



Palestine Polytechnic University
College of Information Technology and Computer
Engineering

Intelligent Humanoid Librarian

Rami Ikhmayyes
Moayad Najjar

Supervised by: Mr. Wael Al Takrouri

Hebron - Palestine
December - 2024

Abstract

Librarians have always been a crucial element of libraries and have provided insight and information to patrons relentlessly across the ages. Still, as time advances, they could use some help to ease their workload, and this is where the Intelligent Humanoid Librarian system comes in. By implementing technological advancements into the library setting, the system aims to introduce a humanoid robot named NAO6, which has been specifically integrated with a large language model's capabilities from OpenAi's product ChatGPT, to function as a Smart Librarian with which library patrons can interact and have conversation with. By using NAO6, it seeks to provide information about the library's collection, all while engaging in clear and well-structured dialogue. In addition, a microprocessor will be used to create a self-checkout station that uses facial recognition to identify users, Radio Identification (RFID) tags are used to make it simpler to borrow and return books. As a result, implementing this system tries to achieve higher user interaction within libraries, improve general access to information, make inquiries easier, as well as meet the various needs of the library patrons.

Keywords: Humanoid, Intelligent librarian, ChatGPT.

الخلاصة

الأمناء في المكتبات كانوا دائماً جزءاً أساسياً من تقديم المعلومات والإرشادات للرواد على مرّ العصور. ومع تقدم الزمن، قد يحتاج هؤلاء الأمناء إلى بعض المساعدة لتخفيف عبء العمل الذي يتقل كاهلهم. هنا يأتي دور نظام أمين مكتبة الذكي شبيه الإنسان. هذا النظام يقدم روباتاً بشرياً يُدعى NAO6، الذي تمّ دمجُه مع تقنيات الذكاء الاصطناعي من نموذج اللغة الكبير ChatGPT الخاص بشركة OpenAI يهدف NAO6 إلى العمل كأمين مكتبة ذكي، يتفاعل مع الرواد ويحيب على استفساراتهم حول مجموعة المكتبة. من خلال هذه التكنولوجيا، يمكن للرواد الحصول على معلومات دقيقة ومفيدة حول الكتب بطريقة سلسة ومنظمة. علاوةً على ذلك، سيتم استخدام معالج دقيق لإنشاء نظام لإعارة الكتب بشكل ذاتي تعتمد على التعرف على الوجه لتحديد المستخدمين، بالإضافة إلى استخدام تقنية التعرف بالتردد الراديوي (RFID) لتسهيل عملية إقراض وإعارة الكتب. من خلال تنفيذ هذا النظام، يمكن تحسين التفاعل بين المستخدمين والمكتبة، وتعزيز الوصول إلى المعلومات، وتسهيل الاستفسارات، وتلبية احتياجات الرواد المتنوعة بشكل أكثر فعالية.

الكلمات المفتاحية: أمين المكتبة الذكي، روبات بشري، نظام إعارة الكتب بشكل ذاتي، مكتبة، الذكاء الاصطناعي.

Contents

Abstract	II
List of Acronyms	VIII
Chapter 1	1
Introduction	1
1.1 Overview	1
1.2 Problem Statement	1
1.3 Expected Results	1
1.4 System Description	2
1.5 System Requirements	2
1.5.1 Functional Requirements	2
1.5.2 Non-Functional Requirements	2
1.6 Objectives	3
1.7 System Constraints and Limitations	3
1.8 Outline	4
Chapter 2: Theoretical Background	5
2.1 Overview	5
2.2 Theories	5
Algorithms used:	6
Convolutional Neural Networks	6
2.3 Literature Review	10
1-Language Models for Human-Robot Interaction	10
2- Smart Library Automation Using Face Recognition	10
3- Defining some criteria for the success of automated library systems	10
2.4 Hardware Components:	12
2.4.1 Tracer WEB008 HD Webcam	14
2.4.2 Raspberry Pi 3B+	14
2.4.3 NAO6 Humanoid Robot	15
2.4.4 RFID Tags	15
2.4.5 VM-5GA UHF RFID Reader	16
2.4.6 RC552 RFID Reader	16
2.4.7 I2C LCD	17
2.4.8 Buzzer	17
2.5 Software Components	18
2.5.1 ChatGPT	19
2.5.2 SQLite	19
2.5.3 Choregraphe	19
2.5.4 LangChain	19
2.5.5 Python	19
2.6 Alternative Design Options	20
2.6.1 Check in/out	20
2.6.2 Single Board Computer	21
2.6.3 Conversation	22
Chapter 3: Design	23

3.1 Overview	23
3.2 Detailed Design and Hardware Setup	23
3.2.1 Block Diagram	23
3.2.2 Activity Diagram	24
3.2.3 Sequence Diagram	25
3.2.4 Schematic Diagram	26
3.3 Algorithms and methodologies	27
3.3.1 Pseudo Code	27
Chapter 4:Implementation	29
4.1 Implementation issue:	29
4.2 Hardware and Software Implementation:	29
4.2.1 NAO6 Robot	29
4.2.2 Automated Library System	32
4.3 Summary	42
Chapter 5: Validation & Results	43
5.1 Choreography Test with NAO6:	43
5.2 Automated Library System:	44
5.3 Summary:	48
Chapter 6: Conclusion & Future work	49
6.1 Conclusion	49
6.2 Future Work	49

List of Figures

2.1	Convolutional Neural Network (CNN) architecture	6
2.2	Feature map	7
2.3	Max pooling	8
2.4	Fully connected layer	9
2.5	WEB008 HD Webcam	14
2.6	Raspberry Pi 3b+	14
2.7	NAO6 Humanoid Robot	15
2.8	RFID Tags	15
2.9	VM-5GA UHF RFID Reader	16
2.10	RC522 RFID Reader	16
2.11	I2C LCD	17
2.12	Buzzer	17
3.1	Block Diagram	24
3.2	Activity Diagram	24
3.3	Sequence Diagram for NAO6 and User Conversation	25
3.4	Sequence Diagram for Automation Library System	25
3.5	Schematic Diagram	26
4.1	NAO6 Robot	30
4.2	Description of Operation	31
4.3	RFID RC522 Reader and LCD Display Connection	32
4.4	Automated Library System	33
4.5	RFID Read Data Code	34
4.6	LCD Display Code	34
4.7	VM-5GA UHF RFID Reader Snippet Code.	38
4.8	VM-5GA UHF RFID Reader Snippet Code.	38
4.9	VM-5GA UHF RFID Reader Snippet Code.	39
4.10	Face Recognition Snippet Code.	40
4.11	Face Recognition Snippet Code.	40
4.12	Face Recognition Snippet Code.	41
4.13	Face Recognition Snippet Code.	41
4.14	Identify faces	42
5.1	Human and Nao6 Conversation	43
5.2	Emulation of NAO Speech using Choregraphe	43
5.2	Raspberry Pi Terminal	44
5.3	LCD Display	44
5.4	Reading Tag ID using RC522 Module and Displaying it on LCD.	44
5.5	Reading Tag ID using VM5GA Module and Displaying it on LCD.	45
5.6	Testing Face Detection.	45
5.7	Images with ID of each user.	45
5.8	User information in database.	46
5.9	User identification based on user ID.	46
5.10	Message for Unregistered user.	46
5.10	Database of borrowed books.	47
5.11	Reading Tag ID of existing book	47
5.12	Reading Tag ID of existing book	47
5.13	Reading Tag ID of not existing book	47

List of Tables

2.1	Comparison between Previous Projects and Our Project	11
2.2	List of Hardware Components And Their Options	13
2.3	List of Software Components And Their Options	18
2.4	Comparison of OCR and RFID	20
2.5	Comparison of Raspberry Pi and BeagleBone	21
2.6	Comparison of NAO6 and Microphone-Speaker System	22
4.1	Get Hardware Version Command	35
4.2	Single polling Command	35
4.3	No Tags Response Command	36
4.4	Multi Polling Command	36
4.5	read and writing only for special tags according to the Select parameter	37

List of Acronyms

AI	Artificial Intelligence
ChatGPT	Chat Generative Pre-trained Transformer
CNN	Convolutional Neural Network
CSI	Camera Serial Interface
CM	Centimeter
DB	Decibel
HOG	Histogram Of Oriented Gradients
I2C	Inter-Integrated Circuit
LBPH	Local Binary Pattern Histogram
LCD	Liquid Crystal Display
LLM	Large Language Models
LUX	Luminous Flux Per Unit
MISO	Master In Slave Out
RFID	Radio Frequency Identification
SDA	Serial Data Line
SVM	Support Vector Machine
UART	Universal Asynchronous Receiver/Transmitter Protocol
UHF RFID	Ultra High Frequency Radio Identification
USB	Universal Serial Bus

Chapter 1

Introduction

1.1 Overview

Libraries are a significant resource for providing knowledge to every human being, therefore, they serve as places where individuals engage with each other, share thoughts, and demonstrate their skills and interests. Despite current technological trends, libraries remain necessary repositories of information in this digital world we live in. This project seeks to bridge the gap between traditional libraries and the digital era in order to enhance user experience. It aims at expanding upon the role of librarians by introducing an interactive system that make information enquiries easier, recommends books, and assists users in determining whether or not the book they want is available for borrowing. The system enables users to engage with libraries in a new way. This initiative takes a step toward enhancing user engagement and experience in today's libraries as they adapt to changing information environments.

1.2 Problem Statement

libraries remain essential in our lives, both for academic and everyday use, but nowadays, with people having increasingly easier access to the information they desire, right at their fingertips, some could argue that traditional libraries would fall short, hence, our system seeks to improve on that part, via utilizing AI-powered conversation, alongside the advanced humanoid robot NAO6, by doing so, it strives to provide accurate responses, increase both interest and engagement and add a modern aspect to modern libraries.

1.3 Expected Results

- Integrate ChatGPT's AI capabilities with the NAO6 humanoid robot to establish an intelligent librarian system for libraries.
- Provide real-time voice assistance, suggest book recommendations based on user input, and offer information on book details such as title, author, year of publication, and availability.
- The system aims to automate library operations and minimize human manual effort.

1.4 System Description

This project visualizes a modern library system where the NAO6 humanoid robot would act as an interactive librarian, engaging in conversations with library patrons. which is made possible by ChatGPT for advanced language processing, NAO6 provides dynamic responses and assistance. The user registration process is streamlined using a Raspberry Pi with facial recognition capabilities. This ensures an easier registration process, without the need for an identification card to be at hand, as each user's information is saved in a database. RFID technology is used to ensure an easier process of registering books to borrow and to identify cases where books are borrowed without proper registration using the checkout system, or cases of theft, additionally, SQLite acts as the underlying database technology that provides efficient handling of data.

1.5 System Requirements

The following is a list of both functional and non functional requirements of the system:

1.5.1 Functional Requirements

- NAO6 should engage in natural language conversations with library patrons.
- ChatGPT integration should provide context-aware and dynamic responses.
- The Raspberry Pi system with the use of the Convolutional Neural Network algorithm must accurately capture and process facial data, recognize registered users, and associate them with their profiles.
- The Raspberry Pi system should detect all outgoing books from the library, and distinguish whether they were properly registered, and if not, sound an audible alarm in the case of not.
- The NAO6 should provide recommendations and tailored assistance for users.

1.5.2 Non-Functional Requirements

- **Reliability:** The system should consistently deliver accurate and reliable performance, minimizing downtime and errors over time.
- **Accuracy:** The system's recognition algorithm and provided information about the library collection are expected to attain high accuracy in identifying users and suggesting relevant materials.
- **Usability:** The system interface is designed to be intuitive and user-friendly, ensuring seamless interaction for all users.
- **Adaptability:** The system is designed to adapt to evolving technology standards and library requirements, ensuring long-term usability.

1.6 Objectives

Book Self-Checkout System Build a book self-checkout system, which performs the following:

- Use RFID tags to check books in and out quickly.
- Utilize facial recognition instead of physical student ID cards.
- Check books leaving the library via UHF RFID tags to ensure they are properly checked in, if not, an alert sound will be generated.
- Record books checked in under users in the database, listing both the date of borrowing and the due date of return.
- Use an output device to provide instructions and system status information.

NAO6 Enable NAO6 to perform the following:

- Assist by voice to library patrons.
- Provide information about books in the library, borrow status, and relevant information about each book.
- Offer information about the university and its history.

