yRépublique Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université Kasdi Merbah Ouargla
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Electronique et de Communication

Mémoire en vue d'obtention d'un :
MASTER ACADEMIQUE
Domaine : Science et technique
Filière : Electronique
Spécialité : Electronique des Systèmes Embarquées
Préparé par :

BENAGGA Chaker Ellah et **KAFI Mohammed Sabri**

BELAOUDMOU Ahmed Rami et **HASSINI Mohammed Abd El Azziz**

Thème :

Cafeteria Management System

Encadrant de mémoire : Bilal BENARABI
Soutenu le : 19/06/2023
Devant le jury :

Mr. Said ACHBI	Président	UKM Ouargla
Mr. Bilal BENARABI	Encadrant	UKM Ouargla
Mr. Abdessalam MESSAID	Examinateur	UKM Ouargla

Année universitaire : 2022/2023

المحتويات

Table des matières

ACKNOWLEDGMENTS

First and foremost, we would like to thank Allah for giving us the courage and the willpower to see this modest work through to its completion.

We would also like to thank our supervisor, Mr. Bilal Benarabi, who gave us special attention and spared no effort or knowledge in providing us with the most valuable assistance possible.

Next, we would like to express our gratitude to all the members of the jury who honored us by carefully examining our work.

Our thanks also go to the entire teaching staff of the Department of Electronics and Communication at the University of Ouargla, as it was through their efforts that we had no difficulty in filling our theoretical training and satisfying our scientific curiosity.

Finally, we would like to thank all the individuals, friends, and family members who helped us, directly or indirectly, in completing this work.

General Introduction

General Introduction

Expanding on the issue of feeding services in university residences, it is worth noting that the poor management and organization of university cafeterias can lead to several problems. One of the most common problems is long waiting times, which can be frustrating and inconvenient for students who need to quickly grab a meal between classes. Another problem is the lack of transparency in expenses, which can make it difficult for students to budget for their meals.

To address these issues, our project aims to implement innovative solutions that improve the management and organization of university cafeterias. We believe that by leveraging technology and data analysis, In the first chapter, we will provide an overview of the essential elements necessary for the implementation of this project. This will be accomplished through a theoretical study of these elements. Then in the second chapter, we will proceed to the design phase and simulations of the devices to be manufactured in our project. Lastly, we will take a look at the software responsible for managing the system and inventory.

Overall, we are committed to enhancing the quality of feeding services in university residences and ensuring that students have access to affordable, nutritious, and convenient meals. We believe that by addressing these issues, we can contribute to the overall well-being and academic success of university students.

Chapter I: General Study of management system components

I.1 Introduction

In this chapter, we will conduct a comprehensive theoretical study to familiarize ourselves with the essential components involved in the production of our product. We will begin by defining controllers and examining their importance in the overall system of the product. We will explore different types of controllers available and their role in effective process control and management. Next, we will delve into a detailed study of RFID technology and its role in the product manufacturing process. Subsequently, we will move on to studying the motors required for operating the conveyor belt.

I.2 MICROCONROLLER

In our project, the controller will play a fundamental role in system implementation as it works on controlling and linking between all the element.

A microcontroller is a compact intertwined circuit designed to govern a specific operation in an embedded system. A typical microcontroller includes a processor, memory and input/ output (I/ O) peripherals on a single chip. occasionally appertained to as an embedded controller or microcontroller unit(MCU), microcontrollers are set up in vehicles, robots, office machines, medical devices, mobile radio transceivers, dealing machines and home appliances, among other devices. They're basically simple atomic particular computers (PCs) designed to control small features of a larger element, without a complex front- end operating system(OS) [1].

I.2.2 The elements of a microcontroller:

The core elements of a microcontroller as showing in (Figure I.1) are:

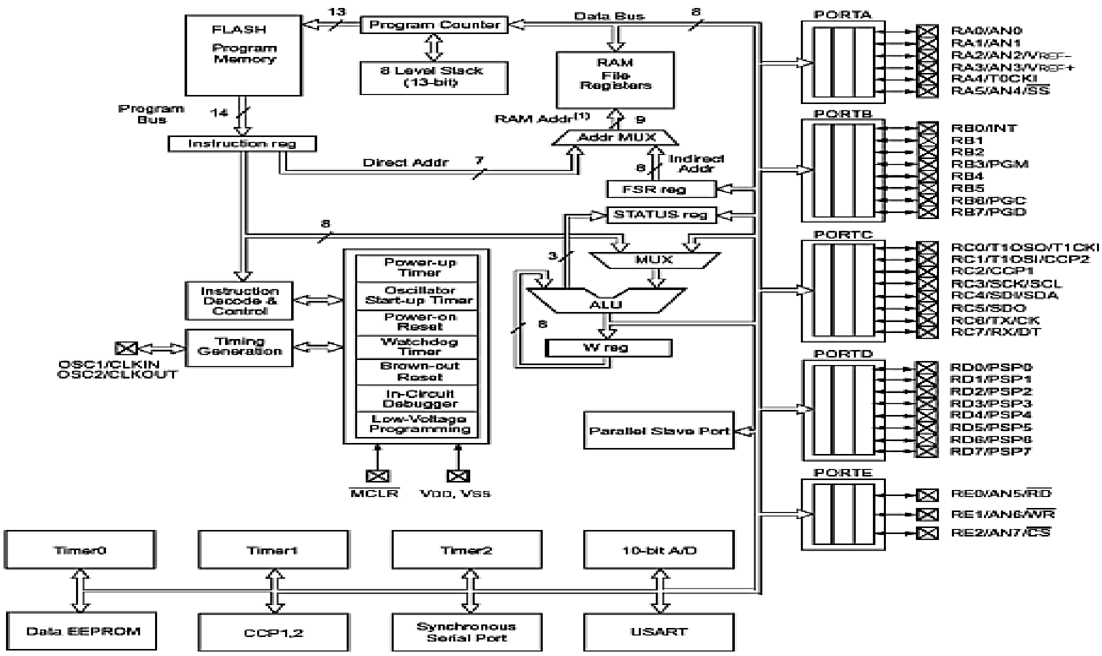


Figure I.1: Parts of Micro Controller **[2].**

1. **The processor (CPU):** A processor can be allowed of as the brain of the device. It processes and responds to colourful instructions that direct the microcontroller's function. This involves performing introductory computation, logic and I/O operations. It also performs data transfer operations, which communicate commands to other components in the larger embedded system**.**

2. **Memory:** A microcontroller's memory is used to store the data that the processor receives and uses to respond to instructions that it's been programmed to carry out. A microcontroller has two main memory types
   - **Program Memory:** which stores long- term information about the instructions that the CPU carries out? Program memory is on-volatile memory, meaning it holds information over time without demanding a power source.
   - ***Data Memory:*** which is needed for temporary data storehouse while the instructions are being executed? Data memory is unpredictable, meaning the data it holds is temporary and is only maintained if the device is connected to a power source.

3. **I/ O peripherals:** The input and output devices are the interface for the processor to the outside world. The input ports admit information and shoot it to the processor in the form of double data. The processor receives that data and sends the necessary instructions to output devices that execute tasks external to the microcontroller**.**

Other supporting elements of a microcontroller include

- **Analogue to Digital Converter (ADC)**: An ADC is a circuit that converts analogue signals to digital signals. It allows the processor at the centre of the microcontroller to affiliate with external analogue devices, similar as detectors.
- **Digital to analogue Converter (DAC)**: A DAC performs the inverse function of an ADC and allows the processor at the center of the microcontroller to communicate its gregarious signals to external analogue components.
- **System bus**: The system bus is the connective line that links all components of the microcontroller together.
- **Periodical port**: The periodical port is one example of an I/ O port that allows the microcontroller to connect to external components. It has an analogous function to a USB or a resembling port but differs in the way it exchanges bits [1] .

I.2.4 Functioning of microcontrollers:

A microcontroller is embedded inside of a system to control a singular function in a device. it does this by interpreting data it receives from its I/ O peripherals using its central processor. The temporary information that the microcontroller receives is stored in its data memory, where the processor accesses it and uses instructions stored in its program memory to decipher and apply the incoming data. It also uses its I/O peripherals to communicate and legislate the applicable action. Microcontrollers are used in a wide array of systems and devices. Devices frequently use multiple microcontrollers that work together within the device to handle their separate tasks. For example, a car might have numerous microcontrollers that control colorful individual systems within, similar as theanti-lock braking system, traction control, fuel injection or suspension control. All the microcontrollers communicate with each other to inform the correct conduct. Some might communicate with a more complex central computer within the car, and others might only communicate with other microcontrollers. They shoot and admit data using their I/ O peripherals and process that data to perform their designated tasks [1].

I.2.5 Types of microcontrollers

Common MCUs include the Intel MCS- 51, frequently appertained to as an 8051 microcontroller, which was first developed in 1985; the AVR microcontroller developed by Atmel in 1996; the programmable interface controller (PIC) from Microchip Technology; and colorful licensed Advanced RISC Machines (ARM) microcontrollers as showing in Figure I.2 [3].

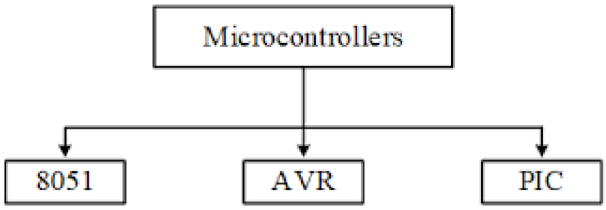


Figure I.2: Types of Microcontroller [4].

I.2.6 Microcontroller features

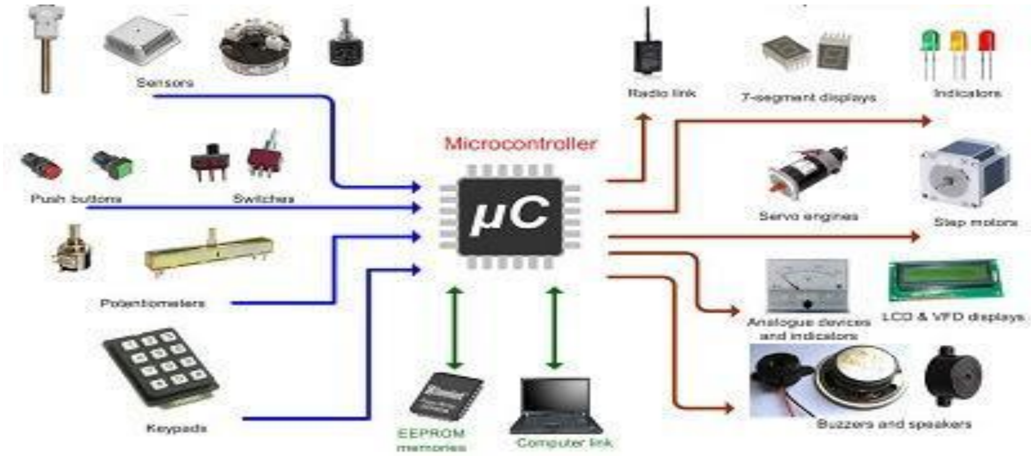
A microcontroller's processor will vary by application. Options range from the simple 4- bit, 8- bit or 16- bit processors to more complex 32- bit or 64- bit processors. Microcontrollers can use unpredictable memory types similar as arbitrary access memory (RAM) and non-volatile memory types-- this includes flash memory, erasable programmable read-only memory (EPROM) and electrically erasable programmable read-only memory(EEPROM).

Microcontroller architecture can be grounded on the Harvard architecture or von Neumann architecture, both offering different styles of swapping data between the processor and memory. With Harvard architecture, the data bus and instruction are separate, allowing for contemporaneous transfers. With Von Neumann architecture, one bus is used for both data and instructions. Microcontroller processors can be grounded on complex instruction set computing (CISC) or reduced instruction set computing(RISC). CISC generally has around 80 instructions while RISC has about 30, as well as further addressing modes, 12- 24 compared to RISC's 3- 5. While CISC can be easier to apply and has more effective memory use, it can have performance declination due to the advanced number of clock cycles demanded to execute instructions. RISC, which places further emphasis on software, frequently provides better performance than CISC processors, which put further emphasis on hardware, due to its simplified instruction set and, thus, increased design simplicity, but because of the emphasis it places on software, the software can be more complex. Which ISC is used varies depending on application.

MCUs feature input and output pins to apply supplemental functions. similar functions include analogue- to- digital converters, liquid crystal display (LCD) controllers, real- time clock(RTC), universal coetaneous/ asynchronous receiver transmitter(USART), timers, universal asynchronous receiver transmitter(UART) and universal periodical bus(USB) connectivity. Detectors gathering data related to moisture and temperature, among others, are also frequently attached to microcontrollers [1].

I.2.7 Microcontroller applications

Microcontrollers are used in multiple industries and applications (**Figure I.3**), including in the home and enterprise, erecting automation, manufacturing, robotics, automotive, lighting, smart energy, artificial automation, communications and internet of things (IoT) deployments. One very specific application of a microcontroller is its use as a digital signal processor. Constantly, incoming analogue signals come with a certain position of noise. Noise in this context means nebulous values that cannot be readily restated into standard digital values. A microcontroller can use its ADC and DAC to convert the incoming noisy analogue signal into an indeed gregarious digital signal.



FigureI.3: Microcontroller applications [5]

The simplest microcontrollers grease the operation of electromechanical systems set up in everyday convenience items, similar as ovens, refrigerators, broilers, mobile devices, crucial fobs, video game systems, televisions and lawn- watering systems. They're also common in office machines similar as photocopiers, scanners, fax machines and printers, as well as Smart meters, ATMs and security systems. More sophisticated microcontrollers perform critical functions in aircraft, spacecraft, and ocean- going vessels, vehicles, medical and life - support systems as well as in robots. In medical scenarios, microcontrollers can regulate the operations of an artificial heart, kidney or other organs. They can also be necessary in the functioning of prosthetic devices [1].

