

An NFC Enabled Student Card System

**Raissa Pululu**

**K00260175**

**Supervised by Brendan Watson**

A Final Year Project submitted in fulfilment of the requirements of the Technological University of the Shannon: Midlands Midwest, for the degree of Bachelor of Science (Honours) in Software Development

**BSc in Software Development – Declaration of Authorship**

I, Raissa Pululu, declare that the software development project titled An NFC Enabled Student Card System submitted by me as part of my BSc in Software Development is entirely my own work, and I have not used unauthorized external assistance in its creation. I affirm that:

I have fully and independently developed the software code, design, and documentation contained within this project.

Any external sources, such as code snippets, libraries, or third-party resources, have been appropriately cited and referenced in accordance with academic citation standards.

This project is a result of my own creative and intellectual efforts, and I have not copied, borrowed, or utilized work produced by others without proper acknowledgment.

I have not used automated content generation tools, including AI-based code or text generation services, in the creation of this project.

Any collaborations or contributions from external individuals are acknowledged and appropriately credited in the project documentation.

The code and documentation presented here reflect my understanding of the concepts and principles learned during the course.

I understand that any breach of academic integrity, including plagiarism or submitting work that is not my own, is subject to disciplinary actions as outlined in the TUS academic policies.

[Signature] Raissa Pululu

[Date] 1/5/2024

# **Acknowledgements**

I would like to thank and acknowledge the staff of TUS that have been there through my continuous growth during the past 4 years. The advice and feedback throughout the development of this project and the 4 years of this course has been incredibly valuable and shows how I have been able to get here

I would like to thank the supervisor of the project Brendan Watson for supporting throughout this project. The aim of this project would not have been conceptualised nor achieved without his support.

I would like to thank my parents for their enormous efforts throughout my childhood to ensure I get to this point in my education. Their support is something I will always be indebted to.

I would like to finally thank the class of BSc Software Development 2024, it’s incredible that so many of us are still here battling to the end. I could not be prouder of the work we have all put in to get here.

# **Abstract**

NFC and RFID are two of the main driving forces moving most of the worlds everyday tasks and taking many users from physical to digital. In the last few decades, we have gone from the physical world to now a contactless world. These efficient ways of moving about the world in a contactless have now been long integrated into our lives such as Transport for Ireland’s Leap Card, payment transactions with contactless cards and mobile wallets such as the ApplePay and GooglePay, even as far as BMW using the Apple Wallet to develop a digital card to unlock their cars.

This project outlines the in-depth research completed into NFC and RFID technologies and how it can be applied to a simple application with the use of a Raspberry Pi. The goal of this project is to model a student card system with the use of RFID enabled tags and chips that can communicate to a Raspberry Pi. The application should be able to complete simple tasks such as modelling unlocking a door, signing in students for attendance, add a new student to the system, top up the balance of each card to allow the user to pay for parking along with showing how we can create a sample API based on the project to display the functionality of how so many of our projects can be created as APIs.

**Table of Contents**

[**Acknowledgements** 3](#_Toc164535874)

[**Abstract** 4](#_Toc164535875)

[**Table of Figures** 7](#_Toc164535876)

[**Table of Tables** 8](#_Toc164535877)

[**Introduction** 9](#_Toc164535878)

[**Literature Review** 11](#_Toc164535879)

[**NFC Technology** 11](#_Toc164535880)

[**History and Evolution** 11](#_Toc164535881)

[**Mechanisms and Operations of NFC and RFID** 14](#_Toc164535882)

[**Applications** 20](#_Toc164535883)

[**NFC Enabled Student Cards** 22](#_Toc164535884)

[**Implementation** 22](#_Toc164535885)

[**API Integration** 26](#_Toc164535886)

[**Definition** 26](#_Toc164535887)

[**Challenges** 27](#_Toc164535888)

[**Future Trends** 29](#_Toc164535889)

[**Conclusion** 30](#_Toc164535890)

[**System Analysis and Design** 31](#_Toc164535891)

[**Introduction** 31](#_Toc164535892)

[**Analysis** 32](#_Toc164535893)

[**Hardware Requirements** 34](#_Toc164535894)

[**Software Requirements** 36](#_Toc164535895)

[**System Analysis** 37](#_Toc164535896)

[**Requirements Analysis** 37](#_Toc164535897)

[**User Analysis** 38](#_Toc164535898)

[**System Design** 39](#_Toc164535899)

[**Hardware Parts** 39](#_Toc164535900)

[**Hardware Design** 42](#_Toc164535901)

[**Architecture Design** 43](#_Toc164535902)

[**Interface Design** 45](#_Toc164535903)

[**Alternative Design** 47](#_Toc164535904)

[**Database Design** 47](#_Toc164535905)

[**Summary** 49](#_Toc164535906)

[**Implementation** 50](#_Toc164535907)

[**Introduction** 50](#_Toc164535908)

[**Technical Information** 51](#_Toc164535909)

[**System and Software Design** 51](#_Toc164535910)

[**Circuit Design** 53](#_Toc164535911)

[**Project Management** 54](#_Toc164535912)

[**Design Quality** 55](#_Toc164535913)

[**Program Reliability** 56](#_Toc164535914)

[**Software Reuse** 59](#_Toc164535915)

[**Design Decision** 60](#_Toc164535916)

[**System Architecture** 64](#_Toc164535917)

[**Overall Structure** 64](#_Toc164535918)

[**Dependencies Directory Structure** 65](#_Toc164535919)

[**Modules Directory Structure** 67](#_Toc164535920)

[**Completed System** 74](#_Toc164535921)

[**Testing** 82](#_Toc164535922)

[**Introduction** 82](#_Toc164535923)

[**Testing Methodologies** 82](#_Toc164535924)

[**Test Cases** 82](#_Toc164535925)

[**Automated Testing** 82](#_Toc164535926)

[**Test Results** 82](#_Toc164535927)

[**Test Conclusion** 82](#_Toc164535928)

[**Conclusion** 82](#_Toc164535929)

[**Summary of Research** 82](#_Toc164535930)

[**Practical Implications** 82](#_Toc164535931)

[**Recommendations** 82](#_Toc164535932)

[**Final Thoughts** 82](#_Toc164535933)

[**Closing Remarks** 82](#_Toc164535934)

[**References** 83](#_Toc164535935)

# **Table of Figures**

[Figure 1 NFC Reader Modes (STMicroelectronics, 2022 15](#_Toc164535936)

[Figure 2 PN532 39](#_Toc164535937)

[Figure 3 PN532 39](#_Toc164535938)

[Figure 4. ARC122U 40](#_Toc164535939)

[Figure 5 RC522 41](#_Toc164535940)

[Figure 6 RC522 41](#_Toc164535941)

[Figure 7 Sample Circuit Diagram 42](#_Toc164535942)

[Figure 8 Sequence Diagram 44](#_Toc164535943)

[Figure 9 Tkinter Sample Interface 45](#_Toc164535944)

[Figure 10 Pyqt Sample Interface 46](#_Toc164535945)

[Figure 11 Database Design 47](#_Toc164535946)

[Figure 12 System Architecture 52](#_Toc164535947)

[Figure 13 Exploratory Software Development 52](#_Toc164535948)

[Figure 14 A top level view of the physical circuit 53](#_Toc164535949)

[Figure 15 A screenshot of naming conventions in the project 55](#_Toc164535950)

[Figure 16 A screenshot of error handling 57](#_Toc164535951)

[Figure 17 A screenshot of error handling 57](#_Toc164535952)

[Figure 18 A screenshot of threads 57](#_Toc164535953)

[Figure 19 Sample code from I2C\_LCD\_driver.py 66](#_Toc164535954)

[Figure 20 app.py Code Snippet 67](#_Toc164535955)

[Figure 21 check\_attendance.py Code Snippet 68](#_Toc164535956)

[Figure 22 main.py Code Snippet 69](#_Toc164535957)

[Figure 23 main.py Code Snippet 70](#_Toc164535958)

[Figure 24 save\_user.py Code Snippet 71](#_Toc164535959)

[Figure 25 scan.py Code Snippet 72](#_Toc164535960)

[Figure 26 unlock.py Code Snippet 73](#_Toc164535961)

[Figure 27 A screenshot of the main system 74](#_Toc164535962)

[Figure 28 A screenshot of the parking system 74](#_Toc164535963)

[Figure 29 A screenshot of the top up system 75](#_Toc164535964)

[Figure 30 A picture of the LCD displaying messages 75](#_Toc164535965)

[Figure 31 A picture of the LCD displaying messages 76](#_Toc164535966)

[Figure 32 A screenshot of the API call for attendance tracking 76](#_Toc164535967)

[Figure 33 A screenshot of the department dropdown 77](#_Toc164535968)

[Figure 34 A screenshot of the Unlock Door method 78](#_Toc164535969)

[Figure 35 A screenshot of the Register User method 78](#_Toc164535970)

[Figure 36 A screenshot of the parking method 79](#_Toc164535971)

[Figure 37 A screenshot of the attendance system 79](#_Toc164535972)

[Figure 38 A screenshot of the top up method 80](#_Toc164535973)

[Figure 39 A screenshot of the top up method 81](#_Toc164535974)

# **Table of Tables**

[Table 1 Hardware Requirements 34](#_Toc164444998)

[Table 2 Software Requirements 35](#_Toc164444999)

[Table 3 Specification of PN532 38](#_Toc164445000)

[Table 4 Specification of ARC122U 39](#_Toc164445001)

[Table 5 Specification of RC522 40](#_Toc164445002)

# **Introduction**

The 21st century is now being described as the digital era. Seamless, contactless, and wireless interactions are now becoming the new norm using technologies such as Wi-Fi, Bluetooth, NFC or Near Field Communication, the internet and so much more. Our methods of communication are changing every day and more and more it seems as though it’s up to us to upkeep to the overwhelming new standards as they rollout. Not only do we see these being implemented into our homes and workplaces, the public sector; public transport, mobile industries, automotive infrastructures have rapidly adopted these technologies to offer continuous ways of staying online using all the above but most importantly NFC and APIs (cj&co, 2023).

NFC in general has been the strongest force conducting real time and real-world applications in terms of going digital and contactless using RFID. Payment transactions, transport, mobile payments are globally integrated with the use of NFC chips integrated onto physical debit/credit cards, NFC integrated basic cards, tags or even QR codes. By 2026, it’s predicted that contactless payments are expected to rapidly increase by 221%, contactless cards expecting to increase by 119% in the same period which will come from majority of mobile payments. (Brophy and King, 2023). As of 2018, there are 3.4 billion active smartphones globally, two billion of these are NFC enabled devices meaning 20% of the global population are using NFC enabled devices. (NFC Forum, 2019).

Software as we know it today has become more expandable. Developers are looking for easier ways to complete projects, apply the newest features and satisfy stakeholders and their end users. APIs have become a growing facilitator in the development of software. They are not only a flexible tool; they aim to assist the developers in creating fast, intuitive, and reliable systems. APIs grant us the ability to plug and play software with minimal effort, reducing the workload as the software has simply been implemented and wrapped around existing projects.

Statistics show that 90% of software developers integrate APIs into their work. 69% using third-party APIs and 20% using internal or private APIs. Important factors considered for using APIs include performance, security, reliability, documentation, and scalability with performance being the highest factor at 72% and scalability being the lowest at 59%. Developers that create APIs find that integration with internal systems are the most important factor at 83% and speeding up development times the least important at 42%. (Kolya Hnatyuk, 2023).

Following these statistics, we can see that NFC and APIs are becoming global tools for all to use and that’s how this final year project aims to take advantage of this technology and proposes an NFC enabled student card system that integrates with existing campus infrastructure. Systems like these already exist and findings of these systems will be discussed in this review. Solutions implemented by other developers see the system as not only an identification system but as solution tailored to enhance not only the student experience but academia experience. Many of these systems can be transformed into APIs which allow the future developers and system administrators to easily implement new programs to existing solutions.

The goal of this Final Year Project is to develop and implement an NFC Enabled Student Card system that will grow and improve student cards based on five use cases. The technology intends to offer a comprehensive solution for employees and students that includes faster class attendance tracking, paid parking, and a secure system with door access control. It will function as a mock system with NFC hardware integration by utilising the Raspberry Pi microcontroller.

The goal of the project is to provide a system to relieve staff from tracking attendance during class times, have shorter lines for parking and have a secure system for door access control for specific areas on campus, offices etc.

The system would proceed to track attendance based on each scan of a student card with information including the name, the date and time of each student card and provide a callable API that would be available to the staff member. Depending on the permissions the administrator sets, the system would allow some cards to only access rooms. Without the use of a cash register, the card may be used to make purchases in parking.

This Final Year Project aims to cover NFC technology integration, with software and hardware and model it on a smaller scale. Reviewing and understanding what NFC is, how it functions, and how to implement access control functions like door access control. Studying the use of NFC and how to use public Python libraries to obtain data from the readers and cards and how to store them. the data it gathers to send to specified users and how admins can view the information. As well as develop an API that can show that this application can be flexible and used in different ways for different purposes.

# **Literature Review**

## **NFC Technology**

This section will provide an overview of the historical implementation and impact Near Field Communication technology has provided over the course of the past decades and how it now used in our daily lives.

### **History and Evolution**

NFC or Near Field Communication is the main force behind your everyday payment facilitator such as Apple Pay, Google Pay and contactless card payments. Its main functionality is providing a wireless technology between devices within proximity of a range between 2 to 5 centimetres.

This technology can be dated as far back as the 1940s as an implementation for World War II using RFID (Radio Frequency Identification) as a means for radar technology marking the beginning of the new wave of communication which was especially significant for the Manhattan Project. This technology was used to detect and locate objects based on the reflection of radio waves. This was quickly adopted by the military. From this finding RFID was born (Landt, 2005). Harry Stockman’s research on RFID technology has been noted as the first or earliest research published in 1948 in a paper “Communication by Means of Reflected Power”. The paper did state that there was tremendous research completed before him regarding the finding of this information stating, “Evidently considerable research and development work has to be done before the remaining basic problems in reflected-power communication are solved, and before the field of useful applications is explored.”. (Stockman, 1948). As there was a lack of hardware during the time of the paper published by Stockman, it took over 30 years for this development to become a physical finding. Transistors, integrated circuits, microprocessors, communication networks and other developments were needed to find the solution to RFID and NFC (Landt, 2005).

The first real world application of RFID was developed between the 1960s and 1970s when companies Sensormatic, Checkpoint and Knogo began their development of EAS or Electronic Article Surveillance which would now be the grandfather of security tags for retail merchandise. They used microwaves or inductive technology which was cheap to create and the first commercially available RFID product (Landt, 2005). The 1970s there was increase of research towards RFID, Los Alamos Scientific Laboratory notably the lab that had managed most of the development of the first atomic bomb, Northwestern University and the Microwave Institute Foundation in Sweden all taking the reins in this development. Los Alamos delivered an important paper in 1975 on their researched named “Short-Range Radiotelemetry for Electronic Identification using Modulated Backscatter.”. This research demonstrated tags that could be operated from a 10-meter range while larger companies developed RFID technology such as Raytheon’s Raytag in 1973 and the RCA’s electronic development system in 1975 (Landt, 2005).

In the meantime, there was some news of an electronic toll collection in the works by the Port Authority of New York and New Jersey in partnership with General Electric, Westinghouse, Phillips and Glenayre. In the 1980s European countries had implemented RFID through their toll roads, animal tracking while the United Stated focused on transportation, personnel access, and animal tracking. The first commercial application of an RFID toll system after years of testing began in Norway in 1987 and the Dallas North Turnpike in 1989, the Port Authority of New York and New Jersey implemented RFID operations for buses passing through the Lincoln Tunnel. The 1990s signified the widespread use of RFID technology through electronic tolls with over 3 million RFID tags implemented on North American rail cars. Similar technology was being implemented in Europe, South America, Asia, and Australia (Landt, 2005). South Korea’s Seoul Bus Transport Association implemented the first contactless transport card called the UPass (Global Payments Integrated, 2014).