1.7 System Constraints and Limitations

For our system to function as expected, the following constraints must be fulfilled:

- **Entry gate width:** If the entry gate exceeds 1 - 1.5 meters the Ultra-High-Frequency Radio Frequency Identification(UHF RFID) reader's effective range, may not be able to read tags efficiently, which may lead to recognition failures.
- **Light Level:** 300 to 500 lux brightness is recommended, lighting conditions within a closed room may vary, and lack of proper indoor lighting could affect the face recognition results, affecting the quality and visibility of facial features.
- **Distance between the camera and the opposing face:** The opposing face cannot be too far nor too near the camera, around 35-40 cm is recommended.
- **The presence of background noise or speech** could diminish the quality of NAO6's voice recognition and result in inaccurate prompts, a level of 30 - 40 dB is recommended for optimal.

1.8 Outline

Chapter 2 introduces a literature review including available image recognition-related projects and their results. It also briefly goes over the theoretical background of the project, as well as hardware and software components. Chapter 3 discusses the conceptual design of the system, block diagrams, pseudo-code, and detailed hardware connections, Chapter 4 demonstrates the hardware and software implementation of the system. Chapter 5 shows the validation and testing process of the project and the final results that came out. Chapter 6 summarizes the project with some suggestions for future work.

Chapter 2: Theoretical Background

2.1 Overview

This chapter mostly gives a brief overview of the technologies that make up our project. We will first go over the theoretical context and define some common phrases. To demonstrate how earlier research accomplished these goals, we will next provide a review of the literature. In the third section, we'll review the primary hardware components we require and their purposes. Lastly, we will discuss the history of software, primarily the programming languages and the algorithms that we will employ.

2.2 Theories

2.2.1 Large Language Models (LLMs):

Large Language Models (LLMs) are advanced artificial intelligence (AI) systems that can comprehend and produce human-like text, made possible by transformer architectures, which were developed by Google researchers in 2017. These models have evolved from older neural networks, like recurrent neural networks (RNNs), to models that can handle more complex language tasks.

A great advancement was with the usage of Transformers, which uses self-attention mechanisms to improve textual context awareness. These models are trained on massive datasets, often consisting of gigabytes of text, and include billions of parameters. Due to this training, they can now perform jobs like question answering, language translation, and even reasoning without the need for previous examples.

LLMs have shown the ability to generalize to multiple tasks and exhibit abilities such as reasoning and decision-making, even in zero-shot settings, where they respond accurately without prior task-specific fine-tuning [1]. The scale and capability of LLMs have made them crucial in pushing the boundaries of what AI can do in natural language processing [1].

2.2.2 Radio Frequency Identification: The discovery and utilization of continuous waveform in the realm of radio-frequency technology, ranging from low-frequency (LF) to high-frequency (HF) and ultra-high-frequency (UHF), have had a profound impact on various fields. This has led to the development and widespread implementation of Radio Frequency Identification (RFID) across a broad spectrum of frequencies. RFID is a technology that uses radio waves within the LF (30 kHz to 300 kHz), HF (3 MHz to 30 MHz), or UHF (300 MHz to 3 GHz) ranges to identify and track objects, people, or different sorts of assets by attaching RFID tags to them [2].

2.2.3 **Facial Recognition:** Using a technique called bio-metrics, facial recognition technology can identify a person’s face. It typically maps a face’s characteristics from an image or video, and then compares the results to a database of recognized faces to locate matches.[3]

2.2.4 **Histogram:** In modern digital image processing, image similarity and recognition are rapidly evolving technologies. One approach involves recognizing a specific person’s face by assessing the similarity between their face images. The joint histogram technique is a key method, relying on local pixel features to construct a multidimensional histogram.[4]

Algorithms used:

Convolutional Neural Networks

Is a subclass of deep neural networks called convolutional neural networks (CNNs) is mostly employed in image analysis. Their ability to acquire hierarchical facial features has made them popular in face recognition applications. photos created directly from pixel data. Since CNNs automatically extract the features needed for recognition from the input images, they are far more useful in a wide range of applications than traditional methods like LBPH. In Figure 2.1 illustrates CNNs extract essential features through layers of convolutions and pooling, which are then used for classification using fully connected layers and a softmax function.

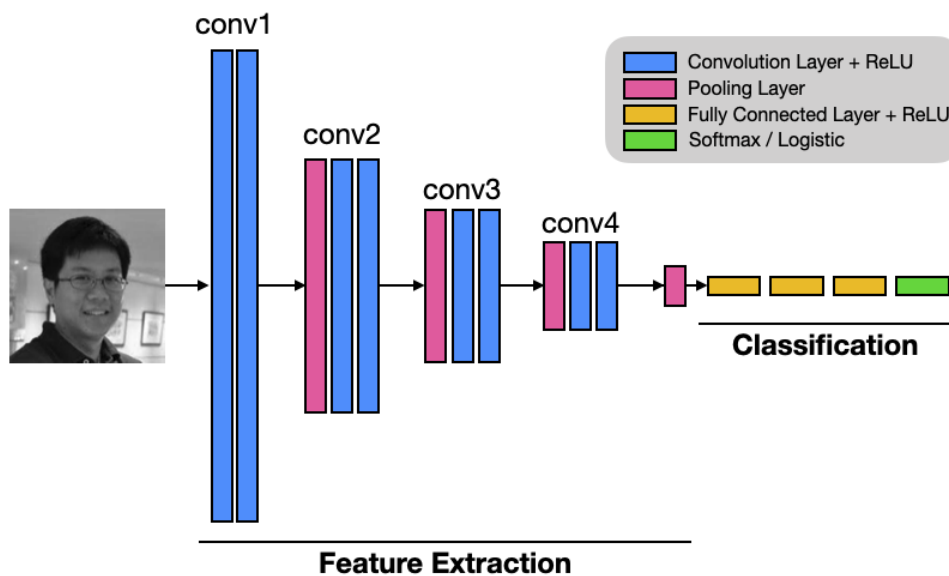


Figure 2.1: Convolutional Neural Network (CNN) architecture [6].

1. Steps to Compute CNN-based Face Recognition:

Image Preprocessing: The photos should be preprocessed to guarantee consistency and improve characteristics important for face recognition applications.

CNN Model Training: Utilizing a sizable dataset of annotated face photos, train a CNN model. The CNN gains the ability to extract discriminative elements from face photos during training that are helpful for identifying between many people.

Feature Extraction: Use the trained CNN model to extract features from the input facial images. The CNN typically consists of multiple layers, including convolutional layers and pooling layers, which collectively learn to represent facial features at different levels of abstraction and below the explanation for each layer:

1. **Convolutional layers:** set of matrices that get multiplied by the previous layer output in a process called the convolution to detect some features these features could be basic features (e.g. edge, color grade or pattern) or complex ones (e.g. shape, nose, or a mouth) so, those matrices are called filters or kernels.

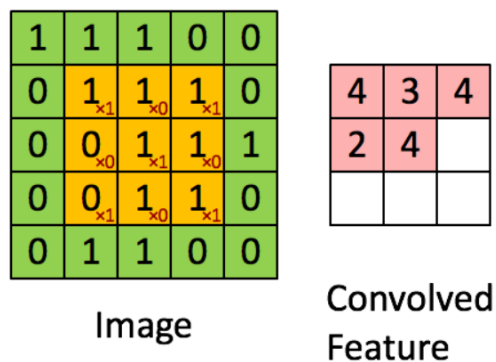


Figure 2.2: Feature map

2. **Pooling layers:** in pooling layer (Max pooling, avg pooling ...), the most common these days is Max pooling maxpooling is a method of pooling in networks where translation variance is provided and parameter number increased dimensionality reduced effectively. It does this by sliding a pre-determined filter/window across the input, giving the highest value within that window as a result.

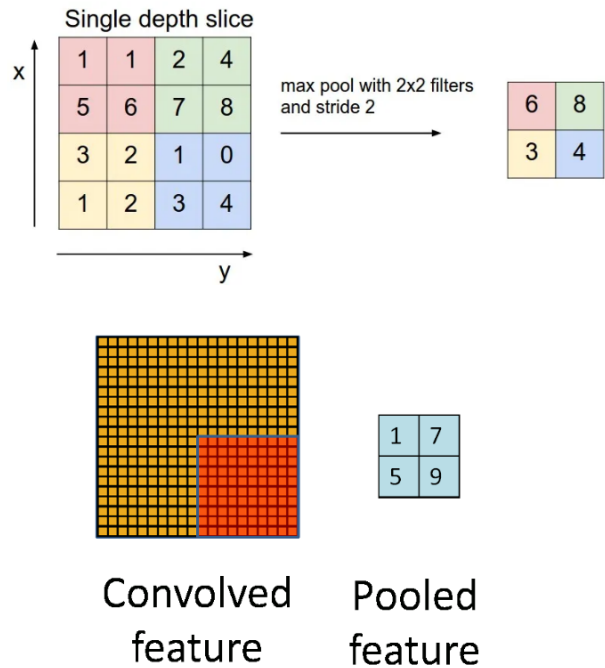


Figure 2.3: Max pooling

Classification: Apply any form of classification algorithm, such as softmax regression or SVMs, over the outputs of fully connected layers to classify the input images into different classes corresponding to different identities. Fully connected layers are typically part of the classification stage, where they help in mapping features extracted by earlier layers (like convolutional and pooling layers) to class labels.

- **Fully connected layers:** In convolutional neural networks (CNNs), the fully connected layer also called the dense layer is essential in the network architecture. It connects each neuron of the previous layer with all neurons of the fully connected layer. The main purpose of this type of layer is to learn the global patterns from the data and, consequently, build relationships between them. Most authors position this one last, after the pooling and convolution stages. In the Figure 2.4 you can see how the fully connected layers are used to classify the features extracted by the convolutional layers into different categories, such as identifying whether a section of an image corresponds to a face or non-face, and further distinguishing facial features like eyes, nose, and mouth.

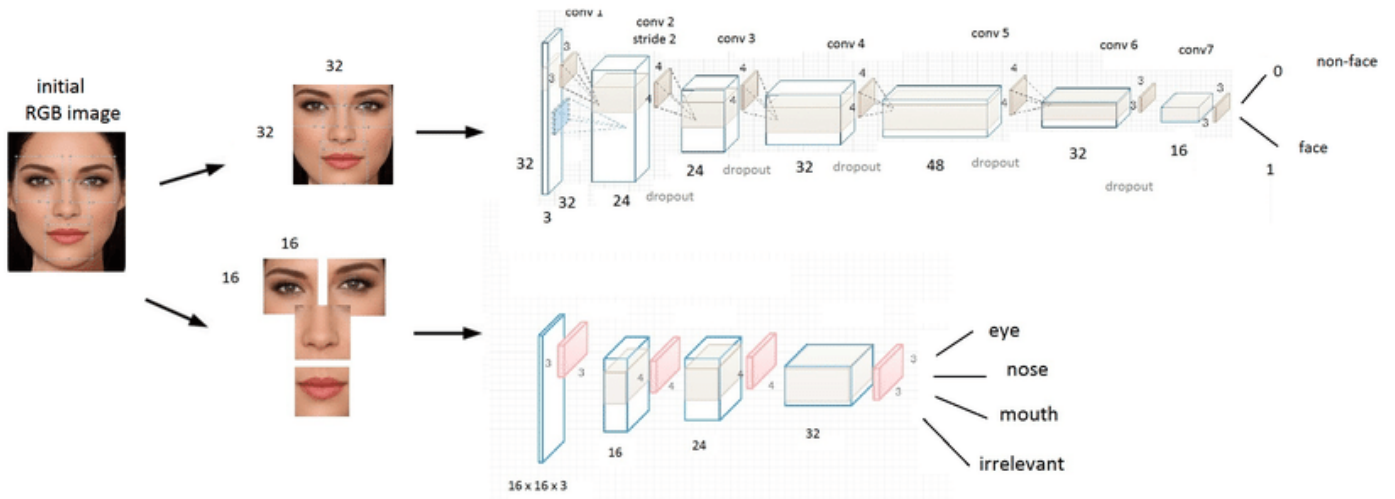


Figure 2.4: Fully connected layer

2. Advantages of CNN-based Face Recognition:

- **Automatic Feature Learning:** CNNs automatically learn hierarchical features directly from the input images, eliminating the need for manual feature engineering.
- **Robustness to Variations:** CNNs are inherently robust to variations in facial appearance due to changes in illumination, pose, facial expression, and occlusion.
- **High Accuracy:** In facial recognition benchmarks, CNNs have shown state-of-the-art performance, with high accuracy rates even in demanding real-world scenarios.
- **Scalability:** It is appropriate for a range of applications, such as biometric authentication and access control, because they can be scaled to accommodate big datasets and real-time processing demands.

