GUJARAT TECHNOLOGICAL UNIVERSITY

Chandkheda, Ahmedabad

Affiliated

Gujarat Power Engineering and Research Institute
Mevad, Mehsana

A Project Report On

RFID Based Entry Management System Using Single Board Computer

Under subject of

DESIGN ENGINEERING – II B

B. E. Semester - VI

(Computer Engineering Branch)

Submitted by:

Sr.	Student Name	Enrollment No.
1.	Parth N. Patel	131040107024
2.	Monish M. Soni	131040107039
3.	Samarth Patel	131040107027
4.	Vandan Shah	131040107038
5.	Shukla Suraj	131040107041

Prof. Harshad Chaudhary
(Faculty Guide)

Prof. Hasmukh Patel
(Head of Department)

Academic Year
(2015-2016)

Acknowledgement

We have taken sincere efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organization. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Gujarat Power Engineering & Research Institute** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project and the guidance from the faculty of computer department **Prof. Harshad Chaudhary** and our head of department (H.O.D.) **Prof. Hasmukh Patel** who constantly supported us through all the guidance related to Project of RFID Based Entry Management System.

Without our team spirit and contribution this project would not be possible.
We would like to express our special gratitude and thanks to External Examiner for giving me such attention and time.

Our thanks and appreciations also goes to our colleague in developing the project and people who have willingly helped us out with their abilities.

GROUP INTRODUCTION:

1. **PARTH N. PATEL**

   Parth is a computer Engineering Student of GPERI. His Interest is in Computer Tweaks and Working of Computer. He loves to code. He is Enthusiastic about Web Designing and Development. He is running a Website "DgComp.in". His Hobbies are to play table-tennis and getting Knowledge about Technologies. He has done many Projects like, Home and Industry Automation, Touch Table using I-R sensing Technology.

2. **MONISH SONI**

   Monish is A Computer Geek and simple guy Loves Around to be Nature and Be always Trying to Connect the Dots of Humanity With the Technology at this Cutting Edge of Revolution. He loves to Play Football and Cycling. His Specific Fields of Interest in Computer Science is Networking, Hardware and Multimedia Animation.

3. **SURAJ SHUKLA**

   Suraj is Computer Student of GPERI. His Hobbies are Swimming and Listening Songs. His specific Related Interest in computer is to code. He had Work on several Projects on Coding from the School and Contests. He also loves to work on embedded system and related to electrical system designing.

4. **VANDAN SHAH**

   Vandan Is a Code Aspiring Student of Computer Department Who Loves to Create New Things and Built the Innovative things .He loves to Play Badminton and also Play Cricket sometimes. Vandan loves doing maths and computer related software testing.

5. **SAMARTH PATEL**

   Samarth Is Computer Student of GPERI Who loves to Organize and Management the Computer Things and Play Hand fully with the Stuffs and Latest Technology Related Current Affairs. His Favorite Sports are Football and cricket. He wants to be a professional Engineer of a Big Software Company.

- **PROF. HARSHAD CHAUDHARY(**GUIDE**):-**

  He is Our Professor of Computer Department at GPERI. He is Very helpful and Kindly Nature of Person who is Enthusiastically Believing in the Technology and Computer Related Innovation and New creative Ideas. He is Person who loves to Code and Made and Effective Contribution towards a Project for making Automatic Rod bending Machine.

INDEX

1. Introduction:

1.1 Domain

- Our project is RFID based entry management system, in this project we are using a single board computer with RFID technology for Entry management system with the core implementation of IOT(Internet of Things) within it.

- The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data.

- A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer. Single-board computers were made as demonstration or development systems, for educational systems, or for use as embedded computer controllers. Many types of home computer or portable computer integrated all their functions onto a single printed circuit board.

- Unlike a desktop personal computer, single board computers often did not rely on expansion slots for peripheral functions or expansion. Some single-board computers are made to plug into a backplane for system expansion. Single board computers have been built using a wide range of microprocessors. Simple designs, such as built by computer hobbyists, often use static RAM and low-cost 8 or 16 bit processors. Other types, such as blade servers, include all the memory and processor performance of a server computer in a compact space-saving format.

- Radio-frequency identification (RFID) is the wireless use of electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects.

- The RFID device serves the same purpose as a bar code or a magnetic strip on the back of a credit card or ATM card, it provides a unique identifier for that object. And, just as a bar code or magnetic strip must be scanned to get the information, the RFID device must be scanned to retrieve the identifying information.

- RFID comes under Automatic identification technology which uses radio-frequency electromagnetic fields to identify objects carrying tags when they come close to a reader. RFID uses several radio frequencies and many types of tag exist with different communication methods and power supply sources.

- RFID tags generally feature an electronic chip with an antenna in order to pass information onto the interrogator (also known as a base station or more generally, reader). The assembly is called an inlay and is then packaged to be able to withstand the conditions in which it will operate. This finished product is known as a tag (transponder).

- The information contained within an RFID tag's electronic chip depends on its application. It may be a unique identifier (UII, Unique Item Identifier or EPC code, Electronic Product Code, etc.). Once this identifier has been written into the electronic circuit, it can no longer be modified, only read. (This principle is called WORM-Write Once Read Multiple). Some electronic chips have another memory in which users can write, modify and erase their own data. These memories vary in size from a few bits to tens of kilobits.