Retreating a few years, in the 1980s Sony and Phillips saw potential in the ability to use the technology to transfer data between devices. Both companies had used their development to transfer music files between devices during that time. (profylecard, 2021). Further on Sony and Phillips join to co-invent NFC which is then approved by the ISO in 2003 as a new standard. In the same year Nordea and Luottokunta released a prototype device that used NFC to make payments. The following year, Nokia modified the Nokia 6210 to be able to read RFID tags. Partnered with JC Decaux, the modified 6210 could read tags and send an SMS once the tag had been read. In the same year Nokia, Phillips and Sony curated the NFC Forum to aid in the development of NFC technology, they hoped it would create new standards in the mobile industry which was achieved now with over 160 members in the forum (Paragon ID, 2017).

The evolution of these developments and studies, companies began getting onboard the new burst of information about RFID and NFC. Nokia kept the mark strong by releasing the first NFC enabled phone in the same year as the first creation of the NFC tag in 2006. 3 years later peer-to-peer communications started to pop up allowing more wireless connections and allowing more functionality in mobile phones. Evolution continued as NFC enabled smartphone the Samsung Nexus S through Android was being sold to consumers (Paragon ID, 2017).

Contactless payments as we know today through our phones and physical cards began in 2008 when Visa American Express and MasterCard offered contactless payments for credit cards (Global Payments Integrated, 2014). Debit cards had followed suit the following year (Statista, 2020). Mobile payments are then launched in 2011 with Google Wallet™ and Android Pay™ aiming to removing need to carry your card everywhere and complete all payments through your smartphone. Apple followed suit in 2014 with Apple Pay® and the next year wearable devices were added into the mobile payment ecosystem, allowing devices such as the smartwatch to complete the same functionality as the smartphone (Global Payments Integrated, 2014).

In our everyday practical lives, we now see NFC integration not only on our mobile devices but even in our homes. Starting with security systems and how they may provide tags to enable and disable the alarm, entering homes and hotel rooms have had a long history of NFC enabled devices working alongside each other.

### **Mechanisms and Operations of NFC and RFID**

NFC or Near Field Communication is an extension of RFID technology Radio Frequency Identification. NFC enabled communication between devices in proximity i.e., short-range radio technology. This technology was standardised by the International Organization for Standardization in 2003 (Prohel, 2013).

The short-range, high frequency, low bandwidth and wireless communication occurs at 13.56 MHz high frequency with data transmission rates of up to 424 kbit/s with up to 20 centimetres, but 4 centimetres is practical which prevents eaves dropping on other NFC-enabled devices. There is usually a maximum of two devices used to complete the transfer of data using NFC. The first device is called the initiator which starts the communication, and the second device is called the target which responds to the requests from the initiator. (Rahul et al., 2015). With the use of electromagnetic induction loop antennas between two devices the process of request and response communication is processed. The standard frequency provided by ISO/IEC 180000-3 for NFC systems for 13.56 MHz can be used without a license (IONOS editorial team, 2023).

NFC provides 2 modes of communication: passive and active mode. Passive mode allows active NFC readers to read passive transponders. The antenna of the active device generated a high frequency magnetic field. If the transponder enters the range of the high frequency field, it is supplied with power but does not generate a feedback signal and is transmitted using a modulating inquiry signal. This can be used for contactless payments through debit/credit cards or mobile devices. Active mode occurs when both the initiator and target device can transmit NFC signals. These devices have their own energy source and generate their own high frequency during communication activating and deactivating fields when necessary or when waiting for a response (IONOS editorial team, 2023).

NFC can be used independently but the rest of the communication can be completed using Wi-Fi or Bluetooth e.g., wireless card readers. The target device may be able to handle multiple carriers such as NFC, Wi-Fi, and Bluetooth. The requesting device can choose which ever carrier is best suitable for both devices when it received the response message. This method of communication is preferred over Bluetooth as it consumes less power and does not require any physical paring compared to Bluetooth (Rahul et al., 2015).

NFC chips can support multiple Radio Frequency (RF) protocols and features such as Read/Write, Peer-to-Peer and Card Emulation as outlined in Figure 1.

1. Reader Mode: One device can act as a reader or writer while the other serves as a tag or card. The tag holds a small amount of data such as a unique identifier or other information. Tap & Pair represents the initiation of a connection between two devices such as when an NFC tag is attempting to connect to a Bluetooth printer. Tap & Exchange represents the exchange of data between the reader and NFC tag such as reading data from a smart poster using a smartphone.
2. Card Emulation Mode: The device emulates an NFC card or tag such as contactless payments where smart devices such as mobile phones or watches emulate debit/credit cards. Tap & Pay represents contactless payments from an NFC enabled device to a POS terminal.
3. Peer-to-Peer Mode: Two NFC enabled devices can communicate and exchange data with each other. Tap & Exchange represents the exchange of data like Reader Mode, but this mainly occurs between two devices like smartphones to exchange contacts, photos, etc...
4. [A few cards with text and symbols

   Description automatically generated with medium confidence](https://www.st.com/content/st_com/en/support/learning/essentials-and-insights/connectivity/nfc/nfc-chips.html)Charging Mode: This is not a traditional NFC operation but represents NFC technology being used as a wireless charging mechanism. Approach & Charge represents an NFC enabled device being brought close to a wireless charger.

Figure 1 NFC Reader Modes (STMicroelectronics, 2022

Due to communication occurring on two devices, there needs to be a security standard already implemented to reduce eavesdropping. Point-of-Sales (POS) terminals that use NFC to initiate contactless payments that use RFID communication protocols have implemented standards such as ISO/IEC 14443 (Badra and Badra, 2016).

The ISO/IEC 14443 is the international standard for proximity cards and contactless smart cards that are used in many NFC applications. It specifies the communication and security protocols for secure data transfer (ISO, 2023). It specified the physical characteristics, radio frequency power and signal interfaces along with initialisation and anticollision protocols. This standard is implemented by Card Emulation Mode when NFC enabled devices emulate ISO/IEC 14443 cards in this mode such as digital wallets making it compatible with POS terminals. ISO/IEC 14443 is split into two types: Type A and Type B. NFC is compatible with both types allowing for broader range of interactions and supporting a large array of applications, particularly contactless payments (ISO, 2023).

Initially in 2001 when the ISO/IEC 14443 standard was released, it was structured into four distinct parts: physical characteristics, radio frequency power and signal interface, initialisation, and anti-collision and transmission protocol. These aimed to ensure consistent behaviour and interoperability(ability to operate and share data with another application) among proximity cards and readers. Over the years the specifications where refined or expanded on. In the 2010s, there was an enhanced security update applied to counter the rise of digital attacks. The standard received updates on encryption, authentication, and other security measures to ensure data being passed across contactless communication protocols were secure and could not be tampered with (ISO, 2023).

The ISO/IEC 18092 defines the communication modes for NFC Interface and Protocol (NFCIP-1) using conductive coupled devices operating at the specified frequency of 13.56 MHz. It defines the Active and Passive communication modes for NFCIP-1. This standard is implemented through Peer-to-Peer modes, allowing two-way communication between two NFC enabled devices. It encompasses the various layers of NFC communication including initialisation, data collision and data exchange (ISO/IEC, 2013). This standard ensures the compatibility with other contactless technologies such as FeliCa which is a contactless smart card system developed by Sony in Japan. It is widely used for public transportation including Suica and Pasmo cards. Universities and companies in Japan use FeliCa as their main system for NFC access control system and ID cards. FeliCa is also implemented into mobile phones in Japan to be used as digital smart wallets or public transport passes (Sony, 2023).

In 2004, the ISO/IEC 18092 standard “Near Field Communication Interface and Protocol(NFCIP-1) was released defining the communication modes for NFC enabled devices. From 2004, the adoption of this standard became widespread particularly in niche applications and electronics. In the 2010s, this standard was adopted on smartphones, tablets, smartwatches, and wearables on systems like Android and iOS. Security and performance improvements were implemented for payment transactions (ISO/IEC, 2013).

Apart from the security that are included in the standards provided by ISO/IEC, many developer/companies do lay their own security protocols on top of the pre-existing ones.

Tokenization is a method of replacing or masking sensitive data such as card information with randomly generated, unique card details used on digital wallets such as ApplePay or Google Pay for e-commerce or in-store transactions on POS terminals. By doing this, it mimics the physical card and replacing it with the token. If said token is intercepted it cannot be used apart from the original transaction. During the token life cycle the Token Service Provider must control it with 4 different statuses:

1. Requested: Initial state of token on digital wallet
2. Active: The token is ready or already has been used
3. Suspended: The token is suspended by the issuer.
4. Ended: The token has permanently been disabled (HST Software Solutions, 2021).

Encryption is another method of security that it used to enhance the privacy of sensitive data being passed along NFC channels. To avoid eavesdropping and interceptor’s encryption can be used to encrypt the data during transmission and decrypted when received providing the receiver has the decryption key. Symmetric and asymmetric encryption can also be used where with symmetric both the sender and receiver use the same key for encryption and decryption, this is faster but a higher risk of exposure. Asymmetric using a public key for encryption and a private key for decryption which is mostly used in NFC transactions as the public key can be openly shared (Hendry, 2014).

SE or Secure Elements is a tamper-proof platform usually in the form of a chip, is capable of security hosting applications on their confidential and cryptographic data such as NFC enabled mobile payments on smartphones. This chip does not have permissions enabled for installation; all software comes preinstalled. Only trust applications such as digital wallets or POS terminals have read/write access to the chip. SE provides detection of hacking and modification attempts, creation of Root of Trust platform for encryption systems, storing private encryption keys and bank details, random number generator that is are secured by cryptography, generation of private and public keys for asymmetric encryption. This is all provided at a hardware level (Kaspersky, 2020).

Introduced around 2014, HCE or Host Card Emulation allows for NFC applications on smart devices without relying on access to the SE making mobile payments more versatile. HCE aimed to provide two solutions, card emulated by the cloud system or emulated by the mobile application. In comparison to SE, SE provided several functions as mentioned above like a cryptography check to validate the card information and verification where as HCE’s implementation would guarantee that any NFC data received by the processing app was delivered from the controller and would remove any other entities. By doing this, this would mean that the mobile device would have to be connected to a network to complete the authorisation of the transaction (Sims, 2014).

While all the above mentioned, intricate security onboarding protocols can be implemented onto NFC chips/devices, it is important to note that there is still the possibility of signal copying or cloning. Since NFC operated at a 13.56 MHz and allows two-way communication between devices and allows devices to read tags or act as one regarding card emulation it is possible for the data to be intercepted using specialised equipment or mobile applications designed for this purpose. Typically, the attacker would need to be close to the victim, centimetres away, they can then intercept the data. Once its intercepted, the data can be analysed using sophisticated tools as it may not be comprehendible immediately. With the captured and decrypted data, the attackers can emulate the same tag or reuse the signal emitted by the device to carry out unauthorised activities such as making fraudulent payments, identity, and data theft (Francis et al., 2012).

Devices such as Proxmark devices or the ChameleonMini can be used for RFID/NFC signal copying. The Proxmark3 can read and emulate a multitude of NFC and RFID tags as ca the ChameleonMini. These devices or similar are readily available on websites like Amazon starting at 40 GBP (Waldman, 2021). Projects like the ChameleonMini had started out as open source projects on GitHub in 2013 but now have become widescale cloning tools coming in the form of credit cards. These devices could clone the UID(Unique Identifiers) and stores the data and can also attack RFID readers to sniff for keys and decrypt them but to be able to do this the device has to be near be able to complete the task (Person, 2022).

### **Applications**

In an era where technological advancements are never ending, NFC stands out as a pivotal enabler of seamless digital interactions. NFC has become the source of most of your daily interactions to hardware, making systems simple and efficient. From going from cash to cashless, from replacing physical public transport tickets to creating interactive access control systems.

The rise of digital wallets and mobile payments solutions have been largely fuelled by NFC technology. NFC-enabled smartphones allow to store their debit/credit card information on their devices virtually by masking the card details but routing back to the account of where the information lies. Using POS systems like Square or SumUp, transactions can be completed without the need of a physical card. Due to the masking of the card details it adds a layer of security by using encrypted tokens. In 2022, 49% of global e-commerce transactions were completed by digital wallets while 20% were completed by 20% by credit card and 12% by debit card (Statista, 2023). Projected to surpass 1.30 billion proximity mobile payments in 2023, in 2019 0.95 proximity transactions were completed (Statista, 2019).

The public transport experience globally is fuelled in NFC technology. Dating as far back as the 1980’s with NFC tags for tolls and rail cars to collect payments for crossing borders and South Korea implementing the first public transport system using NFC in 1996. Transport for London implemented the Oyster card in 2003, allowing contactless public transport payments using an RFID system with a built-in inductor that reads the frequency being projected by the reader (Birbeck, University of London, n.d.). In 2012, TFL had implemented a contactless payment method for the underground and rail network allowing debit/credit cards to be used alongside the Oyster card and cash which has now the biggest payment form on TFL. Finland had implemented the same system on buses in 2021 in its city of Turku. By using their pre-existing ticketing system, they changed the model to match the requirements of payment service providers (Moore, 2023).

Traditional access control has relied on magnetic stripe cards or physical keys. With NFC, access control becomes more digital and more secure. Whether for offices, secures areas in airports, or even gym memberships, NFC devices can grant access based on the embedded data or permissions. These kinds of systems can provide logs of entry and exit times to enhance security. As predicted by market research, the access control market is expected to grow from $175.57 million in 2021 to $787.01 million by 2027 (The Insight Partners, 2020). Companies like Kisi Inc., Salto Systems, Proxy Inc., BlueID, Remotelock, YPTOKEY, Nok, Inc., Openpath, Inc., Unikey Technologies Inc, and Brivio Systems LLC are leading players in the marker that are implementing NFC in NFC access control systems (Research Union, 2023).

## **NFC Enabled Student Cards**

### **Implementation**

To get a complete understanding of this Final Year Project and compare it to the proposed system, we will delve into previous proposed systems or already implemented systems to see how versatile this technology can be.

As part of the proposed system for this final year project, the system should be able to handle door access control, class attendance tracking, paid parking/canteen purchases all handled by an NFC integrated student card. The goal of this implementation is to create a unified, efficient experience for all students and faculty.

As discovered by Duke University, implementing NFC technology with student ID cards has been able to streamline campus life for all its students and staff. As well as using a physical student card, the university has opted for implementing this through the Apple Wallet with iPhone and Apple Watch to “improve the student experience”. They have integrated this as this now allows students to pay for lunch at dining halls through their student cards on their digital wallets which makes the card act as the home of the entire system. They have also implemented door access control through these digital wallets where only students specified to enter a chemistry lab has permissions to enter compared to a business student who would not need to enter that room thus the card would decline them from entering. Florida International University have implemented NFC tags and codes at their library for instant access to digital reading (Zimmerman, 2018).

Companies like Transact offer a multitude of customisable features, starting from having digital NFC enabled student cards that store encrypted student data per card which allows the student to make purchases on campuses which is integrated with pre-existing POS systems. They also provide door access control with can be integrated with security on campus which allows for an admin view where they can set and change permissions as they see fit (Transact, 2023).

Purdue University in West Lafayette Indiana, implemented a similar system like Duke University, allowing students to use their digital wallets as placeholders for the student cards. This digital NFC cards allow students and faculty to have assigned residential and campus access, pay for meal plans and laundry through the card, act as an attendance tracker for class and door access tracker for administrator logs. This is all with the use of the Transact system built around the pre-existing system at Purdue University to offer a smooth and seamless experience for everyone on campus (Purdue News Service, 2023).