3. Applications of CNN-based Face Recognition::

- **Biometric Authentication:** CNN-based face recognition systems are widely used for biometric authentication in security systems, mobile devices, and online platforms.
- **Surveillance:** CNN-based face recognition is employed in surveillance systems for identifying individuals in crowded environments and tracking their movements.
- **Access Control:** CNN-based face recognition systems are utilized in access control systems for granting or denying access to secured areas based on recognized identities.
- **Law Enforcement:** CNN-based face recognition is used by law enforce-

ment agencies for identifying suspects in criminal investigations and monitoring public spaces for security purposes.

2.3 Literature Review

Studies related to this work for the visually impaired have proposed various approaches.

1-Language Models for Human-Robot Interaction

Billing et al., in "Language Models for Human-Robot Interaction," presented an innovative system that integrates large-scale language models like GPT-3 into social robots such as Nao and Pepper. This system transforms traditional text-based interactions into seamless verbal dialogues, enabling robots to engage in more natural and flexible conversations with users. By utilizing cloud-based services for speech recognition and text processing, the system efficiently manages spoken inputs and generates appropriate responses, enhancing the overall human-robot interaction experience. The implementation of this technology not only highlights its potential in the field of robotics but also offers valuable insights into the future of automated dialogue systems. This work is particularly relevant and insightful for our project, as it closely aligns with our objectives of advancing interactive robotic systems.[7]

2- Smart Library Automation Using Face Recognition

Yalagi, in "Smart Library Automation Using Face Recognition," presented a cutting-edge system designed to automate library access and various services through the use of face recognition technology. This system efficiently captures users' facial images to grant entry to the library and manage book checkouts, thereby eliminating the need for traditional physical library cards. By streamlining these processes, the system significantly reduces the Probability of errors and greatly enhances the overall user experience. The successful implementation of this technology demonstrates its increasing importance and utility in the field of automation. This work has proven to be particularly relevant and insightful for our project, as it aligns closely with the objectives we are pursuing.[8]

3- Defining some criteria for the success of automated library systems

According to Hossein's (1999) article titled "Defining some criteria for the success of automated library systems," all users typically look for two inter-related factors when purchasing or developing a new automated library system: effectiveness and success. The identified 26 criteria for the success of automated library systems are important because they are based on available resources, literature, and the opinions of experts in automated library systems. A survey conducted in 1993 looked at how Australian university librarians and systems managers felt about these standards. It was discovered that the sur-

vey sample accepted 23 of the identified criteria while rejecting the remaining three. To obtain greater generalizability, these standards would be put to the test among other cohorts of library automation specialists. [5]

2.3.1 Related Project

In Table 2.1 shows a comparison between previous projects and our project.

Feature	Previous Projects	Our Project	Relevant Literature
Searching Process	Using application for searching about a book and only data that exist in the database	Utilizing NAO6 voice recognition to search for data in our database and also the ChatGPT database	Billing et al., "Language Models for Human-Robot Interaction"
Identifying Users	Using RFID key	Using facial recognition technique	Yalagi, "Smart Library Automation Using Face Recognition"
Security	Mentioned with no specific description	Using VM-5GA UHF RFID and buzzer for anti-theft system	Not explicitly mentioned but relevant to system security
Accessing Database	Using application for access based on RFID key and RFID tags	Using NAO6 voice recognition and RFID tags	Billing et al., "Language Models for Human-Robot Interaction"

Table 2.1: Comparison between Previous Projects and Our Project

2.4 Hardware Components:

This section will cover a wide range of hardware components that are essential for creating the system we need. Each component plays a unique role in enhancing functionality and efficiency. First, let's explore the components we use and the options available for each:

Hardware	Selected Hardware	Option1	Option2	Specification
WebCam	Tracer Web008	Logitech Webcam C930e	Raspberry Pi Camera	Logitech Webcam C930e offers higher resolution and better image quality compared to Tracer Web008. Raspberry Pi Camera is more integrated with Raspberry Pi systems but has lower resolution.
Process Node	Raspberry Pi 3B+	Raspberry Pi 4	Nano Pi	Raspberry Pi 4 offers better performance with more memory and processing power compared to Raspberry Pi 3B+. Nano Pi provides similar or slightly different features but may have less community support.
Robot	NAO6	Pepper by SoftBank Robotics	No other option	Pepper is more advanced and versatile compared to NAO6, with better interaction capabilities. No other option for comparison.
UHF RFID Reader	VM-5GA UHF RFID	SRK-SYMD6L UHF RFID	YANZEO SR791 UHF RFID Reader	SRK-SYMD6L offers similar or improved range and accuracy compared to VM-5GA. YANZEO SR791 has additional features but may be more expensive.

NFC RFID Reader	RC522 RFID Reader	PN532 NFC /RFID Module	ST25R39-11B NFC Reader	PN532 offers more versatility with support for both NFC and RFID compared to RC522, which is primarily for RFID. ST25R3911B offers advanced NFC capabilities with better range and features but might be more complex and costly.
LCD Display	I2C LCD Display	TFT LCD Display	OLED Display	TFT LCD offers better color and resolution compared to I2C LCD. OLED Display provides superior contrast and color depth but can be more expensive and have limited lifespan compared to TFT LCD and I2C LCD.
Alert	Buzzer	Speaker	Active Alarm Buzzer	An Active Alarm Buzzer provides louder, multi-tone alerts, ideal for security systems. A Speaker is versatile for various sounds but needs more power. A Standard Buzzer is simple and cost-effective, with single-tone output.

Table 2.2: List of Hardware Components And Their Options

2.4.1 Tracer WEB008 HD Webcam

The Tracer WEB008 HD Webcam is intended to deliver sharp, high-definition images for a range of uses, including streaming, taking pictures, and making video chats. It probably has features like autofocus, a built-in microphone, and support for widely used streaming and video conferencing services. It may also feature an adjustable mounting option or a wide-angle lens for positioning versatility.



Figure 2.5: Tracer WEB008 HD Webcam [10]

2.4.2 Raspberry Pi 3B+

The Raspberry Pi 3B+ is an inexpensive, versatile single-board computer, It was developed by the Raspberry Pi Foundation. It has a quad-core 1.4 GHz processor, 1 GB random access memory, and connectivity options including Gigabit Ethernet and dual-band Wi-Fi for its hardware capabilities, This product is popular among beginners or students for personal designing flexibility, it has various connection possibilities catering to different hobbies such as programming, media player, etc, sometimes it might be used for more than these purposes. [9]

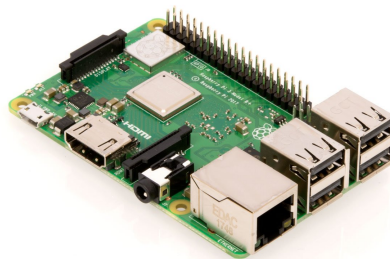


Figure 2.6: Raspberry Pi 3b+ [9]

2.4.3 NAO6 Humanoid Robot

The NAO6 humanoid robot, developed by SoftBank Robotics, stands approximately 58-61 cm tall and weighs around 4-5 kg. It specializes in human communication, featuring speech interaction, bipedal walking, dancing, and facial/object recognition capabilities. Equipped with omnidirectional microphones, tactile sensors, sonar, and 2D cameras, it ensures comprehensive perception. With Wi-Fi and Bluetooth, it's customizable for specific applications via a software development kit, finding use globally in research, education, and healthcare as an interactive humanoid robot.



Figure 2.7: NAO6 Humanoid Robot [11]

2.4.4 RFID Tags

Radio Frequency Identification tags, are passive tags (batteryless tags) that would be placed on the cover of each book, they hold "write once read many" memory, and will store information for each book, required for the anti-theft implementation.



Figure 2.8: RFID Tags [12]

2.4.5 VM-5GA UHF RFID Reader

will be used to read the RFID tags on each book that passes through the front gate of the library. The reader will scan these tags within a range of 1-2 meters and compare them to a database of lent-out books to help prevent theft. The reader operates at a frequency of 860-960 MHz, follows the EPC Gen 2 / ISO 18000-6C protocol, and operate with a working voltage of DC 3.5V-5V. It also has a writing range of 10-50 cm. .



Figure 2.9: VM-5GA UHF RFID Reader [13]

2.4.6 RC522 RFID Reader

The RC522 RFID reader is a popular RFID (Radio-Frequency Identification) module that operates at 13.56 MHz and has a range of 5 cm. It is commonly used for reading RFID cards and tags in various applications, such as access control systems, attendance systems, and electronic projects. and in our project will be used for check-in/out.



Figure 2.10 : RC522 RFID Reader [14]

2.4.7 I2C LCD

I2C communication allows an LCD I2C module to show data with fewer connections. It utilizes the obtained data and instructions from a microcontroller to set the text and symbols it will display on the screen.



Figure 2.11 : I2C LCD [15]

2.4.8 Buzzer

Is an electronic device that generates sound when an electric current passes through it. It typically operates at a voltage range of 3V to 5V DC. Buzzers are widely used in alarms, notifications, and signaling systems to alert users with audible sounds.



Figure 2.12 : Buzzer [16]

2.5 Software Components

This section will cover a wide range of software components that are important to different fields and each uses their own special talents to improve functionality and efficiency. First, let's explore the components we use and the options available for each:

Software	Selected Software	Option 1	Option 2	Specification / Comparison
AI	ChatGPT	Google Bard	Bing AI	ChatGPT offers extensive API support and customizability; Google Bard is tightly integrated with Google services; Bing AI has strong search integration and Microsoft product synergy.
Database	SQLite	PostgreSQL	MySQL/MariaDB	SQLite is lightweight and serverless, PostgreSQL offers advanced features and scalability, MySQL/MariaDB are widely used with strong community support and performance in web apps.
Robot Controlling Application	Choregraphe	ROS (Robot Operating System)	VPL (Visual Programming Language)	Choregraphe is user-friendly for NAO robots, ROS is open-source with extensive robot support, VPL is visual and accessible but less powerful for complex tasks.
Framework	LangChain	Rasa	Hugging Face Transformers	LangChain specializes in chaining and memory management, Rasa is focused on conversational AI with dialogue management, Hugging Face Transformers excel in model training and deployment.
Programming Language	Python	C++	Java	Python is highly readable and versatile, C++ offers high performance and control, Java is platform-independent with a strong ecosystem but it cant work with Nao6 robot.

Table 2.3: List of Software Components And Their Options

2.5.1 ChatGPT

A widely recognized Large Language Model that utilizes AI, was constructed using a deep learning model called Transformer architecture, it utilizes Reinforcement Learning, with human feedback, which allows it to produce human-like text output.

2.5.2 SQLite

a lightweight, open-source relational SQL database engine facilitating efficient storage, management, and retrieval of the library's book collection and user data, ensuring seamless integration with Python programming language for streamlined data manipulation and query execution.

2.5.3 Choregraphe

a software application that's used to program the movements and behaviors of humanoid robots such as NAO and Pepper. It provides a user-friendly interface for designing and controlling various actions, motions, and interactions of the robot, enabling users to create complex and dynamic behaviors.

2.5.4 LangChain

An open-source framework for software developers working with AI and its machine learning subset. combining large language models with other external components to develop LLM-powered applications, allowing the development of LLM-powered applications. This capability enables us to customize LLMs like ChatGPT by incorporating our data, and tailoring it to suit our specific requirements better.

2.5.5 Python

Selected as the programming language for NAO6 due to its user-friendly nature and extensive libraries. Python supports a variety of use cases such as face tracking and facial recognition, which are planned for implementation.