- In this Project we have connected a RFID reader with raspberry pi a Single board computer which is our base of our Server fulfilling the purpose of both Interacting with RFID tag as well as the Client Side request. When the RFID tag is swiped on RFID reader, which is connected to Raspberry PI. It takes the Unique ID of the RFID Tag as an Input which is bind with the Enrollment no. and identifies the User and allocate the System to User. Then the User is allocated the PC and the PC is boot via LAN Boot, and the system shows the client software on it which automatically allocates the enrolment no. of the user and asks for the password.

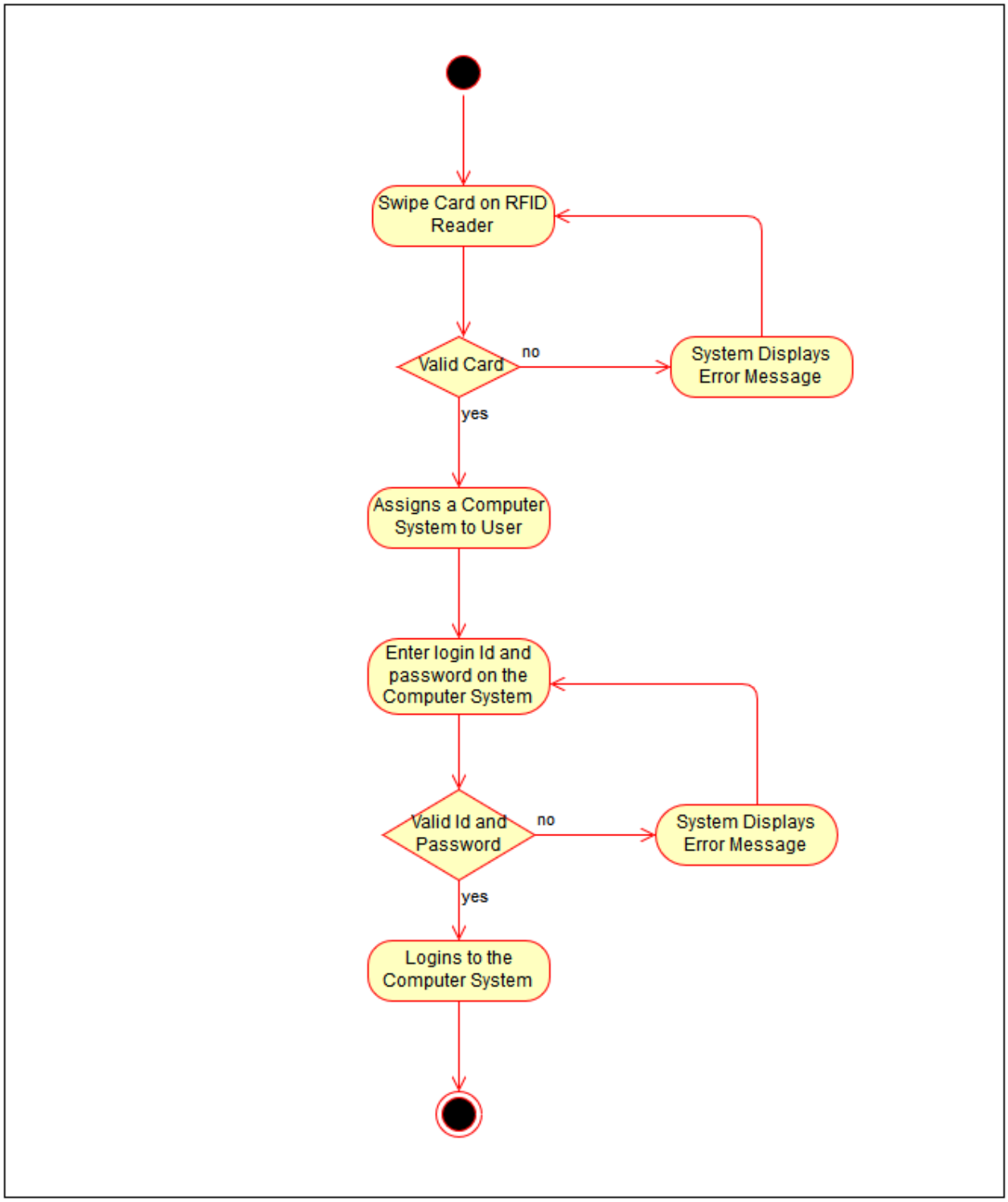
1.2 Conclusion about current System Problem

- We are talking about the current system of lab entry system, if we are taking a glance of that system it consist of one manual register where every have to enter there details like there name, Enrolment number, Date, Time of Enter, & Sign.
- Many time it happen that we need to do some really important work any which is need to be done in very short time.
- At that time we didn't have time to register our details in the register.
- Let us consider we have enter our details correctly but when we trying to start the computer it doesn't getting start or it doesn't have the internet connection as we required that.
- So its takes our so much time just getting our work start instead in that time could even finish our work.
- There could be also possible that some of the student may not register into the register and can misuse the system.
- Let us consider one case where we are trying to identify some issue which was happened in the lab. Consider one such instrument is stolen in the lab, to identify the act we just have the register which doesn't even include entry of the some student any also not include the PC no. where he/she was sited.
- So hence we cannot track or identify the act happen in lab still having the register records.
- Other case is where the computer system could not maintain well with this system.
- User can not submit their issue regarding the computer system failure or not availability of the features to the lab in charge.