The University of Hong Kong have implemented an attendance-taking and event enrolment using their in-house developed AIESEC HKU Tap-in app. Each student card can be scanned by the phone, it collects a string of characters is collected by the phone which contains each student which is now displayed on the app. Before it would take 30 seconds to register each person but now it takes two seconds to register students (Lung, 2015).

Previously implemented functionality by scholars who have designed a similar system start by creating a schema to manage all the data to complete the functionality. In terms of creating a class attendance system, tables such as ‘Course’, ‘Student’, ‘Lecturer’, ‘CourseAttendance’ would exist to manager this functionality. To implement this, a web page as front end including HTML, JavaScript and CSS and MySQL and PHP acting as back end was used. The lecturer would be able to log in and view which students were marked as present based of the NFC tags that were read. To simulate the physical hardware RiFiDi was used to mock the NFC tags. Two antennas would be used to cover the whole classroom and each student would be tagged (Pireva, Krenare R, Siqeca and Berisha, 2013).

Another attendance tracker implemented by students from the University of Bridgeport. Using an Arduino Mega to process tan NFC Module to read a NodeMCU to send IoT messages over Wi-Fi all with an NFC enabled card with the ISO/IEC 14443 standard, if the card is scanner by the NFC Module, the reader will send the ID number and desk to a cloud database to track attendance as well as the student information, course, and sign-in time. Each student ID card has a unique information based on 10-15 students in each class. When the NFC card is tapped on a reader during scheduled class time, the reader sends a message confirming it has received the data before sending the data to an Azure IOT Hub via MQTT message (Dixon and Abdel-shakour Abuzneid, 2020).

Implementing a smart access control for paid parking was implemented by students of the Department of Electrical Engineering, Hasanuddin University. Using a registered NFC student or staff card, it can open or close a parking gate automatically. The proposed parking gate by the students included a PN532 NFC RFID module, a microcontroller, NFC tag, PING sensor and ethernet shield. All information passed through this smart gate would be stores on a server including user data, access history entry and exist using a localhost. The process of interacting with the gate begins with the user tapping their tag or card on the reader of the gate, if the tag/card is registered and identified as having access to the gate, it is opened and closed automatically after the PING sensor has been detected. During this process the database server is updated by ordering the users in chronological order of accessing the gate. This paper looks at this smart gate as a security measure on campus by only allowing certain users to have access to the gate (Mansur and Hasanuddin, Zulfajri B, 2018) .

Students of Electronic and Mechanical Engineering College, Fujian Polytechnic Normal University and Maynooth International Engineering College, Fuzhou University implemented a door access control system using OpenMV(microcontroller), Arduino an RC522(RFID reader), Esp9266 Wi-Fi module and Ali Cloud of Internet of Things. By developing a WeChat program, these students were able to read information between the users and electromagnetic lock when the door was opened and ensuing a non-contact unlock. The OpenMV was able to read and store information based on facial recognition by the connection module and power supply. The RC522 module was able to store and read information from an NFC enabled card through the connection module and power supply. The Esp8266 module was able to connect to the WeChat program through the Alibaba Cloud IOT so they it could interconnect the entire system to the campus network and so it could receive unlock and lock instructions. A magnetic lock is used to simulate the lock of the system. In this system one of two electrical connections is receiving a specified level while the other is grounded. The Arduino UNO board is connected to the computer and the lock is connected to the board; the Arduino has been coded to keep the lock closed. For facial recognition, the OpenMV is connected to the computer is coded to be able to write and modify face storage and face recognition code. Face recognition and radio frequency is set up as a level output and connected to the whole system by writing and modifying the code on the Arduino. The Wi-Fi module is then connected to the Arduino and connected to the Alibaba Cloud IOT and there is code written to interconnect the system to the campus network infrastructure (Zhou et al., 2022).

Students of State Polytechnic of Malang, Malang, Indonesia built an NFC based self-service canteen in a local boarding school. The reason for implanting such system was that students were exceeding the amount of money they could take daily from their residential financial managers which brought up a solution to tackle overspending and long lines of students wanting to withdraw money. Students also kept money in their dormitories which also raised a security concern. By creating a system that uses e-money transactions would eliminate all the issues brought up above. This application used NFC technology combined with a smart card and smartphone to make canteen purchases. The system included a web server and standalone transaction application which would manage the payment streams. The NFC/smart card would replace cash and hold a balance of money for each student. Each card would act a payment method for students and would complete transactions in the canteen (Reva Rikat Asih et al., 2022).

To ensure a safe and reliable NFC enabled student card, one of the best ways would be to use a Multi-Factor Authenticator. MFA has three main factor a PIN/password, device used and behavioural/physical biometrics. EU law states at least two of these must be used to use MFA. The students at Brno University of Technology, Brno, Czech Republic implemented a system using an NFC enabled android smartphone , SAM module in the form of a smart card and a Raspberry Pi 3. The system allows the terminal to connect to an electric door lock to unlock based on a successful authentication from the user. The user triggers the system by bringing the card to the reader, an Android application is triggers asking the user to authenticate themselves by either using a PIN or fingerprint which are encrypted in the backend. Once authenticated, there is a log of successful authentication with a timestamp which is also encrypted. They door then unlocks depending on if the authenticated user has permissions to use this door (Cvrček and Dzurenda, 2022).

## **API Integration**

### **Definition**

As defined by MuleSoft, an API is an Application Programming Interface which is a software intermediary that allows two applications to talk to each other (Frye and MuleSoft, 2023). They provide a simple interface to interact with a complex system like sending a mobile payment or changing the home temperature on your phone without needed to see or know the intricate details of its implementation. APIs can include functions and procedures. These then bridged the gap between the application on your device and other applications. They take up over 80% of internet traffic globally (Kong Inc, 2023).

Web services that allow you to log in with your Google or Facebook account to authenticate you as a user, use the API provided by the host to authenticate you. They operate by a client/server architecture. The application that sends the request acts as the client and the application receiving the response acts a service. In terms of using Google to authenticate you as a user for mobile gaming application, Google would act the server and the gaming app would act as the client (Amazon Web Services, Inc., 2023).

According to AWS there are four types of APIs:

1. SOAP APIs – Simple Object Access Protocol. Client and Server communicate through XML.
2. RPC APIs – Remote Procedure Calls. Client completes function on the server and the server sends the response to the client.
3. WebSocket APIs – Modern web API that uses JSON to pass data. Supports two-way communication between client and server.
4. REST APIs – Representational State Transfer defines set of functions like GET, PUT, DELETE etc. Stateless i.e., do not save client data (Amazon Web Services, Inc., 2023).

APIs have many benefits as they can be flexible depending on the application they are being used on. They provide automation as they can be managed by computers as there is less human effort required and workflows receive faster and more productive updates and can reach end users entirely. APIs enforce developers to reuse code as they don’t need to continuously need to restart from scratch every time, they develop a new application as the API would specify how to assemble the software components (AirFocus, 2020).

### **Challenges**

APIs do come many with challenges. Software integration with APIs do require high level adoption of application languages and maintenance to meet with business requirements are a never ending task. As much as they do help add better functionality and improve services, they do come with their own challenges.

As data breaches become one of the most common security risks in 21-century technology, API technology remains vulnerable. As a lot of these APIs are always connected to a web application it has more opportunities to be susceptible to be targeted. Since these APIs can be complex the level of security and level of communication needs to be detailed from the beginning to avoid any attacks or misuse. A team or task force can be put together to manage the API. They would be responsible for managing security (Nguyen, 2023).

Many developers may be reluctant to implement new APIs as they may not be fully knowledgeable with new systems. Implementing new API would also require training which can become costly and not time effective. Professional or skilled developers may also be necessary to implement new solutions. Maintenance and upgrades should also be taken into consideration when planning to use a new API. It’s important to note that APIs are not a one size fits all. Not all systems connect straight way meaning more time is spent planning and developing a system. Having a skilled developer experienced in the API can reduce the cost and time spent implementing a new API (Curoe, 2021).

Lack of documentation can also provide more issues for developers. Understanding the functionalities, data, and methods of accessing the API may not be available for the developers. These further delays implementation or updates of systems. Contacting the API developers may become last resort if errors or confusion arise during the integration process. There may be data incompatibility with a pre-existing and a new API. Data mapping tools are available to combat this and ensure that you can map the different formats to ensure data between APIs. APIs are usually version controlled which means they may operate differently or use different data structures. Any application using a version controlled API should also be version controlled to ensure that it is keeping up to date with the API. Regularly testing the API integration should be a priority of a developer to ensure it is functioning correctly with the system as well as monitoring the API to ensure it is consistently online and its behaviour is normal (Jagaad Software House, 2023).

Scalability is also important when developing APIs. To avoid overloading on a service rate limiting is implemented but it can impact the functionality of the API during the traffic spikes. Having backwards compatibility is also important for people that use APIs to ensure they can roll back if their system does not work well with a newly updated API (Marcus, 2023).

## **Future Trends**

The NFC market was evaluated to have reached $21.1 Billion in 2022 (PR Newswire, 2023). NFC sales are set to reach $44.7 Billion by 2032 based on automotive and contactless payments (Future Market Insights, 2022). Companies such as Gemalto, Broadcom, Infineon, Inside Secure, NXP and MediaTek are dominating the NFC market. Gemalto leading the secure connectivity solutions with NFC offering services starting from SIM cards to mobile payment solutions (Reach Researcher, 2023).

Beamian, a hybrid event platform, project that NFC will take over event management by using it for ticketing and access control, reducing the need for paper tickets, speeding up the check in process by using wearable devices, mobile phones. NFC tags can be used around the venue for event information, maps, schedules. They can be used to capture attendees’ data for health purposes or safety (Carrilho, 2022).

NFC enabled devices such as thermometers, blood pressure monitors or glucose meters are predicted to be used for health monitoring, diagnosis, treatment, and prevention. NFC enabled test kits, patches and implants are now on the rise for treating conditions and illnesses (Linkedin.com, 2023).

In terms of APIs, as AI and ML are now becoming the most studied technologies, it is expected that APIs built to handle and manage these systems will now start trending as developers will look for systems to build their applications on top of to reduce development time and code manipulation. Since APIs can act a Software-as-a-Service model, developers are now treating it as a product rather than a method of communication between systems (Jaffery, 2022).

It’s been noted that serverless architecture for cloud computing is becoming more popular. APIs will be more focused on as there is no need to have any server maintenance and further reduces costs. API management tools are now becoming more popular to help developer automate tasks or run tests. IOT devices are now becoming more complex with links between cars and smartphones or a smart home and a smartphone, developers are starting to opt for APIs that can act a foundation for their programs (Nguyen, 2022).

## **Conclusion**

This exploration of NFC technology, particularly within the context of NFC-enabled student cards and the implications of API integration, highlights overflowing amount of technology with potential and challenges. Starting with a foundational overview, we have explored NFC's progressive journey from its creation to its current state, acknowledging the pivotal developments that have contributed to its evolution.

The mechanisms and operations of NFC have been demonstrated, showcasing the efficiency of which this technology operates. NFC's versatility was further depicted through a diverse array of applications, from simple data transfer to complex, secure transactions.

Focusing on NFC-enabled student cards, we have examined the multifaceted functionalities they offer, ranging from access control to cashless payment systems, enhancing the educational ecosystem's efficiency and security. The implementation section shed light on the practical aspects, including the technical and logistical considerations vital for successful deployment.

Going into API integration, we discussed the role APIs play in today’s software systems and beyond. While APIs offer a framework for innovation and flexibility, the challenges they present—such as security risks and the need for rigorous management.

Future trends anticipate significant technological advancements, suggesting a trajectory of growth and transformation for NFC and API ecosystems. The potential applications reveal a world of possibilities, from market growth to implemented health systems...

In closing, NFC technology and API integration stands as a testament to the adaptability and potential of digital systems to reshape our interactions and infrastructures. As we continue to embrace these technologies, the importance of strategic implementation, continuous innovation, and consideration of privacy and security becomes increasingly evident. The future trends indicate not only technological progress but also a need for a conscientious approach towards embracing these advancements, ensuring they serve to better our institutions and society.

As this field continues to evolve, it will be imperative to monitor the developments and address the challenges discussed herein. With careful planning and foresight, the integration of NFC and APIs will continue to revolutionize the student experience and beyond.

# **System Analysis and Design**

## **Introduction**

The NFC Enabled Student Card System is an innovative project designed to use Near Field Communication technology to streamline and secure campus operations and services. The NFC Enabled Student Card System aims to not only enhance campus security and efficiency but also to foster a more connected, digitally enabled educational environment. Objectives include seamless access control, accurate attendance tracking, and facilitating cashless transactions. The scope of this system will extend to all areas of campus life impacted by these functionalities, ensuring a comprehensive integration of NFC technology. The approach to this system’s analysis and design is going to be focused on three users; the student, lecturer, and administrative staff. We aim to cover any vulnerabilities of the system as well as study the physical design, software design, coding, and software architecture. Any potential solutions or areas that can be improved and any restrictions will be highlighted. We'll go over the required hardware and software, along with any APIs that can help this project go more smoothly.

NFC technology is being used more and more in a variety of industries, most notably data transfer and security. Research has demonstrated how well it works in educational settings to improve user experience and streamline procedures. The technology is a great fit for university contexts because of its affordability, use, and versatility.

## **Analysis**

When researching previous implementations of an NFC integrated systems for my Final Year Project, I examined several different projects to gain a sense of what previous projects looked like and what kind of hardware and software was utilised to accomplish even the most basic functions, such as reading or writing from an NFC card. According to my research, there are numerous factors to consider when attempting to construct a system like this, such as microcontrollers, NFC chips, software languages, APIs, and so on. User requirements are important to the design of our system. We are committed to providing students, teachers, and administrative personnel with an intuitive experience. Reliable user authentication, efficient access management, and quick data retrieval are all functional needs. Non-functional criteria including system performance, security, stability, scalability, and usability are just as important. These will ensure that our system is not only functional, but also sturdy and easy to use. Taking all these aspects into account, I will present a breakdown of the hardware required, components, software, and technologies that can assist in producing a functioning project in line with the proposed system.

1. The total cost of components, hardware and software licenses needed can play a part in deciding how to begin a project. Depending on whether you choose cheaper components or microcontrollers they may not be able to handle what you intend them to execute. Same with software licences or even cloud storage may play a part. During the decision of the project, it is important to consider whether you would like to store the data on a cloud server or store it locally.
2. The user requirements are important as well. While I am coming up with my own user requirements, it’s important to document and attempt ones that encapsulate the focus of the project.
3. Functional requirements including user authentication, access control, data retrieval must be detailed. Specified user requirements must match the functional requirements and they must work as the system is intended and documented to do.
4. Identifying non-functional requirements like performance, security, reliability, scalability, and usability should be considered if this is something that is intended to be created for an API. APIs must be able to handle the data load or usage specified or that does not offer a working product. As this is not a production level project, the non-functional requirements may be a bit relaxed.
5. The data model should be designed to represent all the identified models in the system. This along with the database design and data storage requirements should be analysed.
6. Since there will be sensitive data handled in the system, the system will have to have adequate security and encryption settings in place. As well as having a tamper proof system for the access control to prevent unauthorized access.
7. UI design for the administrative view must be user-friendly. Considering it will manage student cards, door authorisation, view logs etc… it must be laid out clear and understandable for administration to view.
8. Choosing hardware depends on the requirements needed to complete the project. Certain microcontrollers may only be able to handle a specific workload while another may be able to handle everything which also comes down to price. As well as any external modules chosen, it’s important to consider that some modules will not be able to support all types of NFC cards/tags.