I.3 RFID TECHNOLOGY

RFID in this project will have an important and fundamental role in our project as it determines the student's identity and sends it for processing .RFID is an acronym that stands for “radio- frequency identification” and refers to a technology whereby digital data decoded in RFID markers are captured by a reader via radio waves. RFID is analogous to barcoding in that data from a tag or marker are captured by a device that stores the data in a database. The tag contains a chip that can carry lower than one megabyte of data and is used by interrogators for identification. this technology is considered one of the stylish option to enable the digitization process due to its low- cost, battery-less asset and long life compared to other ways [6][7].

I.3.1 Embedded Passive RFID Tags

Passive RFID tags (**FigureI.4)** are decreasingly used to automate systems, identify products, trace means, monitor, measure the process, and control their physical parameter. But bedding unhesitant RFID tags requires numerous considerations related to the RFID itself, the object or systems, and/ or the space that contains the RFID. The effectiveness of the RFID tag design can be decided by parameters in terms of size, shape, and configuration, the material of the antenna and chip, and the objects that RFID tags install on or are bedded.

FigureI.4: RFID Tags [5]

The RFID tag design parameters are essential, so bedding unhesitant RFID tags in the analog pointers and manometers is a practical challenge and implicit research area. Digital transmitters could replace these manometers, but safety standards emphasize the need to keep them as redundancies for analogical ones. These restrictions insure that operators administrators can cover critical parameters similar as boiler temperature and pressure in both styles, on- site/ physically and in an automation system/ digitally. In this regard, the purpose of our practical operation is to digitize hotspots and trigger points in an analogy manometer passively and wirelessly [6].

I.3.2 RFID functions

The main components of RFID are tags, tag antenna, reader antenna, reader, and savant (**Figure I.5**). A tag carries data and substantially contains identification number and a unique code. The tag antenna is a line or conductive ink attached to the chip in the tag. The reader antenna is a coil incorporated in a plastic. A reader detects the data stored in the RFID tag within its sphere. A reader can accept different information from different tags at a time. A savant acts as a middleware that links the readers with external databases In general; RFID system has three parts a scanning antenna, a transceiver that interprets data, and a transponder or the RFID tag (**Figure I.4**) that contains applicable information. The scanning antenna emits radio frequency signals within a short domain. The radio frequency radiation provides the capability of communicating with the tag and energy needed by the tag to communicate. This is an important aspect of RFID technology since RFID tags need not to have batteries and can therefore operate for longer time (In case of unresisting tags). The scanning antenna can be placed on the ground and in other cases some are movable. For case, they can be placed on a door frame to identify visitors. When an RFID tag enters in the sphere of the scanning antenna, it discovers the launch signal from the antenna. That triggers the RFID chip to convey information to the scanning antenna. Also, RIFD tag can be active or unresisting. Active RFID tags contain some kind of energy source substantially batteries. This implies that data transmission is nonstop. Passive RFID tags calculate on the power- radio signal transmitted by the reader through the scanning antenna. Active tags can transmit data in long ranges and are less sensitive to hindrance as compared to unresisting tags. Despite their difference, both tags carry the same information, which is appertained to as the Electronic Product Code [7].

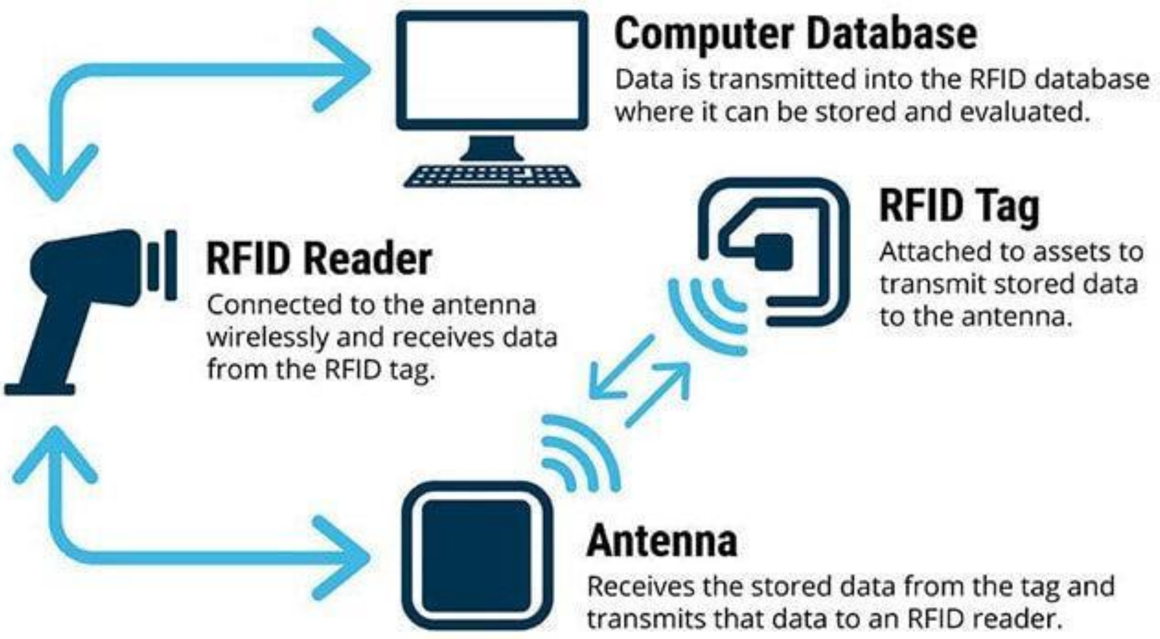


Figure I.5: RFID Work Process [8]

I.3.3 Applications of RFID

RFID tags are made in colorful shapes and sizes in respects to different materials, animal tracking tags can be rice- sized and placed beneath the skin; screw shaped tags can be used to identify trees or rustic objects; access applications use credit- card shaped tags; hard-plastic tags can be attached on products for security purposes, and heavy- duty tags are used to cover shipping holders or heavy objects(**FigureI.6**). In essence, the operations of RFID technology are hand identification and access control, airline baggage identification, livestock identification, parts identification, vehicle identification and tracking, supply chain automation, asset tracking, identification of widgets, and numerous further[6].

FigureI.6 : RFID Application[9]

The lack of internal power supply made Passive RFID tags dependent on the energy absorption transferred from the RFID reader antenna, making the tag's performance largely linked to the environment and the installation position. For this reason, the RFID must be tuned precisely for its installation substrate. For this reason, the research methodology is divided into three sections [7].

- Experiment and abstract test.
- Design and tune a new RFID tag.
- Electromagnetic isolation covers simulation and optimization.

The study methodology starts with modeling EM isolation effects on marketable RFID tags. Due to the large dimension of general purpose RFID markers (a many centimeters), a test bench was designed on a large scale. Modeling was done to test the original conception and determine the influential variables. Eventually, we execute a compound EM simulation to validate the digitization concept for the embedded RFIDs, device, and metallic cover [7].

I.3.4 Problems with RFID

The implementation of RFID technology varies across different companies. This is because global standards for RFID are still in their design phase. Generally, some RFID devices cannot be moved down from their area, and thus can beget difficulties for associations. Customers in numerous cases have problems with RFID standards. However, it means that he or she'll be needed to only pierce the services or products of that company, If an individual contains a tag registered by one company. It's noted that, a consumer will thus need to have different cards to pierce different RFID systems. Accordingly, RFID systems can be disintegrated because they use electromagnetic field, similar as the Wi-Fi networks. Disruption in this sense implies that the systems can jam. This leads to adverse problems in sensitive places like hospitals or military. Again, active RFID tags can be disintegrated and rendered inactive when used constantly. In other circumstances, signals from two readers might collide, making the tag to fail to induce a response. RFID tags can also collide in cases where numerous tags are available in a small area; this depends on the speed of reading the tags. There are different security, privacy and ethical issues in relation to RFID. First, the information of RFID tag can be penetrated after the item departs the supply chain. An RFID cannot distinguish readers and therefore information can be penetrated by movable RFID scanners. Second, tags used in consumer products are delicate to remove. Still, new technologies give a way of bedding removable tags on items. Incipiently, RFID tags can be penetrated without the owner’s knowledge because tags need not to be swiped; this leads to loss of privacy [7].

I.4 AC Motors

The function of the motor in this project is rotating the conveyor belt. The alternating current (AC) motor is a remarkable engineering invention that has revolutionized various industries and shaped our modern world. Developed in the late 19th century, AC motors are widely used in diverse applications, ranging from industrial machinery to household appliances. This essay explores the working principle, types, and key applications of AC motors, highlighting their significant impact on the global economy and technological advancements [10].

I.4.1 Working Principle

AC motors are based on the principles of electromagnetism and electromagnetic induction **(Figure I.7)**. The primary components of an AC motor include a stator, rotor, and a power source that supplies alternating current. The stator consists of stationary coils, while the rotor contains a set of conductive bars or coils. When an AC voltage is applied to the stator, it induces a rotating magnetic field. As a result, the rotor experiences a magnetic force, causing it to rotate [10] .

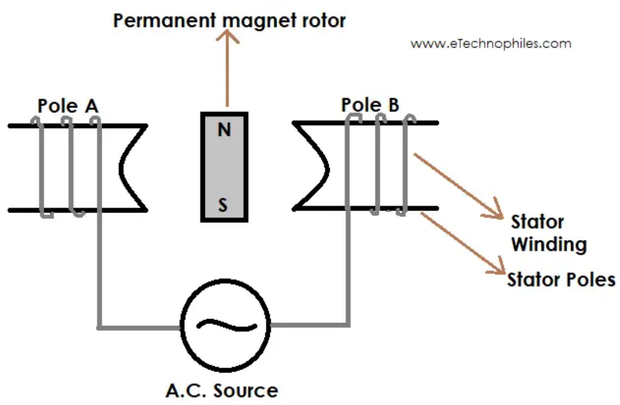


Figure I.7: Working principle of AC motor [11] .

I.4.2 Types of AC Motors

AC motors can be classified into various types, each designed for specific applications. The two most common types are induction motors and synchronous motors **(Figure I.9)**. Induction motors **(Figure I.8)**, also known as asynchronous motors, are extensively used due to their simplicity, reliability, and cost-effectiveness.

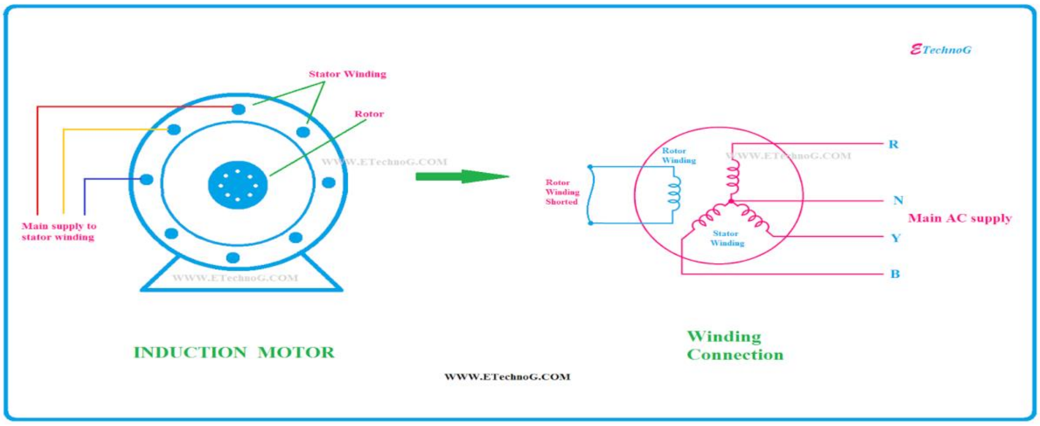


Figure I.8: induction motor[12]

They operate on the principle of induction **(Figure I.8)**, where the rotor rotates at a slightly slower speed than the rotating magnetic field produced by the stator. On the other hand, synchronous motors(**Figure I.9)** operate at a constant speed by synchronizing the rotor's speed with the frequency of the applied AC power [10].

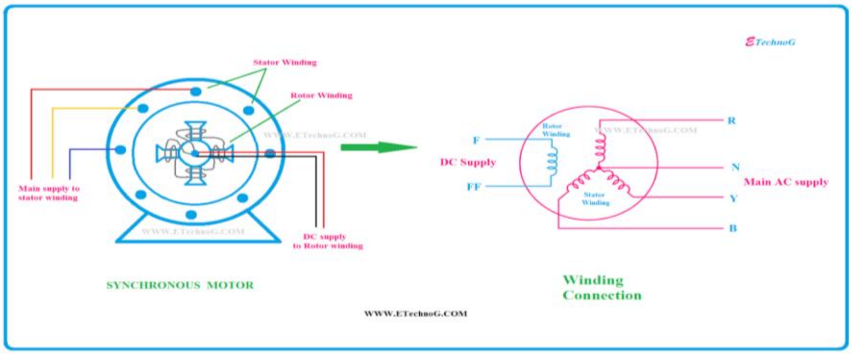


Figure I.9: synchronous motor [12]

I.4.3 Applications of AC Motors

AC motors find application in numerous industrial sectors, contributing to enhanced productivity and efficiency. In manufacturing industries, AC motors power conveyor belts, pumps, compressors, and various types of machinery. They are crucial in the operation of HVAC systems, refrigerators, washing machines, and other household appliances. AC motors are also utilized in electric vehicles, elevators, and renewable energy systems such as wind turbines [13].

I.5 Serial Communication

Serial communication is an essential method for transmitting data bit-by-bit over a communication channel(**Figure I.10)**, enabling efficient data transfer in various electronic devices, including microcontrollers, sensors, and embedded systems. It serves as a widely utilized communication protocol in computer networking and other applications.