2. Modelling and Analysis using Software:

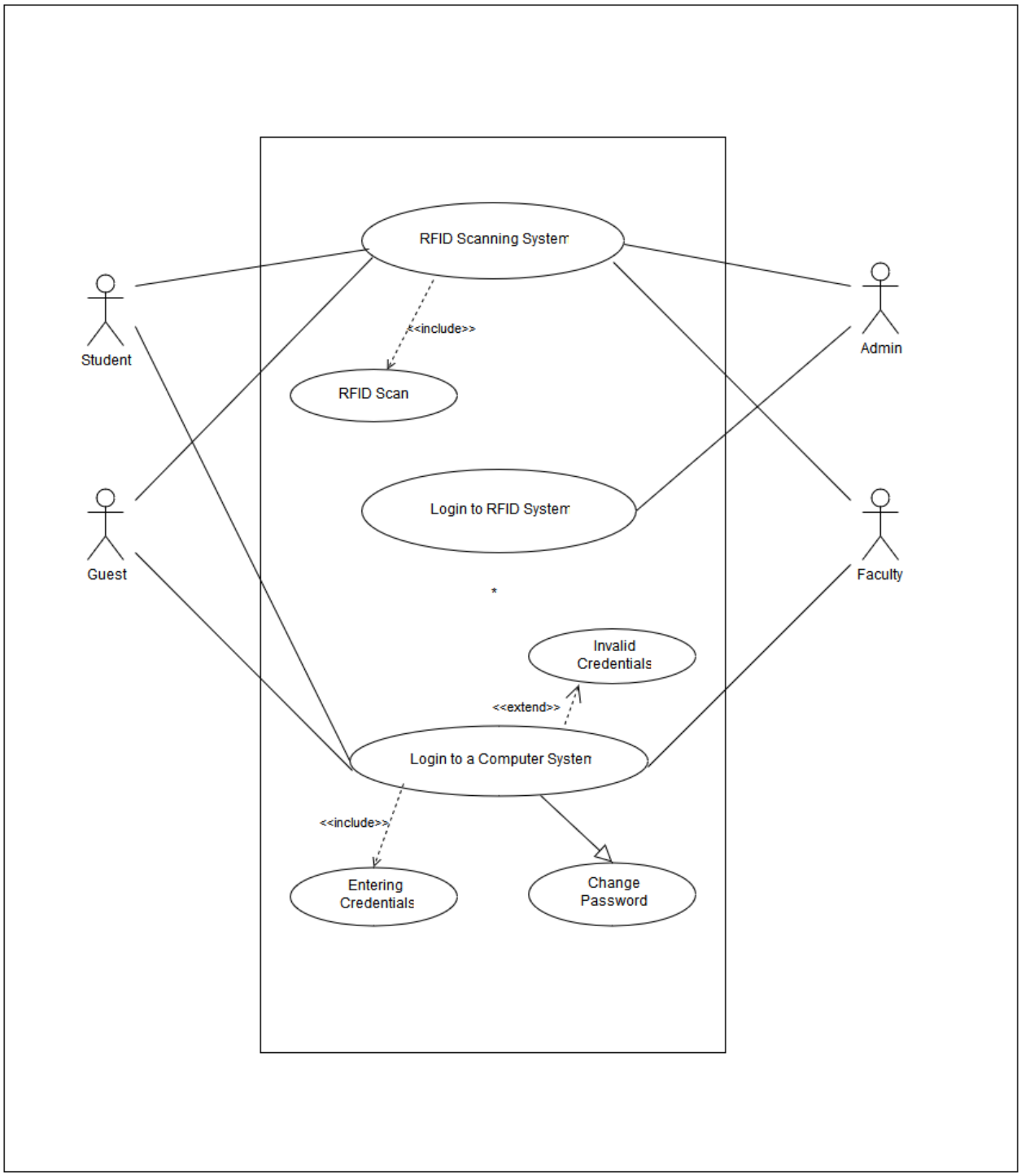
2.1 Activity Diagram:

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. Below given is a Activity Diagram for our RFID based entry management System.