## **Hardware Requirements**

The chosen hardware components, including the NFC RFID Reader Writer and the Raspberry Pi 4 Model B, are selected for their reliability and compatibility with our system's requirements. These components are integral to ensuring efficient data processing and user interaction with the NFC system.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Hardware** | **Manufacturer** | **Quantity** | **Function** | **Reason for choosing** | **Connection** | **Cost** |
| NFC RFID Reader Writer ACR122U ISO 14443A / B | Advanced Card Systems | 1 | A module managing the read and write to the NFC cards, will be able to talk to microcontroller | You can read and write from the module | USB A | £30.98 |
| Electronic Kit | Rk Education | 1 | Includes LED nodes, breadboards, capacitors, transistors etc.. | This will be used to connect the microcontroller to any external modules we have | N/A | £10.23 |
| NFC White Tags | cobee | 20 | NFC compatible cards | Can read and write from these cards | NFC/RFID | £7.16 |
| Raspberry Pi 4 Model B | Raspberry Pi | 1 | The microcontroller controlling the system | The microcontroller controlling the system | N/A | £66.10 |
| Arduino Starter Kit | Arduino | 1 | The microcontroller controlling the system | The microcontroller controlling the system | N/A | 0.00 |
| 5V 1 Channel Relay Board Module | BuxiuGK | 6 | Can act as a switch | Simple mechanism that does not require a lot of manipulation | N/A | £9.21 |
| DC12V Solenoid Lock | Tangxi | 1 | Type of lock used for electronic lock modules | It is a simple lock that can be connected to a breadboard and written through to lock and unlock. | N/A | £6.81 |
| LCD 1602 Module | FREENOVE | 1 | Display messages from microcontrollers | It can display any messaged when we execute functions through the microcontroller | N/A | £8.14 |
| NFC NXP RFID Module V3 Kit | Hailege | 2 | Supports NFC/RFID reading and writing | You can read and write from the module | N/A | £6.38 |
| **SUM** |  |  |  |  |  | **£145.01** |

Table 1 Hardware Requirements

## **Software Requirements**

The software stack will include robust database management systems and programming languages such as Python or Java. These tools are chosen for their flexibility and wide support in handling complex data structures and user interfaces essential for our NFC system.

|  |  |  |
| --- | --- | --- |
| **Software Component** | **Purpose** | **Justification** |
| Database Management System (e.g., MySQL, PostgreSQL) | To store and manage user data, access logs, and transaction records | Provides robust, secure, and scalable data management capabilities |
| Programming Language (e.g., Python, Java) | For developing the system backend and processing NFC data | Versatile and widely supported, suitable for backend development and NFC integration |
| Web Framework (e.g. Flask for Python) | To build the administrative web interface for system management | Facilitates quick development of secure and scalable web applications |
| API Development Tools (e.g., Swagger, Postman) | For creating and testing APIs used in the system | Streamlines the development and testing of RESTful APIs for system integration |
| Version Control System (e.g., Git) | To manage codebase changes and collaborate among development team members | Essential for tracking revisions and collaborating in a team environment |

Table 2 Software Requirements

## **System Analysis**

### **Requirements Analysis**

For this project, the NFC interface serves as the fundamental driving power of the entire concept, using it to demonstrate key principles of real-time data processing and user interaction. While real-time processing remains important, to ensure that the system is secure there can be an implemented slight delay to ensure that the system does recognise the scanned NFC tags as valid. The system could demonstrate a variety of different applications based on locations such as basic access control in classrooms to simple transaction processing in canteens. The system might be able to handle basic authentication and simple transaction processing such as a balance reduction system for paid parking and canteen purchases which can be implemented without using a complex financial solution in addition to the system.

Data management can focus on fundamentals aspects of storing and retrieving data efficiently without needing a high level of database management. Using a database system like MySQL or Microsoft SQL Server for data storage can alleviate any complexities. The database can demonstrate storing, retrieving data and logs. Security should also be considered during the database management process as there is sensitive information being processed about each student. All sensitive information should be encrypted to ensure security throughout the database.

As this is concept idea, the project should be able to demonstrate potential to show that it is scalable and reliable. The system architecture should be designed in a way that shows it can be scaled for example how we can integrate other NFC readers into the system in the future. The system should generally be reliable under normal usage conditions, but it should have redundancies or fail mechanisms implemented or documented. To ensure of a flexible system, the system should be coded using modular practices and follow all coding standards.

### **User Analysis**

The user groups of this system are designed for two: students and staff. These users have distinct needs and interactions with the system and understanding the different is crucial for the system design.

Many of the users if it was expanded to a production level software, would be students. Their interactions with the system are expected to be frequent but short such as using their student card for access to classrooms, labs etc…, making purchases in the canteen or paying for parking. The system should allow for these interactions to happen smoothly and quickly to ensure that the system fast paced as intended. The staff group includes faculty such as lecturers and administrative staff. They will handle different areas of the system in a complex manner. For administration, they would be working with the system to restrict access, monitor logs and create new cards with new functionalities. The system would not be used as frequently for all staff compared to students but would still need to be as robust and intuitive as the student side.

Interactions need to be straightforward for both groups. Tapping the NFC card should be simple and consistent on all the different modules and should provide clear indications of successful or unsuccessful scans. The system should be able to adapt as per the user group and provide a tailored experience for each. The system should be capable of handling transactions in a matter of seconds to prevent bottlenecks.

## **System Design**

### **Hardware Parts**

Choosing the parts for the system depends on what you are trying to achieve. In relation to NFC/RFID modules some may be read only, some may be read and writable. For this project I’ve chosen two different types of modules.

1. PN532 NFC NXP RFID Module

The PN532, is an NFC RFID module that supports reading, writing and peer-to-peer communication. It is a small sized module that is compatible with Arduino and Raspberry Pi. It supports I2C, SPI and HSU(High Speed UART) equipped with 5V TTL for I2C and UART, and 3.3V TLL for SPI.

|  |  |
| --- | --- |
| SPECIFICATION | |
| Operating voltage | 5V/3.3V |
| Interface | I2C, SPI, HSU |
| Compatibility | Arduino, Raspberry Pi |
| Supports | MIFARE 1k, ISO/IEC 14443-4 cards |
| Dimensions | 42.7mm x 40.4mm x 4mm |

Table 3 Specification of PN532A red circuit board with a white border

Description automatically generated

Figure 2 PN532

Figure 3 PN532

1. ACR122U USB NFC Reader

The ACR122U is a contactless smart card reader/writer using 13.56 MHz RFID technology which is compliant to the ISO/IEC 18092 standard for NFC. It supports MIFARE and ISO/IEC 14443 A and B cards and all types of tags.

|  |  |
| --- | --- |
| SPECIFICATION | |
| Protocol | USB CCID |
| Power Source | USB Port |
| Interface | 12C, SPI, HSU |
| Supports | ISO/IEC 18092 NFC, ISO 14443 Type A & B, MIFARE®, FeliCa |
| Dimensions | 98.0 mm x 65.0 mm x 12.8 mm |

Table 4 Specification of ARC122U

A white electronic device with a cord

Description automatically generated

Figure 4. ARC122U

1. RC522 Module

The RC522, is an RFID module with MIFARE transponders which is based off the Philips MF522-AN. It supports reading, writing and peer-to-peer communication. It is a small sized module that is compatible with Arduino and Raspberry Pi. It supports I2C.

|  |  |
| --- | --- |
| SPECIFICATION | |
| Operating voltage | 13,56MHz |
| Interface | I2C |
| Compatibility | Arduino, Raspberry Pi |
| Standard | ISO/IEC 14443A |
| Dimensions | 40 x 60mm |

Several blue electronic components

Description automatically generated with medium confidenceTable 5 Specification of RC522

Figure 5 RC522

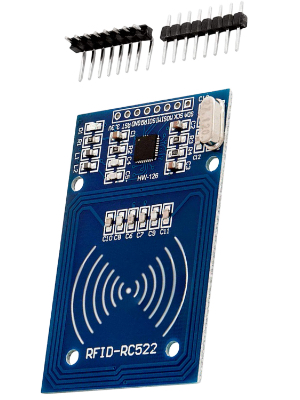


Figure 6 RC522

## **Hardware Design**

A diagram of a circuit board

Description automatically generatedIn Figure 4, we have provided a sample circuit diagram of what the hardware should look like when all the components are connected. It outlines the Raspberry Pi has as the heart of the system with the NFC Module connected to it which will read the tags and cards, the LCD display to display access messages, the relay module to act as an electronically operated switch controlled by the Raspberry Pi and would then control the power to the lock mechanism based on the validity of the NFC card.

Figure 7 Sample Circuit Diagram

To setup the Raspberry Pi, an OS must be installed. To do this, we download the Raspberry Pi OS. Then we format an SD card using an SD formatting software and then drag over the downloaded OS into the SD card. We then power on the Raspberry Pi, insert the SD card, and install the software on the microcontroller.

## **Architecture Design**

This system's architecture consists of a combination of hardware and software elements, each of which is essential to its operation. The purpose of this architecture is to guarantee secure operations, easy-to-use interfaces, and effective data processing.

NFC/RFID readers are the primary contact points for the users. Since this is a model of a system, each reader will have a different function and model a classroom, canteen area etc…, which will enable users of the system to interact with their NFC enabled cards or tags. Microcontrollers such as an Arduino Uno or Raspberry Pi can act as intermediaries between NFC readers and the server. They will process the data coming from the reader and forward it to the server for further processing. The server will act as the system’s backbone. It will process the data from the microcontroller, interact with the database and manage user interfaces and API calls.

The servers will run on a reliable and secure operating system based on the demands of the system. A database management system like MySQL or Microsoft SQL Server can be used for data storage and management. Software developed on the server side will handle tasks such as user authentication, data processing and API management. All data processing between components must be encrypted to ensure data protection and security.

Based on this, in Figure 4, we have outlined a UML sequence diagram which depicts the interactions between various components of the system during specific use-case scenarios.

The actors and components include a User Interface which the administrators and users interact with, Microcontroller which is the hardware controlling the logic of the NFC module and access control through the door lock, NFC Module which reads data from the student cards, Door Lock which is the model of the access control which can be unlocked by the NFC card, Server which is the central system that manages all the data and authorisation, Attendance System is the subsystem responsible for tracking attendance, Parking System which manages parking access and payments and Canteen System which manages canteen transactions for payments and records.

The scenarios and interactions identified are as follows:

* For Access Control, the process begins when the User Interface sends a request to set permissions and rules to the Microcontroller.
* When the student card is authenticated and validated by the NFC Module, the Microcontroller can unlock the door if access is granted.
* The Server logs the entry and if the attendance system is part of the interaction, it records the attendance.
* If the access is denied, the door remains locked, and the User Interface displayed an access denied message.
* For System Configuration admins can manage the system settings through the User Interface, updating the configurations on the server which then updates the Microcontroller settings.
* For Paid Parking the Server manages the parking payments by processing fees and logs the transactions.
* A screenshot of a computer screen

  Description automatically generatedIt does the same for the Canteen Transactions, where it processes the payments and records them.

Figure 8 Sequence Diagram

## **Interface Design**

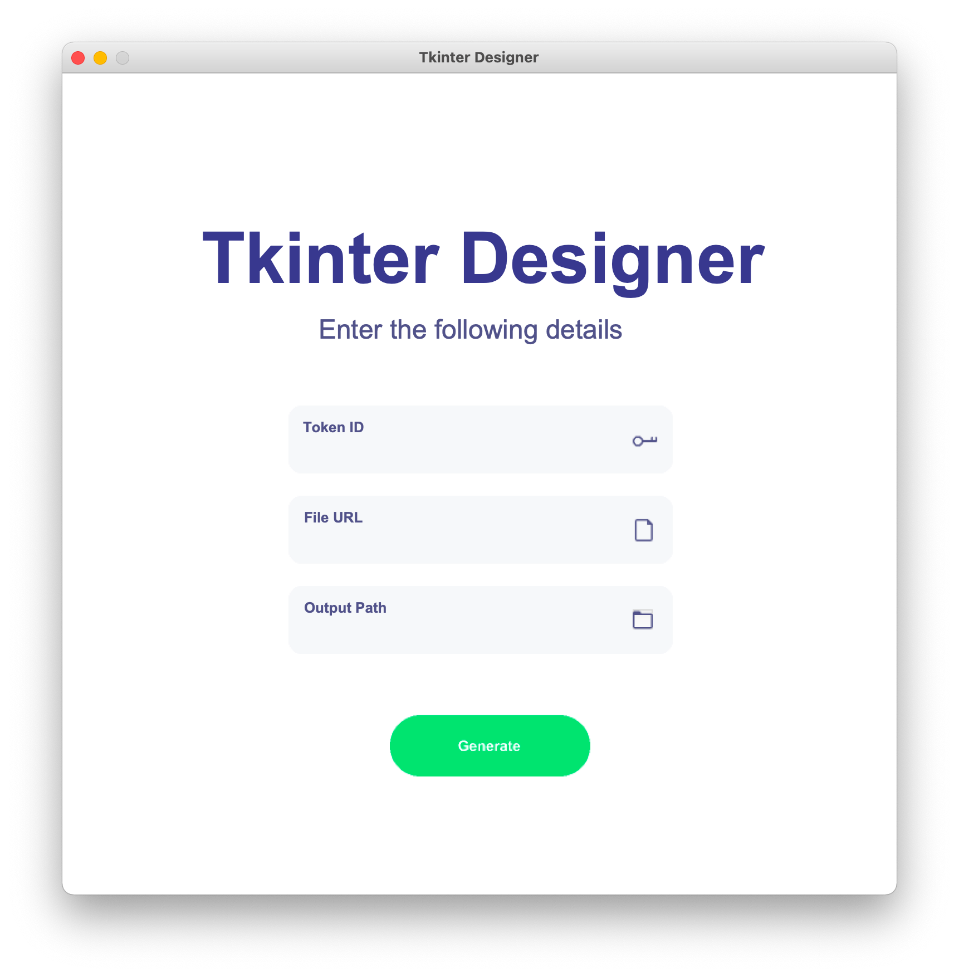
[](https://pythonrepo.com/repo/ParthJadhav-Tkinter-Designer-python-graphical-user-interface-applications#google_vignette)Ensuring that the UI design is clear, efficient, and intuitive for the administrators ensures that they can have a good user experience, reduce errors, and streamline system operations. Administrators should be able to manage access rights, view logs and conduct other necessary operations in an effortless manner. The UI should be laid out logically and ensure all features are easily accessible. It should have relevant information upfront such as recent logs or transaction records which should be presented in a well-organised table format. It should be easy to navigate between different sections with the use of icons and labels to guide the users through the system. The interfaces buttons should be clearly labelled and controls for adding new users or updating new permissions should be clear from the rest of the operations. With these actions there should also be confirmation prompts to ensure the admins are aware of any changes they are making. To accelerate development process and since this is only a model project, we can Tkinter or PyQT for simple GUI design and implementation.

