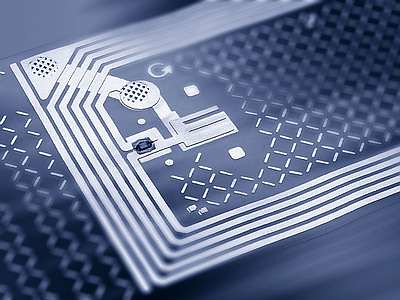
Herzing University

Joseph Cann

|  |
| --- |
| Implementation of RFID access controlled database system  By Alec Rasmussen |



[This Photo](http://www.blacklistednews.com/San_Antonio_NISD_Schools_Tracking_Kids_with_RFID_pilot_program/21219/0/38/38/Y/M.html) by Unknown Author is licensed under [CC BY](https://creativecommons.org/licenses/by/3.0/)

# Abstract

The use of RFID tags is everywhere, you may not see them, but they exist and have many different functions. If you use an ID tag to enter a building or hear a beep as you leave a store with a product, that is the use of RFID tags at work. They are used for many applications, from security cards to product tracking, and commonly as anti-theft devices. RFID tags are cheap, easy to make, and have different applications based on what function they are designed for. But RFID is not perfect; many places try to limit there use as they add a security risk of the card information being copied by an intruder allowing the intruder to get access to whatever the card is used for. This risk is something many companies do not work to fix when deciding on security and instead take it as a risk as it is an excellent way to keep many out of places they do not belong. Instead of relying on other security measures to make systems and buildings more secure such as the use of security guards, surveillance cameras,etc., there should be a focus on making the RFID tags themselves more secure.

Table of Contents

[Abstract 2](#_Toc27207962)

[Introduction 4](#_Toc27207963)

[Problem Definition 6](#_Toc27207964)

[Proposed Solution 8](#_Toc27207965)

[*Figure 1 WBS of Project* 9](#_Toc27207966)

[Technical Requirements 10](#_Toc27207967)

[Design MethOdologies/Project Design 11](#_Toc27207968)

[Figure 2 Agile Development Life Cycle (Image from (Anurina, 2019)) 11](file:///G:\capstone%20progress%20folder\Implementation%20of%20RFID%20access%20controlled%20database%20system%20final%20pape4r.docx#_Toc27207969)

[*Figure 3 Conext Diagram* 13](file:///G:\capstone%20progress%20folder\Implementation%20of%20RFID%20access%20controlled%20database%20system%20final%20pape4r.docx#_Toc27207970)

[*Figure 4 DFD Level-0* 14](file:///G:\capstone%20progress%20folder\Implementation%20of%20RFID%20access%20controlled%20database%20system%20final%20pape4r.docx#_Toc27207971)

[*Figure 5 DFD Level 1 Verify System* 15](#_Toc27207972)

[*Figure 6 Verify Database ER Diagram* 16](#_Toc27207973)

[*Figure 7 Project Timeline Gnatt Chart* 17](file:///G:\capstone%20progress%20folder\Implementation%20of%20RFID%20access%20controlled%20database%20system%20final%20pape4r.docx#_Toc27207974)

[Implementation 18](#_Toc27207975)

[*Figure 8 Verify card key against database key* 20](#_Toc27207976)

[*Figure 9 Creating new card user* 21](#_Toc27207977)

[*Figure 10 Basic UI* 21](#_Toc27207978)

[Project Testing 22](#_Toc27207979)

[Conclusion 24](#_Toc27207980)

[References 25](#_Toc27207981)

[*Figure 11 Verify Successfully result* 26](file:///G:\capstone%20progress%20folder\Implementation%20of%20RFID%20access%20controlled%20database%20system%20final%20pape4r.docx#_Toc27207982)

[*Table 1 Project Timeline Table* 26](#_Toc27207983)

[*Figure 12 Card denied three time* 27](file:///G:\capstone%20progress%20folder\Implementation%20of%20RFID%20access%20controlled%20database%20system%20final%20pape4r.docx#_Toc27207984)

[*Figure 13 Server Conenction failed screen* 27](file:///G:\capstone%20progress%20folder\Implementation%20of%20RFID%20access%20controlled%20database%20system%20final%20pape4r.docx#_Toc27207985)

[*Figure 14 Formatting function for keys* 28](#_Toc27207986)

# ****Introduction****

**Terminology**

**RFID**- Radio Frequency Identification Device

**Scanner**- RFID antenna that scans cards to read and write to them

**Raspberry Pi**- Small all in one computer that can be programmed to do many things

**Key**- Special sequence of numbers that are made uniquely for each card with each use

**UID**- Special number sequence that is hardcoded into an RFID card to help identify it

**User**- Employee that can access data from the database

**Database**- A type of computer service that organizes data given to it in a way that can be access in an easy manner. These systems are used usually for massive amounts of data organized in an easy to search way.

**Introduction**

Using RFID tags as a security device for a database is something that could make accessing easier. The project I have developed is to show how such a system could be built. The system developing is going to be a portable python based application running on a Raspberry Pi. The Raspberry Pi allows access to a database using secured RFID cards that have changing keys with each use. The system encompasses many skills and ideas that were taught and learned during the completion of the curriculum. The project also shows how I was able to learn new technologies and skills while working on the project.

The project's entire scope was to make a simple working system to demonstrate the simplicity of creating an encryption program for RFID cards. The system shows how using a secure RFID card system can add a layer of security for database access. The project is to show how such a system could be quickly done and possibly integrated into an existing system.

**Backstory of the Idea**

The use of RFID cards is everywhere. These devices work on a simple concept. The RFID tags are small devices that are made of two items a small microchip and two large antennas made of aluminum. These tags microchip are about the size of a needle's eye. These small chips are made in different ways based on there use. When the RFID tags are wanting to be scanned, a device called a transceiver is used. These devices are any size but are designed to read specific RFID frequencies. This is done by sending our radio waves that the antennas on the RFID tags pick up the microchips uses these waves to power them enough to send out the data stored on them. This system works because the RFID tags use tiny chips and massive antennas. This project came to be after messing with a Raspberry Pi acquired one day. Messing with the machine and the given attachments,one being an RFID scanner and cards, the idea of using the RFID cards for accessing something became the goal. After looking through many different articles, papers, and journals on the use of RFID cards, the flaw that this system is here to solve came to be and that was the lack of security on the cards.