2.6 Alternative Design Options

In this section, we present an alternate strategy to achieve the intended goals using a different technique:

2.6.1 Check in/out

Because of the unique solutions in the data capture technology the process of selecting one is not easy some of these technologies are RFID and optical character recognition (OCR). The process of extracting the text from the image is called optical character recognition. On the other hand, RFID uses radio waves to detect physical items making it such an important tool, especially in libraries among others like personal identification systems, Table 2.4 shows the difference between these two technologies.

	OCR	RFID
Components	Scanners, mobile OCR apps, document cameras, and multifunction printers.	RFID readers, RFID tags, RFID antennas, and some smartphones with built-in RFID capabilities.
Technology	Converts images and scanned documents into editable text	Uses radio waves to identify and track objects
Data Capture	Visual interpretation of characters from images	Captures data through radio frequency signals with RFID tags
Application	Digitizing and cataloging printed materials	Library automation for tasks like self-checkout, inventory, and security
Speed	Processing time influenced by document complexity	Fast and efficient data capture, suitable for high-speed operations
Cost	No Cost	Implementation costs include RFID tags, readers, and infrastructure
Environment	Requires clear and readable text for visual recognition	Operates wirelessly, non-line-of-sight, with varying read ranges

Table 2.4 : Comparison of OCR and RFID

One possible alternative solution would be to use two UHF RFID readers. One reader would be placed at the gate to check the status of books leaving the library, while the other would be positioned at the checkout station, which would be surrounded by a confined space to limit the reading radius. This setup would ensure that the reader only scans books placed vertically on top of it, allowing for the scanning of multiple books in a single action and eliminating the need for two different types of tags on each book. However, this was not possible due to the limited availability of parts locally.

2.6.2 Single Board Computer

Among the single-board computers existing at the moment, Raspberry Pi and Beaglebone are known best. They are suited for different types of projects, so they are having their unique strengths. Table 2.5 gives a short feature comparison between Raspberry Pi, known for its flexibility and great computational power, and Beaglebone, known for its efficiency and very good support for embedded systems.



	Raspberry Pi  [9]	BeagleBone  [19]
Technology	Single-board computer	Single-board computer
Use Cases	Broad range including IoT, home automation, and multimedia applications	Embedded systems, robotics, and industrial automation
Processing Power	Multicore ARM Cortex processors, higher performance	ARM Cortex-A8 processor, 1GHz, moderate performance
Programming	Supports multiple languages: Python, C, Java, etc.	Primarily C/C++ with support for other languages
Connectivity	Built-in Wi-Fi, Bluetooth, GPIO, USB, HDMI	GPIO, USB, Ethernet; limited built-in wireless connectivity
Flexibility	Full Linux OS support, versatile for general computing	Real-time OS options, specialized for specific tasks
Cost	\$75 to \$150 (varies by model)	\$50 to \$100 (varies by model)

Table 2.5: Comparison of Raspberry Pi and BeagleBone
[17][18][19][20]

2.6.3 Conversation

In Table 2.6 shows the NAO6 Humanoid Robot and a common Microphone-Speaker system comparison. The Robot NAO6 is known for its advanced capabilities in robotics against the very simple versatility of an average audio communication setup. Their functionalities, applications, and potential synergies are explored to offer a comprehensive understanding of the unique qualities of each technology.

	NAO6 Robot	Microphone and Speaker System
Interaction	Engages in interactive conversations, responding to voice commands and questions	Primarily designed for audio communication; lacks interactive conversational abilities
Language Processing	Utilizes natural language processing for understanding and responding in a conversational manner	Basic voice recognition, limited language understanding capabilities
Social Engagement	Provides a social and engaging conversation experience with gestures, expressions, and contextual responses	Focuses on delivering clear audio output and capturing input; lacks social engagement features
Learning	Has the ability to learn and adapt to user preferences and conversation patterns	Limited learning capabilities; relies on external systems for updates and improvements
Context Awareness	Understands context during conversations, providing more meaningful and context-aware responses	Lacks inherent context awareness; responses may not be contextually driven
Multi-Modal Interaction	Supports multi-modal interaction with speech, gestures, expressions, and visual cues	Primarily focuses on audio-based communication without additional interactive modalities

Table 2.6 : Comparison of NAO6 and Microphone-Speaker System

Chapter 3: Design

3.1 Overview

This chapter describes the system's general architecture, how its parts work together, and how they differ from previously implemented systems. It also displays the block, schematic, and state diagrams for the system. In addition, we'll discuss the fundamental algorithms we'll implement.

3.2 Detailed Design and Hardware Setup

This provides a broad outline of how the primary components are arranged and interact with one another in this design:

- The system will be stationary, NAO6 robot and raspberry pi would be designated at the front desk of the library, and once of the RFID reader would be installed at the front entrance gate and the second will be near to Raspberry Pi.
- The NAO6 robot will establish a connection with ChatGPT, and the ChatGPT data will be customized to suit the project requirements.
- The LCD will display relevant messages based on user and system interaction.
- The webcam will capture and recognize the user's face, and the LCD will display the user's name upon recognition.
- The RC522 reader utilizes short-range RFID tags operating at 13.56 MHz.
- The VM5GA UHF reader uses high-frequency RFID tags operating between 840-960 MHz.

Each book will have two tags: one for scanning the book at the checkout station and another for anti-theft measures at the library exit. This setup was chosen to ensure proper tracking of books with the corresponding user.

3.2.1 Block Diagram

The main components of the system are illustrated in Figure 3.1's block diagram. The Raspberry Pi serves as the hub through which other system components connect. Its components include a camera that sends images for facial recognition and display indications to indicator components, two RFID components with different tag work with for data capture, and a SQLite database that serves as the main database. Additionally, we have Nao6 that is connected to intermediate hardware(laptop) to create librarian Nao6.

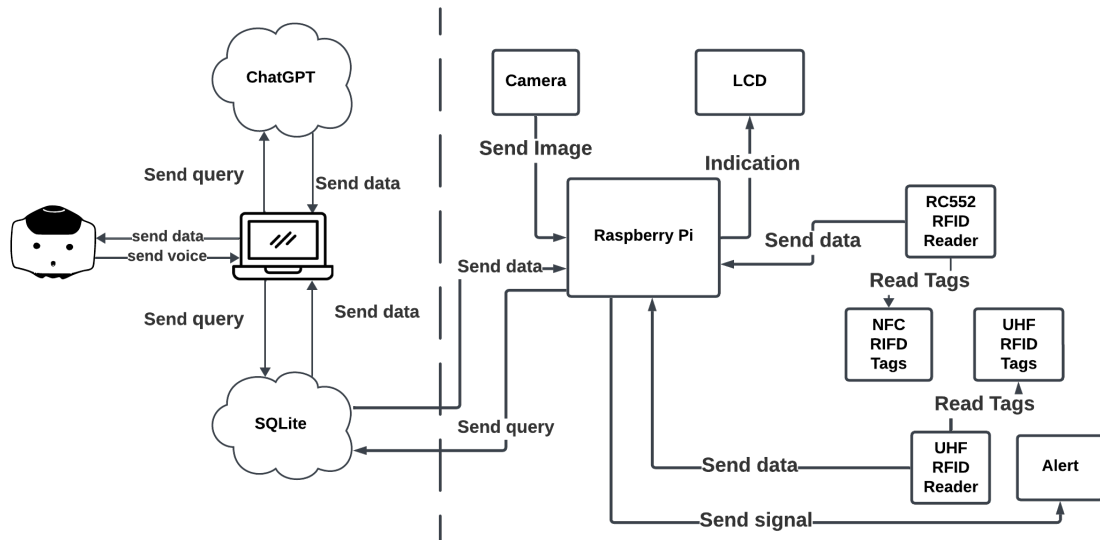


Figure 3.1: Block Diagram

3.2.2 Activity Diagram

in Figure 3.2 shows the user interaction with the system, starting with the Raspberry Pi, serving as the central communication hub. The interface seamlessly enables users to engage with various components such as the camera for facial recognition, RFID for book-related interactions, and the NAO6 humanoid robot for natural language conversations. users can initiate commands, input queries, and receive feedback.

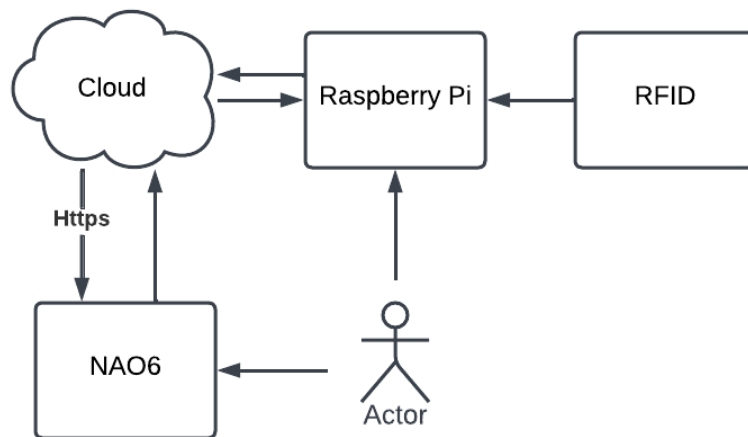


Figure 3.2: Activity Diagram

3.2.3 Sequence Diagram

In this section will demonstrate the process of how user interact with system .In Sequence 1 the user speak with Nao6 asking for something than user voice transmitted to laptop for process and deliver the response, In Sequence 2 the user approach the camera to gain access to use system for check in/out also their anti-theft system for check the book status.

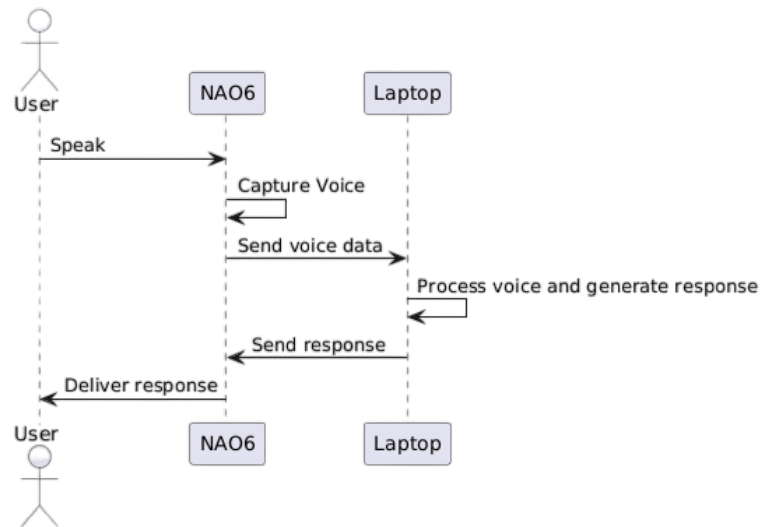


Figure 3.3: Sequence Diagram for NAO6 and User Conversation

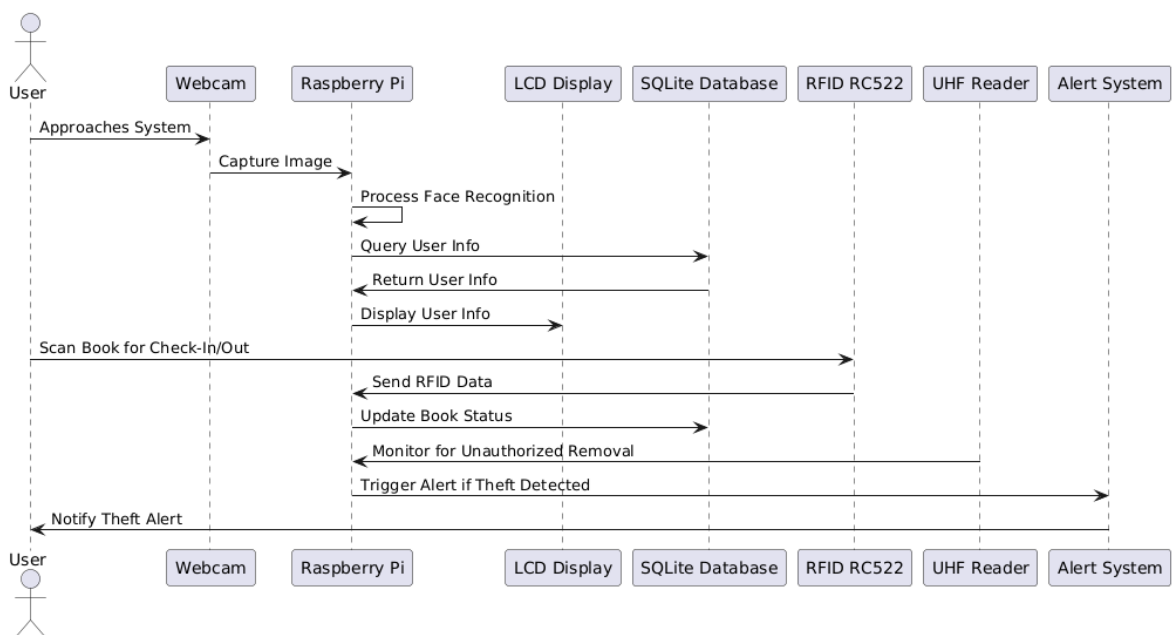


Figure 3.4: Sequence Diagram for Automation Library System

3.2.4 Schematic Diagram

The schematic diagram of the system is illustrated in Figure 3.3, outlining the components of the system and their interconnections. At the heart of the system is the Raspberry Pi 3B+, The VM5GA UHF and RC522 RFID Readers are connected to the Raspberry Pi, with one situated alongside the Raspberry Pi and the other strategically placed at the entrance gate with a buzzer next to it for anti-theft measures:

- The ground pin and 5V pin of Raspberry PI will be connected to the power supply
- The TRACER Webcam is connected to Raspberry Pi through a USB cable
- The RC522 RFID is connected to Raspberry Pi through 7 pins: SDA connects to GPIO 2, SCK connects to GPIO 11, MOSI connects to GPIO 10, MISO connects to GPIO 9, GND connects to once of the available ground pin, RST connects to one of available GPIO, 3.3v connects to once of available 3v3 power pin.
- The LCD is an indicator for indicating user identification, The ground pin and VCC pin of LCD is connected to Raspberry PI to one of the available pins, SCL connects to GPIO 3, and SDA connects to GPIO 2.
- The VM5GA UHF RFID is connected to Raspberry PI through UART TTL to USB port.
- The two pins of the Buzzer are connected to one of the available ground pins of Raspberry PI, and the other connects to one of the available GPIO.

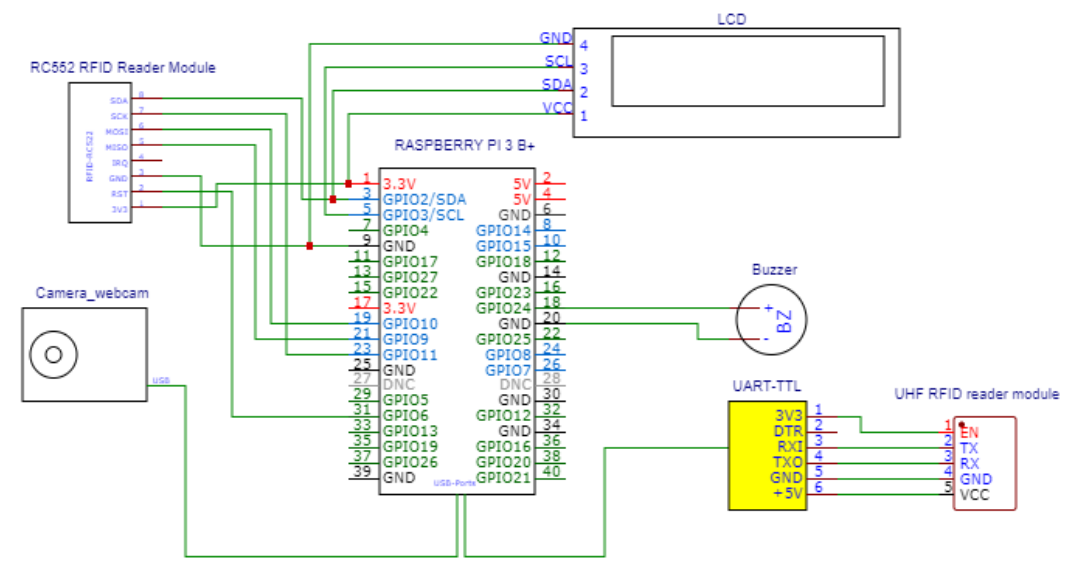


Figure 3.5: Schematic Diagram

3.3 Algorithms and methodologies

The system's algorithms for both project subsystems are described using flowcharts. Initially, the CNN facial recognition algorithm will be initialized and triggered at the Raspberry Pi. Then NAO6 Voice Recognition's speech recognition feature will be used to answer user voice inquiries, the following pseudo code describe them.

3.3.1 Pseudo Code

Voice Recognition plays a crucial role in the seamless interaction within the NAO6 in Algorithm 1, titled "NAO6 Voice Recognition and Query Processing," this process involves capturing and processing audio inputs from the user. The algorithm identifies recognized voice commands and performs actions accordingly, for data queries, the algorithm interacts with the project database, retrieving and displaying relevant information. Additionally, it interfaces with ChatGPT for additional details and executes predefined actions based on voice commands.

Facial Recognition is capturing and processing facial images when a user approaches. In Algorithm 2, titled "Facial Recognition System," the system operates continuously in Facial Recognition Mode. By utilizing the CNN model, the algorithm identifies the user, displays relevant information, and updates the database upon successful recognition.

Algorithm 1 NAO6 Voice Recognition and Query Processing

```
1: while NAO6 Voice Recognition Mode is on do
2:   Activate voice recognition module
3:   Capture and process audio input
4:   Identify recognized voice commands
5:   if Voice command matches "Query for data" then
6:     if Voice command matches predefined action then
7:       Execute corresponding action
8:     else if Send query to project database for relevant data then
9:       Retrieve and display data from the project database
10:    else if Send query to ChatGPT for additional information then
11:      Retrieve and display ChatGPT response
12:    else
13:      Display unrecognized command message
14:    end if
15:  end if
16: end while
```

Algorithm 2 Facial Recognition System

```
1: while True do
2:   while StartButton is not Clicked do
3:     System is Idle
4:     Display Message: "Push Button to Use System" on Screen
5:   end while
6:   if StartButton is Clicked then
7:     Activate Facial Recognition Mode
8:     if User Approaches then
9:       Initialize Attempt Counter:  $attempts \leftarrow 0$ 
10:      while  $attempts < 3$  do
11:        Capture Facial Image using Web Camera
12:        Preprocess the Captured Image
13:        Identify the User using the Trained CNN Model
14:        if User is Recognized then
15:          Display User Name on Screen
16:          Break from Recognition Loop
17:        else
18:          Display Message: "Unknown User"
19:          Wait for 3 Seconds
20:          Increment Attempt Counter:  $attempts \leftarrow attempts + 1$ 
21:        end if
22:      end while
23:      if  $attempts = 3$  then
24:        Display Message: "User Not Recognized."
25:        Return to the Initial State
26:      end if
27:    else
28:      Display Message: "No Face Detected"
29:    end if
30:  end if
31: end while
```

Chapter 4:Implementation

In this chapter, we will discuss the implementation section of the project, with a closer look into the system components and how the performance tests were performed.

4.1 Implementation issue:

- **Version Compatibility:** NAO6 Choreographe and Naoqi currently use Python 2.7, which did not allow for the use of many desired features that were introduced later on in newer versions, for that reason, an intermediary laptop was required.
- **Limited Computing Resources:** NAO6 has many features, but for the most part are only elementary and are intended for basic uses, for example, The resolution of Cameras is not fitting for accurate facial recognition, also, raspberry pi is capable of running the System software, but not including Choreographe and Langchain.
- **Information Availability:** Some hardware, such as the UHF RFID tag reader, has very limited documentation, and information in general regarding connection schematic, and had to figure out by try and error, also, High-Frequency Hardware is very limited in the region, both for the reader and tags, and being able to find and use them proved to be a challenge, on NAO6 part, resources on projects sometimes proved to be outdated, or not fulfilling our requirements to benefit from.

4.2 Hardware and Software Implementation:

In this section, we will discuss the implementation of both the hardware and software components of the system. The system is divided into two subsystems: the NAO6 Robot and the Automated Library System.

4.2.1 NAO6 Robot

4.2.1.1 Hardware Implementation

To configure the NAO6 robot every component has to be detected and is performing as expected. A connection to laptop is necessary for this purpose because it enables us to control and monitor the behavior of the robot.



Figure 4.1: NAO6 Robot

4.2.1.2 Software Implementation

For the NAO6 Robot to connect to ChatGPT, there are a few steps that need to be taken when it comes to software implementation. This is an NAO6 robot that is powered by the NAOqi framework hence making it have its functionality basis. Below are the steps to be followed so that ChatGPT can be integrated into NAO6 Robot powers:

1. Installing necessary libraries: `langchain`, `openai`, `chromadb`, `tiktoken`, `unstructured` to allow ChatGPT to understand our SQLite database.
2. Establishing connectivity between the NAO6 robot and laptop via IP, using the NAOqi library.
3. Transmitting audio recordings captured by the NAO6 Robot's microphone to external devices using the NAOqi library.
4. Since the NAO6 voice recognition does not recognize all words we use the Google Speech Recognition library that converts the recorded audio input to text form (Speech to Text).
5. To access the ChatGPT feature we use the OpenAI API-KEY.
6. Sending the converted text input to ChatGPT for query execution, searching both our SQLite database and the ChatGPT database.
7. We send the response that comes from ChatGPT in text format to the NAO6 Robot using the NAOqi library enabling the robot to vocalize the response (Text to Speech) with the appropriate movement of NAO's hands and face.

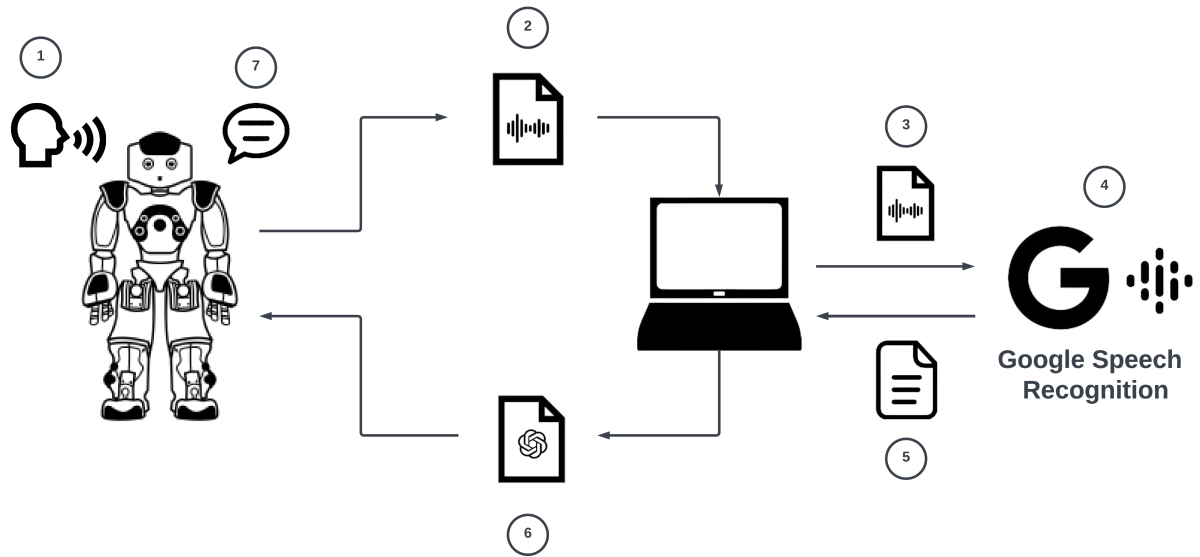


Figure 4.2: Description of Operation

Steps of Nao6 Operation:

1. The user provides input via voice through the Nao6 microphone.
2. The voice is captured and sent to the laptop as a .wav file.
3. The laptop sends the .wav file to Google Speech Recognition.
4. The recorded voice is processed by Google Speech Recognition for transcription.
5. The transcribed text is sent back to the laptop.
6. The laptop sends the transcribed text (representing the user's input) to ChatGPT. ChatGPT generates an appropriate response and sends the response as text back to the Nao6.
7. The Nao6 converts the received text response into voice and informs the user of the result.