Figure 9 Tkinter Sample Interface

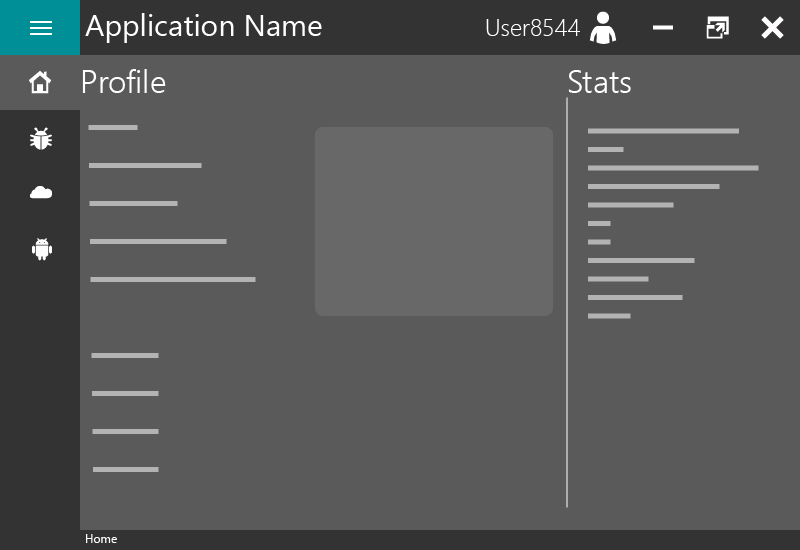
[](https://github.com/anjalp/Minimalistic-Flat-Modern-GUI-Template)

Figure 10 Pyqt Sample Interface

## **Alternative Design**

In the event where some of the hardware fails or the implementation does not go to plan, it is important to have a plan in place to substitute the hardware if needed. In this case if the Raspberry Pi is unusable, we can substitute for an Arduino and in the case of the NFC modules, we can substitute the modules for the reader if they fail. During the implementation process we can highlight any failure and make substitutions throughout to ensure that the project is completed in a timely and effective manner which will ultimately be seen in the final design and solution.

## **Database Design**

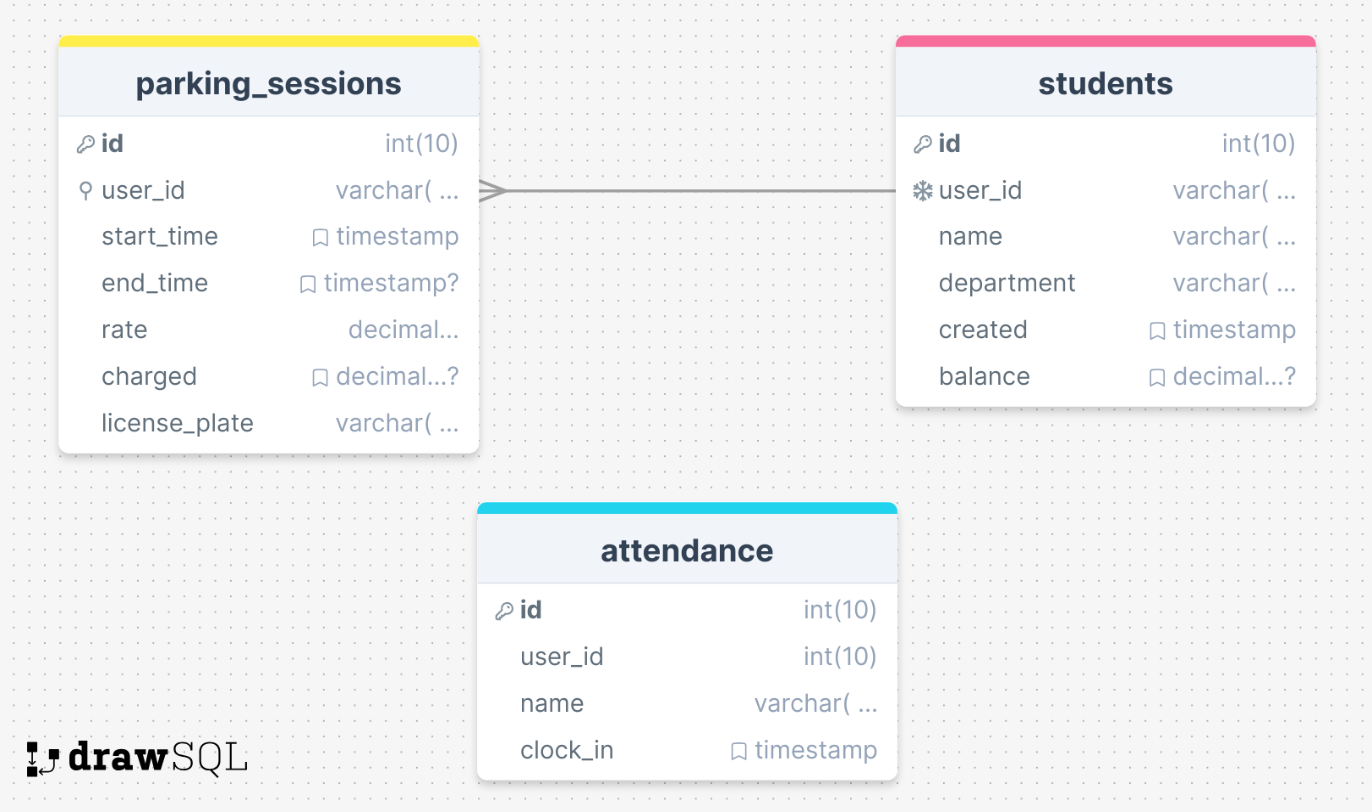
[](https://drawsql.app/teams/nfc/diagrams/nfcstudentsys)As this application will be reading information from each NFC enabled card or tag and will be storing information based each user, the application will need a database management system to store all of this information. For this system the RaspberryPi will act as the server and the client so all of the application does not need to run on a network unless you are trying to call the API from a different device. Since the whole system is relying on hardware integration and fast responses from a database, opting to use a local database on the microcontroller is best for this system. In a more centralised application, it would be more beneficial to use a cloud database as there would be multiple hardware points in a building.

Figure 11 Database Design

The database design of the system is shown in figure 10. As stated before, this entire database is being run locally on the RaspberryPi. This database does use MariaDB which makes it suitable for a small-scale application such as this one. It is open source and free to use which reduces the overall cost of the system.

The database only consists of 3 tables with the students table being the reference point of where all the user information is stored. All registered students and staff go into the students table where they are identified by their user\_id which stores the RFID\_UID of their designated NFC card, their name and department which is necessary in different parts of the application. The created attribute stores the information of when the user was registered and the balance references, how much money the user has on their card which is used in the parking and top up applications.

The parking\_sessios table stores all the information relating to a parking session. It does take information from the students table based on the RFID\_UID of each card. From there it can use the balance in order to deduct money from the balance when a user pays for parking, as well as storing the charged amount and license plate.

The attendance table keeps track of the attendance. While there is no relationship between attendance and students, the application does abstract the name in the code and stores it there and compared the scanned RFID\_UID to one in the students’ table.

## **Summary**

The NFC Enabled Student Card System project proposes using Near Field Communication technology to improve campus operations' usefulness and security. The system is aimed to serve students, lecturers, and administrative staff by providing smooth access management, accurate attendance tracking, and cashless transactions, encouraging a more connected, digital campus environment. The project considers both hardware and software requirements, with a focus on intuitive user experience and solid system performance. The NFC RFID Reader Writer and Raspberry Pi were chosen for their dependability and interoperability, while the software stack is based on strong database systems and programming languages like Python. The system's design emphasises modularity, scalability, and security, with an administrative user interface that provides a simple and effective management interface. To ensure project flexibility, an alternate design is being studied, which suggests using Arduino as a backup for the Raspberry Pi and additional NFC modules if necessary. This strategy ensures a durable and adaptive system capable of overcoming any implementation challenges, with the goal of achieving a decisive solution that effectively incorporates NFC technology into the fabric of campus life.

# **Implementation**

## **Introduction**

This chapter of the Final Year Project will discuss the implementation of the NFC Student Card System. We will outline the technical components, design decisions and practical implementation of this system. The NFC Student Card System is designed to improve and streamline processes for students, providing easy access control management and attendance tracking as well with API functionality. It combines hardware and software components to create a reliable and user-friendly experience for both students and staff.

The Technical Information section aims to provide an in-depth analysis of the technical foundations of the system. It will discuss the hardware elements such as NFC and RFID readers, RFID enabled tags and cards as well as the software architecture that enabled efficient data management and security. The design decision focuses on the crucial design choices that influenced the system, emphasizing being adaptable and scalable to meet the need of potentially changing the system if required.

The System Structure section will provide a detailed explanation, including the arrangement of the folders and files that are essential for it functioning. This section will go through the system’s architecture which includes the logic on the sever side, the interfaces that clients communicate with, and the protocols that regulate NFC/RFID transactions. This section discusses the key components of the system which include main.py for the primary application logic, check\_attendance.py for attendance tracking, save\_user.py for assigning a new student to a new student card, unlock.py for access control functionality, and app.py to demonstrate API functionality of the system. Their respective functions within the system are explained in detail.

The System Deployment and User Interaction section aims to provide an understanding of how the system is deployed and operated. It includes a demonstration of the interface that students use and the administrator dashboard that allows for system oversight. This section will provide visual examples and code snippets to provide a clear and detailed representation of the system’s functionality, encompassing student registration, daily access, and attendance tracking.

In conclusion, the Summary section will reflect on the process of implementing the NFC Student Card System. The section emphasizes the system’s influence on simplifying campus operations and improving the student experience. It also suggests the possibilities for future improvements and adjustments in response to the changing technology environment and campus requirements.

## **Technical Information**

### **System and Software Design**

Adopting agile and flexible methodologies in software development is essential for meeting specific requirements in system development. This project adopts the Explorative Software Development Methodology. This approach deviates from the traditional linear models by promoting a more iterative approach and incremental approach that encourages continuous feedback and adaptation throughout the development lifecycle.

The Explorative Software Development Methodology is distinguished by its iterative cycles of exploration, prototyping, and refining. The initial phase of this methodology includes developers working closely with stakeholders to define and understand the core requirements and objectives. In contrast to the rigid requirements phase in the Waterfall approach, this stage is unrestricted, enabling a thorough and flexible collection of requirements that can develop and change over the course of development. Following the initial research, the project progresses to a phase of fast prototyping. The primary objective is to create functional prototypes that encapsulate the identified features and functionalities. Following the prototyping phase, the system goes through a process of iterative testing and development. This phase resembles the system testing stage of the Waterfall methodology but is implemented continuously across each feature or module.

The verification phase entails the integration and validation of all system components to ensure that they meet the project’s goals and perform as expected. This phase is a repetitive process that takes place during the entire project life cycle, guaranteeing that the system is constantly matching the user requirements. Finally, the maintenance phase focuses on providing continuous support and improving the system. Given the explanatory nature of this methodology, maintenance not only involves bug fixing but also adapting and extending the system to meeting new requirements and incorporating new features as they become relevant. By utilizing the Explorative Software Development Methodology, the project gains an advantage from a flexible and user-focused approach, ensuring the system is functional and reliable but in line with requirements (GfG, 2023).

A diagram of a computer program

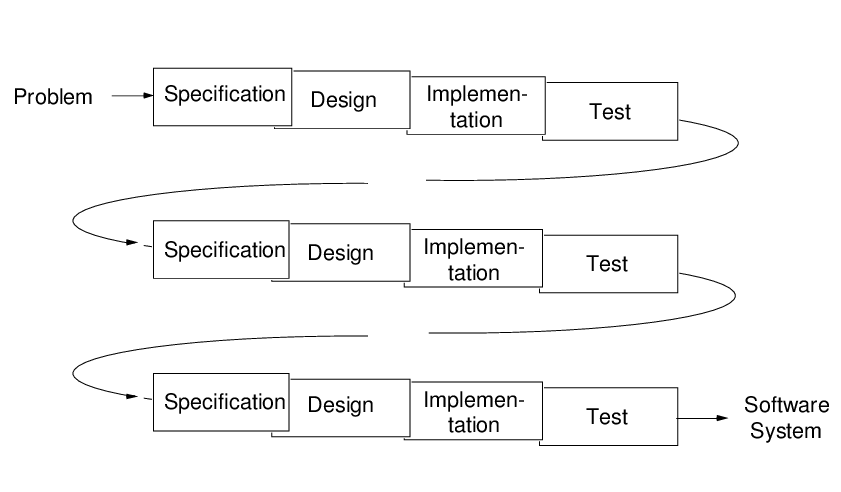
Description automatically generatedAlongside this development approach, software design is also necessary to produce a seamless combination of functionality, dependability, and user-friendliness. This can include and is not limited to architectural design, system definition, interface design, component design, data structure design and algorithm design. By following these, the system is designed to fulfil the requirements of the proposed system, enabling efficient access control management, attendance tracking and all the other proposed functionalities. Below in Figure 11 is an overview of the system architecture.

Figure 12 System Architecture

Figure 13 Exploratory Software Development

### **Circuit Design**

The circuit design of this system is centralised around a RaspberryPi connected to multiple peripheral devices for the applications proposed for the system. As shown in Figure 14, the RaspberryPi serves as the central processing unit and acts as the primary controller for the system. The RC522 RFID reader is the unit used to process all the RFID functionality in the system such as access control, top up of balance. The LCD display is used to present messages to the user derived from the system such as user prompts or error messages. The relay module is linked to an electronic solenoid lock to regulate power to activate the lock mechanism. The buzzer is used to notify the user if they have not been granted access for the access control method. The power source is used for the relay to provide power to the lock. The wiring establishes the connections between the different components, and the GPIO pins on the RaspberryPi, enabling the transmission of data and control signals.

A close-up of a computer

Description automatically generated

Figure 14 A top level view of the physical circuit

### **Project Management**

Within the field of software project management, the primary goals of following established timelines and remaining within the allocated budget are of the highest priority. Despite being motivated by academic and practical reasons; this project recognised the importance of quickly providing a fully functional system to end-users by a deadline. Punctuality was essential to prove the project's practical significance within the educational technology field.

Software development, especially in the field of educational technology, sometimes faces the difficulty of precisely predicting project deadlines, especially when it involves innovative technologies or approaches. Planning and schedule estimation in this case were flexible processes that adjusted as the project progressed. Strategic project milestones were set up to track the different stages of development and guarantee continuous development.

The project required a developer to handle several responsibilities such as project management, design, programming, and testing. This integration of tasks provided a complete view but also required the developer to possess a strong sense of self-motivation and discipline, in addition to external supervision.

A significant milestone in the project was the creation and implementation of the user interface using Tkinter which is Python’s native GUI toolkit. This package facilitated the development of a clear user interface within the codebase itself, ensuring smooth integration with the underlying Python software. The integration of a graphical user interface into the system improved user interaction and experience.

The following developments included the creation of Python modules for managing users, tracking attendance (check\_attendance.py), and controlling access (unlock.py). These modules established communication with both the Raspberry Pi server and the local database. Determining the timeframe of development was most noticeable for the components that interacted with hardware. Modules like the I2C LCD display (I2C\_LCD\_driver.py) and the SimpleMFRC522 package provided quicker solutions to building the project as they had come with predefined functionalities to interact with the hardware.

The project progressed through many stages, starting with its initiation, after the establishment of the Raspberry Pi server and the development of a local database, and concluding with the deployment of the user interface and functional modules. Each step was planned and implemented, resulting in reaching the proposed requirements of the project. Each achievement served as more than simply a point of evaluation, but rather as a chance to improve, acquire knowledge, and adjust, guaranteeing that the result not only achieved its original goals but also was strategically prepared for future improvements.

### **Design Quality**

The project placed a strong focus on the aspect of design quality. Factors like cohesion, interaction, comprehension, and flexibility have an impact on the quality of a software design. Cohesion is the extent in which the components inside a module are interconnected and contribute to a unified functionality. The Python modules were designed with a focus on achieving high unity. This means that each module, such as the user interface (main.py) or attendance tracking (check\_attendance.py), was specifically created to handle a single component of the system's operation and keep its concentration on that aspect.

Coupling, which measures the level of interconnection among various modules, was controlled to maximize the project places a high emphasis on software reliability, given its intended demographic of inexperienced users who may not have availability of technical assistance. To ensure dependability, the system was developed with the intention of reducing the likelihood of errors. The loose coupling played a crucial role in enabling the modification or replacement of modules with minimal errors for the rest. For example, changes made to the registering a user logic in the ‘save\_user.py’ script did not have any impact on the parking system procedures in ‘parking\_system.py’.