# Problem Definition

Currently, RFID systems are like the early internet to many people. (atlasRFID, 2019) By this, they describe it as how the early internet worried very little about security as there were not that many threats to systems. Using such devices with weak security can lead to people getting into systems and databases that they should not have access to. In February of 2019, an article was published by *Forbes* in this article. The writer talks about a time he was at a hospital for an appointment. He went on how he had to wait for a long time and, while bored, found that his phone could be used as an RFID card. Out of a whim and to see what would happen, he used it on the card scanner in the hospital to see if the scanner would react. To his surprise, the door unlocked; he found it odd and tried other doors and found everything that he tried with a scanner opened up to his phone. In the end, he talked about how he talked to the hospital's IT security and showed them his discovery. (Rockman, 2019)

The project itself is not as straight forward as the coding languages used were not covered by any class taken at Herzing University. So understanding the language and how to code using it became a part of this project. The whole system also works on a device never taught about at Herzing as well which is the Raspberry Pi. This device and the system it uses to function were never fully taught as it runs a Linux system but custom for the device, which is a system never taught about at Herzing.

For a company that may use an RFID card system for accessing data such as bank information, company secrets, patient/customer personal information, the security of these devices should be more focused on rather than changing the technology out for a different version of the same technology. What many companies do when a breach in RFID systems happen; they change it out for a different version of RFID cards. This costs a lot of money to do each time.

To a technical person, the use of better security on RFID is excellent. This system will provide a secondary authentication for each card that will be a reliable solution. The keys are even more secure as they change with each use making the cards even harder to duplicate by an intruder. The system is robust and will not let access to anyone without the proper card with proper key, and if the key fails several times, the system locks the card to prevent it from being used.

Many companies and organizations can benefit from a secondary authentication to RFID cards. This is because when RFID tags are used in places to allow and limit access to areas and systems, it creates weak points in the security network. By using RFID tags that use a secondary authentication, it can help reinforce those weak points in the security network.

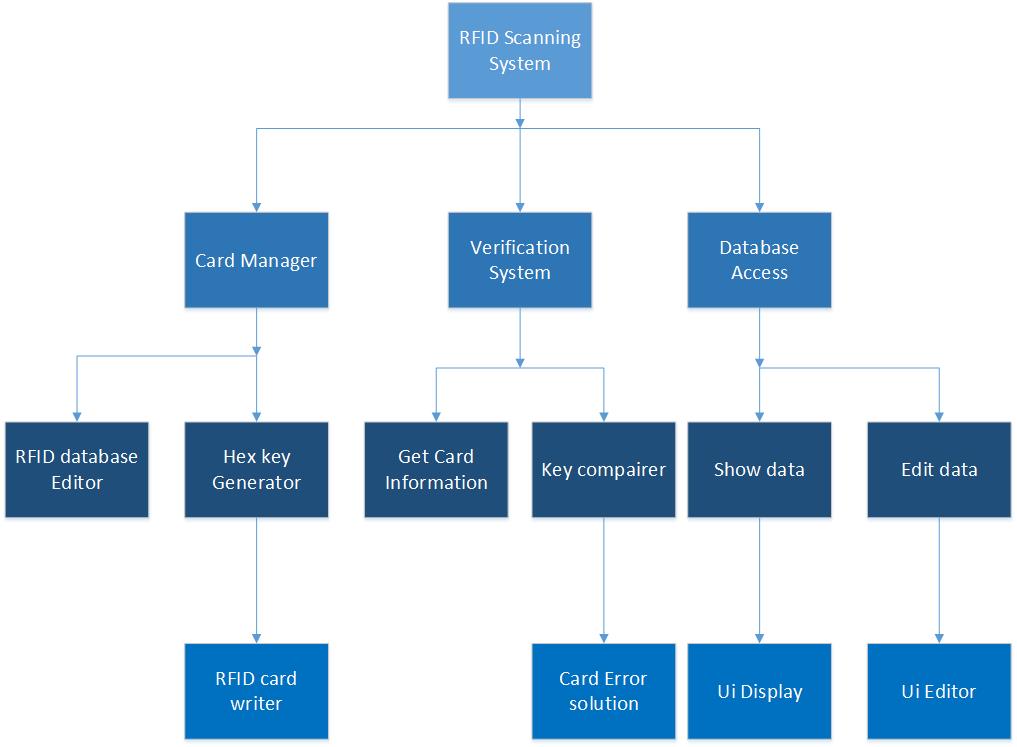
# Proposed Solution

The problem for not only door security but mainly database access is to use an RFID system with a verification system. This verification system developed can be used with any RFID card using the same frequency of 13.5MHz. These cards allow reading and writing to them, which allows a system to be able to make a verification system that uses with read-write capabilities.

The system will use the storage of the cards to store a key that changes with each use. The system will use this key as a secondary authentication for the system to ensure the person trying to access the system is supposed to be able to. This key is perfect as it gives the system a way to verify the person wanting access is the person who has the card. Any fake cards will have a hard time getting in as the card key changes with each successive use. The system will lock out any card that fails three times to authenticate. This is done by changing the database key to a random key making the real key not work. This feature makes hacking in herder as if the intruder does not get the key right with three tries; it would change again. Lastly, the system has a way to add a user to the system where it stores the user's access credentials and database they can access.

**Work Breakdown Structure**

A work breakdown structure(WBS) is a diagram to show how the system would be broken down into more basic parts for development. Doing this makes working out what needs to be done and how to distribute to work out so that the project gets done in a reasonable schedule.



## *Figure 1 WBS of Project*

# Technical Requirements

**Technologies**

The system will be a Python-based application using MySQL, Python 3, and Python Tkinter. Python is the language the program being developed will be written entirely in and using Python Tkinter to create the user interface(UI). This language is the only language that can work with the software that the Raspberry Pi came with to use the supplied RFID scanner. Python Tkinter is a very basic and easy to use Python UI development library. This is key as it allows ease of coding a UI for the system. MySQL is the database server that is being used as it is a few database systems that allow Python to work with it via a python MySQL connection library. MySQL is one of several databases that Python can connect to but is the one that is easy for me to use and setup, as well as being completely free.