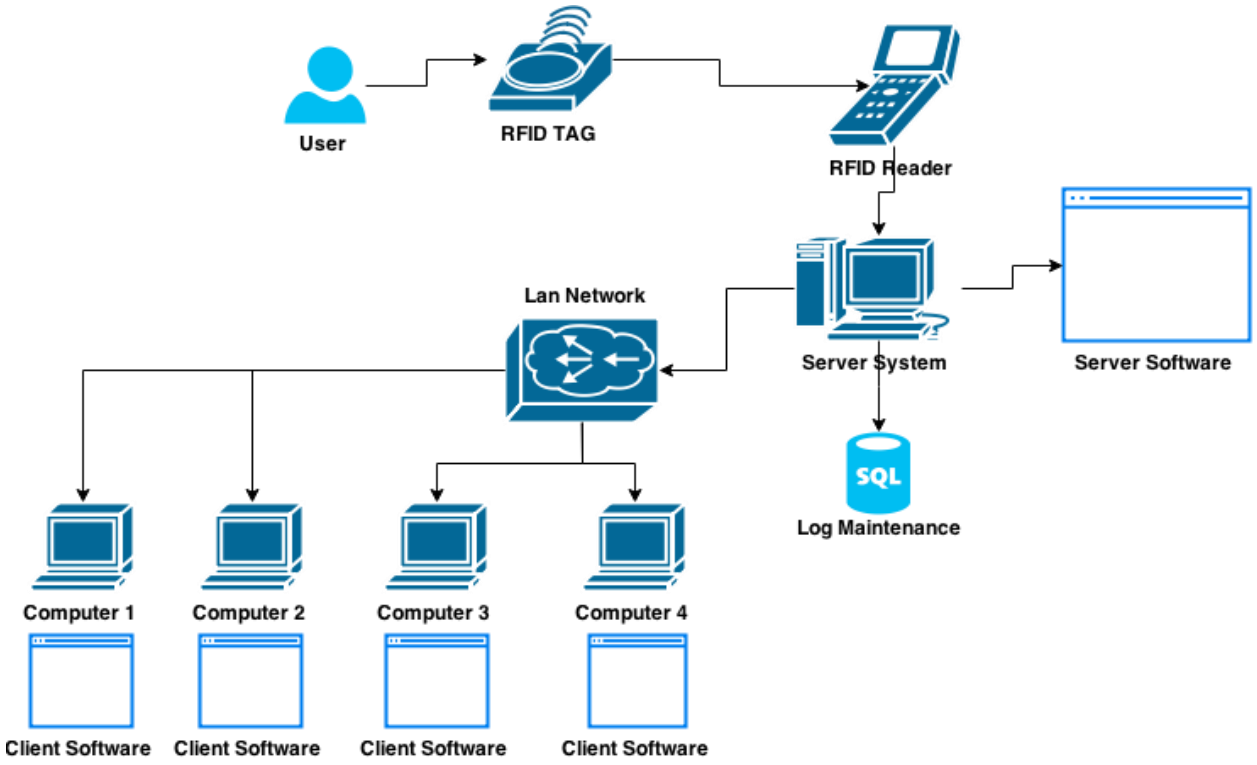


2.2 Use Case Diagram:

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.



2.3 System Flow Diagram:



3. Engineering Economics of Design:

3.1 Labor, Material & overhead coast:

3.1.1 Labor:

We don't require any labor to run the system as the system is fully automated.

E.g. : The user has to just swipe the card and he/she will be allocated the PC all the data uses related to that PC will be maintained in the log after using PC the user will shut down and session will be completed.

3.1.2 Material:

The material required to make the system are listed below:

No.	Equipment Name
1	raspberry pi 2
2	RFID Module with RFID Tags
3	Micro SD 16 GB Class 10
4	Lan Cable
5	Pizzo Buzzer
6	LCD 16X2
7	Keypad
8	2 Amp. Micro usb Charger
9	Jumper Cables
10	HDMI to VGA Converter
11	Router
12	1 Computer System

3.1.3 Overhead Cost:

The electricity charge to run the raspberry pi is 5V is only requirement for this project other than that there is no overhead coast.

3.2 Time Value of Money:

To start the normal computer system it take up to 2 to 3 min to start the use of computer where as in our system , System will already started when user swipes the card so it will be time saving for the user to getting stared PC.

- E.g. : The user Enter in the lab, do the registration in the register manually and switch on the PC.
- In the RFID based entry management the user has to just swipe the card the registration details are filled automatically in the log on server side and the system automatically allocate the PC to the user and start the allocated PC.

3.3 Cost Estimation

A software project is not just about writing a few hundred lines of source code to achieve a particular objective. The scope of a software project is comparatively *quite large*, and such a project could take several years to complete. However, the phrase "quite large" could only give some (possibly vague) qualitative information. As in any other science and engineering discipline, one would be interested to *measure* how complex a project is. One of the major activities of the project planning phase, therefore, is to estimate various project parameters in order to take proper decisions. Some important project parameters that are estimated include:

- **Project size:** What would be the size of the code written say, in number of lines, files, modules?
- **Cost:** How much would it cost to develop a software? A software may be just pieces of code, but one has to pay to the managers, developers, and other project personnel.
- **Duration:** How long would it be before the software is delivered to the clients?
- **Effort:** How much effort from the team members would be required to create the software?

COCOMO

COCOMO (Constructive Cost Model) was proposed by Boehm. According to him, there could be three categories of software projects: organic, semidetached, and embedded. The classification is done considering the characteristics of the software, the development team and environment. These product classes typically correspond to application, utility and system programs, respectively. Data processing programs could be considered as application programs. Compilers, linkers, are examples of utility programs. Operating systems, real-time system programs are examples of system programs. One could easily apprehend that it would take much more time and effort to develop an OS than an attendance management system.
The concept of organic, semidetached, and embedded systems are described below.