4.2.2 Automated Library System

4.2.2.1 Hardware Implementation

For this project, we will connect several components to the Raspberry Pi to create a library system with facial recognition, RFID tracking, and other functionalities. The hardware components consist of a web camera, an RFID RC552 reader, an LCD, a UHF reader and a buzzer.

- **Raspberry Pi:**

- The central processing unit of the system. We ensure the Raspberry Pi is properly powered using a suitable power adapter.
- Install Raspberry Pi OS Raspbian GNU/Linux 11 (Bullseye) with version 6.6.18-v7+ to microSD than Insert microSD card with the Raspberry Pi OS into the slot.

- **USB Webcam:**

- We connect the camera module to the Raspberry Pi's through USB port.
- Ensure the connection is secure by locking the connector properly.

- **RFID RC522 Reader and LCD Display :**

- Connect the RFID reader and LCD Display to the Raspberry Pi's GPIO pins as follows:

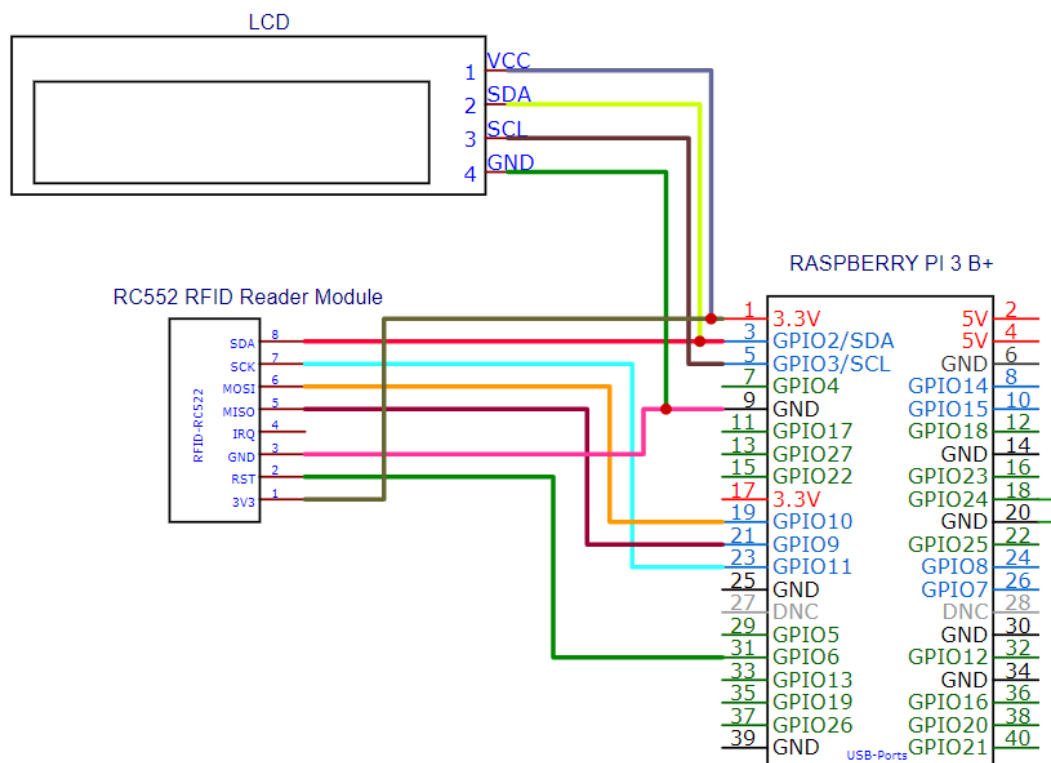


Figure 4.3: RFID RC522 Reader and LCD Display Connection

- **VM-5GA UHF RFID reader:**
 - Connect the UHF reader to the Raspberry Pi according to the manufacturer’s specifications, usually through a serial interface (TTL).
 - Ensure proper connections for TX, RX, GND, and power.
- **Buzzer:**
 - Connect the buzzer to a GPIO pin and GND.

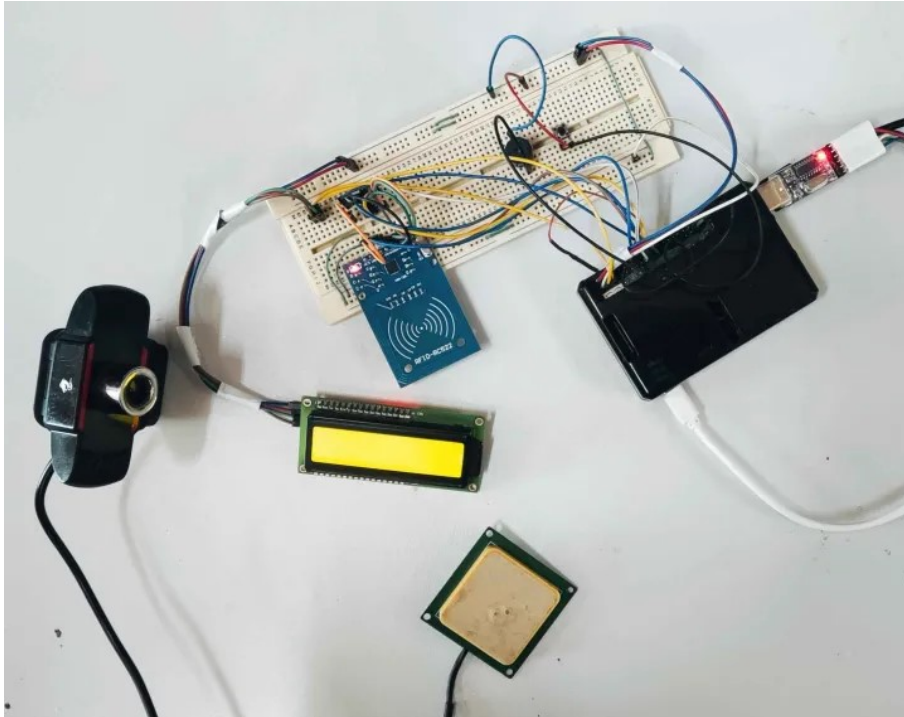


Figure 4.4: Automated Library System

4.2.2.2 Software Implementation Raspberry Pi:

- Install Raspbian OS on Raspberry Pi.
- Install and configure necessary Python libraries to interact with components that connect to it each component has its own library to work with.

USB Webcam:

- Ensure the connection is secure by locking the connector properly.
- Install and configure the necessary drivers and `Open CV` library for the camera module.

RFID RC522 Reader:

- Using the MFRC522 library to interface with the RFID reader.
- Enable SPI using the command: `sudo raspi-config`.
- Check if `spi_bcm2835` is loaded with the command: `lsmod | grep spi`.
- Run the Read Data script and place the RFID tags near the RC522 to read them.

```
1 import time
2 from mfrc522 import MFRC522
3
4 reader = MFRC522()
5 def read_rfid_uid():
6     start_time = time()
7     while time() - start_time < 5:
8         # Scan for cards
9         status, TagType = reader.MFRC522_Request(reader.PICC_REQIDL)
10        if status == reader.MI_OK:
11            print("Card detected")
12            # Get the UID of the card
13            status, uid = reader.MFRC522_Anticoll()
14            if status == reader.MI_OK:
15                # Select the first 4 bytes of the UID
16                uid = uid[:4]
17                # Format the UID as a string
18                uid_str = "".join("{x:02X}" for x in uid)
19            return uid_str
```

Figure 4.5: RFID Read Data Code

LCD Display:

- Install and configure the `drivers` libraries to control the LCD.

```
1 import time
2 import drivers
3
4 # Initialize the LCD
5 display = drivers.Lcd()
6
7 # Display message in line 1
8 display.lcd_display_string("Message in line 1", 1)
9 # Display message in line 2
10 display.lcd_display_string("Message in line 2", 2)
11
12 # Wait for n# seconds
13 time.sleep(n)
14
15 # Clear the display
16 display.lcd_clear()
17
```

Figure 4.6: LCD Display Code

VM-5GA UHF RFID Reader:

- Ensure proper connections for TX, RX, GND, and power.
 - Install and configure the **Serial** library to interface with the UHF reader.
 - Check the manufacturer manual to use the appropriate commands to communicate with VM-5GA UHF RFID reader [22]:
1. In Table 4.1 is The command frame "BB 00 03 00 01 00 04 7E" is used to request the hardware version, with specific values assigned to each field to ensure correct communication.

Header	Type	Command	PL (MSB)	PL (LSB)	Parameter	Checksum	End
BB	00	03	00	01	00	04	7E

Table 4.1: Get Hardware Version Command

- **Frame Type: 0x00**
 - **Command: 0x03**
 - **PL: 0x0001**
 - **Parameter: 0x00(get the hardware version)**
 - **Checksum: 0x04**
 - **Response frame type: 0x01**
 - **Command 0x03**
 - **DataVariable(ASCII code)**
2. In Table 4.2: The "Single Polling Command" uses the frame "BB 00 22 00 00 22 7E" to check the device status. It includes fields like Frame Type (0x00), Command (0x22), and Checksum (0x22). The response would include a frame type of 0x01, indicating that the polling request was received and processed correctly by the device.

Header	Type	Command	PL (MSB)	PL (LSB)	Checksum	End
BB	00	22	00	00	22	7E

Table 4.2: Single polling Command

- **Frame Type: 0x00**
- **Command: 0x03**
- **PL: 0x0000**
- **Checksum: 0x22**

3. In Table 4.3 uses the frame BB 01 FF 00 01 15 16 7E to indicate that no tags were detected or there was a data CRC parity error. It includes fields such as Frame Type (0x01), Command (0xFF), and Parameter (0x15). The response will include a frame type of 0x01, signaling that the command was received and processed correctly but no valid tags were found.

Header	Type	Command	PL (MSB)	PL (LSB)	Parameter	Checksum	End
BB	01	FF	00	01	15	16	7E

Table 4.3: No Tags Response Command

- **Frame Type: 0x01**
- **Command: 0xFF**
- **PL: 0x01**
- **Parameter: 0x15**
- **Checksum: 0x16**

4. In Table 4.4 uses the frame BB 02 22 00 11 C9 34 00 30 75 1F EB 70 5C 59 04 E3 D5 0D 70 3A 76 EF 7E to indicate that the chip has received and processed a single polling command and successfully read the tag data. The response includes fields such as RSSI, PC, EPC, and CRC. If multiple tags are read, multiple response commands will be returned.

Header	Type	Command	PL (MSB)	PL (LSB)	RSSI	PC (MSB)	PC (LSB)
BB	02	22	00	11	C9	34	00
EPC (MSB)							
30	75	1F	EB	70	5C	59	04
			EPC (LSB)	CRC (MSB)	CRC (LSB)	Checksum	End
E3	D5	0D	70	3A	76	EF	7E

Table 4.4: Multi Polling Command

- **Frame Type: 0x02**
- **Command: 0x27**
- **PL: 0x0011**
- **RSSI: 0xC9**
- **PC: 0x3400**
- **EPC: 0x30751FEB705C5904E3D50D70**
- **CRC : 0x3A76**

- **Checksum: 0xEF**

5. In Table 4.5 is used to set the Select parameters before performing polling operations. When the Select mode is set to 0x02, this command allows you to specify which tags should be targeted for subsequent operations like reading and writing. If multiple tags are detected, you can filter and operate only on tags that meet the criteria defined by the Select parameters.

Header	Type	Command	PL (MSB)	PL (LSB)	Sel-Param	Ptr (MSB)	
BB	00	0C	00	13	01	00	00
	Ptr (LSB)	Mask-Len	Truncate	Mask (MSB)			
00	20	60	00	30	75	1F	EB
							Mask (LSB)
70	5C	59	04	E3	D5	0D	70
Check-Sum	End						
AD	7E						

Table 4.5: read and writing only for special tags according to the Select parameter

- **Frame Type: 0x00**
- **Command: 0x0C**
- **PL0x0013**
- **SelParam: 0x01 (Target: 3'b000, Action: 3'b000, MemBank: 2'b01)**
- **Ptr: 0x00000020(unit is bit, not word) start from EPC bit.**
- **Mask Length: 0x60(6 words, 96bits)**
- **Mask: 0x30751FEB705C5904E3D50D70**
- **Checksum: 0xAD**
- **SelParam is with Byte and Target owns 3bits, Action owns the 3bits in middle, MemBank owns the last 2bits,MemBank definition as below:**
 - 2'b00: RFU data storage area of the tag.
 - 2'b00: EPC data storage area of the tag.
 - 2'b00: TID data storage area of the tag.
 - 2'b00: User data storage area of the tag.

