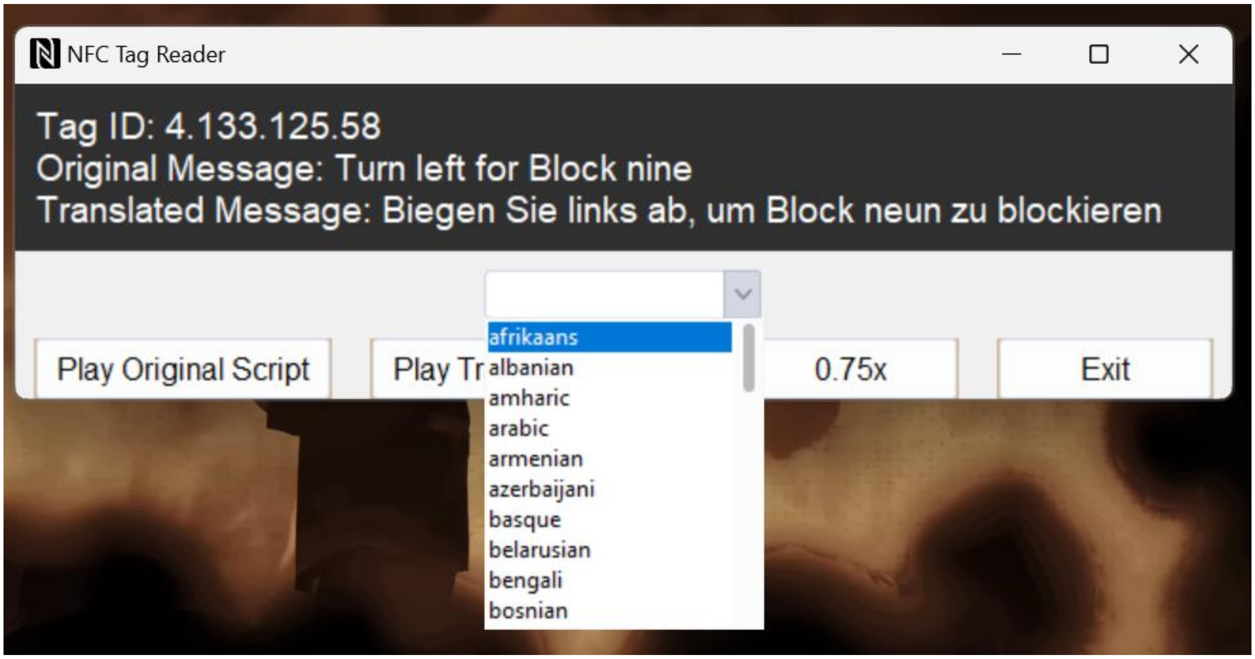
1. The first pop box after the NFC tag being scanned.



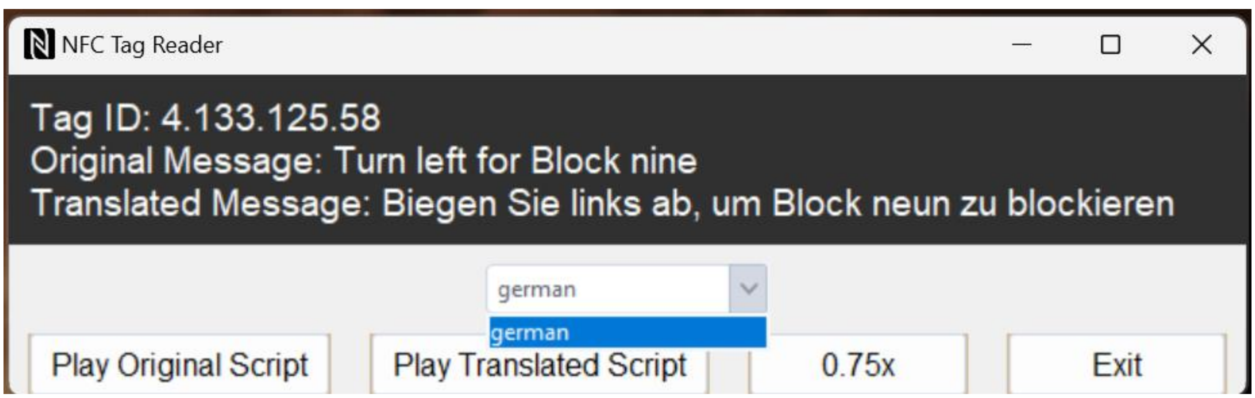
2. The audio output

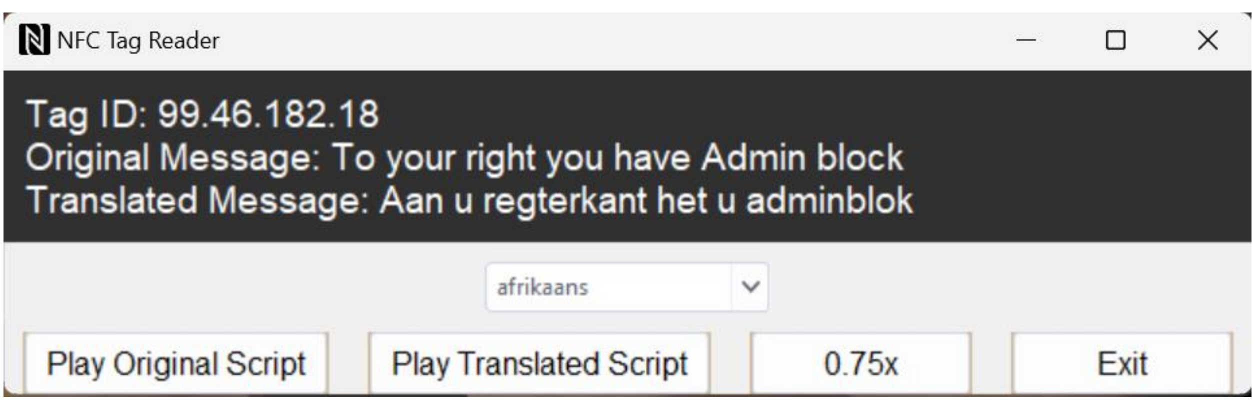


3. The screenshot of preferred language option settings



4.The audio output in the preferred language.





REFERENCES

1. Interfacing PN532 NFC RFID Module with Arduino :

https://how2electronics.com/interfacing-pn532-nfc-rfid-module-with-arduino/