- **Organic**: A development project is said to be of organic type, if
    - The project deals with developing a well understood application.
    - The development team is small.
    - The team members have prior experience in working with similar types of projects.

- **Semidetached:** A development project can be categorized as semidetached type, if
    - The team consists of some experienced as well as inexperienced staff
    - Team members may have some experience on the type of system to be developed
- **Embedded:** Embedded type of development project are those, which
    - Aims to develop a software strongly related to machine hardware
    - Team size is usually large

RFID Project Estimation (Assumption):

1. Project Implementation Techniques Chosen: COCOMO
2. COCOMO Project Model Category: Complete COCOMO Model
3. Project Category: Embedded Project

Complete COCOMO Model:

Both the basic and intermediate COCOMO models consider a software to be a single homogeneous entity -- an assumption, which is rarely true. In fact, many real life applications are made up of several smaller sub-systems. (One might not even develop all the sub-systems -- just use the available services). The complete COCOMO model takes these factors into account to provide a far more accurate estimate of project metrics.
To illustrate this, consider a very popular distributed application: the ticket booking system of the Indian Railways. There are computerized ticket counters in most of the railway stations of our country. Tickets can be booked / cancelled from any such counter. Reservations for future tickets, cancellation of reserved tickets could also be performed. On a high level, the ticket booking system has three main components:

- Database
- Graphical User Interface (GUI)
- Networking facilities

Among these, development of the GUI is considered as an organic project type; the database module could be considered as a semi-detached software. The networking module can be considered as an embedded software. To obtain a realistic cost, one should estimate the costs for each component separately, and then add it up.

We are dividing our Project in into 2 sub-Project,

1. Client-Server Software (Basic Category).
2. RFID Scanning System Hardware Implementation (Semi-Detached Category).

Calculation of Effort/Time using Basic Model:

The basic COCOMO model helps to obtain a rough estimate of the project parameters. It estimates effort and time required for development in the following way:
Effort = a * (KDSI)b PM
Tdev = 2.5 * (Effort)c Months

Where,

- KDSI is the estimated size of the software expressed in Kilo Delivered Source Instructions
- a, b, c are constants determined by the category of software project
- Effort denotes the total effort required for the software development, expressed in person months (PMs)
- Tdev denotes the estimated time required to develop the software (expressed in months)

The value of the constants a, b, c are given below:

Software project	***a***	***b***	***c***
Organic	2.4	1.05	0.38
Semi-detached	3.0	1.12	0.35
Embedded	3.6	1.20	0.32

Now,

As our Project is Embedded Project the value of **a=3.6** , **b=1.20**, **c=0.32**.
-As per our assumption the total KDSI(Kilo Delivered Source Instructions),
KDSI = 3.5
Now putting this KDSI in the equation above,

Effort = a * (3.5) ^b
=(3.6) * (3.5) ^ (1.20)
=(3.6) * (4.4966)
Effort = 16.1877 PM

Tdev = 2.5 * (Effort)c Months
= 2.5 * (16.1877) ^ (0.32)
= 2.5 * (2.4375)
Tdev = 6.0937 Month

People required (P) = Effort Applied / Development Time
= 16.1877 / 6.0937
= 2.6565

Calculation of Effort/Time using Intermediate Model (Embedded System):

The basic COCOMO model considers that effort and development time depends only on the size of the software. However, in real life there are many other project parameters that influence the development process. The intermediate COCOMO take those other factors into consideration by defining a set of 15 cost drivers (multipliers) as shown in the table below. Thus, any project that makes use of modern programming practices would have lower estimates in terms of effort and cost. Each of the 15 such attributes can be rated on a six-point scale ranging from "very low" to "extra high" in their relative order of importance. Each attribute has an effort multiplier fixed as per the rating. The product of effort multipliers of all the 15 attributes gives the Effort Adjustment Factor (EAF).

Cost Drivers	Ratings					
	Very Low	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	**1.40**	
Size of application database		0.94	1.00	**1.08**	1.16	
Complexity of the product	0.70	0.85	**1.00**	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			**1.00**	1.11	1.30	1.66
Memory constraints			**1.00**	1.06	1.21	1.56
Volatility of the virtual machine environment		**0.87**	1.00	1.15	1.30	
Required turnabout time		**0.87**	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	**1.00**	0.86	0.71	
Applications experience	1.29	1.13	**1.00**	0.91	0.82	
Software engineer capability	1.42	1.17	**1.00**	0.86	0.70	
Virtual machine experience	**1.21**	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	**0.95**		
Project attributes						
Application of software engineering methods	1.24	**1.10**	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	**0.83**	
Required development schedule	1.23	1.08	**1.00**	1.04	1.10	

Now, Taking the Product of all cost Drivers Multipliers highlighted in **Bold & Underline**.
Total EAF = 1.2011

Software project	a_i	b_i
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

Now using equation,

E=ai(KLoC)(bi)(EAF)

KLOC = 3.5 , ai = 2.8 , bi = 1.20

E = (2.8) * ((3.5) ^ (1.20)) * (1.2011)
= (2.8) * (4.4966) * (1.2011)
E = 15.122 PM

Tdev = 2.5 * (Effort)c Months
= 2.5 * (15.122) ^ (0.32)
= 2.5 * (2.3845)
Tdev = 5.962 Months

People required (P) = Effort Applied / Development Time
= 15.122 / 5.962
= 2.5364

Total Manpower Required,

P = 2.6565 + 2.5364
P = 5.1929

Advantages of COCOMO:

COCOMO is a simple model, and should help one to understand the concept of project metrics estimation.