**Software**

For coding the system, It uses Python, Python Tkinter, and SQL. This system uses Python 3 for coding the logic of the system which makes the functions all work. Python is chosen as it allows cross-OS compatibility and has a library for accessing MySQL databases. Python Tkinter is used to create the graphical user interface(GUI) that the system uses for accessing the database and adding new users. SQL is used as the database sued is an MSQL database.

For coding, Visual Studio Community 2019 and Geany are used throughout developing the system. Visual Studio is used for building a basic framework as it allows the coding to be done on the same machine as the database server is on. This makes coding a framework easier as it allows testing database access and manipulation without swapping from different machines. Geany is used for final coding as it is a nice IDK for the Raspberry Pi which is used for scanning the cards. Lastly, MySQL is used to create and host the database. This is because of the ease of using one software for hosting and making the database creating fewer points of failure when accessing data.

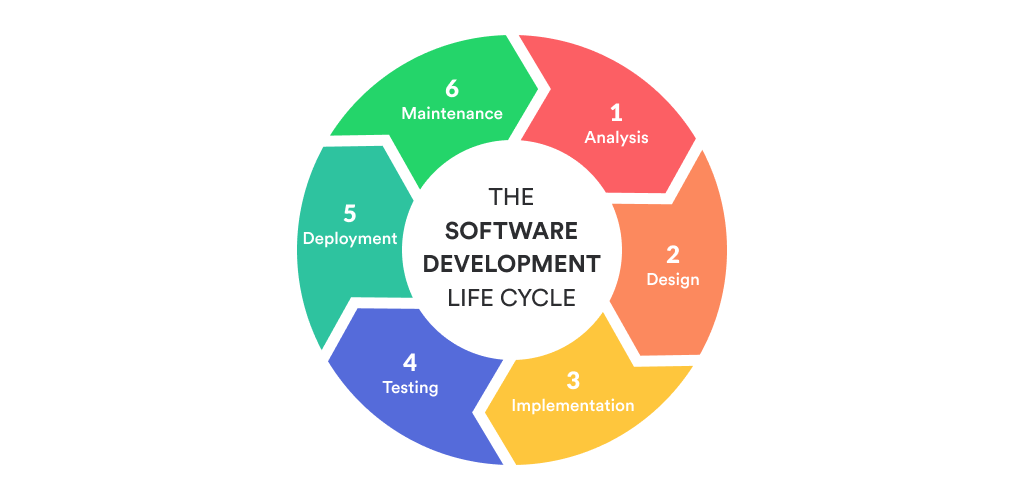
**Hardware**

The hardware needed is simple; the database is hosted on a VMware workstation that has access to the same network with a wifi access point as the Raspberry Pi will be on. The workstation is also used to run Visual Studio for framework developing. A Raspberry Pi 3 B+ is a small Linux based computer that is used to run the users end of the system and is what the card scanner plugs into. The Scanner is an MFRC522 scanner, which is a scanner that is compatible with the Raspberry Pi and the 13.56MHz RFID tags. Lastly using 11 RFID cards at 13.56 Mhz frequency. These cards are chosen as being the only cards that have a scanner for the Raspberry Pi that has internal storage on the RFID tags.

# Design MethOdologies/Project Design

**Methodology**

For this project, Agile Development was used to guide how the system will be developed. This methodology focuses on the rapid development of a project, testing it, then evaluating the results before going back and starting from the beginning and editing it more before finishing.



## Figure 2 Agile Development Life Cycle (Image from (Anurina, 2019))

Agile is ideal for this project as it will allow the project to be made as sections of code, and each section tested first, then added to the rest, then tested as a whole. This process is repeated until the entire project is completed.

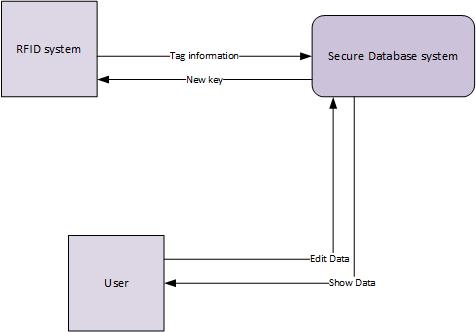
**Context Diagram**

The Conext Diagram shows a high-level view of the system to show what external input and outputs interact with the system. Figure 3 shows the Context Diagram showing how the interaction of the RFID system that I am connecting to and the user that will use the system. The RFID system will be talked to read and write to the cards that are scanned by the system. The user will be given the data from the database they accessed and allow the user to edit the information in that database.