VM-5GA UHF RFID Reader Code:

1. In this snippet we calculate a checksum for a given list of bytes then we construct a command packet to be sent over the serial port then we Send a command to the serial port and wait for a response.

```
def calculate_checksum(data):
    return sum(data) & 0xFF

def construct_command(command_body):
    header = 0xBB
    end = 0x7E
    command = [header] + command_body
    checksum = calculate_checksum(command_body)
    command.append(checksum)
    command.append(end)
    return bytes(command)

def send_command(ser, command):
    ser.write(command)
    ser.flush()
    response = ser.read_until(b'\x7E')
    return response
```

Figure 4.7: VM-5GA UHF RFID Reader Snippet Code.

2. In this snippet we Send a specific polling command to the serial port and process the response to extract the tag UID and we continuously poll for tags until a tag UID is found.

```
def single_polling(ser):
    command_body = [0x00, 0x22, 0x00, 0x00]
    command = construct_command(command_body)
    response = send_command(ser, command)
    print(command.hex())
    tag_uid = response.hex()[28:-8]
    if tag_uid:
        print(f"{tag_uid}")
        return tag_uid
    return None

def read_tags(ser, interval=2):
    while True:
        tag_uid = single_polling(ser)
        if tag_uid:
            break
        time.sleep(interval)
```

Figure 4.8: VM-5GA UHF RFID Reader Snippet Code.

3. In this snippet after we define the functions to work with VM-5GA we establish a serial connection to `/dev/ttyUSB0` with a baud rate of **115200** since the VM-5GA works at that rate if the connected success we start reading tags.

```
def main():
    ser = None
    try:
        ser = serial.Serial('/dev/ttyUSB0', 115200, timeout=1)
        if ser.is_open:
            print("Connected to /dev/ttyUSB0")
            read_tags(ser, interval=4)
        else:
            print("Failed to open serial port.")
    except Exception as e:
        print("Error:", e)
    finally:
        if ser and ser.is_open:
            ser.close()

main()
```

Figure 4.9: VM-5GA UHF RFID Reader Snippet Code.

Buzzer:

- We connect the buzzer to a GPIO pin and GND.
- We install and configure the `gpiozero` library to control the buzzer.

Face Recognition and SQLite Process:

- We install and configure the `Open CV`, `Face Recognition` librarys .
- After configuring the camera module, we use it to capture images of users.
- We store these images in a dataset file and associate information in the SQLite database.
- To identify user identities, we use both the dataset file and the SQLite database.
- We check if user information exists in both the dataset file and the SQLite database.
- If the user exists, they are allowed to use the system (check-in/checkout). If not, we prompt them to register their face to gain system access.

Face Recognition Code:

1. In Figure 4.9 shows a code snippet of face recognition code in this snippet we prepare a set of known face encodings and their associated names, which can later be used for face recognition.

```
known_face_encodings = []
known_face_names = []

for image_name in os.listdir(known_faces_dir):
    image_path = os.path.join(known_faces_dir, image_name)
    image = face_recognition.load_image_file(image_path)
    face_encodings = face_recognition.face_encodings(image)

    if face_encodings:
        known_face_encodings.append(face_encodings[0])
        known_face_names.append(image_name.split('.')[0])
```

Figure 4.10: Face Recognition Snippet Code.

2. In Figure 4.10 shows a code snippet of face recognition code in this snippet we create a list of face encodings. This list can then be used to compare and recognize faces.

```
for image_name in os.listdir(check_faces_dir):
    image_path = os.path.join(check_faces_dir, image_name)
    image = face_recognition.load_image_file(image_path)
    face_locations = face_recognition.face_locations(image)
    face_encodings = face_recognition.face_encodings(image, face_locations)
```

Figure 4.11: Face Recognition Snippet Code.

3. In this snippet the purpose of this code is to identify faces in an image by comparing the face encodings found in the image with known face encodings.

```

image_bgr = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

face_names = []

for face_encoding, face_location in zip(face_encodings, face_locations):

    matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
    name = "Unknown"

    face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
    best_match_index = face_distances.argmin()
    if matches[best_match_index]:
        name = known_face_names[best_match_index]

    face_names.append(name)

```

Figure 4.12: Face Recognition Snippet Code.

4. In this snippet the purpose of this code is to add a bounding box around each detected face and label it with the associated name if a match is found in the known face encodings. If no match is found, it labels the face as "Unknown"

```

top, right, bottom, left = face_location
cv2.rectangle(image_bgr, (left, top), (right, bottom), (0, 0, 255), 2)

cv2.rectangle(image_bgr, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
font = cv2.FONT_HERSHEY_DUPLEX
cv2.putText(image_bgr, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)

output_path = os.path.join(output_dir, image_name)
cv2.imwrite(output_path, image_bgr)

```

Figure 4.13: Face Recognition Snippet Code.

5. In Figure 4.13 shows how the recognition be:

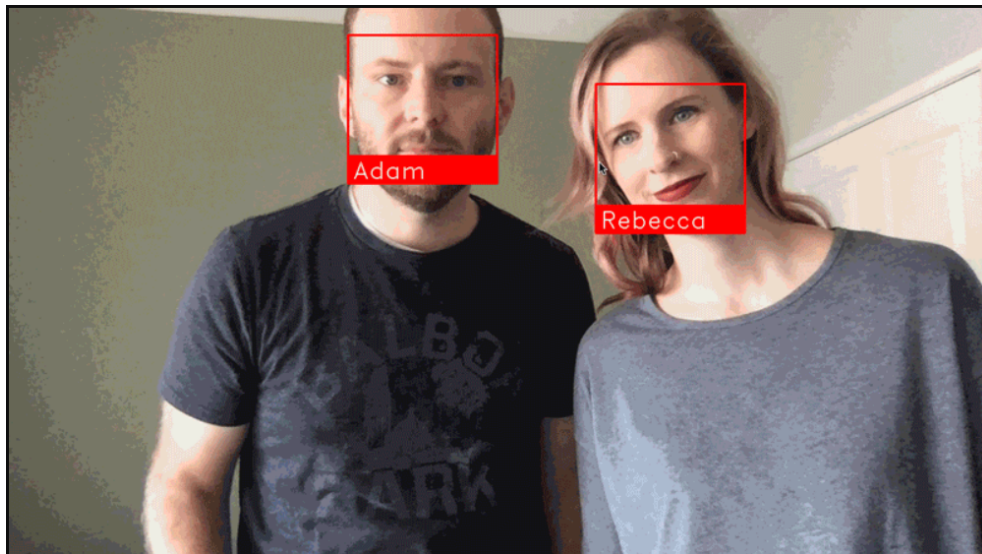


Figure 4.14: Face Identification [23].

4.3 Summary

The project's goal is to modernize the library experience by integrating cutting-edge technologies. Users engage in natural language conversations with the NAO6 humanoid robot, acting as a librarian, thanks to the ChatGPT language model. RFID is utilized for theft prevention and book tracking upon departure from the library, while Raspberry Pi's facial recognition ensures smooth user identification. Registering is user-friendly and involves the capture of patrons' facial images. The database is efficiently managed using SQLite. Ultimately, the project aims to craft an interactive, futuristic library that adapts to users' evolving digital needs.

Chapter 5: Validation & Results

In this chapter, we will discuss the validation of all system components and how results are obtained.

5.1 Choreography Test with NAO6:

By using Choregraphe, we test the NAO6 by asking it some questions, Figure 5.1, In Figure 5.2 show the results of the questions:

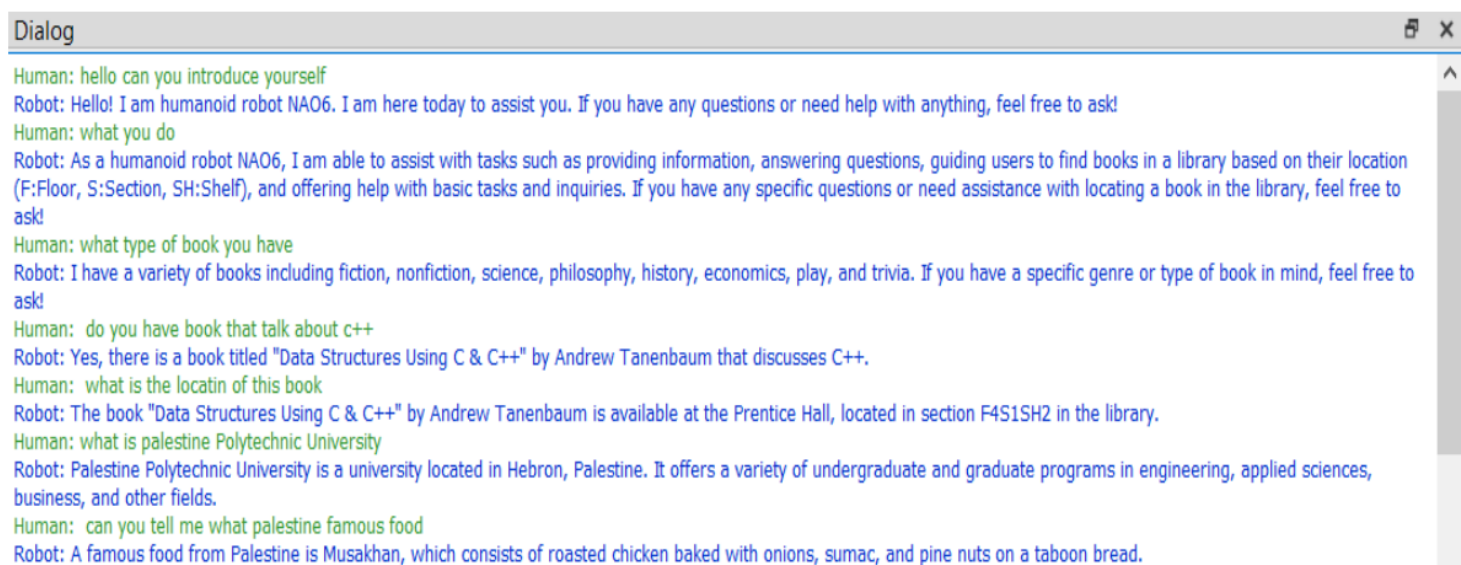


Figure 5.1: Human and Nao6 Conversation

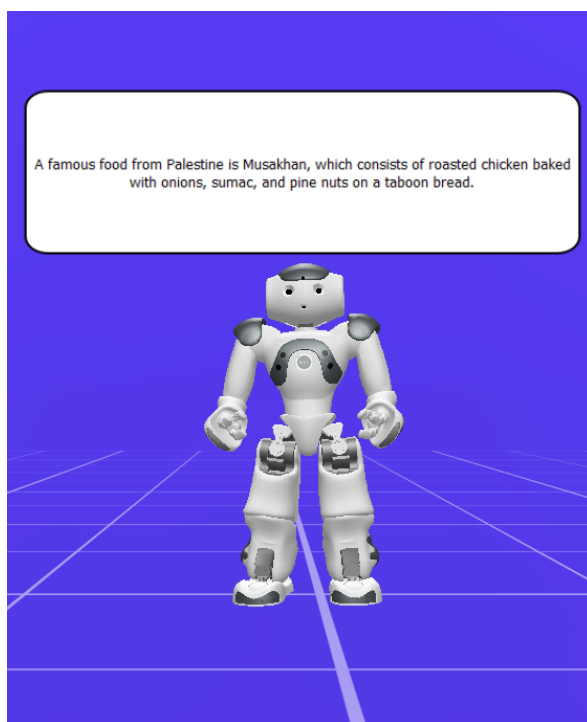


Figure 5.2: Emulation of NAO Speech using Choregraphe

5.2 Automated Library System:

- **Raspberry Pi:** We tested the Raspberry Pi by running it and accessing the terminal.