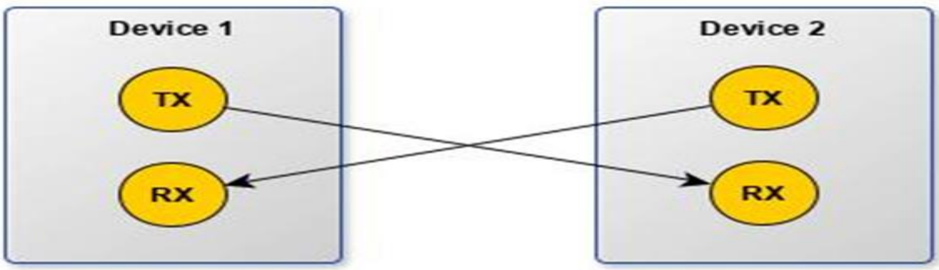


Figure I.10: Serial Communication [14]

When transmitting data over long distances, serial communication proves advantageous over parallel communication due to limitations in wire count and potential signal degradation. By transmitting data sequentially over a single wire, serial communication ensures reliable and efficient data transfer.

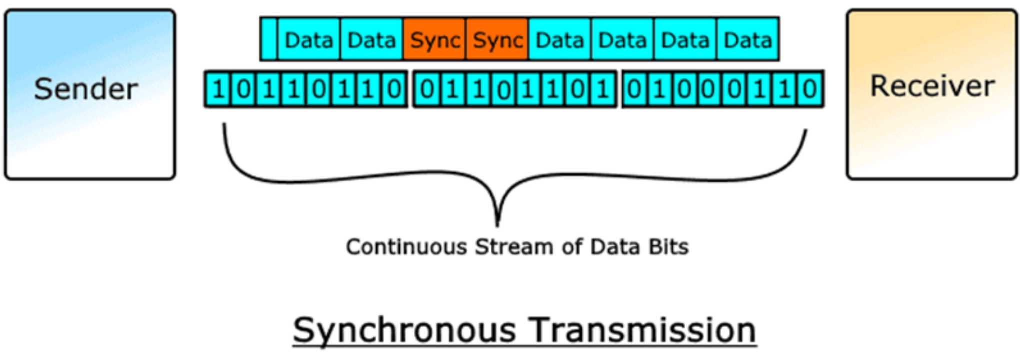
The receiving device in a serial communication setup interprets the incoming data by reconstructing it back into its original form. This process allows for the accurate retrieval and utilization of the transmitted information.

I.5.1 Types of Serial Communication:

There are two main types of serial communication: synchronous and asynchronous.

A. Synchronous Serial Communication:

In synchronous serial communication, the sender and receiver are synchronized to a common clock signal. This means that data is sent and received at a specific rate determined by the clock signal (**Figure I.11**).



FigureI.11: SYNCHRONOUS TRANSMISSION [16]

Synchronous Serial Communication has a range of applications, for example Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), and Universal Serial Bus(USB) [15]:

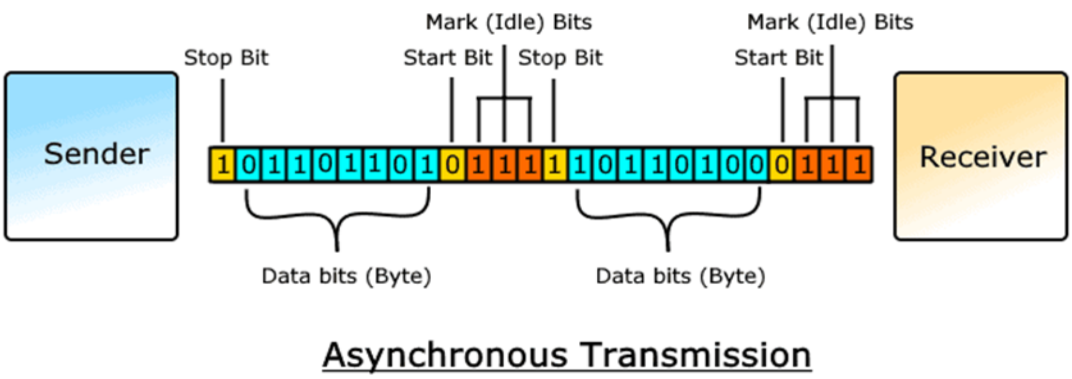
I2C (Inter-Integrated Circuit): This serial communication protocol is commonly used in embedded systems for communication between microcontrollers and peripheral devices such as sensors and actuators [17].

SPI (Serial Peripheral Interface): This protocol is widely used in embedded systems for high-speed communication between microcontrollers and peripheral devices like flash memory and sensors [18].

USB (Universal Serial Bus): USB is a popular serial communication protocol employed for high-speed communication between computers and external devices like printers, scanners, and storage devices [19]. .

B. Asynchronous Serial Communication:

In asynchronous serial communication, data is sent one byte at a time without a clock signal to synchronize the sender and receiver. Each byte is preceded by a start bit and followed by a stop bit to indicate the beginning and end of the byte (***FigureI.12***).



FigureI.12: Asnychronous Transmission [20]

Asynchronous serial communication protocols Communication has a range of applications, for example : Recommended Standard 232(RS-232) and Universal Asynchronous Receiver-Transmitter(UART) [15]:

- **RS-232** (Recommended Standard 232): RS-232 is a widely used serial communication protocol for communication between computers and external devices such as printers, scanners, and modems [21].

RS-485: RS-485 is a serial communication protocol commonly employed in industrial automation and control systems for communication between multiple devices over long distances [22].

I.5.2 UART Communication Protocol

UART (Universal Asynchronous Receiver/Transmitter) is a widely used serial communication protocol for transmitting and receiving data between microcontrollers and other devices (**FigureI.13**). It is supported by almost all microcontrollers and serves as a fundamental protocol in embedded systems [24].

UART is a full-duplex communication protocol, enabling simultaneous transmission and reception of data. It utilizes two communication lines: a transmit line (TX) for sending data and a receive line (RX) for receiving data [24].

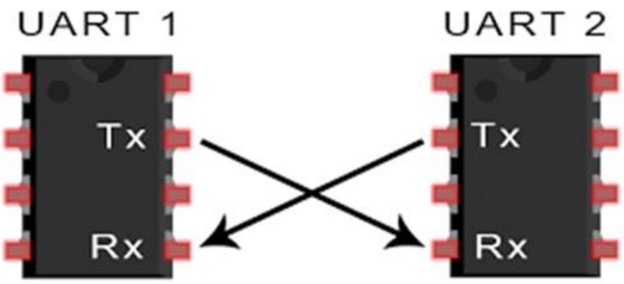


Figure I.13 : UART Comuncation Protocole [23]

Being an asynchronous protocol, UART does not rely on a shared clock signal between the transmitter and receiver. Instead, it employs start and stop bits to delineate the beginning and end of data packets. Data is transmitted in a byte-by-byte manner, where each byte is preceded by a start bit and followed by one or more stop bits [26].

UART is a versatile and reliable communication protocol extensively used in embedded systems for establishing communication between microcontrollers and various devices like sensors, motors, and displays. Its simplicity and ease of implementation, requiring only a few external components, make it a cost-effective solution for numerous applications [26].

I.5.3 Modes of UART Communication

UART communication can operate in two modes: half-duplex and full-duplex.

A. Half-Duplex Mode

In half-duplex mode, data transmission is possible in both directions (***FigureI.14***), but not simultaneously. The transmitter and receiver share the same data line, which allows for bidirectional data transfer. However, only one device can transmit or receive data at any given time. The transmitter disables its output when it receives data, and the receiver disables its input when it transmits data [23].

FigureI.14 *: HALF- DUPLEX MODE [28]*

Half-duplex communication is commonly employed in applications where data needs to be transmitted in both directions but not simultaneously, such as in walkie-talkies or remote controls.

B. Full-Duplex Mode

In full-duplex mode, data can be transmitted simultaneously in both directions(***Figure I.15***). Each device has its own dedicated data line, enabling independent transmission and reception of data. This allows for concurrent bidirectional communication without the need for switching or disabling the transmitter and receiver [23]

Full-duplex communication is utilized in applications where data needs to be transmitted in both directions simultaneously, such as in computer networks or serial ports.

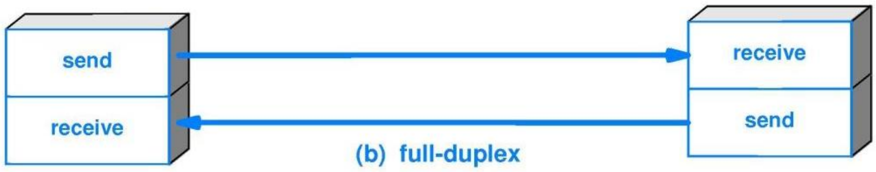


Figure I.15: FULL-DUPLEX MODE [28]

Both half-duplex and full-duplex modes have their own advantages and disadvantages. Half-duplex communication requires fewer wires and is simpler to implement. However, it is slower compared to full-duplex communication and may introduce errors if data is transmitted simultaneously in both directions. On the other hand, full-duplex communication is faster and more efficient but requires additional wires and is more complex to implement [23].

The choice of communication mode depends on the specific requirements of the application, taking into account factors such as data transfer speed, resource availability, and complexity of implementation.

I.5.4 UART Transmitter:

A digital block diagram of a UART transmitter (***Figure I.16***) typically consists of the following components :

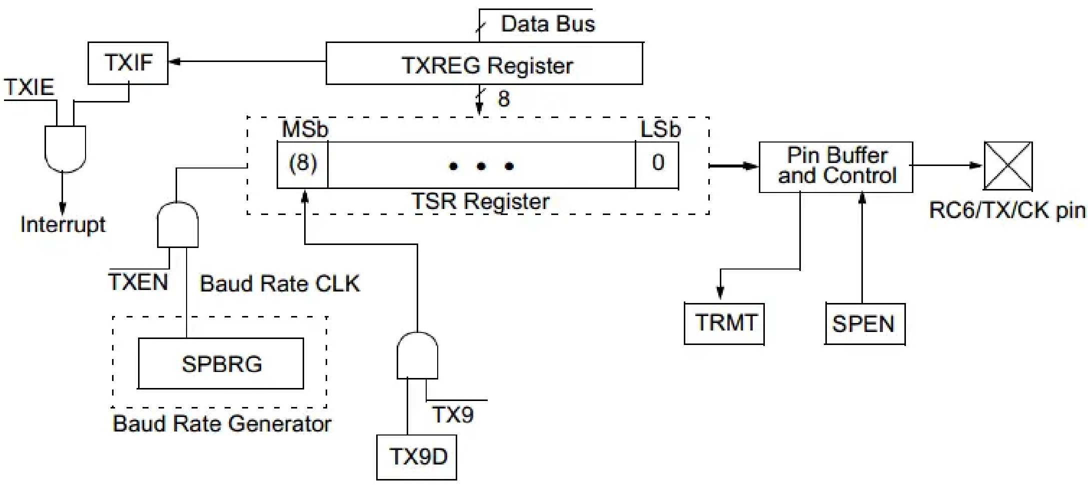


Figure I.16: Block Diagram (UART Transmitter) [25]

Data Source: The data source provides the data payload that is to be transmitted. The data can come from any digital source, such as a microcontroller or a sensor.

Data Formatter: The data formatter formats the data into the required packet format for transmission. This includes adding start and stop bits, configuring the data payload length, and setting the parity bit (if used).

Baud Rate Generator: The baud rate generator generates a clock signal that controls the rate at which the data is transmitted. The baud rate is set by the device settings and is typically between 1200 and 115200 baud.

Transmit Buffer: The transmit buffer temporarily stores the formatted data before it is transmitted. The buffer allows the transmitter to continue sending data while the previous packet is being transmitted.

Serial Interface: The serial interface is the physical interface that connects the transmitter to the data line. The interface typically consists of a transmitter output and a ground connection.

Timing and Control Logic: The timing and control logic generates the control signals that synchronize the transmitter operation. This includes generating the start and stop bit signals, controlling the transmit buffer, and synchronizing the clock signal with the data stream.
Together, these components form a complete UART transmitter that can transmit data in a serial format over a single data line. The digital block diagram can be implemented using discrete components or as a dedicated UART integrated circuit [26].

I.5.5 UART Receiver:

A digital block diagram of a UART receiver (**Figure I.17**) typically consists of the following components:
Data Line Interface: The data line interface receives the serial data stream from the transmitter. The interface typically consists of a receiver input and a ground connection.

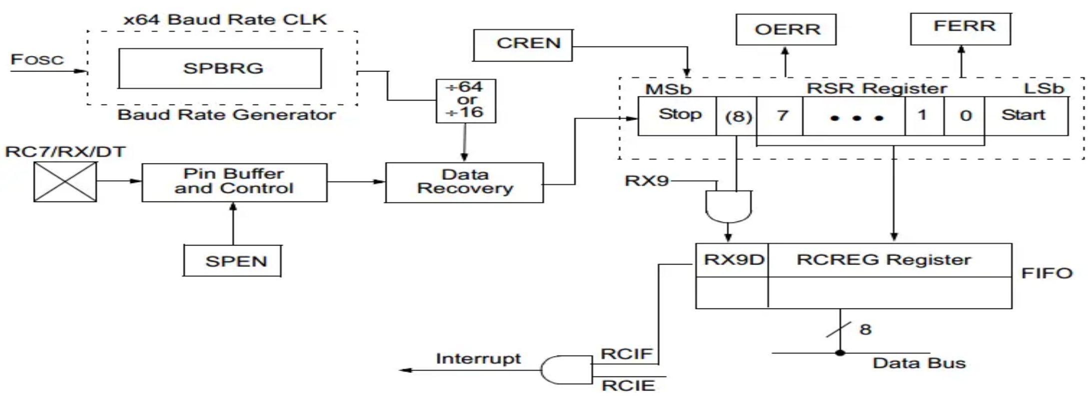


Figure I.17: Block Diagram (UART Receiver) [25]

Baud Rate Generator: The baud rate generator generates a clock signal that controls the rate at which the data is received. The baud rate is set by the device settings and must match the baud rate used by the transmitter.
Receive Buffer: The receive buffer temporarily stores the received data before it is processed by the receiver. The buffer allows the receiver to receive data while the previous packet is being processed.
Data Recovery: The data recovery circuitry decodes the received signal and recovers the original data payload. This includes detecting the start and stop bits, synchronizing the clock signal with the data stream, and validating the parity bit (if used).
Error Detection: The error detection circuitry checks for errors in the received data. This includes checking for framing errors, parity errors, and overrun errors.