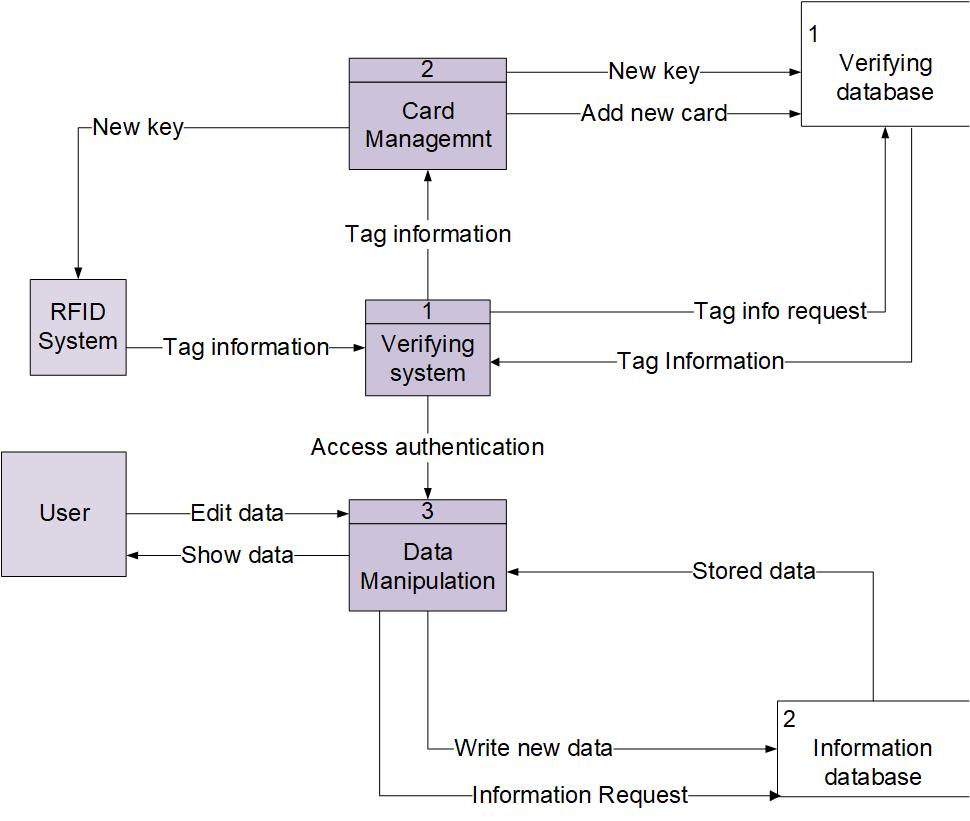
**Data Flow Diagram**

A Data Flow Diagram (DFD) shows the internal workings of the system by showing the connections between different sections of the system and show how they interact with each other. Figure 4 shows a level-0 DFD. This shows the connections of the major parts of the system and how they all interact with each other and what interacts with the user and RFID scanner system. The use of the level-0 DFD is to break down the Context Diagram into the general parts of the system.

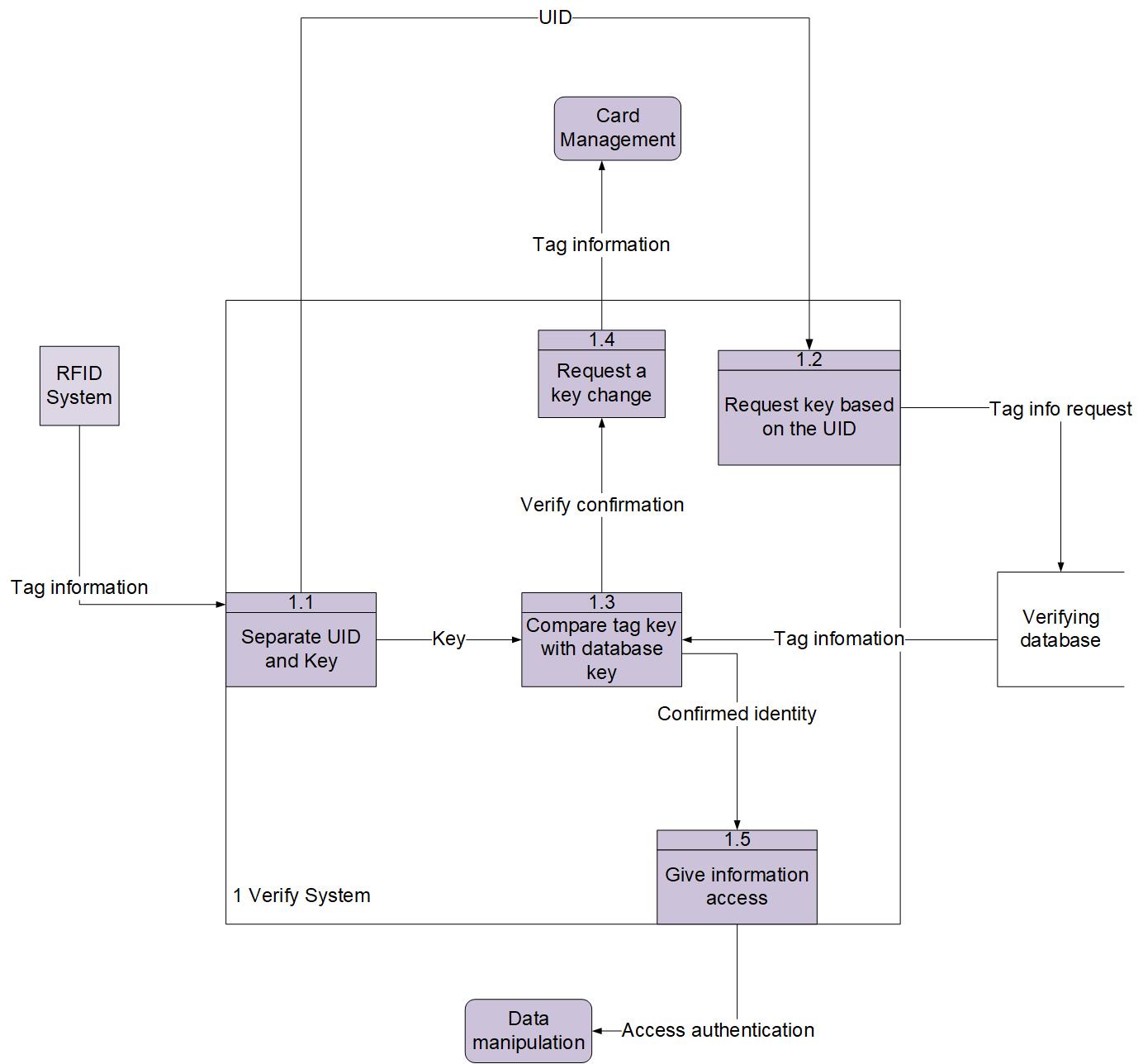
DFD can be broken down even further to shows in more details of a specific part of a DFD Level-0. Figure 5 is a broken down of Level-0 DFD’s Verification system. This is called a Level-1 DFD and shows in more detail how that part of the system will function and how the parts of that part will interact with itself.