Drawbacks of COCOMO:

COCOMO uses KDSI, which is not a proper measure of a program's size. Indeed, estimating the size of a software is a difficult task, and any slight miscalculation could cause a large deviation in subsequent project estimates. Moreover, COCOMO was proposed in 1981 keeping the waterfall model of project life cycle in mind [2]. It fails to address other popular approaches like prototype, incremental, spiral, agile models. Moreover, in present day a software project may not necessarily consist of coding of every bit of functionality. Rather, existing software components are often used and glued together towards the development of a new software. COCOMO is not suitable in such cases.
COCOMO II was proposed later in 2000 to many of address these issues.

4. Design for Use, Reuse and Sustainability

4.1 Design for USE – How long this design will work?

- As the system uses the embedded hardware components (raspberry pi) and the server/client software the system can be in used till any malfunction occur in the components, or occurrence of network issue.
- The system can remain functional up to 24*7 without any problem.

4.2 Design for reuse:

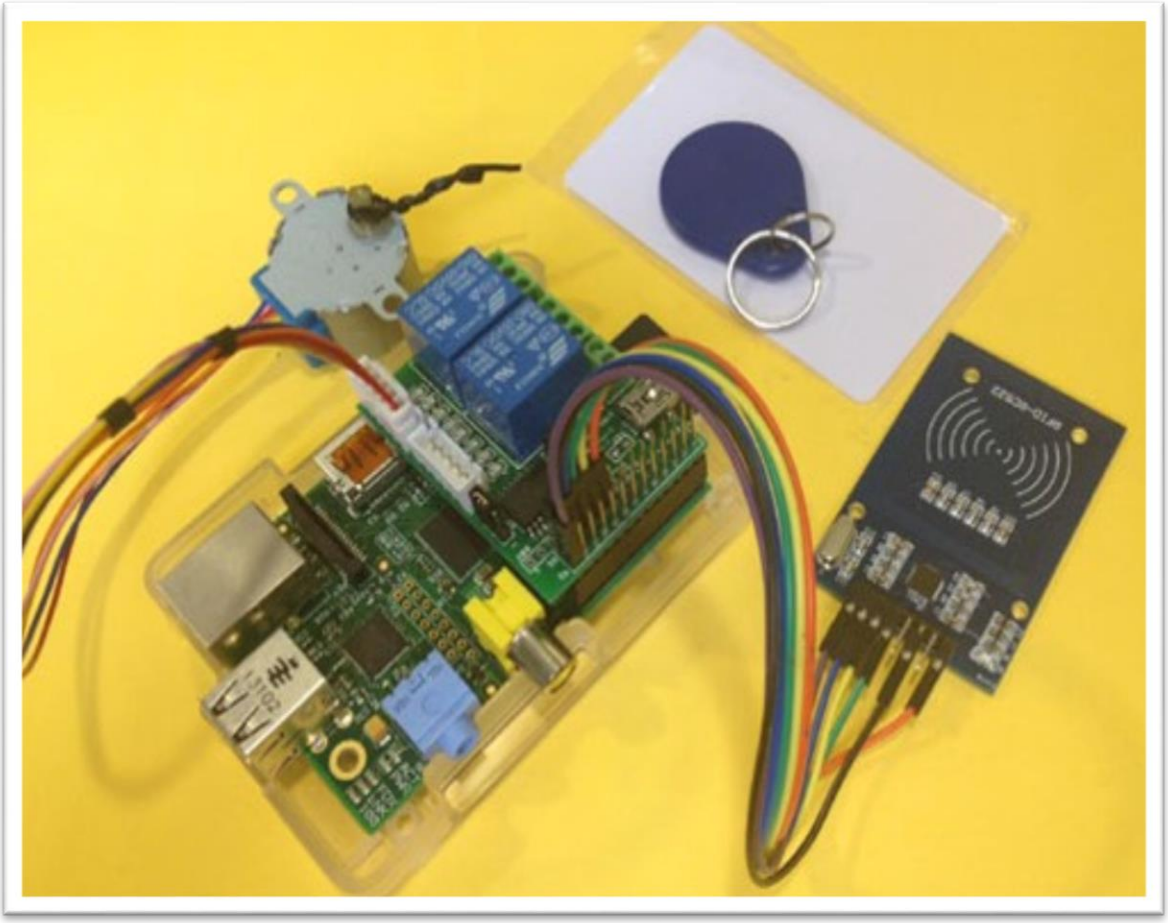
The system is reusable, as the system is an embedded, we can simply remove the system from the current location to any desirable place by doing little modification. The mobility of this system gives the freedom to do so. Other than that Raspberry Pi Technology also give us facility to include other modules like, Webcame, FingerPrint Scanner, Etc.Which make our system easily extendable.