Data Sink: The data sink receives the processed data payload and sends it to the destination device. The data can be sent to any digital device, such as a microcontroller or a display.

Together, these components form a complete UART receiver that can receive data in a serial format over a single data line. The digital block diagram can be implemented using discrete components or as a dedicated UART integrated circuit [26].

I.5.6 Advantages and Disadvantages Of UART Serial Communication

The UART (Universal Asynchronous Receiver/Transmitter) protocol has both advantages and disadvantages, as follows (**Table I.1)**:

Table I.1: Advantages and Disadvantages Of UART

Advantages of UART Serial Communication	Disadvantages of UART Serial Communication
Simple and Versatile	Limited Distance
Low Cost	Limited Number of Devices
Low Power Consumption	No Built-In Addressing
High Data Rates	Vulnerable to Interference
Robust	Requires Synchronization
Widely Used	
Easy to Implement	

Chapter II: Cafeteria Management system: Design and simulation

II.1 INTRODUCTION

In this chapter, we focus on the practical aspects of the project, including its real-world application and objectives. We explore the goals of the system and the impact it aims to achieve, providing a clear understanding of its purpose. Additionally, we present an initial conceptual model of the project to visually represent its design and structure. This model serves as a foundation for further development. We also explain the system management method employed to ensure efficient functioning of the system. Furthermore, we delve into project simulations, which utilize advanced techniques to mimic the project's behavior and assess its performance. Throughout the chapter, we provide valuable insights into the implementation and management of the project, aiming to offer a comprehensive understanding of its practicality and potential impact.

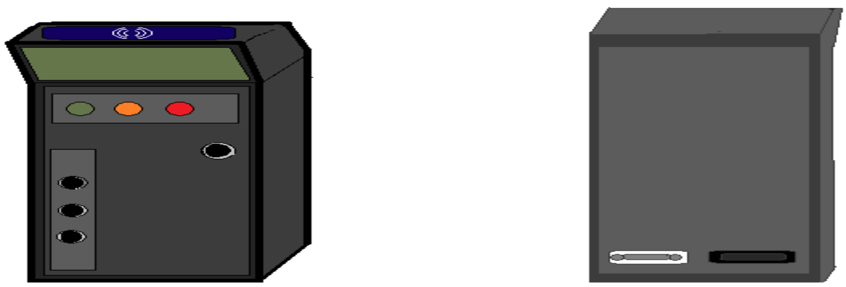
II.2 Design Cafeteria Management system

Our project aims to address one of the problems that university students face in dormitories, in general, which is the issue of dining services. Dining services are among the services that generate anger and problems for both students and staff due to poor management, such as overcrowding in queues or shortage of meal cards. How can we effectively address this problem and what are the solutions?

Our project aims to solve these problems through a radical transformation in the structure of these transactions, following three fundamental steps:

II.2.1 Authentication and Verification Machine:

The device plays a crucial role in the intended goal and facilitates the digitization of paper tickets and food cards using RFID technology. It captures the relevant information from these cards and sends it for verification and processing. Figure II.1 show an illustrative figure that represents the approximate shape of the device.

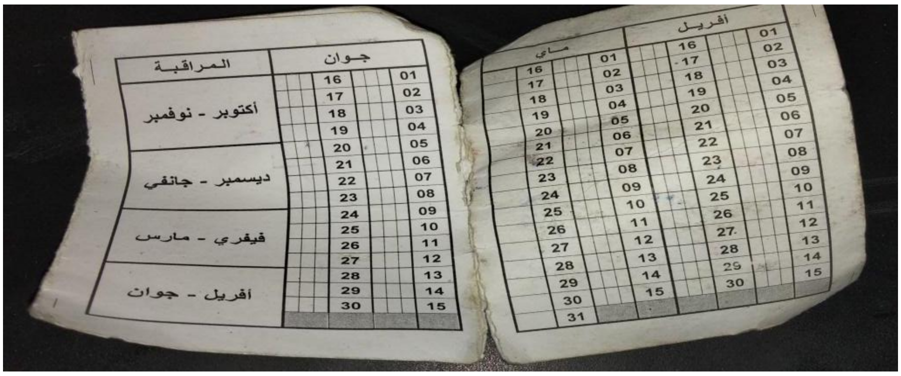


FigureII.1: Authentication and Verification Machine

To ensure that students receive their meals, several procedures need to be followed, including:

- ❖ **Meal Card:**

This card ***FigureII.2*** is essential for every student to authenticate themselves and obtain their meal. It helps determine whether the student has already taken their meal or want to repeat the meal.



FigureII.2: Meal Card

- ❖ **Verification Ticket:**

Without the ticket ***FigureII.3***, students cannot obtain their meal. After authenticating their meal card, they must also verify the ticket to receive their meal.

FigureII.3: **Verification Ticket**

Meal cards (***FigureII.2***) and verification tickets(***FigureII.3***) are paper cards that can be easily damaged throughout the year. In case of a shortage or absence of verification tickets or if a student is unable to purchase a ticket, they are denied access to their meal. The authentication machine converts these tickets into non-transferable and easily countable numeric tickets, making it convenient for students to obtain them at any time.

The problem with meal cards lies in their frequent damage due to students carrying them repeatedly, and sometimes they get misplaced. This causes issues for both staff and students. By integrating the meal card into the student's residence card, which is always with the student, authentication can be performed more quickly, eliminating the traditional and time-consuming verification process that contributes to long queues. This, in turn, reduces frustration among staff and students.

II.2.2 serving and Organizing Machine:

After the authentication process by the authentication and verification machine (***Figure II.1***), the student proceeds to obtain their meal. This stage often leads to the biggest problems among students, including conflicts arising from queue-cutting or a lack of respect for the queue order. By implementing an ordering and organizing machine ***FigureII.4***, an automated system can regulate student behavior during the meal collection process.

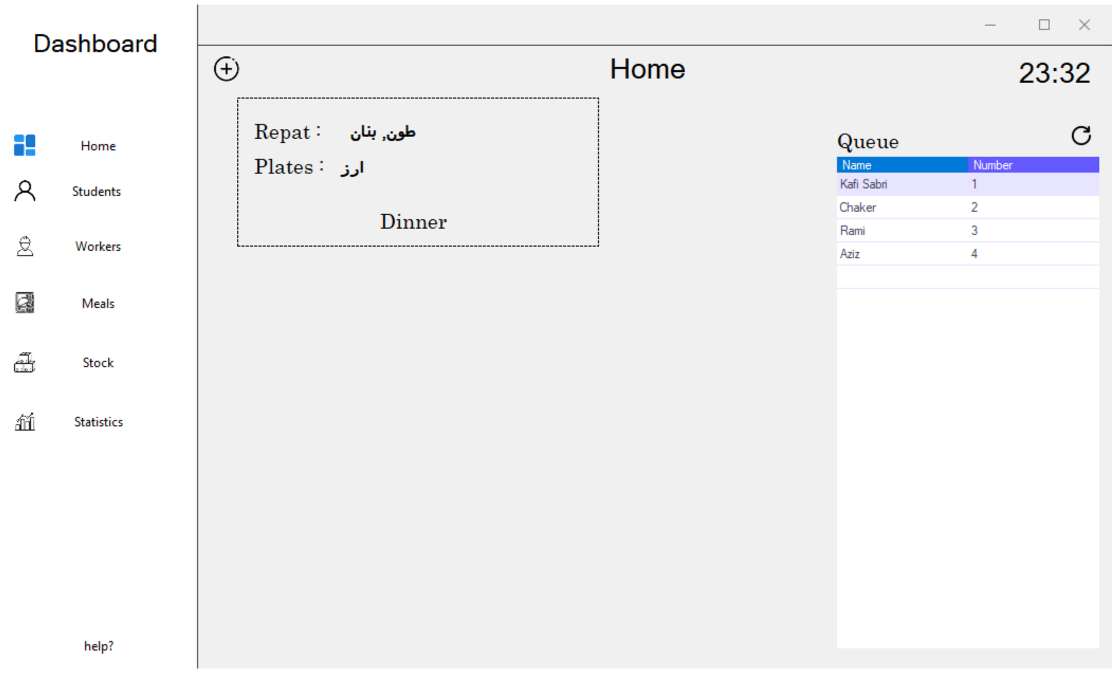
Upon the initial authentication, the machine organizes the students automatically, ensuring that they can only receive their meal when it is their turn in the queue. When the student reaches the machine to collect their meal, their verification is checked once again, ensuring compliance with the queue order. This eliminates the possibility of any student bypassing the queue or disrespecting the order.

By implementing this machine, student behavior can be better managed and conflicts arising from queue-related issues can be significantly reduced. This helps create a more orderly and respectful environment during meal collection

Figure II.4: Ordering and Organizing Machine

II.3 Management and Administration Software:

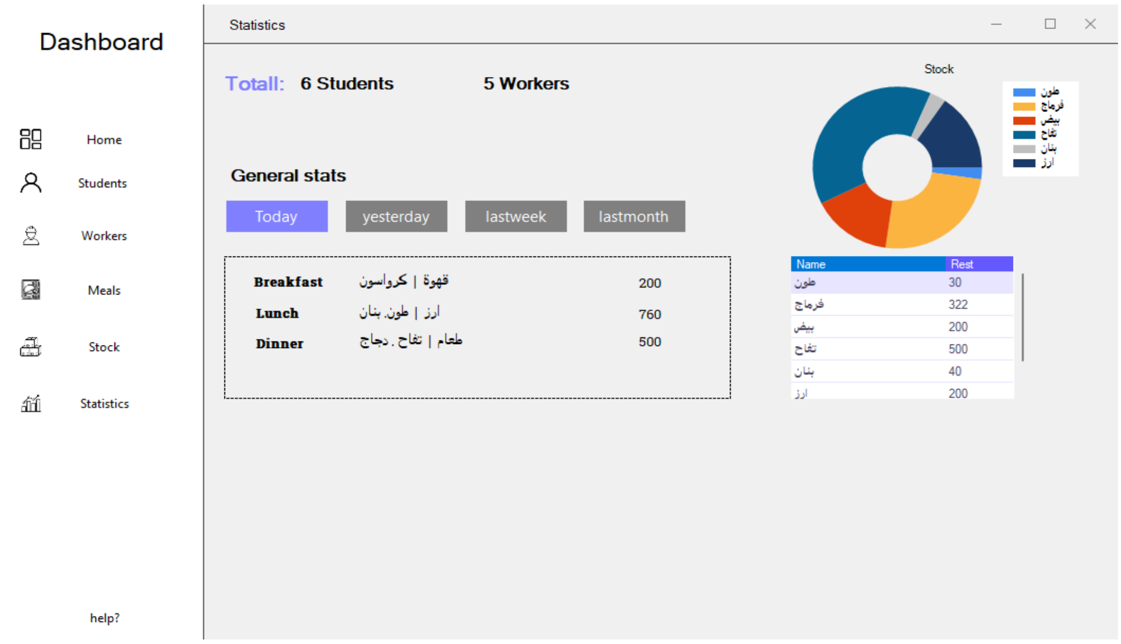
The software serves ***FigureII.5*** as the focal point and backbone of this project. After the ticketing and verification process, the machine checks the student's information that requires authentication.



FigureII.5: The Management and Administration Software

Once the student's information is validated, it is forwarded to the ordering machine for queue placement. The software performs several tasks, including:

- ❖ **Verification**: After the initial authentication, the software verifies the accuracy of the student's information.
- ❖ **Queue Management**: After the verification process, the software sends the student's information to the ordering machine to assign them a place in the queue**.**
- ❖ **Statistics:** The software keeps track of the number of students who have received their meals and those who haven't ***(Figure II.6)***. It also records the types of meals distributed (hot or cold) for efficient inventory management. The software calculates the number of tickets sold and the number of tickets verified.



FigureII.6: Statistics interface

- **Inventory Management:** The software manages the inventory by tracking the inputs, outputs, and remaining stock.

After performing reconciliation and statistical analysis, the software automatically sends this information to the relevant authorities periodically, either daily or monthly, for further study and performance evaluation of each residence facility.

Additionally, the software provides various features, such as statistics on the number of students who have received meals versus those who haven't, categorized by medical requirements or meal types. This allows for future projections to ensure efficient meal distribution and minimize waste.

Overall, this software plays a vital role in managing and administering the dining services, ensuring accurate verification, efficient queue management, and comprehensive data analysis to optimize the overall dining experience and reduce food waste.

II.4 Design of the authentication machine

In order to design the authentication machine, we will go through several steps:

-The controller used in the device is of the PIC 16F877A type ***(Figure II.7)***. It is characterized by the following features: it contains counters and a sensor for comparison, as well as another sensor for pulse modulation. It also provides protocols for synchronous data transmission, including SPI and I2C, as well as asynchronous data transmission through UART. Additionally, it consumes low power and operates within a voltage range of 5V to 2V.