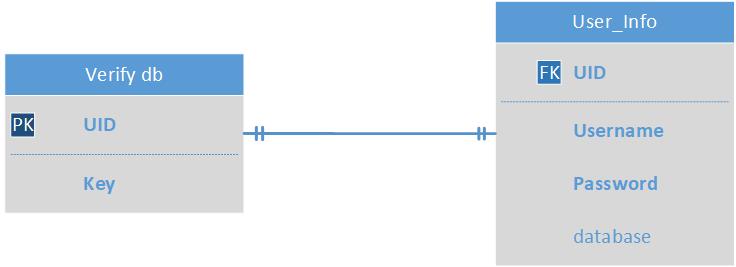
## *Figure 3 Conext Diagram*



## *Figure 4 DFD Level-0*



## *Figure 5 DFD Level 1 Verify System*

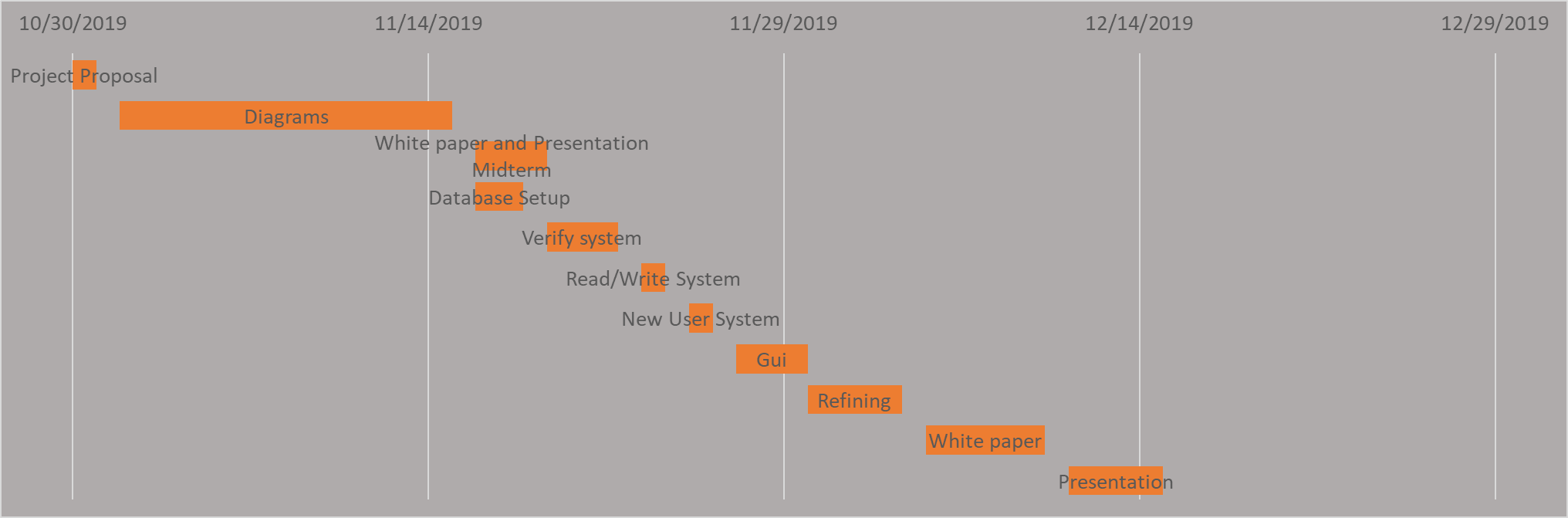


## *Figure 6 Verify Database ER Diagram*

**ER Diagram**

An entity-relationship diagram, more commonly called an ER diagram it is a diagram to show the relation of tables in a single database and how they interact with each other. The idea of it is to give a visual representation of how all tables in a database interact and give a nice view of all the attributes in each table. The connections are identified by the different lines used to connect each table together. The ends of the lines mean different things. In the ER diagram, *Figure 6* shows the interaction of the tables that are made specifically for this project. In this diagram, there are two tables being used by the system; these are Verify db and User\_info.

In this project, these two are linked with a one to one relation. This means that for each item in one table, a relating item in the other table must exist. In my project, the way the system knows what makes the items related is the attribute called UID. In Verify db, the UID is labeled as a primary key. This means that it is an attribute used to connect the table to others and give something that both tables will have in common. So in User\_info, the UID is labeled as a foreign key. This makes this UID having to match a UID from Verify db has. This is important as the system will use the UID to pull the information from User\_info for the user who is accessing the system. The last thing I have is bold words and standard text. The bold text represents unique attributes; this means that these attributes must be unique in the database and cannot match another in the same table.



## *Figure 7 Project Timeline Gnatt Chart*

**Project Timeline**

A project timeline is used to show the progress of the project over time. Using a gnat chart allows a visual chart of what has been worked on and when it was started and for how long it has been worked on. *Figure 7* shows the timeline of my project and how each major step is taken to create the whole project took and when it was worked on and for how long. This is an excellent way to show my project's workflow and how the time and efforts to finish the project have been divided.

# Implementation

Implementation of a project is the step of development where a functioning system is created that will take the design stage and make it all real. Prior to this step, all documents created are made as a framework to know what needs to be made, how the system will interact when each part is expected to be finished, and how the system should work.

**Technologies**

The project will be a Python-based system using Python 3 and Python Tkinter; these will be used to create the user interface(UI) and the logic of the system. This will allow the project to run on a Raspberry Pi with ease and work with valid RFID scanners. Python 3 will be for all the logic of the system and is the central part of the whole project. Python Tkinter is used to create a basic UI to show how the system will work and how you can add new users to the system. On the server-side of the system, it uses MySQL to host and operate a database server. This is used as MySQL works easily with Python and allows easy relational database structure.

**Software**

Using Visual Studio to rough out most of the logic and UI of the system in Python and Python Tkinter as it allows easy testing when developing for the database as it allows the work to be done on the same system making checking results easy.. This makes it easy to develop the project in sections and test each section. The database and hosting server was all made and ran using MySQL Developer edition Workbench. This is a MySQL program that allows creating, editing, and host a database with one program.

The last thing used was Geany, which is a programing IDE for the Raspberry Pi. This was used as the final coding was needing to be done on the Raspberry pi to ensure the card scanner system would work. This is because of the specific Python libraries that are only used on the Raspberry Pi to make it talk to the scanner and cards. By using Geany, the project will be made fully functional on the device that it will be expected to run on, allowing me to ensure the project will work as a whole system.

The use of Visual Studio and Geany was chosen over programs such as Notepad or other text editors as they gave syntax error tracking and allowed easy viewing of any errors that may occur when running the parts of the project. This is important as getting these errors to help refine the system to eliminate as many projects breaking errors as possible.