Simplicity is also essential; the code needed to be easily comprehendible to facilitate maintenance. This was accomplished by sticking to consistent naming conventions, creating thorough documentation, and keeping a manageable degree of complexity. Every Python file was given a logical name (e.g., ‘unlock.py’ for access control) that clearly indicated its purpose inside the system.

A screenshot of a computer

Description automatically generated

Figure 15 A screenshot of naming conventions in the project

Another important goal in the design was to provide adaptability, considering future requirements and the possibility of system evolution. The project's design facilitated the implementation of updates, whether in reaction to new needs or the addition of additional hardware components, with little difficulty.

The initial configuration and database scripts of the Raspberry Pi server were created with identical ideas in consideration. The bash scripts developed for database management were designed to be uncomplicated and modular, facilitating comprehension and modification as required. This approach prioritised clarity and simplicity over the utilisation of the most advanced or complex solutions, even when incorporating new technologies.

The system's architecture was designed to embody these concepts, with every component, ranging from the Python modules to the Raspberry Pi server configuration, demonstrating robust internal unity, little external interconnection, high comprehension, and the ability to be easily modified. By adhering to these design qualities, the project established a strong basis for a resilient, sustainable, and expandable system.

### **Program Reliability**

The project places a high emphasis on software reliability, as the intended demographic of users may not have immediate technical help or will be very new to the system. The system interactions are designed to be user-friendly, including straightforward mouse clicks or typed inputs for specified functions, such as recording attendance or granting access. The minimalistic design minimizes the probability of errors caused by users. It is essential to guarantee the dependability and user-friendliness of all features. If users experience problems or see the software as unreliable, it could greatly impact their trust and willingness to interact with the system.

Integrating a strong error management is crucial for improving system dependability, particularly when the users of the system may not technically proficient. The project does provide comprehensive error logging to ensure that the user and developer understand what the errors are. The system provides user-friendly error messages that offer clear, non-technical explanations and meaningful recommendations to ensure the user know how to action the next time they run the application and avoid these failures.

A screen shot of a computer program

Description automatically generated

Figure 16 A screenshot of error handling

A computer screen shot of text

Description automatically generated

Figure 17 A screenshot of error handling

For this system, threading was essential to ensure a responsive user interface while managing background tasks such as database access and hardware interfacing. By running various tasks in separate threads, the system ensures that operations such as RFID scans and database changes do not block the main user interface, enabling it to remain responsive. This methodology significantly enhances user experience by minimising waiting times and enabling simultaneous processing of multiple tasks. Threading plays a key role in improving system performance and reliability ensuring that even during complex processes, the user interface remains responsive and efficient. (Bauer, 2017)

A screen shot of a computer program

Description automatically generated

Figure 18 A screenshot of threads

Using these design methods, the project guarantees a dependable and user-focused experience, which enhances the users' trust and creates an ideal environment for technological engagement, especially for individuals with minimal computer literacy.

### **Software Reuse**

The goal of the software produced in this project is to ensure its reusability across multiple scenarios. To accomplish this, the project follows specific design principles. It was created with a strong focus on modularity, using Python files as the basic components of the system. The naming rules were carefully deliberated to guarantee clarity and consistency throughout the codebase. The reusable components, including distinct functions and classes, were contained within separate modules for easy maintenance and enable potential reuse in various sections of the system as well as the future.

When designing our system, we took into consideration the built-in cross-platform capabilities of Python, ensuring that it was not just modular but also portable. The solution is meant to function across various computing settings and operating systems by utilizing Python's adaptability and the Raspberry Pi's extensive connectivity. This forward-thinking perspective defines the project as a viable option for future adaptations and integrations, regardless of technology improvements.

Another factor to consider was the system's capacity to manage data in a manner that would ensure long-term accessibility and convertibility. The principles of data independence and futureproofing were of importance. The system stores data, such as user information and access logs, in a standardized format that allows for simple export and manipulation if necessary.

Python scripts and the Raspberry Pi platform provide a promising opportunity for future research and development. With the constantly evolving technological environment and the development of new data management approaches, the project's organized approach and the decision to utilize widely supported technologies ensure that the data and software components may be easily adjusted or transferred to other formats and systems.

Essentially, the project includes the fundamental concepts of modularity, portability, and future adaptation, which are crucial for ensuring the software's durability and ongoing significance in the ever-changing technology landscape.

## **Design Decision**

**Why a Python-Based Dashboard?**

The choice to use a Python-based dashboard (main.py) for the RFID functionality of the project was inspired by various practical factors. The RFID system has a separate dashboard interface that provides a streamlined controlled environment for users to interact with as well has the circuit of the hardware. The system enables the integration of immediate input and interactive components, which are essential to controlling and overseeing operations that rely on the use of RFID.

Using a Python-based dashboard customised for the Raspberry Pi allows improved communication and user involvement, surpassing the capabilities of a web-based interface, especially in offline environments or areas that lack internet connectivity. It enables the user to easily move through various features, hence improving their independence and the overall usability of the system.

The use of the dashboard method is advantageous due to Python's extensive collection of libraries and frameworks. This resilient support system allows for the development of a highly adaptable and user-friendly interface for RFID interactions. The modular structure of Python code facilitates the reuse and scalability of the dashboard, enabling it to adapt to the evolving needs of the system.

Additionally, the widespread knowledge of Python within the development community ensures that developers with different degrees of skill may effectively maintain and enhance the dashboard. As the system usage expands, the dashboard can be improved to incorporate other functionalities such as data visualisation, user administration, and system configuration choices.

**Benefits of a Dashboard Approach**

1. Centralised Control: The dashboard serves as the central hub for all RFID-related functions, providing users with a single point of interface with the system.
2. Improved User Experience: A carefully developed dashboard can significantly improve the user experience by providing information and controls in a structured and easily accessible manner.
3. The Python dashboard shows adaptability and extensibility, enabling seamless adjustment to new demands, such as the addition of additional hardware functionalities or the enhancement of the user interface.
4. Operational Independence: In contrary to web-based solutions, the dashboard functions independently of web connectivity, enabling its operation in diverse situations, such as remote or isolated places.
5. Community Support and Resources: Python's vast community provides developers with a diverse range of resources for resolving issues and improving the dashboard.

The project's design is based on the expectation of future expansions and the requirement for a dependable, user-focused interface. The dash.py dashboard demonstrates a dedication to these principles, guaranteeing that the software stays a robust and adaptable tool for managing RFID systems.

**Why an API?**

JSON (JavaScript Object Notation) and APIs (Application Programming Interfaces) are foundational technologies that facilitate the integration and compatibility of modern software systems. Unlike HTML (HyperText Markup Language) is primarily intended for organising and displaying material on the web, JSON is specifically created for data exchange, offering a compact, text-based format that is easily readable by human and machines. APIs, especially those that use JSON, provide seamless communication between different software components, systems, or services, allowing them to exchange data and functionality. This section examines the justification for using JSON and API technologies in this project.

The core principles underlying JSON and APIs is to promote decoupled architecture allowing the exchange of data and services can occur independently of their internal implementation. This separation enhances adaptability, expandability, and ease of maintenance across multiple systems. JSON can be used to serialize and transmit structured data across a network, especially between server and web applications. APIs, in contrast, establish the protocols and data structures that other systems can use to exchange information or use each other’s capabilities. Together, they facilitate a modular approach for system design, allowing for the development, modification, or substitution of components without impacting the entire system.

**Main Advantages of JSON and APIs**

1. Interoperability: JSON is independent of any specific computer language, meaning it can be generated and interpreted by various programming languages, making it a suitable format for data exchange across different systems. APIs that employ JSON further enhance interoperability by offering a standardized method for applications to interact.
2. Efficiency and Performance: JSON's efficiency and performance requires less bandwidth, leading to faster data transfer rates and better performance compared to other data interchange formats. APIs that are designed for JSON data exchange contribute to this efficiency, enabling real-time data processing and responsiveness.
3. Simplicity and Accessibility: JSON's format is straightforward, making it easy to write and understand without the need for specialized tools or parsers. APIs that communicate via JSON inherit this simplicity, allowing for quick development cycles and easier integration.
4. Flexibility: JSON data offers flexibility, allowing it to easily adapt to the specific needs of the application. APIs can use this flexibility to evolve over time, adding or modifying data fields without disrupting existing services.
5. Security: APIs can implement robust security protocols to protect data exchange, including authentication, authorization, and encryption. When combined with JSON's simplicity, this ensures secure yet accessible data communication pathways.

Although JSON and APIs provide multiple benefits, it is also important to consider their limitations and challenges, such as security implications of open APIs, the need for thorough documentation, and potential performance issues with poorly designed API interfaces. However, with proper design and implementation strategies, JSON and APIs remain pivotal technologies that support the modular, interconnected, and efficient systems that define the current world of software.

**Disadvantages of a Python Based Dashboard**

While a Python based dashboard for RFID functionality has several advantages using Tkinter, like centralised access control, enhanced user experience, and operational independence, it also presents certain challenges that need to be addressed. It is vital to recognise these issues to have a full understanding of the technical decision to use this type of technology and minimise any issues that may arise.

**Challenges and Limitations of a Python based Dashboard**

1. Performance Constraints: Python’s ease of use and versatility can be compromised when compared to compiled languages such as C or C++. Where real time data is being processed and high speed interactions are occurring, Python may impose limitations especially on low-powered devices such as the Raspberry Pi or Arduino.
2. Resource Intensive: Python applications, particularly those with GUIs (Graphical User Interfaces), tend to require more resources compared to alternatives written in lightweight or native frameworks. This can impact the performance of the system especially in environments like embedded systems or IoT devices.
3. Security Vulnerabilities: Any software project imposes security risks including applications built in Python. Python libraries and frameworks that are open-source can be vulnerable to attacks if there are not constantly updated secured.

While choosing a Python based dashboard for managing RFID functionalities present many advantages, it may also come with challenges that need to be accounted for. Understanding these limitations enables the development of more effective and secure ways for developing and implementing the dashboard ensuring it is useable and user-friendly as the project progresses.

**Disadvantages of JSON and APIs**

JSON and APIs also provide many benefits for modern software systems, including interoperability, efficiency, and flexibility but they also have their own issues and challenges that need to be addressed. These disadvantages are necessary for developers to identify to minimise issues that arise during the design and implementation phase.

**Challenges and Limitations of JSON and APIs**

1. Data Overhead: JSON being as lightweight as it is compared to other data exchange formats it still incurs some additional data due to its textual structure. This can have a considerable impact on performance and bandwidth utilisation, especially in application involving high volume transactions or the transmission of complex or deeply nested data structures.
2. Security Risks: APIs that can be publicly accessible pose vulnerabilities. Insufficient security measures, such as absence of rate limitation, encryption etc… can make APIs susceptible to data breaches, unauthorized access and DDOS attacks (Denial of Service).
3. Complexity in API Management: As systems grow, managing a collection of APIs can becoming more complex. Maintaining consistency, implementing API versioning without causing disruption to users and providing complex documentation for developers can demand substantial dedication and resources.

JSON and APIs are powerful tools for building interconnected software systems. Even though they provide many challenges, developers can counter this by designing fool-proof APIs with strong security measures and comprehensive documentation.

## **System Architecture**

The created system consists of various important components, each contributing to the overall functionality and user experience. The purpose of this section is to offer a brief overview of the system's design. This project uses Python scripts and a Raspberry Pi server to oversee and conduct its fundamental operations, adopting a distinct methodology.

### **Overall Structure**

The NFC Student Card System has an organisational structure that effectively separates and categorises its different functional components. This structure reflects the complex of the system operations and make it easier to navigate and maintain. The directory and file structure follows a hierarchical arrangement.

At the highest level, the system includes:

* The ‘FYP’ root directory: This serves as the central location for the project related materials, including the project’s source code and accompanying documentation and support files.

Within the root, the directory structure branches out into several key components:

* The ‘dependencies’ directory contains external dependencies.
* The ‘modules’ directory acts as the core of the system’s functionality. It contains the Python modules that constitute the system’s operations.
* The ‘tests’ directory hosts all the unit tests developed for the project.

The system is structured as:

* There are multiple functionalities in their own directory,
* Each one of these functionalities are then brought together into one main file which is encompassed by a GUI,
* There is an API file that displays a JSON file based on its functionality.

### **Dependencies Directory Structure**

The project's dependencies directory plays a crucial role in maintaining and storing the essential external libraries and drivers needed for the system to function properly. The layout and contents of this directory are intentionally selected to encapsulate the external dependencies of the system, enabling a separation from the core application functionality.

* The directory houses essential drivers or dependencies required for hardware connection. The ‘I2C\_LCD\_driver.py’ which was obtained from [vay3t](https://gist.github.com/vay3t/8b0577acfdb27a78101ed16dd78ecba1)’s implementation of [Denis Pleic’s ‘RPi\_I2C\_driver.py’](https://gist.github.com/DenisFromHR/cc863375a6e19dce359d), provides a means of connecting and communicating with the LCF display via I2C communication protocol. It simplifies the direct contact with hardware by abstracting its complexity, offers a streamlined approach for the project.
* The ‘\_init\_.py’ file located in the dependencies directory, serves as a marker that designates the directory as a Python package module. This allows the content of the directory to be imported and used modularly in other portions of the project. It signifies the directory’s role in the overall framework.

The directory for dependencies can be organised to incorporate version details or configuration files that describe precise versions or customised settings of the dependencies utilised. This facilitates the maintenance of a consistent development and deployment environment.

A screenshot of a computer program

Description automatically generated

Figure 19 Sample code from I2C\_LCD\_driver.py

### **Modules Directory Structure**

The modules directory in the project, is structured to contain the essential features needed for the system’s operations. The main logic and process flows are located here.

* The ‘\_init\_.py’ file located in the dependencies directory, serves as a marker that designates the directory as a Python package module. This allows the content of the directory to be imported and used modularly in other portions of the project. It signifies the directory’s role in the overall framework.
* The ‘app.py’ modules as the forefront for the API logic and includes the functionality of displaying students that signed in for attendance within the last hour.

A screenshot of a computer program

Description automatically generated

Figure 20 app.py Code Snippet

* The check\_attendance.py’ module is responsible for the attendance tracking feature of the system. It handles the logic for recording and verifying the students that have signed in for class, interfacing with the RFID readers to collect and validate attendance data.

**A screenshot of a computer program

Description automatically generated**

Figure 21 check\_attendance.py Code Snippet

* The ‘main.py’ module houses all the functionality in one file to allow for flexibility in code and be wrapped around a GUI. In includes threading, control flows, setting up the environment etc…

**A computer screen shot of a program

Description automatically generated**

Figure 22 main.py Code Snippet

**A screenshot of a computer program

Description automatically generated**

Figure 23 main.py Code Snippet

* The ‘save\_user.py’ module contains the logic for adding new students to the system and assigning their department as it is crucial for the access control part of the system.

**A screenshot of a computer program

Description automatically generated**

Figure 24 save\_user.py Code Snippet

* The ‘scan.py’ module is basic module that contains logic for the RFID tag IDs for the admin to see.

**A screenshot of a computer program

Description automatically generated**

Figure 25 scan.py Code Snippet

* The ‘unlock.py’ module contains the logic unlocking based of the authentication of the card presented.

**A screenshot of a computer program

Description automatically generated**

Figure 26 unlock.py Code Snippet

### **Completed System**

A screenshot of a computer

Description automatically generated

Figure 27 A screenshot of the main system

A screen shot of a parking system

Description automatically generatedFigure 28 A screenshot of the parking system

A screenshot of a computer

Description automatically generated

Figure 29 A screenshot of the top up system

A small electronic device with a blue screen

Description automatically generatedThe completed system works as 3 part system. The first application is where the users can either interact with the main system which hosts the API call, unlocking a door, registering a user and logging attendance. The second system is where the user can log their parking session. The third system is where the user can update the balance to allow for more money for parking. All three systems have been developed using Python as the lanaguage and Tkinter as the GUI design. As well as the GUI, the user does have the hardware in front of them, the LCD will be displaying information to them as they progress through the application.