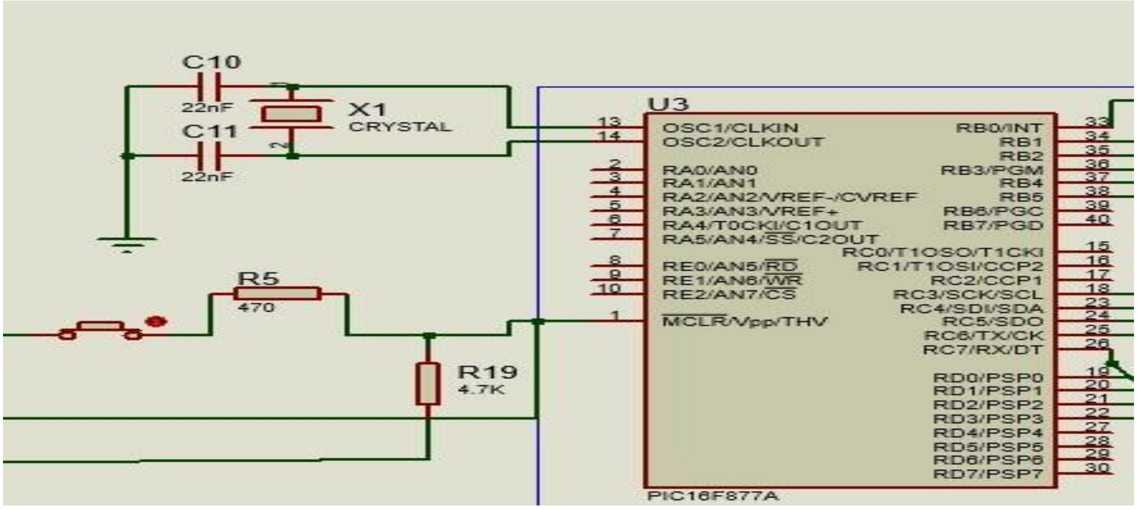


Figure II.7: simulation of Microcontroller [18]

It is connected to an external pulse generator, and this circuit improves the generation of pulses to control the data transmission speed in the device, resulting in faster data transfer. This circuit is connected to pins 13 and 14 of the controller, which are designated for interfacing with an external power source such as a crystal.

As for input number 1, which is used for supplying power to the controller, it can be connected to a power source through a battery or directly via electrical wires. Additionally, a reset switch can be used for restarting the controller.

-This diagram ***(Figure II.8)*** is designed to create a power supply for the device, and it consists of three stages:

The red zone: Power is taken from an AC source of 220V and converted to DC power with a peak ranging from 16V to 17V.

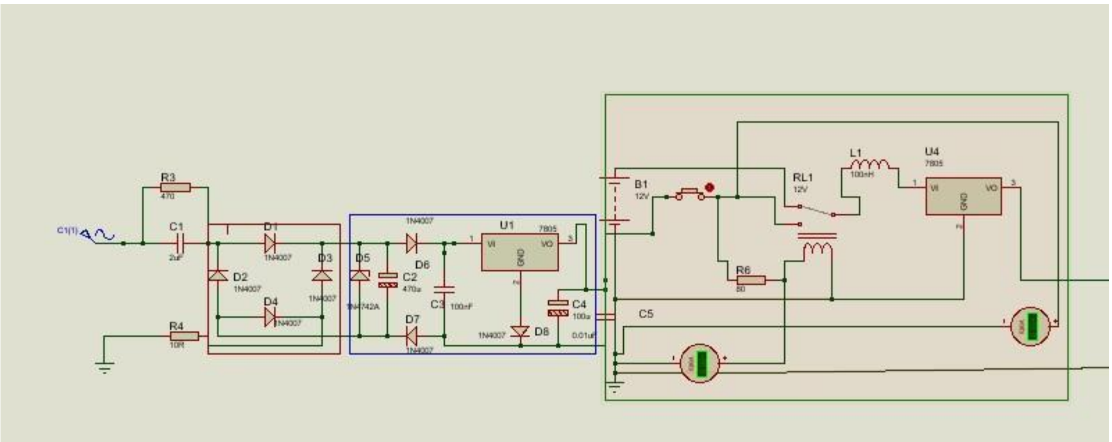


Figure II.8: Simulation of power source

The blue zone: The current is then converted from 17V to 12V using a voltage regulator.

The green zone: After converting the current from AC to DC, it becomes usable as a power supply for electrical devices. We have added a 12V battery to ensure device operation in case of a power outage. The battery is connected to the power source through a relay, allowing automatic and seamless switching between power sources without interrupting the device. The output of the relay is connected to a 5V voltage regulator, ensuring the appropriate voltage for the controller.

-The student's information, including name and ticket number, is displayed on a display screen ***(Figure II.9)***. The screen has 7 data inputs, which requires the same number of inputs on the controller. To reduce the number of inputs, we used PCF8574, which operates using the I2C protocol and requires only two inputs: SDA (data transfer input) and SCL (clock control input).

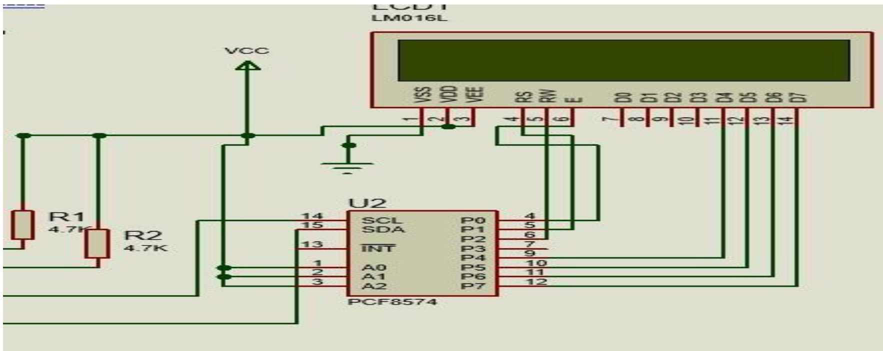


Figure II.9: *simulation of LCD*

The remaining four inputs on the display screen are connected to four inputs on the PCF8574 to transfer data using a 4-bit protocol over this interface

-To authenticate a student, their student ID card needs to be swiped to obtain their unique ID number. This process can be accomplished using an element that operates on the SPI protocol (***Figure II.10)***. The SPI protocol is a versatile protocol that allows communication with multiple devices simultaneously.

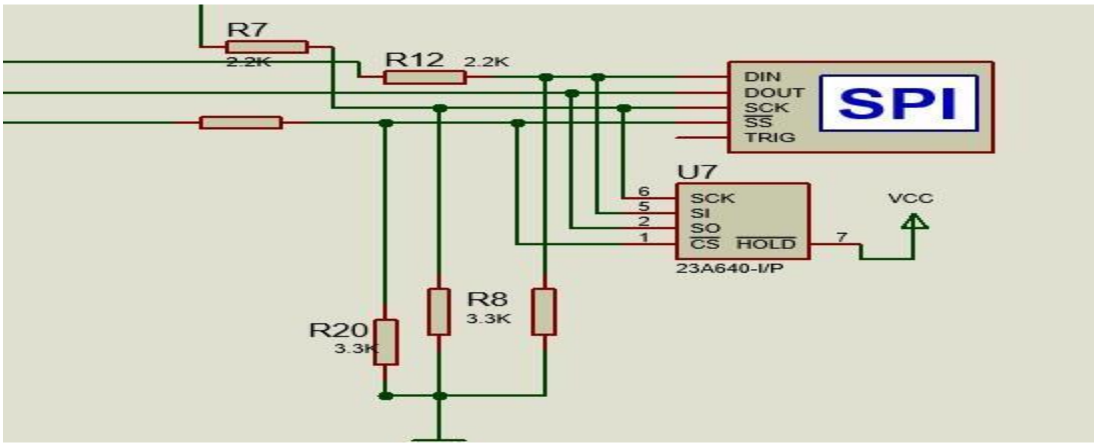


***Figure II.10**: simulation of CARD RFID*

It is known for its low power consumption. The element is connected using the SDA (data) and SCL (clock) inputs for synchronized data transfer, as well as the SDO input for receiving data. The CS input is used to select the recipient device.

-In this device, we observe the use of two protocols that share the same inputs on the controller, namely SDA and SCL. The element used for controlling the signal transmission in this scenario is a multiplexer (***Figure II.11)*** . In our device, we will utilize this element as a switching mechanism.

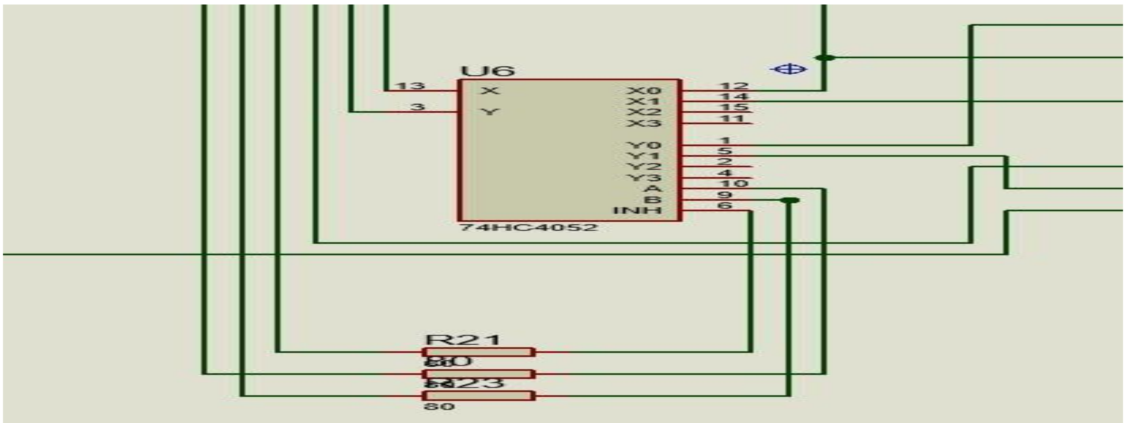


***Figure II.11**: Simulation of* multiplexer *74HC4052*

The multiplexer (***Figure II.11)*** has four channels and two inputs to determine which channel is open for transmission. The SPI and I2C protocols are connected to separate channels, and their control is governed by inputs A and B, depending on the program's requirements.

-The device features a digital ticket selling process, which is carried out through the buttons shown in (***Figure II.12)***. Each button has a specific value that is added to the student's database.

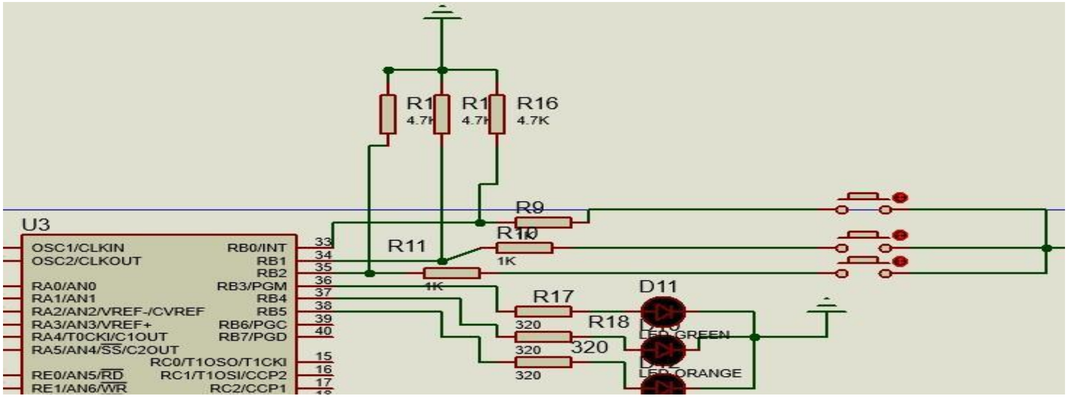


Figure II.12: Simulation of control and results

The purpose of the illuminated lights depicted in the design is to indicate the status of the student, whether they are present in the database or if they have the required number of tickets.

- In the end, the final design of the authentication device is show in ***Figurer II.13*** :

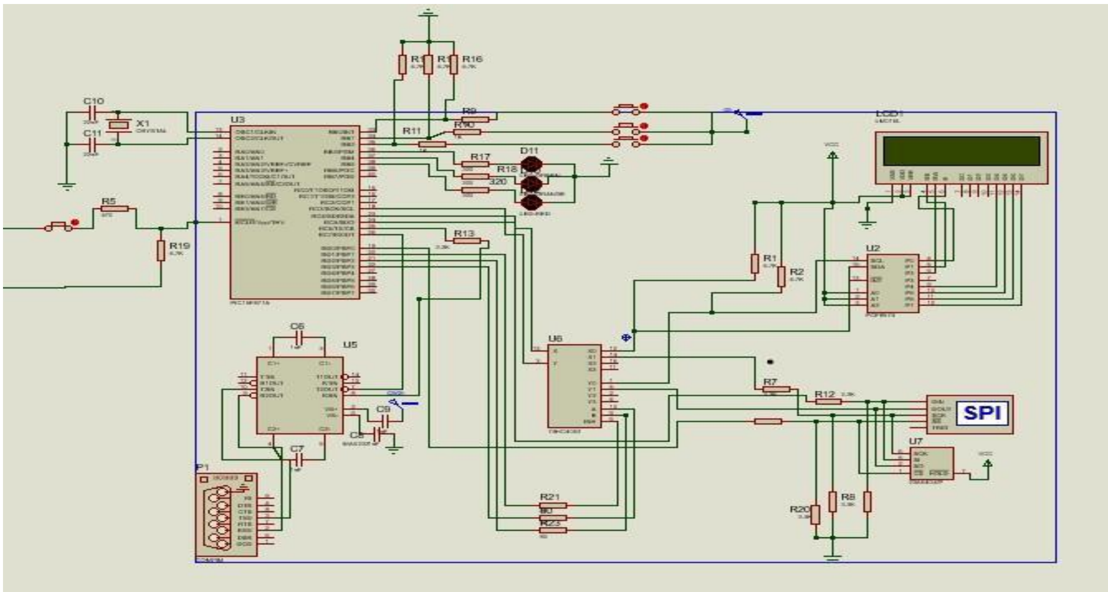


Figure II.13: Simulation of Authentication Machine.

II.5 Design of the Serving Machine

After the authentication process, the meal serving takes place. This process is carried out by the student swiping their card using an RFID module. The system then checks the student's position in the queue and displays their status using a red light and alerts with a buzzer in case of queue-jumping, while a green light indicates their position in the queue is valid.

Once the student's queue status is confirmed, the dish is automatically presented through a conveyor belt. The role of the sensor is to detect the status of the dish, whether it has been taken or left untouched (***Figure II.14***).