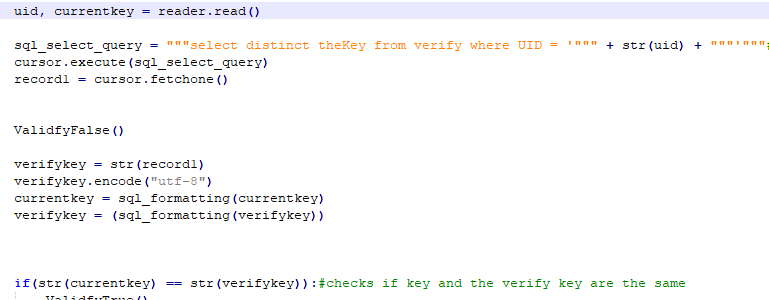
**Hardware**

The hardware for this system is going to be basic. First is a computer or server that has windows installed and MySQL database installed onto it. This machine needs to be able to run the program and connect to the same network as the Raspberry Pi will be on. Ideally, the MySQL system is running on a network with a wireless access point that the Raspberry Pi can connect to. This will allow the Raspberry Pi to be as portable as possible.

The primary device will be a Raspberry Pi 3 b, which is, at this time, the most advanced Raspberry Pi on the market. The ideal for this project is that the whole system can be portable, so having some sort of small battery pack for the Pi is ideal and having a touch screen. Using 7” LCD screen found that will work as a portable display. Finally, an RFID scanner and matching cards will be needed. This project used an MFRC522 scanner, which works with any 13.56MHz RFID tags.

**System Parts**

The project's core is the verifying system. This system is where the main logic for the verifying of the user takes place, which is the main point of this project. Verifying is used to take the information from the user's card, check it against what is in the database, and then must decide if it does match or not. This is where it gets tricky as there are three significant points where the system needs to be able to recover from in case of an error. First is when the system cannot connect to the database; this is the first thing the system tries to do. If it cannot connect, it will print out an error saying connection cannot be made. The reason there is no detail is that the system, by default, then prints out an error code indicating where the error came from.

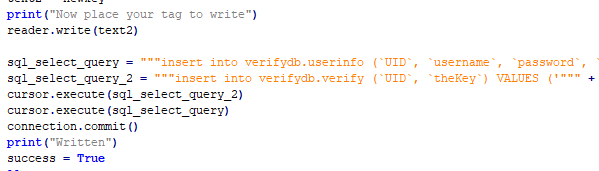


## *Figure 8 Verify card key against database key*

So when it does verify a connection has been made, it will continue. *Figure 9* is the next part and is the primary function of the system. It will first read in the card's data and pull from the database a key that matches the UID of the card. This is where the second error that can occur is if the card is not in the database what it should do. In this project, the system will print out the card failed and say they have 2 more tries. If not, the system will end the action and close connection to the database. If the card is correct and the keys do not match, the system will again give the user two more tries. If the scan fails and the card is in the system, it will scramble the database key, locking out even the original card.

Lastly, the system pulls out the user's username, password, and database they have access to. This is then sent off to any system you wish to use it to control the access. I have it set to a basic database to show it will work.

The second part of this project is the card management system. What this is is sections of the code that are used to add or manipulate data on the card, or the verify the database. In the verify system, this is used only as calling card data and changing the card keys. But in the writing system, it allows adding of a new card to the system. To do this is different then the verify as I do not need to verify a user as I am adding a user that was not in the system, to begin with. The writing system creates a new entry into the verify databases verify table and user\_info table. In the verify table, it inputs the new cards UID and creates a new random key that is stored in the database and the new card.

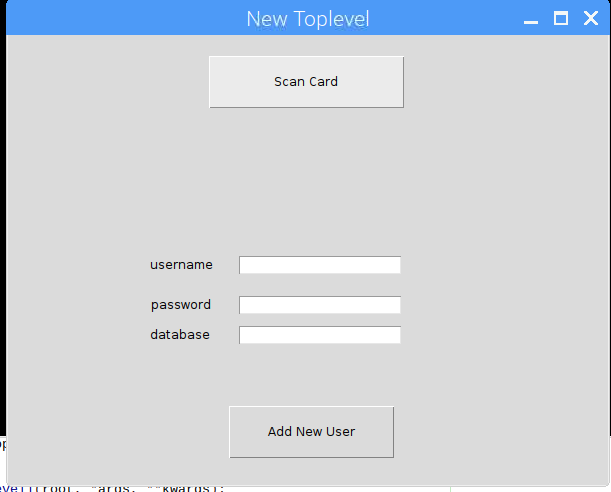
**

## *Figure 9 Creating new card user*

The system then takes the input from the UI of the new user's username and password as well as the database they re allowed access to and saves it to user\_info where there is a new entry with the new cards UID already in. the system just fills in the input into the database so that the user\_info is now linked to the cards UID.

**User Interface**

The user interface developed was made as a proof of concept. It has essential functions of scanning current cards and adding new users to the system with a customizable username, password, and database. The final product would not have the scan to access and add a new user on the same UI.



## *Figure 10 Basic UI*

# Project Testing

Testing, in any project, is considered the most vital step during development. When creating any application, no matter how well you plan and document the steps, if the application does not function, the project was a fail. Testing is the part of developing where you ensure the project functions correctly, and you meet the goals you set for the project that you have set.

Deciding what you are looking for in a test is a big part. As setting goals you want to reach and then to test for those goals is how to ensure a project is going to be successful. For this project, five significant tests have been developed. These test plans where developed around one of each of the Project needs I had decided on earlier. The results were reached by using unit testing throughout the entire project. This is done by testing modules of the code to see if they will work, then adding them to others and testing that larger module to ensure nothing broke. This was done until the final product passed each test set.

First test was to show the keys from the card need to match the database. The plan set was to have the system show an indication that the scan and verification either succeeded or failed. *Figure 11* shows the system correctly reading and verifying a proper user. *Figure 12* shows what happens when a card is scanned, and the keys do not match.

The second test was to show the system verifies the connection to the server. To test this building a basic frame was done as this ability is used whenever the system accesses the database. *Figure 13* shows what happens when the system fails to connect to the database for any reason. *Figure 14* shows what would be expected of the system to give as an output if the connection was successfully made.

The third and fourth tests are connected as the third test will fail if the fourth test fails as well. The third test was to verify that a card was actually scanned before any data is pulled from the server. This was simple as the system does not continue after verifying the connection to the server until a card is successfully scanned. This worked until trying to use a different card, and the system had errors. Soon I realized that this was due to the database key, and card keys were not in the same format. Fixing this was simple by making a function that can be called on to format any input into identically formatted output. *Figure 14* shows the code I came up with to correctly format all the keys to an identical format.