4.3 Design for sustainability:

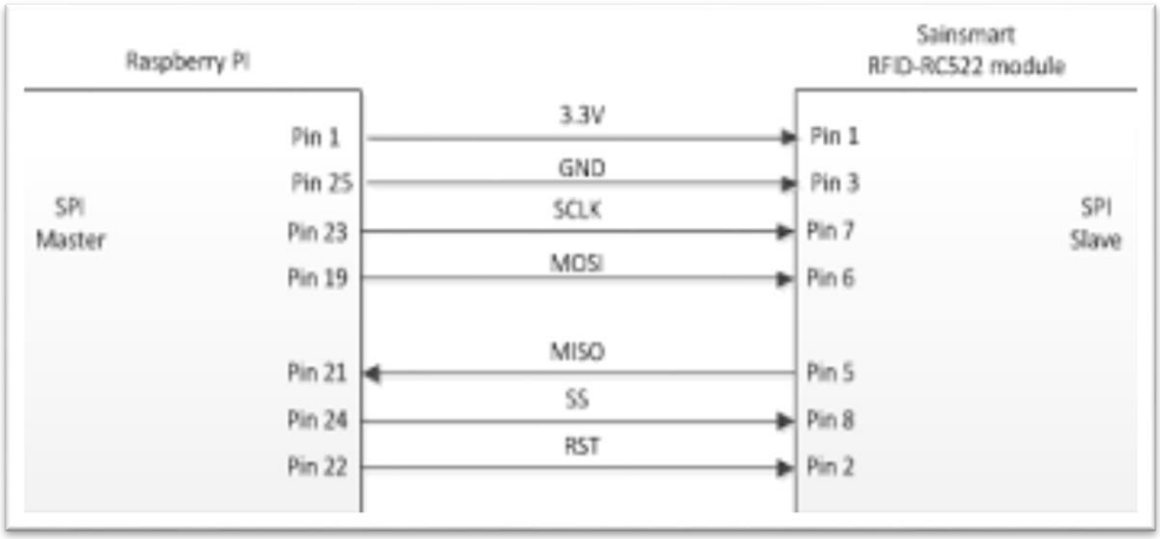
The raspberry pi uses 5v supply to run, and require network connectivity so it is sustainable as long as the network connectivity is available.

5. Prototyping:

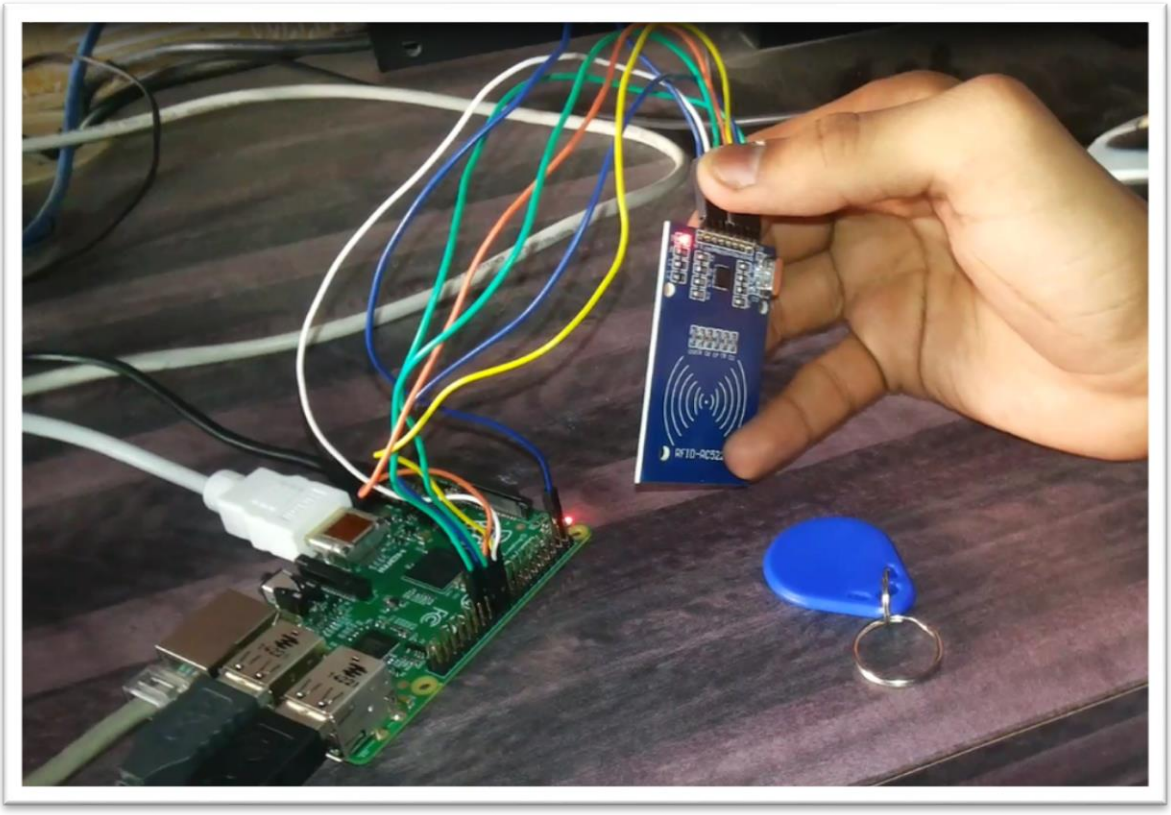
- Our team have begun our work towards our goal and we have achieve some of the result which are displayed here,