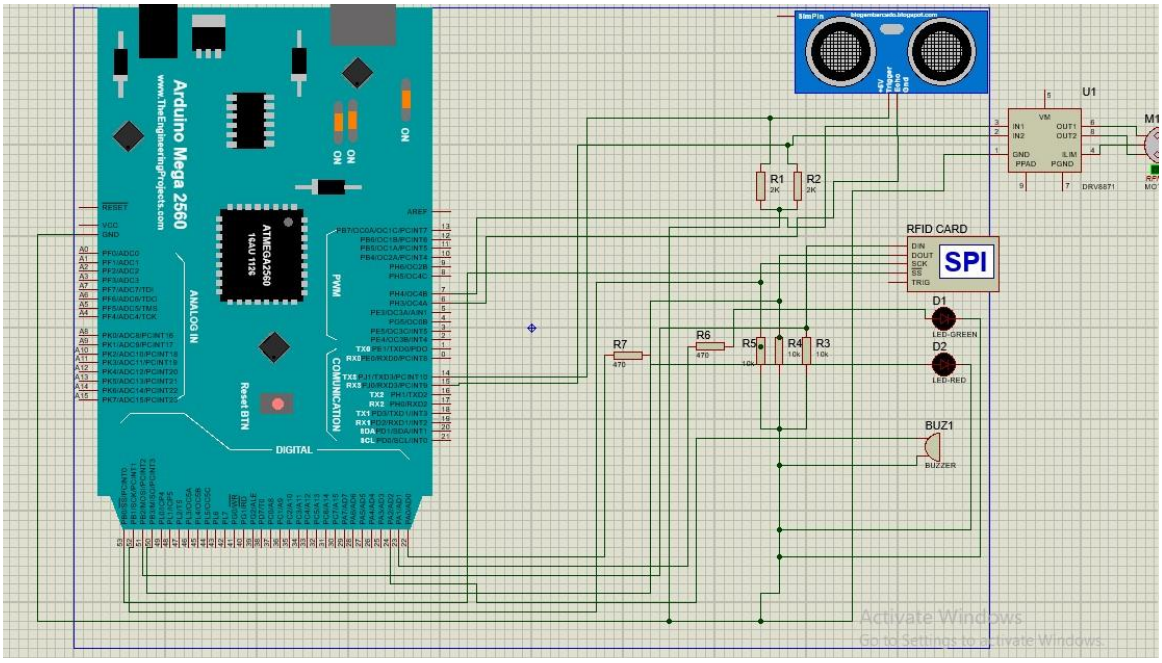


Figure II.14: Simulation of Serving Machine

Chapter III: Software Application of System

III.1 Introduction

In this chapter, we will delve into the practical study of the project, the system's objectives, the initial conceptual model of the project, an explanation of the system management method, and project simulations

We will explore the practical study of the project, examining how it can be applied in a real-world context, as well as identifying its goals and the impact it aims to achieve. Additionally, we will provide an initial conceptual model of the project, offering a preliminary visualization of its design. We will also explain the system management method, detailing how the system is operated and organized. Finally, we will discuss project simulations, which involve using simulation techniques to simulate the project's functionality and behavior

III.2 The software application interface:

Defining the purpose and scope of a desktop application is a crucial step in its development process. The purpose refers to the reason behind creating the application, while the scope encompasses its features and capabilities. It is essential to clearly define these aspects to ensure that the application meets the needs of its intended users. This involves understanding the target audience and their requirements. A project proposal or requirements document should outline the purpose, scope, features, user interface, and functionality of the application, as well as any constraints such as time and budget limitations. By doing so, developers can create a roadmap that guides the development process and ensures the application's timely delivery with the desired features and functionality.

III.3 Importance of the program in the system

The administrative system of the university restaurant greatly benefits from a program that accelerates inventory management and minimizes waste. The program provides detailed reports on stored materials, reducing production costs and enhancing productivity. It also simplifies the creation of general statistics, allowing supervisors to obtain accurate reports effortlessly. By eliminating paper tickets, the program saves time and effort in ticket calculation and meal management, facilitating quicker and more effective sales and meal management processes. Additionally, the program retains old information for analysis, enabling the identification of patterns, trends, and informed strategic decisions to improve the restaurant's service..

III.4 Overview of the desktop application architecture

III.4.1 High-Level Architecture: Components & Interactions

When designing a desktop application, considering the architecture is crucial. The architecture refers to the overall structure and organization of the software system, including its components and interactions. Typically, a desktop application architecture consists of layers such as the presentation layer, application layer, and data layer. The presentation layer handles the user interface, the application layer contains the business logic, and the data layer manages data storage and retrieval. Additional layers may be added based on specific requirements, such as real-time data processing. These layers interact with each other to form a cohesive architecture, allowing the desktop application to function effectively. For instance, user actions in the presentation layer can trigger business logic in the application layer, which may interact with the data layer to retrieve or update information [27].

The architecture of a desktop application is an important consideration when developing software. By understanding the various components and how they interact with each other, developers can design an architecture that meets the specific requirements of the application. This can help to ensure that the application functions as intended and provides a positive user experience.

III.5 Design Decisions for Architecture: Tech & Framework Choices

When designing desktop application architecture, there are many factors that influence the choice of technologies and frameworks used. Some of the main considerations include the functionality and performance requirements of the application, the development team's expertise and experience, and the availability and suitability of the technology or framework for the project. Additionally, factors such as cost, licensing, and compatibility with existing systems may also play a role in the decision-making process , to ensure that the desktop application architecture is well-suited to the project's requirements, it is important to carefully consider the available options and evaluate them based on a set of criteria. For example, when choosing a programming language, developers may consider factors such as the language's syntax, its performance, and its popularity in the industry. Similarly, when selecting a database management system, the decision may be based on factors such as scalability, security, and ease of integration with other systems.

the design decisions made for the desktop application architecture should reflect a balance between meeting the functional requirements of the project and ensuring that the architecture is scalable, maintainable, and cost-effective. By carefully evaluating the available options and selecting technologies and frameworks that are well-suited to the project's needs, developers can ensure that the desktop application architecture is a solid foundation for building a high-quality software system

III.6 Description of the programming and database tools

The development of a desktop application requires the use of various programming languages, libraries, and frameworks to create a robust and feature-rich software system. In the case of the desktop application being discussed, the programming language used is C#, which is widely used for Windows application development due to its ability to create native code that can take full advantage of the Windows operating system features [34]. In addition to the programming language, several libraries and frameworks were used to enhance the functionality of the desktop application. One of the most important libraries is .NET, which is a development framework that provides a comprehensive set of libraries and tools for developing Windows applications [29]. Other libraries used in the development of the desktop application include Windows Forms, a graphical user interface (GUI) library, and Windows Presentation Foundation (WPF), a UI framework that allows for the creation of rich, multimedia applications [29].

The database management system used in the development of the desktop application is Microsoft SQL Server. SQL Server is a relational database management system (RDBMS) developed by Microsoft and widely used for enterprise-level databases due to its scalability and robustness [30]. The integration of the SQL Server into the desktop application was achieved through the use of Entity Framework Core, a lightweight, open-source ORM framework that simplifies database programming and provides a higher level of abstraction for working with databases [31].

III.7 DBMS integration in the desktop application.

The database management system (DBMS) used in the development of the desktop application is a crucial component that provides the necessary infrastructure for storing and managing data. In the case of the application being discussed, the DBMS used is Microsoft SQL Server(***Figure III.1)***.

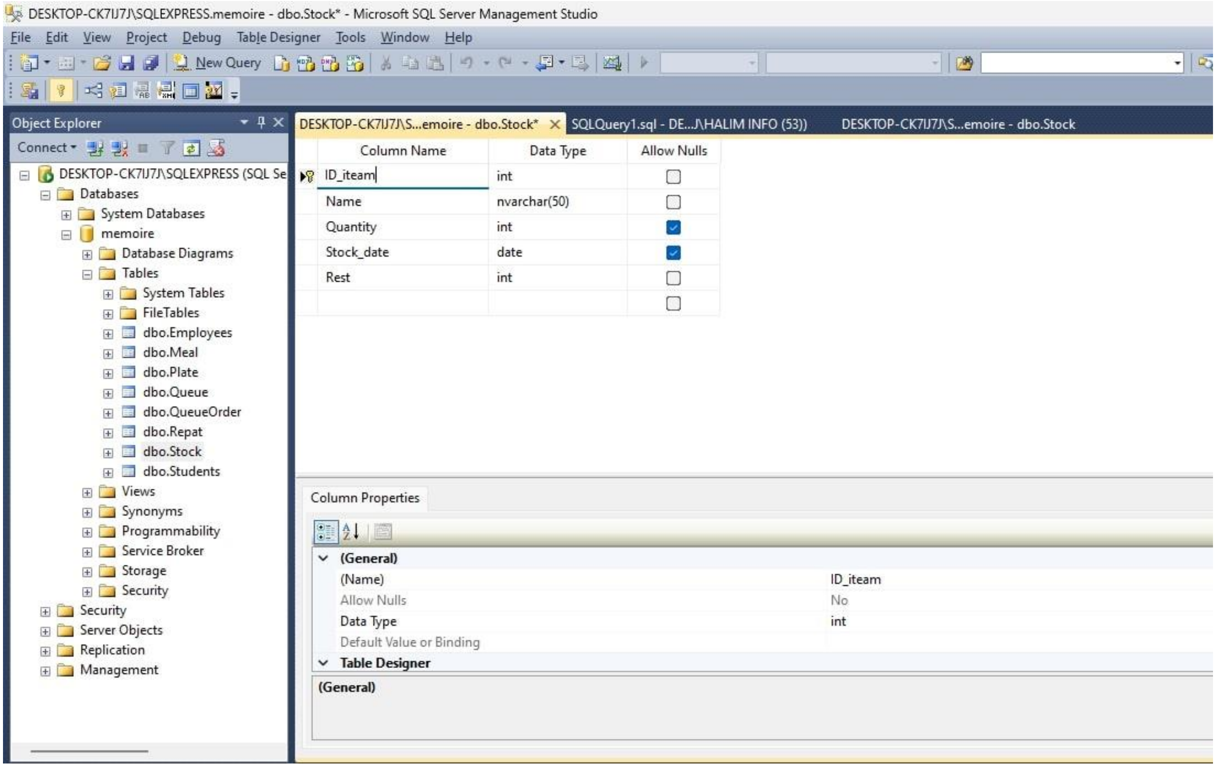


Figure III.1: SQL Server

Microsoft SQL Server is a relational database management system that is widely used in enterprise-level applications due to its robustness, scalability, and security features. It provides a comprehensive set of tools and features that enable efficient management of large amounts of data.

To integrate the SQL Server database into the desktop application, the development team used a variety of tools and techniques. The primary tool used was the Entity Framework, which is an object-relational mapping (ORM) framework that allows developers to interact with databases using object-oriented code. The Entity Framework provides a high-level abstraction of the database schema, making it easier to work with databases in object-oriented programming languages such as C# [32].

The Entity Framework also provides a range of features for managing the database schema, such as migrations and code-first development. These features enable developers to evolve the database schema over time without the need for manual SQL script modifications. In addition to the Entity Framework, the development team may have used other tools and techniques to integrate the database into the desktop application, such as stored procedures, triggers, and views. These database objects can be used to enforce business logic and improve application performance by reducing the amount of data that needs to be transferred between the application and the database [33] .

III.8 Provide details of IDEs used for development.

The primary tool used for development was Visual Studio 2022(***Figure III.2)*** , a popular integrated development environment (IDE) used for creating applications using the C# programming language[35]. Visual Studio 2022 provided a comprehensive set of tools, including code editors, debugging and testing tools, and other features that helped streamline the development process.

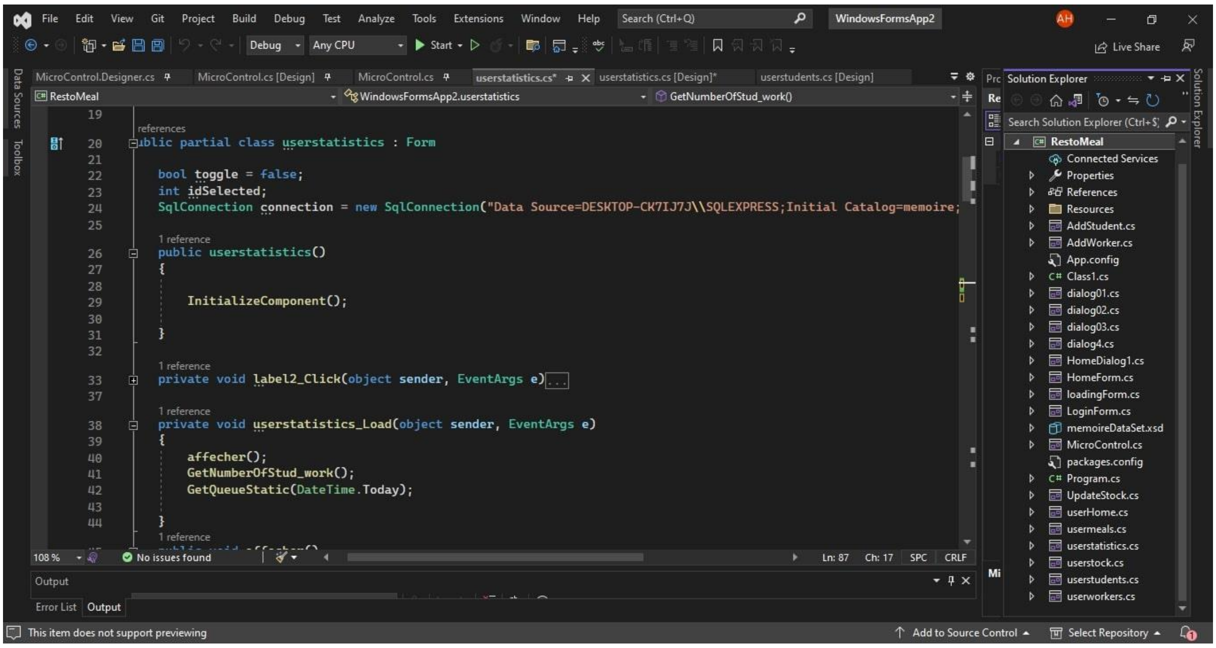


Figure III.2: Visual Studio

We also used SQL Server as the database management system for the application. SQL Server is a popular relational database management system developed by Microsoft, and is widely used in enterprise-level applications. The integration of SQL Server with Visual Studio was seamless, allowing for easy database management and integration with the application.