The fourth test is to ensure that the UID of the card is one that is in the database. This test was successful, but for a different reason then what I expected. As the system will ignore trying to verify and pulling any key form the database is the select query does not find any key that belongs to that UID. I did not make any code to do this; it is just a thing that the code does by default. Instead of pulling, it runs my built-in card error and gives the user 2 more tries to get in.  
 Test five was the hardest test to build for and get to the point that it won't fail. The test is to ensure the card and database keys must change after each successful usage. To test this, I used two sperate programs I made to read and write to the cards outside the system. This was done to change the key to ensure it won't validate but also to see if the key does change. This test is the most significant part that made me change how I authenticate the cards as the cards would not change their key at all, so I had to find a new way to write to them. But after finding a new library to use, it allowed the creation of a working read-write system for the RFID cards.

# Conclusion

The work done during this project has taught a great deal and has been a very productive time. As proof of concept, the creation and implementation have been a great success. What has been done should be adequate proof to show a client how such a project would be done.

This project was a great learning experience, not only in how to do a multi-stage project like this project is. But also show how a project that uses systems and technologies never looked at before can be taken on how to handle the learning of the new items as well as being productive.

The program is far from a perfect system and shouldn’t be directly used for a database. This was to show and create a starting stage where a real product could be made from and how such a technology could be used in different systems to make them more secure. The next thing would be to work with a client who would like this product and see what they need as requirements that would have to built onto the current system created.

# References

Anurina, O. (2019, May 10). *Agile SDLC: Skyrocketing Your Project with Agile Principles*. Retrieved from MLSDev: https://mlsdev.com/blog/agile-sdlc

atlasRFID. (2019, February 28). *WHAT IS RFID? | THE BEGINNER'S GUIDE TO RFID SYSTEMS*. Retrieved from atlasRFIDstore: https://www.atlasrfidstore.com/rfid-beginners-guide/

Grover, A., & Berghel, H. (2011). A Survey of RFID Deployment and Security Issues. *Journal of Information Processing Systems, 7*(4), 561-580. doi:10.3745/JIPS.2011.7.4.561

Gus. (2017, October 03). *How to setup a Raspberry Pi RFID RC522 Chip*. Retrieved from PiMyLifeUp: https://pimylifeup.com/raspberry-pi-rfid-rc522/

Knoche, R. (2016, August 21). *Making a MySQL GUI with Tkinter*. Retrieved from Dealing Data: http://www.dealingdata.net/2016/08/21/Python-MySQL-GUI/

Rockman, S. (2019, February 13). *How I Hacked Hospital RFID*. Retrieved from Forbes: https://www.forbes.com/sites/simonrockman1/2019/02/13/how-i-hacked-hospital-rfid/#26d49fa55da9

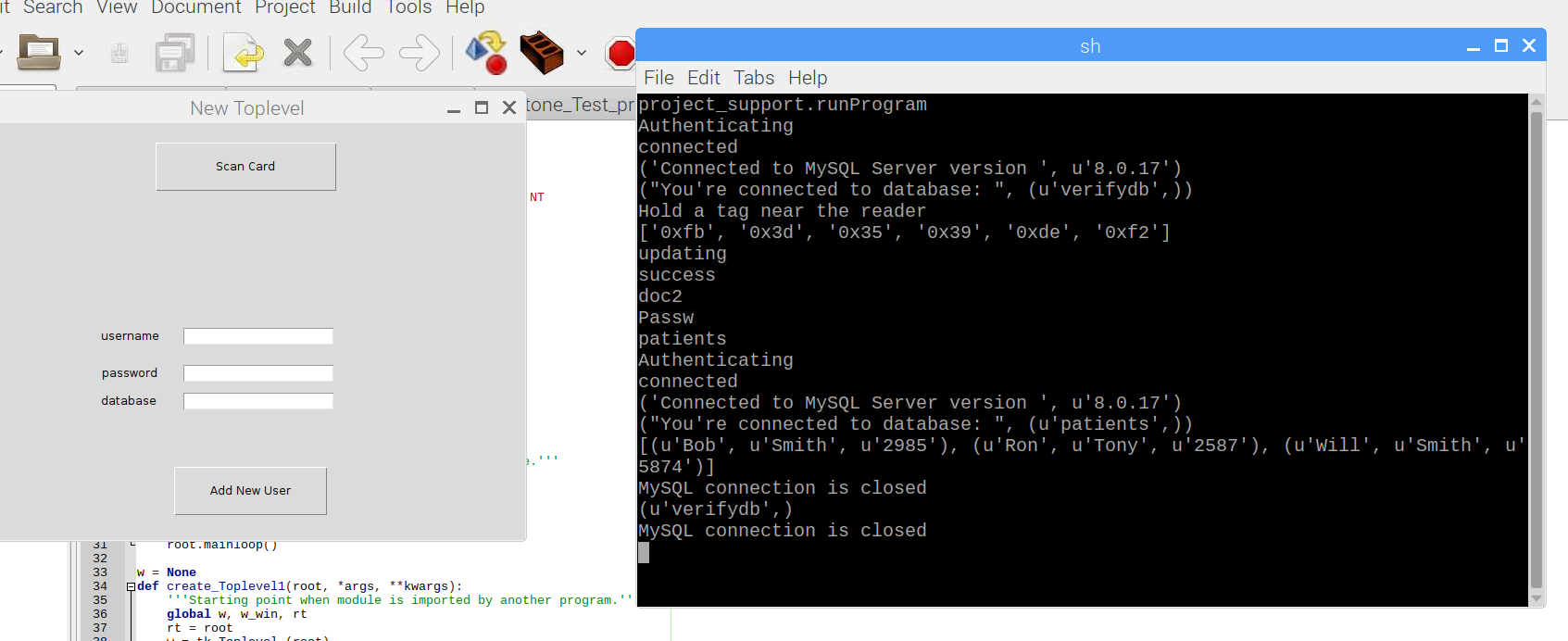
Thrasher, J. (2013, August 22). *How is RFID Used in Real World Applications?* Retrieved from RFID INSIDER: https://blog.atlasrfidstore.com/what-is-rfid-used-for-in-applications