Figure 30 A picture of the LCD displaying messages

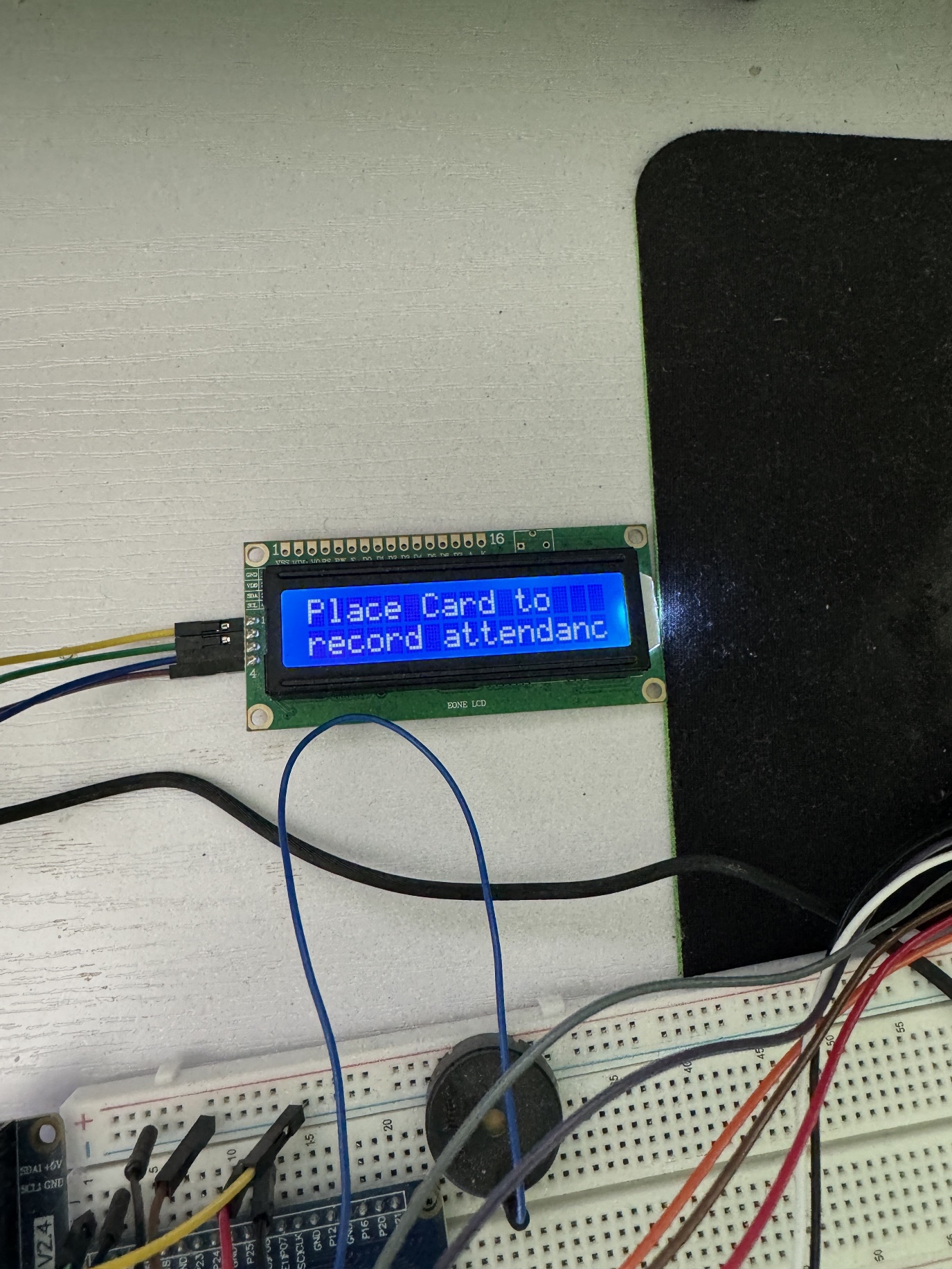
All the functionality for the user is laid out in one place for each program. There is no authentication required for the system, the user just has to click the buttons on the application in order to run the different methods. Each button corrersponds to each method call and is clearly labelled. When the user runs main.py, the first thing they will see 4 buttons that are clearly labelled for their assigned methods, a drop down of the department, a text box to input names and an output box to display messages.

Figure 31 A picture of the LCD displaying messages

The first button on main.py is the button to start the API server and open the browser. It will take the last hour of recorded attendance from the database and display it on the browser. If there is nothing in the past hour to be called it will just return an empty page.

A screenshot of a computer

Description automatically generated

Figure 32 A screenshot of the API call for attendance tracking

From there the user can choose between, ‘Unlock Door’, ‘Register User’, or ‘Attendance’. The first two methods do require inputs from the user. For ‘Unlock Door’, the input required is the department. As default IT is set as the default department, but there is a dropdown menu that the user can select from in order to set the department.

A screenshot of a computer

Description automatically generated

Figure 33 A screenshot of the department dropdown

A screenshot of a computer

Description automatically generatedOnce the department is set the user clicks ‘Unlock Door’ and they are prompted to scan their card. Once the card is scanned, it matches the scanned RFID\_UID to one in the database and checks if the department matches the one that has been selected. If the RFID\_UID is found and the department for that user matches what as been selected in the application, a message is displayed informing that they have received access, as well as a buzzer in the hardware to notify the user that access have been granted.

Figure 34 A screenshot of the Unlock Door method

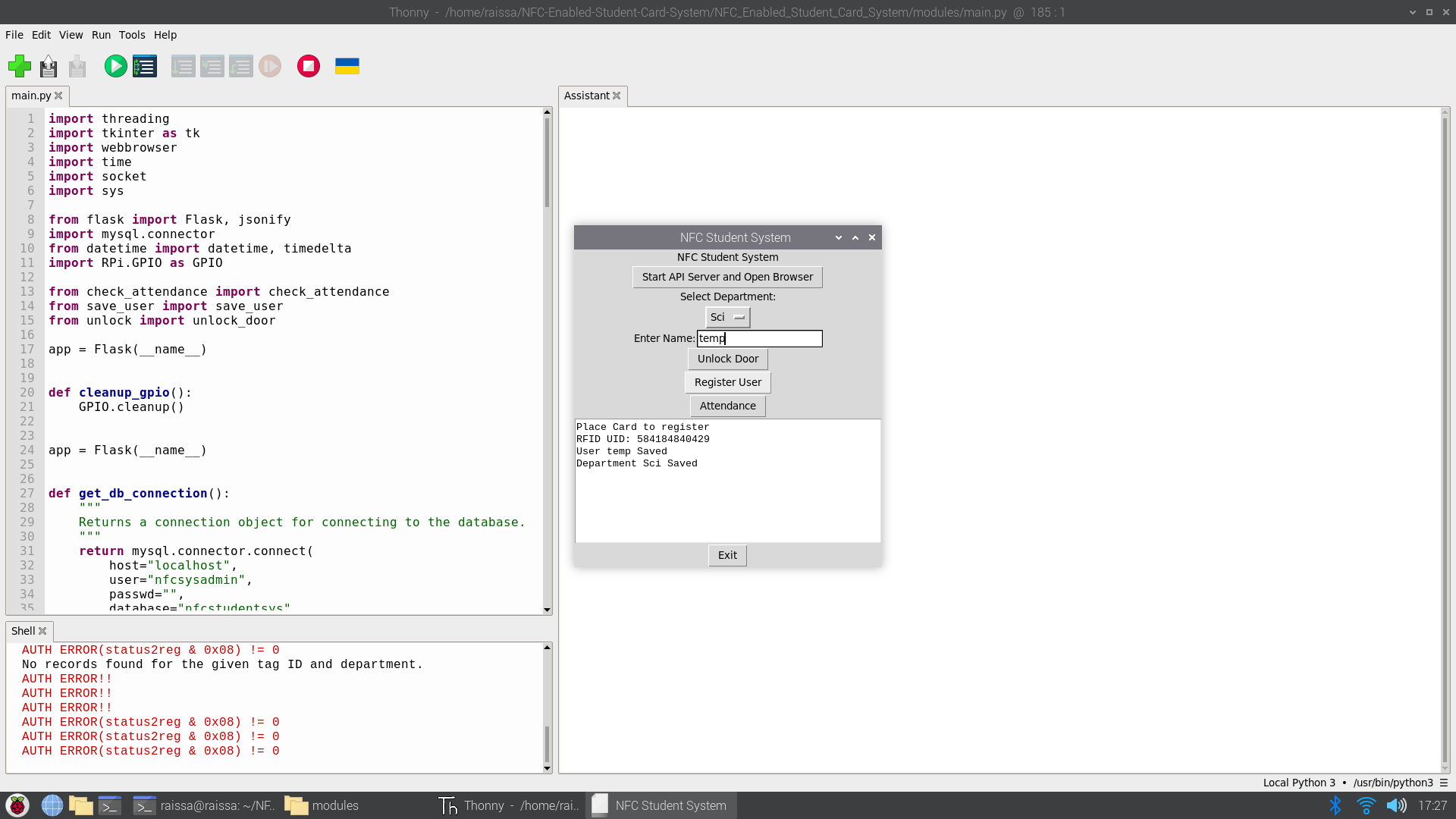
From there, the user can then set the department once again and then enter a name in the text box provided. The user then clicks ‘Register User’ which prompts the user to scan their card. It takes the scanned RFID\_UID and finds it in the database, if the RFID\_UID already has been assigned to a user in the database, the system displays a message that there is already a registered user. If there is not a registered user, the system will display a message that the user and their department has been found.

Figure 35 A screenshot of the Register User method

From there, the user can click ‘Attendance’. There are no inputs required from the user apart scanning the card. When the user clicks the button, they are prompted to scan their card to record attendance. If the user is found it responds that they have signed in, other wise it will display that the user is not found.

A screenshot of a parking system

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure 36 A screenshot of the parking method

Figure 37 A screenshot of the attendance system

For parking\_system.py, it required 3 inputs; the amount of hours required for parking, the license plate and the card to scan. Before the user clicks ‘Start Parking Session’, the must input the amount of hours for parking and the license plate. Once this information has been entered and the user clicks the button, it prompts the user to scan their card. Once the card is scanned, it completes the calculation and displays all the information related to parking. To the data base it saves the parking id, the RFID\_UID, the start time of the parking, end time of the parking, the rate charged and how much they’ve been charged as well as inputted license plate. The rate of charging is 50 cent for the first 2 hours and then increases to €1 per hour, which is then deducted from the balance of the user which is stored in the students table. The system does flag an error if there is no hours inputted or if the hours are less that 0.

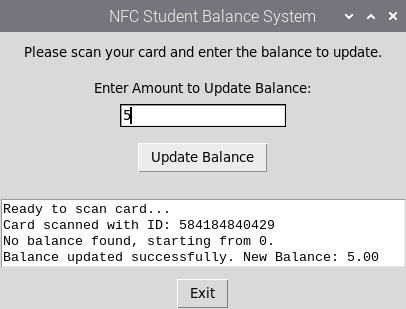


Figure 38 A screenshot of the top up method

For top\_up.py, it requires 2 inputs; the amount the user wishes to top up by and the card to scan. Similary to the parking system, the user must input the amount they wish to update their balance by. Once that has been entered the user can click the button. It will then prompt the user to scan their card. Once it has been scanned, it will find whether or not the balance is null or has a prexisting balance. If the balance is null it will inform the user that they are starting a new balance or it will be added onto the current balance.

A screenshot of a computer

Description automatically generated

Figure 39 A screenshot of the top up method

All of the methods in this application are threaded meaning its able to execute multiple parts of the system at the same time. For example, when the API is executed it does not interfere with the GUI and other processes, the user is able to continue to use the application without interference. When the user clicks the ‘Exit’ it ends all the processes, clears the GPIO pins and closes the window.

# **Testing**

## **Introduction**

## **Testing Methodologies**

## **Test Cases**

## **Automated Testing**

## **Test Results**

## **Test Conclusion**

# **Conclusion**

## **Summary of Research**

## **Practical Implications**

## **Recommendations**

## **Final Thoughts**

## **Closing Remarks**

# **References**

AirFocus (2020). *What Is an API? API Definition, Examples, Benefits, Challenges & FAQ*. [online] Airfocus.com. Available at: <https://airfocus.com/glossary/what-is-an-api/> [Accessed 5 Nov. 2023].

Amazon Web Services, Inc. (2023). *What is an API? - Application Programming Interface Explained - AWS*. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/what-is/api/> [Accessed 5 Nov. 2023].

Badra, M. and Badra, R.B. (2016). A Lightweight Security Protocol for NFCbased Mobile Payments. In: *Procedia Computer Science*. [online] Elsevier, pp.705–711. doi:<https://doi.org/10.1016/j.procs.2016.04.156>.

Bauer, R. (2017). Understanding Programs, Processes, and Threads - Roderick Bauer - Medium. [online] Medium. Available at: https://medium.com/@rodbauer/understanding-programs-processes-and-threads-fd9fdede4d88 [Accessed 20 Apr. 2024].

Birbeck, University of London (n.d.). *Oyster -Smart Card Payment Systems on the London Public Transport Network*. [online] Available at: <https://titan.dcs.bbk.ac.uk/lo/fi/learningresources/session2/fit/oyster.pdf> [Accessed 2023].

Brophy, M. and King, M. (2023). *22 Contactless Payment Statistics for 2023*. [online] Fit Small Business. Available at: <https://fitsmallbusiness.com/contactless-payment-statistics/> [Accessed 14 Oct. 2023].

Carrilho, R. (2022). *5 Reasons Why NFC Technology Will Thrive In Future Events*. [online] beamian. Available at: <https://beamian.com/why-nfc-technology-will-thrive-in-future-events/> [Accessed 6 Nov. 2023].

cj&co (2023). *NFC Meaning: How This Tiny Tech Is Revolutionizing Our World | CJ&CO*. [online] CJ&COTM. Available at: <https://www.cjco.com.au/article/nfc-meaning/> [Accessed 28 Oct. 2023].

Curoe, M. (2021). *5 Common API Integration Challenges And How To Overcome Them.* [online] Redwoodlogistics.com. Available at: <https://www.redwoodlogistics.com/insights/what-are-some-of-the-barriers-facing-api-integration> [Accessed 5 Nov. 2023].

Cvrček, T. and Dzurenda, P. (2022). Access Control System Using Multifactor Authentication. In: *Proceedings II of the Conference Student EEICT*. Brno University of Technology, pp.190–193. doi:<https://doi.org/10.13164/eeict.2022.190>.

Cvrček, T., Dzurenda, P., Mansur, K., Hasanuddin, Zulfajri B, Pireva, Krenare R, Siqeca, J., Berisha, S., Francis, L., Hancke, G., Mayes, K., Markantonakis, K., Birbeck, ISO/IEC, Rahul, A., G, Gokul Krishnan, H, Unni Krishnan, Rao, S., Prohel, G., Tabuk, University of and and, E. (2020). ‘Digital era’: Impact on the economy and the education system (country analysis). *Utopia y Praxis Latinoamericana*, [online] 25(Extra10), p.170937053109325113352137190-193-140-52-144-60-46-120-11-711-93-186. doi:<https://doi.org/10.5281/zenodo.4155437>.

Dixon, J. and Abdel-shakour Abuzneid (2020). An NFC Based Student Attendance Tracking/Monitoring System Using an IoT Approach. [online] doi:<https://doi.org/10.1109/csci51800.2020.00201>.

Francis, L., Hancke, G., Mayes, K. and Markantonakis, K. (2012). *Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones*. [online] Available at: <https://eprint.iacr.org/2011/618.pdf>.