The development process was stream lined, and the testing and debugging process were made easier with the use of these tools and IDEs.

III.9 Implementation

III.9.1 Detailed description of the programming process

The development process of desktop applications involves several stages, including requirements analysis, design, coding, and testing.

During the requirements analysis stage, we gathered and documented the functional and non-functional requirements for the desktop application. This involved understanding user needs and translating them into software requirements.

After defining the requirements, we proceeded with the design stage, where we created a high-level design of the application's structure and functionality. This stage included creating wireframes, user interface models, and architectural diagrams to provide a blueprint for the development process.

The coding stage involved writing the actual code for the desktop application using the C# programming language and various libraries and frameworks, as mentioned earlier. We utilized Visual Studio 2022 as an integrated development environment (IDE) to code, test, and debug the application.

We also focused on key aspects such as statistics, where we implemented a window (***Figure III.3)*** that provides statistics on remaining stock and consumed meals during different time periods.

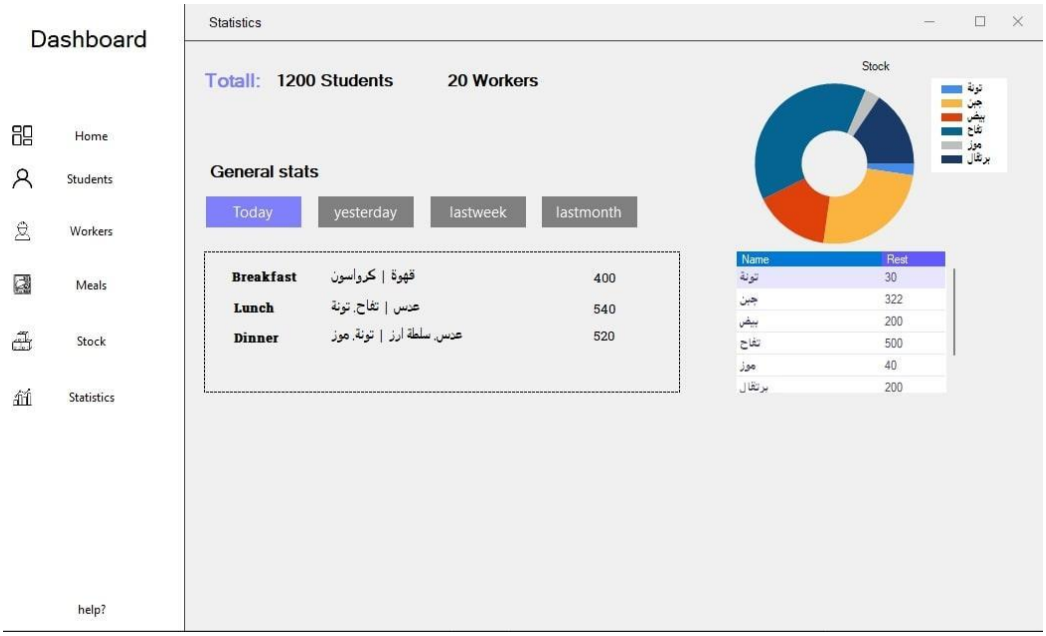


Figure III.3: statistic interface

Additionally, we emphasized inventory management, which allowed full control over inventory operations, including adding, modifying, and deleting items (***Figure III.4)***. Automatic inventory updating was implemented as a fundamental feature, where the inventory is automatically updated every day based on the number of consumed meals deducted from the stock. Overall, the development process involved thorough requirements analysis, careful design planning, and precise coding to ensure the successful implementation of the desktop application.

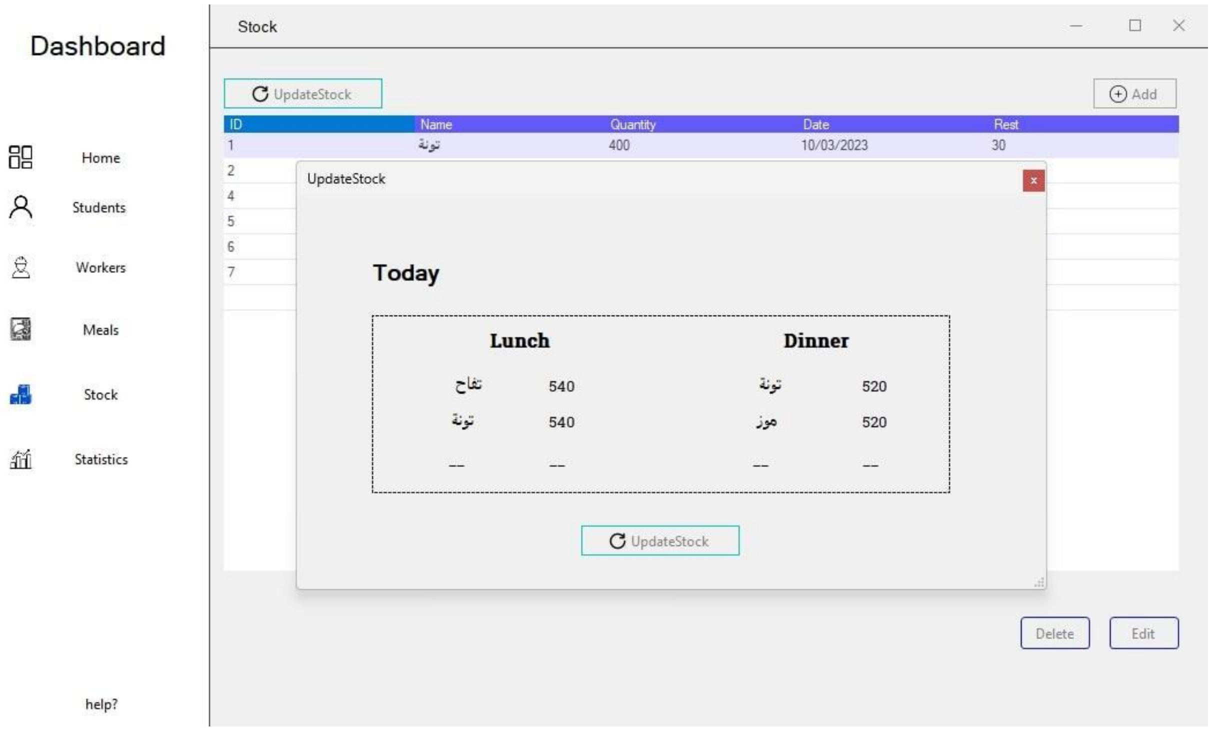


Figure III.4: update stock

III.9.2 Overview of the database (Diagrammes) design and implementation:

Figure III.5 illustrates all the tables present in the program's database. This image helps in understanding the organization and structure of data in the database. Each table is represented by a rectangle containing the necessary fields for each table, along with the primary keys.

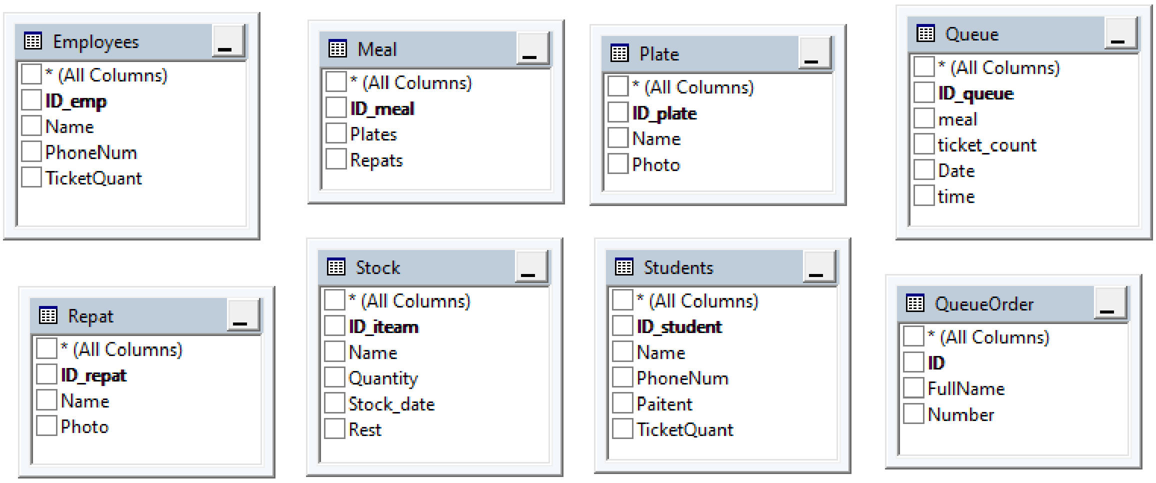


Figure III.5:Tables Diagram

Databases consist of numerous tables that can be organized in a diagram for easier understanding .One of these diagrams is the class diagram as in (**Figure III.6)**, which describes the tables in the database as objects and how they are related to each other. This diagram helps in understanding the organization and relationships between tables, making it easier to design and improve the database's performance.

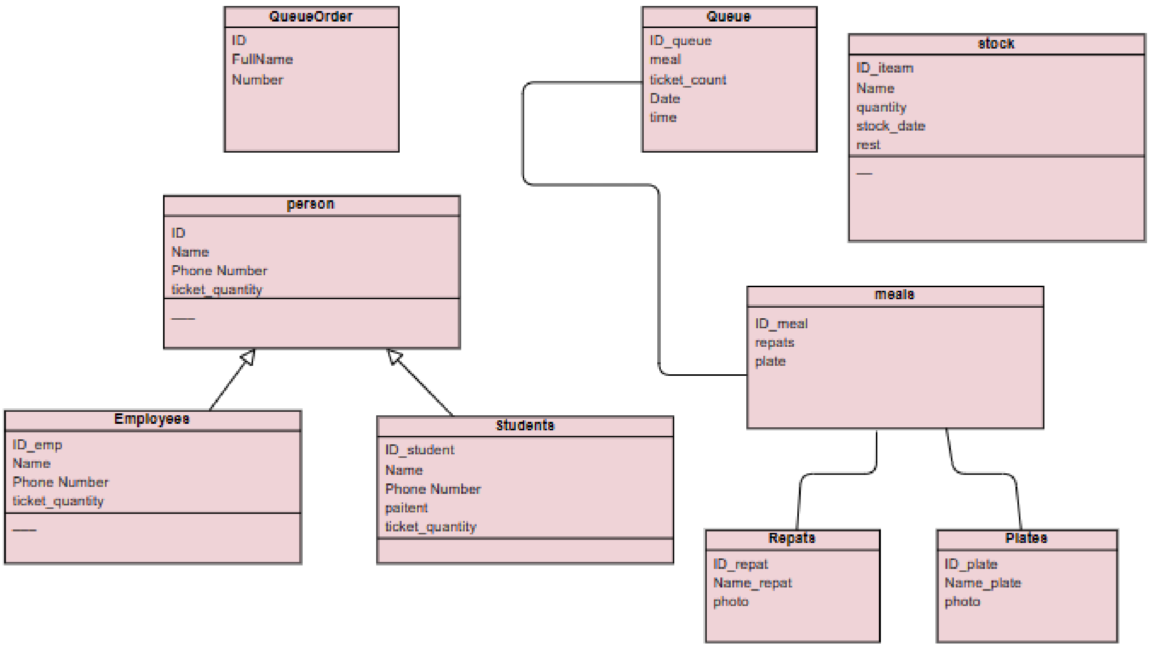


Figure III.6: Class Diagram of the Database Tables

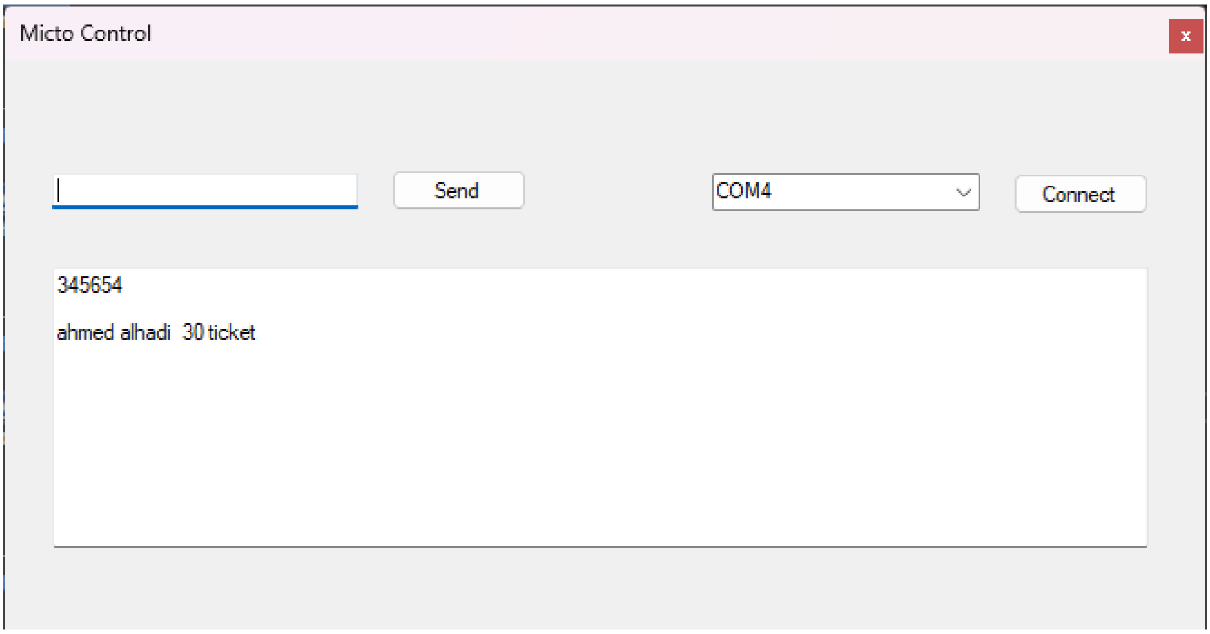
The class diagram is based on the Object-Oriented Programming (OOP) technique and represents each table as an object consisting of properties and methods that can be applied to it. The relationships between tables in the diagram are expressed through the links between the objects representing the tables.