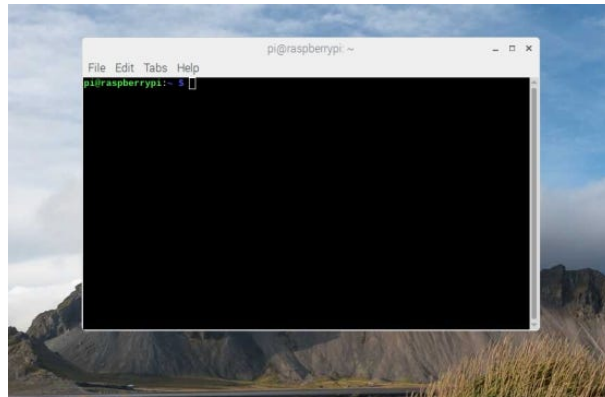


Figure 5.2: Raspberry Pi Terminal

- **LCD Display:**

We tested the LCD by connecting it to the Raspberry Pi and displaying test messages on it.



Figure 5.3: LCD Display .

- **RFID RC522 Reader:**

We tested the RFID RC522 by connecting it to the Raspberry Pi than reading the RFID tag and displaying it on LCD.



Figure 5.4: Reading Tag ID using RC522 Module and Displaying it on LCD.

- **VM-5GA UHF RFID Reader:** We tested the VM-5GA UHF by connecting it to the Raspberry Pi than reading UHF tags and displaying it on LCD.



Figure 5.5: Reading Tag ID using VM5GA Module and Displaying it on LCD.

- **Face Recognition:** We tested the face recognition system by allowing it to identify users based on stored data.

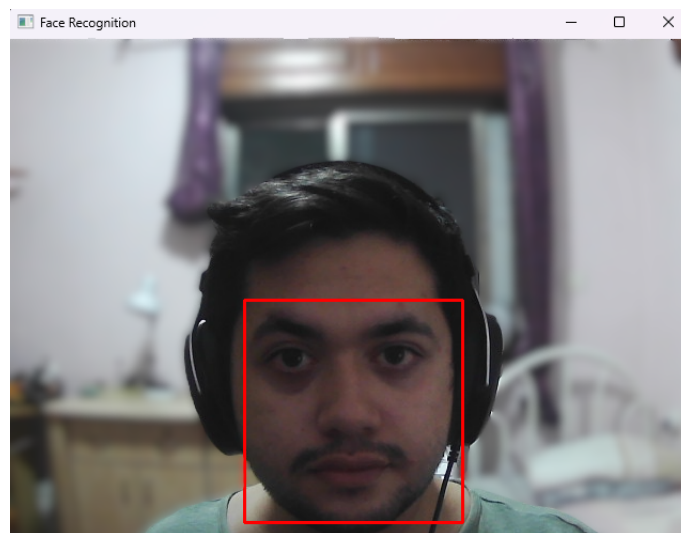


Figure 5.6: Testing Face Detection.

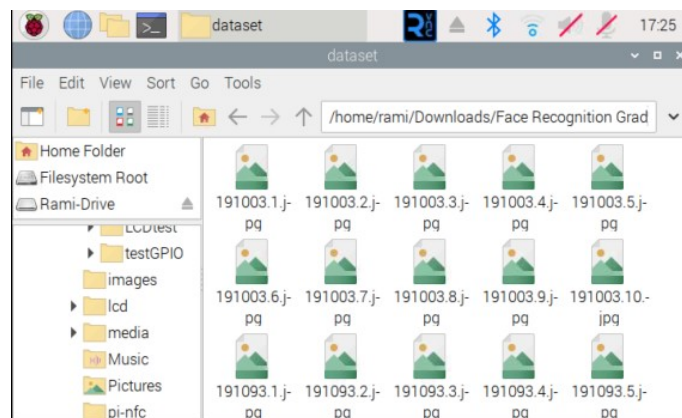


Figure 5.7: Images with ID of each user.

	student_id	student_name	student_email
	Filter	Filter	Filter
1	100010	Alice Johnson	alice.johnson@gmail.com
2	100020	Bob Smith	bob.smith@gmail.com
3	100030	Charlie Brown	charlie.brown@gmail.com
4	100040	Dana White	dana.white@gmail.com
5	100050	Eva Green	eva.green@gmail.com
6	100060	Frank Miller	frank.miller@gmail.com
7	100070	Grace Lee	grace.lee@gmail.com
8	100080	Harry Davis	harry.davis@gmail.com
9	191003	Rami	Rami@gmail.com
10	191093	Moayad Anjjar	MoayadAnjjar@gmail.com

Figure 5.8: User information in database.



Figure 5.9: User identification based on user ID.



Figure 5.10: Message for Unregistered user.

- **Anti-thief System:** We tested the anti-theft system by passing three tags in front of the VM-5GA UHF RFID Reader the alert is work when the once of tags not registered in the database.

book_id	book_uid	student_id	borrow_date	return_date	student_name
Filter	Filter	Filter	Filter	Filter	Filter
B3DF4402	5001835518	191003	2024-07-24	2024-07-31	rami
D18E3C02	4001835510	191003	2024-07-24	2024-07-31	rami

Figure 5.10: Database of borrowed books.



Figure 5.11: Reading Tag ID of existing book



Figure 5.12: Reading Tag ID of existing book



Figure 5.13: Reading Tag ID of not existing book

5.3 Summary:

Within this chapter, we confirmed that the functionalities of diverse elements were okay in the system. Our quest included trials that made certain that each component was functioning normally. The task of the Choregraphe tool was to check if NAO6 can answer queries. As for the automated library system, we checked the terminal access of Raspberry Pi for the Raspberry Pi ensures the LCD's ability to display information together with the reads tags data using RFID RC522 and VM-5GA UHF RFID readers.

Chapter 6: Conclusion & Future work

6.1 Conclusion

In this project, we developed an automated library system with NAO6 Robot as the librarian. It allows the NAO6 Robot, built for hardware integration and software design, to engage in different library functions like providing information power or interacting with users. Therefore, leveraging recent technologies such as ChatGPT, we innovated a new method for library automation.

6.2 Future Work

For future work on the project, we will focus on expanding its capabilities, improving overall performance, and incorporating user feedback to enhance functionality. Our goal is to refine the system, address any limitations, and introduce new features to meet user needs and expectations better.

- **Enhanced User Interaction:** Enhance the conversational capabilities of the NAO6 Robot, so that it can have more flexible and instinctive interactions with library users.
- **Expansion of Services:** Investigate more services and features available on NAO6 such as different guided tours, playing games, and controlling the library system like turning OFF/ON the light.
- **Accessibility Features:** Implement accessibility features, such as voice commands for navigation, text-to-speech capabilities for visually impaired users, or speaking with deaf users by programming Nao6 to use sign language.
- **Performance Optimization:** Optimize the performance of the system by improving hardware efficiency and reducing response times for user queries.
- **Multi-Language Support:** Allow users to choose the language that NAO6 should recognize, such as Arabic, Spanish, English, and others. Upon selecting a language, the system will adjust its interactions accordingly, including speech recognition, natural language understanding, and responses, to match the chosen language. This feature ensures that NAO6 can effectively communicate and interact in the preferred language of the user.

Appendix

References

- [1] Younis, H. A., Ruhaiyem, N. I. R., Ghaban, W., Gazem, N. A., & Nasser, M. (2023). A Systematic Literature Review on the Applications of Robots and Natural Language Processing in Education. *Electronics*, 12(13), 2864.
- [2] BAS-IP, *RFID Meaning Unveiled: A Comprehensive Guide to Radio Frequency Identification*, June 8, 2023, <https://bas-ip.com/articles/rfid-meaning/>,
- [3] Clare Stouffer , July 21, 2023 *What is facial recognition and how does it work?*, <https://us.norton.com/blog/iot/how-facial-recognition-software-works>, Accessed: 20/01/2024
- [4] *Mohammed Abdulameer Aljanabi, Noor Abdalrazak Shnain, 2017*, https://www.researchgate.net/publication/324605665_An_Image_Similarity_Measure_based_on_Joint_Histogram_-_Entropy_for_Face_Recognition,
- [5] Farajpahlou, A. H. (1999). Defining some criteria for the success of automated library systems. *Library Review*, 48(4), 169-180. <https://doi.org/10.1108/00242539910276451>
- [6] WeiT-Meng Lee. *Implementing Face Recognition Using Deep Learning and Support Vector Machines*. Published: February , 2023. Retrieved from <https://www.codemag.com/Article/2205081/Implementing-Face-Recognition-Using-Deep-Learning-and-Support-Vector-Machines>. Accessed: 19/01/2024
- [7] Billing, E., Rosén, J., Lamb, M. (2023). Language Models for Human-Robot Interaction. *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (HRI '23 Companion)*, 905-906. ACM. <https://www.diva-portal.org/smash/get/diva2:1744270/FULLTEXT01.pdf>.
- [8] Yalagi, V. (2021). Smart Library Automation Using Face Recognition. *Journal of Physics: Conference Series*, 1854, 012041. <https://doi.org/10.1088/1742-6596/1854/1/012041>.
- [9] Raspberry Pi Foundation. *Raspberry Pi 3 Model B+ Product Image*. Retrieved from <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>. Accessed: 20/01/2024
- [10] MEGABAJT SP. Z O.O.. *Tracer WEB008 Webcam* . Published: 2016. Retrieved from <https://en.tracer.eu/tracer-pc-cam-hd-web008>. Accessed: 10/01/2024
- [11] Aldebaran. *NAO6 Humanoid Robot*. Retrieved from <https://www.aldebaran.com/en/nao>. Accessed: 18/01/2024
- [12] Indiamart. *Passive RFID Tags*. Retrieved from <https://www.indiamart.com/proddetail/passive-uhf-rfid-tags-22301558662.html>. Accessed: 18/01/2024

- [13] Shenzhen VANCH Intelligent Technology Co. Ltd. *UHF RFID Reader Module VM-5GA*. Retrieved from <https://www.vanch.net/products-list/uhf-rfid-reader-module-vm-5ga/>. Accessed: 21/12/2023
- [14] Components101. *RC522 RFID Module*. Published: June 12, 2019. Retrieved from <https://components101.com/wireless/rc522-rfid-module>. Accessed: 20/12/2023
- [15] LCD Display Module. [Online]. Available: <https://www.amazon.in/CentIoT%C2%AE-Display-Module-LCD-1602-Arduino/dp/B07N8T5CC3?th=1>. Accessed: 18/01/2024
- [16] RS Components Ltd. Birchington Road, Corby, Northants, NN17 9RS, UK. [Online]. Available: <https://uk.rs-online.com/web/p/piezo-buzzers/2596141>. Accessed: 18/01/2024
- [17] BeagleBoard.org Foundation, 2929 Campus Drive, Suite 107, San Mateo, CA 94403, USA. [Online]. Available: <https://www.beagleboard.org/boards/beaglebone-black#:~:text=BeagleBone%C2%AE%20Black%20is%20a,just%20a%20single%20USB%20cable>. Accessed: 17/04/2024.
- [18] Raspberry Pi Foundation. (2022). Official Raspberry Pi Documentation. [Online]. Available: <https://www.raspberrypi.org/documentation/> Accessed: 18/01/2024
- [19] Digi-Key Electronics, 701 Brooks Avenue South, Thief River Falls, MN 56701, USA. [Online]. Available: <https://www.digkey.com/en/product-highlight/t/texas-instruments/beagleboard>. Accessed: 17/04/2024.
- [20] John Wiley & Sons Ltd. (2012). *Raspberry Pi: A Comprehensive Guide*. Official Raspberry Pi Guide. [Online]. Available: <https://www.cs.unca.edu/~bruce/Fall14/360/RPiUsersGuide.pdf>. Publisher: Eben Upton and Gareth Halfacree. Accessed: 22/12/2023
- [21] Scot Fitzgerald & Michael Shiloh, (2012). *Arduino Projects Book*. [Online]. Available: <https://www.uio.no/studier/emner/matnat/ifi/IN1060/v21/arduino/arduino-projects-book.pdf>. Accessed: 18/01/2024
- [22] ShenZhen VANCH Intelligent Technology Co. *UHF VM-5GA MANUAL*. Available: <https://drive.google.com/file/d/1SDbIEZeiBmGZFPUSVpF1QE2Vwg7SBrvT/view>. Accessed: 2/06/2024
- [23] Adam getigey, (Feb 20, 2020). *face-recognition*. Available: <https://pypi.org/project/face-recognition/>. Accessed: 18/03/2024