Uszkai, S., Papp, B., Nemes, Á., & Erdei, T. (2019, April 7). Arduino and Rapsberry Pi Based Hybrid RFID Access Control System with Mobile Platform. *Műszaki Tudományos Közlemények, 9*(1), 251-254. doi:https://doi.org/10.33894/mtk-2018.09.58

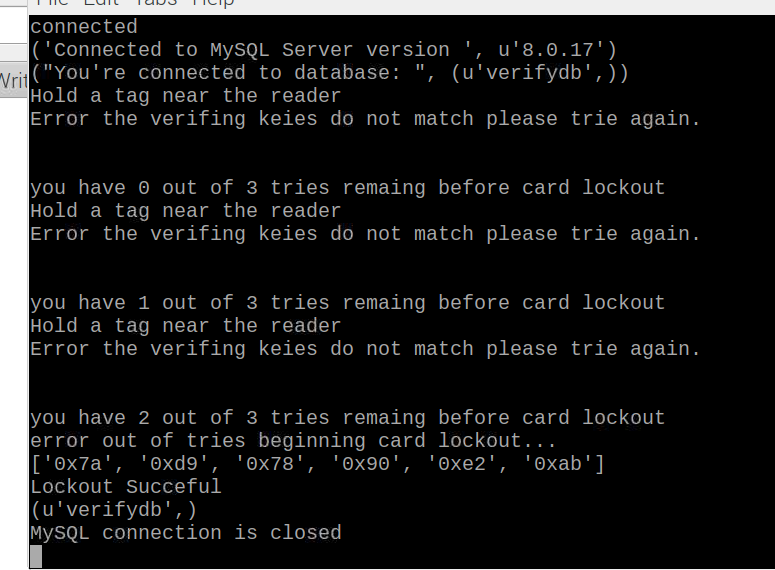
Want, R. (2006, January). An Introduction to RFID Technology. *IEEE Pervasive Computing, 5*(1), 25-33. doi:10.1109/MPRV.2006.2

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Start Date | End Date | Duration |
| Project Proposal | 10/30/2019 | 10/31/2019 | 1 |
| Diagrams | 11/1/2019 | 11/15/2019 | 14 |
| White paper and Presentation Midterm | 11/16/2019 | 11/19/2019 | 3 |
| Database Setup | 11/16/2019 | 11/18/2019 | 2 |
| Verify system | 11/19/2019 | 11/22/2019 | 3 |
| Read/Write System | 11/23/2019 | 11/24/2019 | 1 |
| New User System | 11/25/2019 | 11/26/2019 | 1 |
| Gui | 11/27/2019 | 11/30/2019 | 3 |
| Refining | 11/30/2019 | 12/4/2019 | 4 |
| White paper | 12/5/2019 | 12/10/2019 | 5 |
| Presentation | 12/11/2019 | 12/15/2019 | 4 |

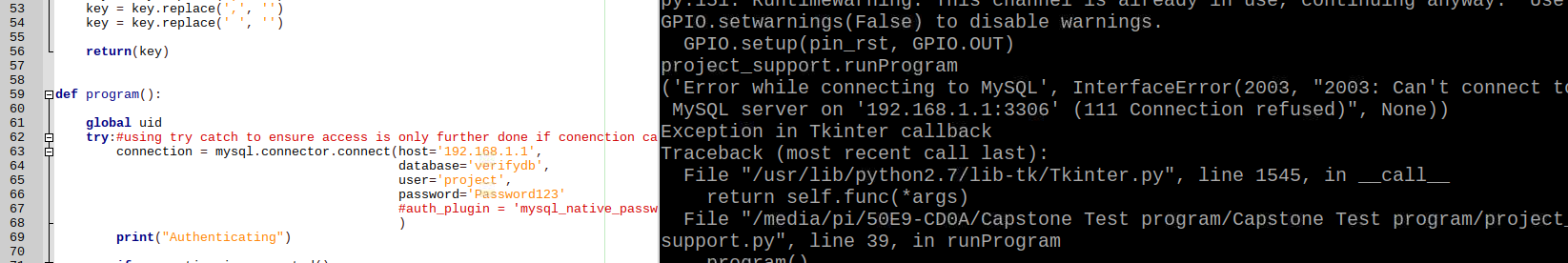
## *Table 1 Project Timeline Table*



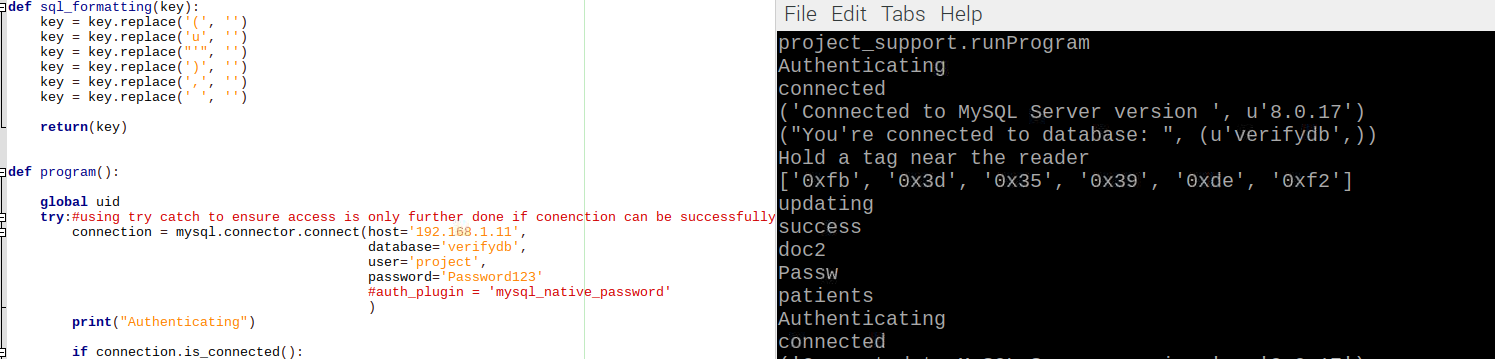
## *Figure 11 Verify Successfully result*



## *Figure 12 Card denied three time*



## *Figure 13 Server Conenction failed screen*



## *Figure 14 Formatting function for keys*