The class diagram simplifies the process of modifying and maintaining databases, as the main tables and their relationships can be clearly and simply defined, reducing the number of errors that may occur during modification.

(Figure III.6) shows a simplified class diagram that does not include some of the features of class diagram.

III.9.3 Explanation of the data flow between the application and the materials

Micro Control is a software application developed using Visual Studio and C#, specifically designed to create a Serial Monitor (***Figure III.7)*** that enables the connection between electronic devices and computers for data transmission, reception, and analysis. The program typically utilizes the COM4 port for connection purposes.



- **Figure III.7**: Serial Monitor

Also, the card is used to add students to the database by uploading the ID that is stored on the card**.**

Updating the number of student and employee cards is done through the **ticket reader device**, where the connection is made by sending the ID and the number of added tickets to the program through the microcontroller. The ID and number of tickets are extracted and the data is automatically updated using SQL commands as appear in ***Figure III.8.***

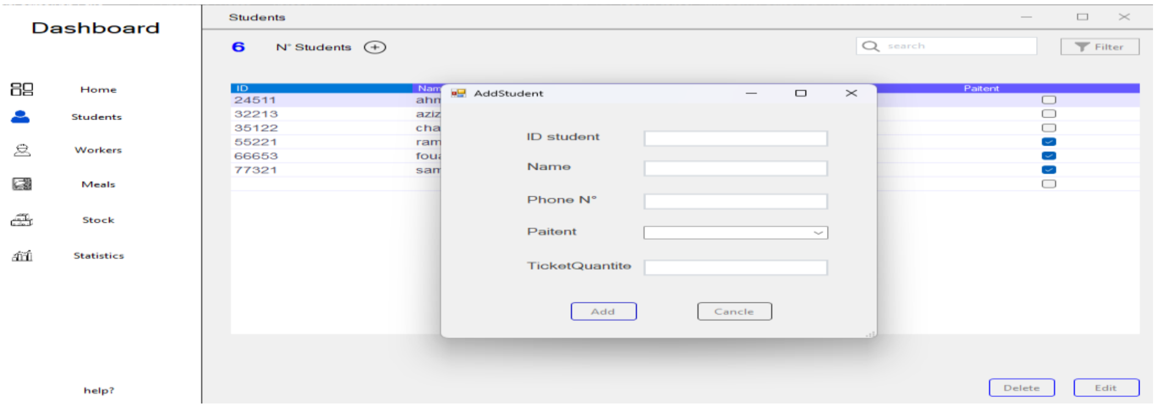
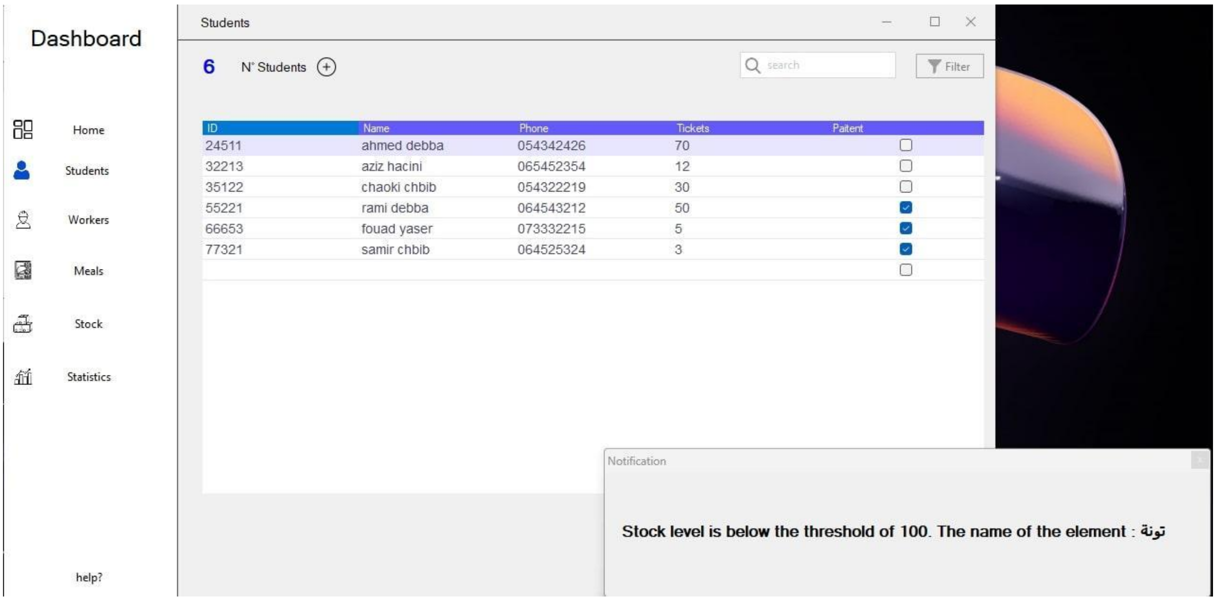


Figure III.8: Add Student Interface

The ***Figure III.9*** represents student data and the number of cards for each student, which changes automatically when updated.



. ***Figure III.9****: students page information and notification*

This is a description of a process whereby a requester scans their RFID card on a device when they join a queue. At this point, the requester's ID is sent and received by the micro controller then the computer, and the requester is automatically added to the queue. The queue can also be reset when finished. This is represented in ***Figure III.10***

Figure III. 10: Dashboard and queue

General conclusion

General conclusion

Quarrels, ticket shortages, and restaurant overcrowding are all problems faced by the food sector. To solve these issues, we have introduced a restaurant management system, a digital system aimed at reducing and controlling these problems.

Chapter 1 introduces the theoretical study of essential components in product production, including controllers, RFID technology, and conveyor belt motors. It explores microcontrollers as controlling elements in embedded systems and discusses RFID technology and its applications. The chapter also addresses challenges associated with RFID implementation.

Chapter 2 focuses on the practical aspects of the project, presenting the authentication and verification machine, ordering and organizing machine, and the management and administration software. It highlights the real-world application and objectives of these components, emphasizing their impact on the meal collection process.

Chapter 3 delves into the practical study of the project, providing insights into its objectives, conceptual model, system management method, and project simulations. It emphasizes the significance of the software application interface and the desktop application's role in the university restaurant's administrative system. The chapter details the architecture, programming and database tools, database management system integration, and implementation process. It also explains key features related to statistics and inventory management and showcases the data flow between the application and materials.

Generally, the goal of the restaurant management system project is to develop the restaurant services sector through the application that relies on digitizing this sector. It will make the operations digital, from ticket purchase to confirmation and recording, as well as maintaining a special inventory record to ensure constant budget monitoring. The application will directly transfer this information to the relevant party in the future. Our aim is to add a set of devices to enhance this system. These devices will handle certain tasks such as authentication and will also introduce new tasks like automated food delivery. Additionally, they will establish a communication network between establishments to facilitate the transfer of information.

Bibliographies

[1]https://www.techtarget.com/iotagenda/definition/microcontroller definitionmicrocontroller (MCU) byben Lutkevich,

[2] datasheet of pic 16F877A

[3] موسوعة المكروكونترولر ل م.حمدي سلطان عبد الخالق

[4]https://hardwarebee.com/types-of-microcontrollers/

[5]https://www.electronicsandcommunications.com/2021/04/applications-of-microcontrollers.html

[6]Mohammadamin Hosseinifard ,Salam Alzubaidi ,Andrea Michel ,Gualtiero Fantoni" RFID Technology as a Low-Cost and Passive Way to Digitize Industrial Analogic Indicators" https://www.mdpi.com/2076-3417/12/3/1451

[7] studycorgi.com/radio-frequency-identification-technology-concepts" Radio Frequency Identification Technology Concepts"

[8] https://www.shopify.com/retail/rfid-technology

[9] https://www.integrasources.com/blog/differences-between-rfid-and-nfc-systems-their-applications-and-alternatives/

[10]. Fitzgerald, A. E., Kingsley, C., Umans, S. D., & Kusko, A. (2016). Electric Machinery (7th ed.). Mcgraw-Hill Education.

[11]https://www.etechnophiles.com/working-principle-of-ac-motor-with-basics-and-construction/motor/

[12] https://www.etechnog.com/2019/02/actual-difference-between-synchronous.html

[13]. Boldea, I., & Nasar, S. A. (2017). Electric Drives (3rd ed.). CRC Press.

[14] https://pi4j.com/documentation/io-examples/serial/ [14]

[15] HELEN DONELAN,KAREN KEAR,MAGNUS RAMAGE ,ONLINECOMMUNICATION AND COLLABORATION:A READER, Retrieved from https://books.google.com/books?Hl=ar&lr=&id=DDI_xtmxi3mc&oi=fnd&pg=PA83&dq=related:eaeb7x4db3kj:scholar.google.com/&ots=3HD8Tug4mc&sig=hyha0jmtkzpeh8ntp3qtvagtay4

[16]https://www.electricaltechnology.org/wp-content/uploads/2020/05/Synchronous-Transmission.png

[17] I 2C-bus specification and user manual. Retrieved from ht09tps://www.nxp.com/docs/en/user-guide/UM10204.pdf
[18] MPC5121e Serial Peripheral Interface (SPI) Retrieved from .https://www.nxp.com/docs/en/application-note/AN3904.pdf
[19].usb IJRET: International Journal of Research in Engineering and Technology Retrieved from
[20]https://www.electricaltechnology.org/wpcontent/uploads/2020/05/Asynchronous-Transmission.png
[21].The RS-232/DALI Bridge Interface, Microchip relative fromhttps://ww1.microchip.com/downloads/en/Appnotes/00811a.pdf
[22].Hein Marais,RS-485/RS-422 Circuit Implementation Guide relative from https://atnel.pl/download/poradniki/rs485/AN-960.pdf
[23]https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-UART-Basic-Connection-Diagram.png
[24] https://www.infineon.com/dgdl/Infineon-Component_UART_V2.0-Software%20Module%20Datasheets-v02_05-EN.pdf?Fileid=8ac78c8c7d0d8da4017d0e7f69251131
Figure [25]: microchip PIC16F87XA Data Sheet
[26]Uart Basics of UART Communication From: Circuit Basics https://cs140e.sergio.bz/notes/lec4/uart-basics.pdf
[27] S. Ambler and P. J. Sadalage, Refactoring Databases: Evolutionary Database Design. Addison-Wesley Professional, 2006.
[28]:https://www.blackbox.fr/_appdata/cms/Default%20pages/techinfo/BBE/BBE_
[29] NET Documentation, Microsoft Docs, https://docs.microsoft.com/en-us/dotnet/
[30] devexpress. (n.d.). Data visualization controls. Retrieved from https://www.devexpress.com/products/net/controls/winforms/data-visualization/
[31] Bouncy Castle. (n.d.). Bouncy Castle cryptography apis. Retrieved from https://www.bouncycastle.org/
[32] "Entity Framework Core." Microsoft Docs, Microsoft, 2021, docs.microsoft.com/en-us/ef/core/.
[33] "SQL Server - Relational Database Management System (RDBMS)." Microsoft SQL Server, Microsoft, 2021, www.microsoft.com/en-us/sql-server/sql-server-downloads.
[34] C# Programming Guide, Microsoft Docs, https://docs.microsoft.com/en-us/dotnet/csharp/

[35] Microsoft. (n.d.). C# programming guide. Retrieved from https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/

Résumé :

Notre projet vise à améliorer le service de restauration dans les résidences universitaires en développant ce système qui organise les files d'attente et numérise les transactions afin d'éliminer les cartes en papier et de les remplacer par des cartes numériques. Cela permet d'éviter la surpopulation et les conflits, en plus de faciliter les opérations statistiques liées au processus. Nous avons réussi à construire avec succès une base de données spécifique au système, où toutes les informations seront stockées et seuls les personnes autorisées auront accès. De plus, nous avons conçu des schémas préliminaires pour les appareils qui soutiendront les performances de ce système, permettant ainsi un meilleur contrôle du flux des files d'attente.

Mots-clés : Service de restauration, Organisation, Collecte de données, Cartes numériques, Communication série, Communication via le protocole UART.

:الملخص

يهدف مشروعنا لتحسين خدمة الاطعام في الاقامات الجامعية من خلال تطوير هذا النظام الذي يعمل على تنظيم الطابور و رقمنه التعاملات للتخلص من البطاقات الورقية و استبدالها بأخرى رقمية مما يؤدي الى تفادي الاكتظاظ والمناوشات بالإضافة لتسهيل عمليات الاحصاء المتعلقة بالعملية، قد استطعنا و بنجاح ان نبني قاعدة بيانات خاصة بالنظام ،سيتم تخزين كافة المعلومات فيها ولن يستطيع الولوج اليها الا المعنيون بالأمر كما تمكنا من تصميم مخططات اولية للأجهزة التي ستدعم اداء هذا النظام من خلال تحكم اكبر بتدفق الطابور .

الكلمات المفتاحية: خدمة طعام تنظيم ، جمع بيانات ، بطاقات رقمية, الاتصال التسلسلي ,برتوكول UART

Abstract:

Our project aims to improve the food service in university dormitories by developing this system that organizes queues and digitizes transactions to eliminate paper cards and replace them with digital ones, thus avoiding overcrowding and conflicts. Additionally, it facilitates the statistical operations related to the process. We have successfully built a database specifically for the system, where all the information will be stored and only authorized personnel will have access to it. Furthermore, we have designed preliminary diagrams for the devices that will support the performance of this system, allowing for better control of the queue flow.

Keywords: *Dining service , Organizing , Data collection , Digital cards , Serial Communication, UART Protocol Communication*