Frye, M.-K. and MuleSoft (2023). *What is an API? (Application Programming Interface) | MuleSoft*. [online] MuleSoft. Available at: <https://www.mulesoft.com/resources/api/what-is-an-api> [Accessed 5 Nov. 2023].

Future Market Insights (2022). *Near Field Communication (NFC) Market by Product Type, Application & Region - Forecast 2022 to 2032*. [online] Futuremarketinsights.com. Available at: [https://www.futuremarketinsights.com/reports/near-field-communication-nfc-market.](https://www.futuremarketinsights.com/reports/near-field-communication-nfc-market#:~:text=Near%20Field%20Communication%20(NFC)%20Market%20Overview%20(2022%2D2032)&text=The%20sales%20of%20NFC%20are,automotive%20and%20contactless%20payment%20sectors.) [Accessed 5 Nov. 2023].

Gapsalamov, A.R., Bochkareva, T.N., Akhmetshin, E.M. and Vasilev, V.L. (2020). ‘Digital Era’: Impact on the Economy and the Education System (Country Analysis): ‘Era digital’: impacto en la economía y el sistema educativo (análisis de país).. *Utopia y Praxis Latinoamericana*, [online] 25, pp.170–189. doi:<https://doi.org/10.5281/zenodo.4155437>.

GfG (2023). *Introduction to Exploratory Style of Software Development*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/introduction-to-exploratory-style-of-software-development/> [Accessed 18 Mar. 2024].

Global Payments Integrated (2014). *The History of Contactless Payments*. [online] Globalpaymentsintegrated.com. Available at: <https://www.globalpaymentsintegrated.com/en-us/blog/2020/09/15/the-history-of-contactless-payments> [Accessed 15 Oct. 2023].

Hendry, M. (2014a). Modes of operation. In: *Near Field Communications Technology and Applications*. Cambridge University Press, pp.31–46. doi:<https://doi.org/10.1017/cbo9781107446854.006>.

Hendry, M. (2014b). Security. In: *Near Field Communications Technology and Applications*. Cambridge University Press, pp.109–120. doi:<https://doi.org/10.1017/cbo9781107446854.012>.

Hendry, M. (2014c). What is NFC? In: *Near Field Communications Technology and Applications*. Cambridge University Press, pp.3–11. doi:<https://doi.org/10.1017/cbo9781107446854.002>.

HST Software Solutions (2021). *Tokenization of Cards - What it is, how it works and the main benefits -*. [online] HST Software Solutions. Available at: <https://www.hst.com.br/tokenization-of-cards-what-it-is-how-it-works-and-the-main-benefits/> [Accessed 28 Oct. 2023].

Hussein al ofeishat (2012). *Near Field Communication ( NFC )*. [online] ResearchGate. Available at: <https://www.researchgate.net/publication/316543104_Near_Field_Communication_NFC> [Accessed 14 Oct. 2023].

IONOS editorial team (2023). *Near-Field Communication (NFC)*. [online] IONOS Digital Guide. Available at: <https://www.ionos.com/digitalguide/server/know-how/nfc-near-field-communication/> [Accessed 24 Oct. 2023].

ISO (2023). *ISO/IEC 14443-1:2018*. [online] Iso.org. Available at: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:14443:-1:ed-4:v1:en> [Accessed 25 Oct. 2023].

ISO/IEC (2013). *Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)*. [online] *standards.iso.org*, Switzerland: ISO, p.52. Available at: <https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html> [Accessed 25 Oct. 2023].

Jaffery, A. (2022). *5 API Trends to Look Forward to in 2023*. [online] Astera. Available at: <https://www.astera.com/type/blog/5-api-trends-to-look-forward-to-in-2023/.> [Accessed 6 Nov. 2023].

Jagaad Software House (2023). *5 Challenges in API Integration and How to solve or avoid them | JAGAAD Digital Solutions*. [online] Medium. Available at: <https://medium.com/jagaad/overcoming-5-common-challenges-when-integrating-apis-e1e10d865d22> [Accessed 5 Nov. 2023].

Kaspersky (2020). *Secure Element*. [online] Kaspersky.com. Available at: <https://encyclopedia.kaspersky.com/glossary/secure-element/> [Accessed 25 Oct. 2023].

Kolya Hnatyuk (2023). *130+ API Statistics: Usage, Growth & Security*. [online] MarketSplash. Available at: <https://marketsplash.com/api-statistics/>[Accessed 14 Oct. 2023].

Kong Inc (2023). *What is an API? Use Cases and Benefits*. [online] Kong Inc. Available at: <https://konghq.com/learning-center/api-management/what-is-api> [Accessed 5 Nov. 2023].

Landt, J. (2005). *IEEE Xplore Full-Text PDF:* [online] Ieee.org. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1549751> [Accessed 14 Oct. 2023].

Linkedin.com. (2023). *What are the latest trends and innovations in NFC technology and how can you keep up with them?* [online] Available at: <https://www.linkedin.com/advice/1/what-latest-trends-innovations-nfc> [Accessed 6 Nov. 2023].

Lung, N. (2015). *Does NFC make some smartphones smarter than others?* [online] TecHKU. Available at: <https://www.engineering.hku.hk/tecHKU/2015/02/09/does-nfc-make-some-smartphones-smarter-than-others/> [Accessed 5 Nov. 2023].

Mansur, K. and Hasanuddin, Zulfajri B (2018). *Implementation of NFC for Smart Gate Access Control in Campus Area*.

Marcus, K. (2023). *Five Common API Integration Issues and How to Avoid Them - Spiceworks*. [online] Spiceworks. Available at: <https://www.spiceworks.com/tech/devops/guest-article/avoiding-apiintegration-issues/> [Accessed 5 Nov. 2023].

Marforio, C., Ritzdorf, H., Francillon, A. and Capkun, S. (2012). Analysis of the communication between colluding applications on modern smartphones. In: *Proceedings of the 28th Annual Computer Security Applications Conference*. [online] New York, NY, USA: ACM, pp.51–60. doi:<https://doi.org/10.1145/2420950.2420958>.

Microsoft Devices Blog and Microsoft Devices Team (2012). *Nokia’s NFC phone history*. [online] Microsoft Devices Blog. Available at: <https://blogs.windows.com/devices/2012/04/11/nokias-nfc-phone-history/> [Accessed 15 Oct. 2023].

Moore, J. (2023). *How have other countries managed contactless bus payments while Ireland has not?* [online] TheJournal.ie. Available at: <https://www.thejournal.ie/contactless-bus-payments-public-transport-ireland-6077000-May2023/> [Accessed 25 Oct. 2023].

NFC Forum (2019). *Fresh SmartPhone Statistics and What They Mean For You, NFC And The World*. [online] Nfc-forum.org. Available at: <https://nfc-forum.org/news/2019-12-fresh-smartphone-statistics-and-what-they-mean-for-you-nfc-and-the-world/> [Accessed 14 Oct. 2023].

Nguyen, S. (2022). *11 API Trends to Watch for in 2023 - DreamFactory Software- Blog*. [online] DreamFactory Software- Blog - API Management, Enterprise Integrations, Data Security and More. Available at: <https://blog.dreamfactory.com/11-api-trends-to-watch-for-in-2022/> [Accessed 6 Nov. 2023].

Nguyen, S. (2023). *The Top Four Challenges With API Development - DreamFactory Software- Blog*. [online] DreamFactory Software- Blog - API Management, Enterprise Integrations, Data Security and More. Available at: <https://blog.dreamfactory.com/the-top-four-challenges-with-api-development/> [Accessed 5 Nov. 2023].

Organ, P. (2024). *The Hidden Costs: How Manual Attendance Tracking Damages Schools and Disrupts Parent Engagement | Orah Blog*. [online] Orah.com. Available at: <https://www.orah.com/blog/hidden-costs-of-manual-attendance> [Accessed 11 Mar. 2024].

Paragon ID (2017). *The history of NFC | Paragon RFID*. [online] Paragon-rfid.com. Available at: <https://www.paragon-rfid.com/en/the-history-of-nfc/> [Accessed 14 Oct. 2023].

Person, C. (2022). *The ChameleonMini is a skeleton key for RFID*. [online] The Verge. Available at: <https://www.theverge.com/23411372/chameleon-mini-rfid-nfc-attack-proxmark3-keyless-card-reader> [Accessed 28 Oct. 2023].

Pireva, Krenare R, Siqeca, J. and Berisha, S. (2013). RFID: Management system for students’ attendance. In: *IFAC Proceedings Volumes (IFACPapersOnline)*. IFAC Secretariat, pp.137–140. doi:<https://doi.org/10.3182/201306063XK4037.00057>.

Pireva, Krenare R, Siqeca, J., Berisha, S., Francis, L., Hancke, G., Mayes, K. and Markantonakis, K. (2012). *Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones*. [online] *IFAC Proceedings Volumes (IFACPapersOnline)*, IFAC Secretariat, pp.137–140. doi:<https://doi.org/10.3182/201306063XK4037.00057>.

PR Newswire (2023). *Near Field Communication (NFC) market size to grow by USD 33.1 billion from 2023 to 2030; Humungous preference for cashless payments to augment the global market trends - Facts & Factor*. [online] Yahoo Finance. Available at: <https://finance.yahoo.com/news/near-field-communication-nfc-market-213000434.html?guccounter=1> [Accessed 5 Nov. 2023].

profylecard (2021). *A history of NFC technology. — Profyle Card*. [online] Profylecard.com. Available at: [https://www.profylecard.com/a-history-of-nfc-technology/.](https://www.profylecard.com/a-history-of-nfc-technology/#:~:text=Near%20Field%20Communication%20(NFC)%20technology,transfer%20music%20files%20between%20devices.) [Accessed 14 Oct. 2023].

Prohel, G. (2013). *STMicroelectronics Introduction to NFC*. [online] Available at: <https://www.mouser.com/pdfdocs/STMicroelectronicsIntroductiontoNFC.pdf>.

Purdue News Service (2023). *Purdue launches Purdue Mobile ID for students, allowing them to get around campus with just a simple tap of their mobile device*. [online] Purdue.edu. Available at: <https://www.purdue.edu/newsroom/releases/2023/Q2/purdue-launches-purdue-mobile-id-for-students-allowing-them-to-get-around-campus-with-just-a-simple-tap-of-their-mobile-device.html> [Accessed 5 Nov. 2023].

Rahul, A., G, G.K., H, U.K. and Rao, S. (2015). Near Field Communication (NFC) Technology: A Survey. *International Journal on Cybernetics & Informatics*, 4(2), pp.133–144. doi:<https://doi.org/10.5121/ijci.2015.4213>.

Reach Researcher (2023). *LinkedIn*. [online] Linkedin.com. Available at: <https://www.linkedin.com/pulse/near-field-communication-nfc-market-size-2023-2030/> [Accessed 5 Nov. 2023].

Research Union (2023). *Decoding the Mobile Access Control Platform Market: A Deep Dive into the Latest Market Trends, Market Segmentation, and Competitive Analysis*. [online] Linkedin.com. Available at: <https://www.linkedin.com/pulse/decoding-mobile-access-control-platform-market-deep-dive/> [Accessed 25 Oct. 2023].

Reva Rikat Asih, Tania Izza Sholikhah, Nabila Dhiya Ulhaq, Rahman, T., Mil'atuttoyyibah, Moechammad Sarosa and Putri Elfa Mas’udia (2022). NFC (Near Field Communication)-Based Canteen Self Service Application at SMA Pomosda Nganjuk. [online] doi:<https://doi.org/10.1109/ieit56384.2022.9967855>.

Roberti, M. and Violino, B. (2005). *The History of RFID Technology*. [online] RFID JOURNAL. Available at: [https://www.rfidjournal.com/the-history-of-rfid-technology.](https://www.rfidjournal.com/the-history-of-rfid-technology#:~:text=The%20Germans%20discovered%20that%20if,the%20first%20passive%20RFID%20system).) [Accessed 11 Mar. 2024].

Sametinger, J. and Stritzinger, A. (1996). *Exploratory Software Development with Class Libraries*.

Sims, G. (2014). *Everything you need to know about Host Card Emulation*. [online] Android Authority. Available at: <https://www.androidauthority.com/everything-need-know-host-card-emulation-347626/> [Accessed 25 Oct. 2023].

Sony (2023). *Sony Corporation - FeliCa - Overview of FeliCa - What is FeliCa ?* [online] Sony.net. Available at: <https://www.sony.net/Products/felica/about/> [Accessed 28 Oct. 2023].

Stanford.edu. (2023). *NFC Technology*. [online] Available at: <https://cs.stanford.edu/people/eroberts/cs201/projects/2010-11/NFCChips/rfid.html> [Accessed 14 Oct. 2023].

Statista (2019). *Global proximity mobile payment users 2023 | Statista*. [online] Statista. Available at: <https://www.statista.com/statistics/557959/global-mobile-proximity-payment-users/> [Accessed 25 Oct. 2023].

Statista (2020). *UK: contactless card issuance 2007-2020 | Statista*. [online] Statista. Available at: [https://www.statista.com/statistics/859911/contactless-cards-in-issuance-in-the-uk/](https://www.statista.com/statistics/859911/contactless-cards-in-issuance-in-the-uk/#:~:text=Since%20contactless%20cards%20were%20first,debit%20cards%20also%20became%20available.) [Accessed 15 Oct. 2023].

Statista (2023). *E-commerce payment methods market share 2026 | Statista*. [online] Statista. Available at: <https://www.statista.com/statistics/1111233/payment-method-usage-transaction-volume-share-worldwide/> [Accessed 25 Oct. 2023].

STMicroelectronics (2022a). *What is an NFC Chip? - STMicroelectronics*. [online] STMicroelectronics. Available at: <https://www.st.com/content/st_com/en/support/learning/essentials-and-insights/connectivity/nfc/nfc-chips.html> [Accessed 25 Oct. 2023].

STMicroelectronics (2022b). *What is an NFC Chip? - STMicroelectronics*. [online] STMicroelectronics. Available at: <https://www.st.com/content/st_com/en/support/learning/essentials-and-insights/connectivity/nfc/nfc-chips.html> [Accessed 25 Oct. 2023].

Stockman, H. (1948). *IEEE Xplore Full-Text PDF:* [online] Ieee.org. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1697527> [Accessed 14 Oct. 2023].

The Insight Partners (2020). *Mobile Access Control Platform Market to Grow at a CAGR of 27.1% to reach US$ 782.01 million from 2020 to 2027*. [online] The Insight Partners. Available at: <https://www.theinsightpartners.com/reports/mobile-access-control-platform-market> [Accessed 25 Oct. 2023].

Transact (2023). *Campus ID Solutions*. [online] Transact Campus. Available at: <https://transactcampus.com/solutions/campus-id> [Accessed 1 Nov. 2023].

University of Tabuk and and, E. (2020). *2020 International Conference on Computing and Information Technology (ICCIT1441).* [online] Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9213758>.

Waldman, A. (2021). *Researchers hack Apple Pay, Visa ‘Express Transit’ mode*. [online] TechTarget. Available at: <https://www.techtarget.com/searchsecurity/news/252507530/Researchers-hack-Apple-Pay-Visa-Express-Transit-mode> [Accessed 28 Oct. 2023].

Zhou, M., Chen, N., Yang, Y., Pan, C. and Zhou, X. (2022). Intelligent Access Control System. *2022 7th International Conference on Communication, Image and Signal Processing (CCISP)*. [online] doi:<https://doi.org/10.1109/ccisp55629.2022.9974224>.

Zimmerman, E. (2018). *Near-Field Communication Technology Supports Contactless Student ID at Universities*. [online] Technology Solutions That Drive Education. Available at: <https://edtechmagazine.com/higher/article/2018/12/near-field-communication-technology-supports-contactless-student-id-universities-perfcon> [Accessed 1 Nov. 2023].