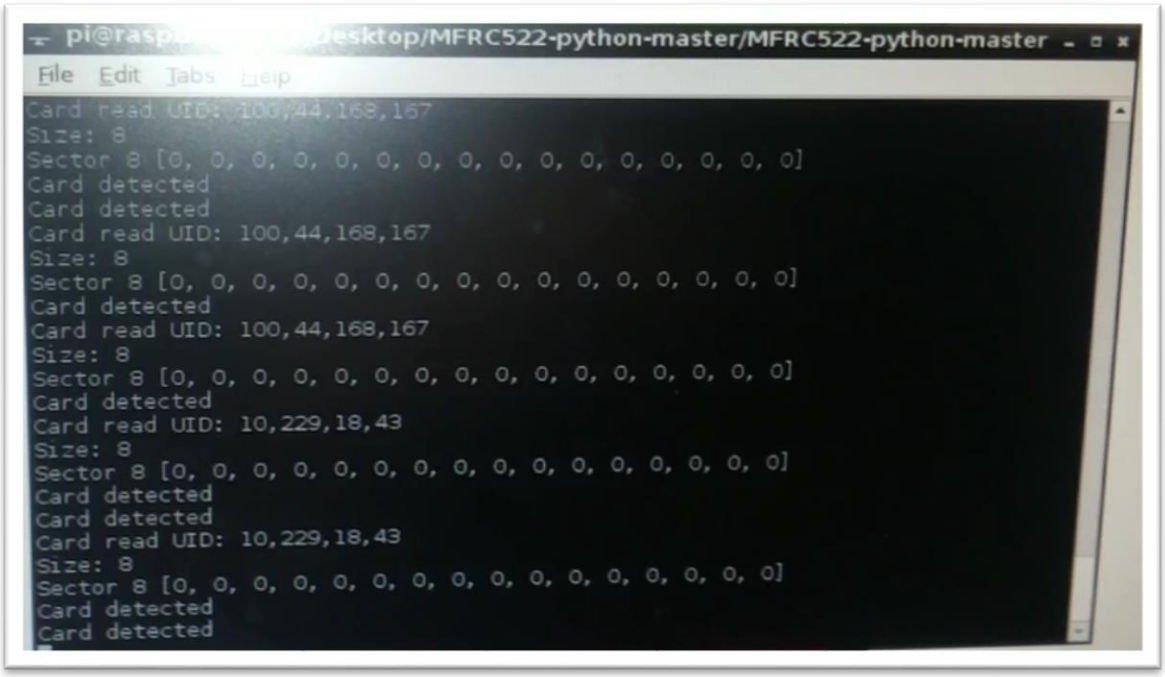
This is the basic configuration of our Raspberry pi server with RFID Module. We have setup and connect the GPIO pins with RFID this way.



Now After setup the environment we have load our python script to the raspberry pi. Now, We are running the python script Read.py which is read the Tag number from the RFID card.



After getting the card detect the screen shows like this way,



5.1 LAN Booting the Network Computer:

For Performing LAN Booting we are using Wakeonlan python API.

What is wakeonlan?

It can be used to wake up computers on your LAN. With this script you no longer have to keep your powerful, jet power hungry, computers on 24/7 if you only need them occasionally. Even when you're not at home to push the ON button, the Raspberry Pi can do that for you.

Almost all modern computers can be powered up by sending a so called magic packet over the LAN network to them. For that the computer must be connected with an ethernet cable. Most computers also have some BIOS setting and maybe even a network card setting which should be set to enable Wake On LAN, otherwise it won't work. But usually, on most computers it will work out of the box.

Usage

To wake up a computer using wake on lan it must first be enabled in the BIOS settings. Please note the computer you are trying to power on does not have an ip address, but it does have a mac address. The package needs to be sent as a broadcast package.

As a python module

- **Import the module**

```
>>> from wakeonlan import wol
```

- **Wake up a single computer by its mac address**

```
>>> wol.send_magic_packet('ff.ff.ff.ff.ff.ff')
```

- **Wake up multiple computers by their mac addresses**.

```
>>> wol.send_magic_packet('ff.ff.ff.ff.ff.ff', '00-00-00-00-00-00',
...                       'FFFFFFFFFFFF')
```

- **An external host may be specified. Do note that port forwarding on that host is required. The default ip address is 255.255.255.255 and the default port is 9.**

```
>>> wol.send_magic_packet('ff.ff.ff.ff.ff.ff',
...                       ip_address='example.com',
...                       port=1333
```

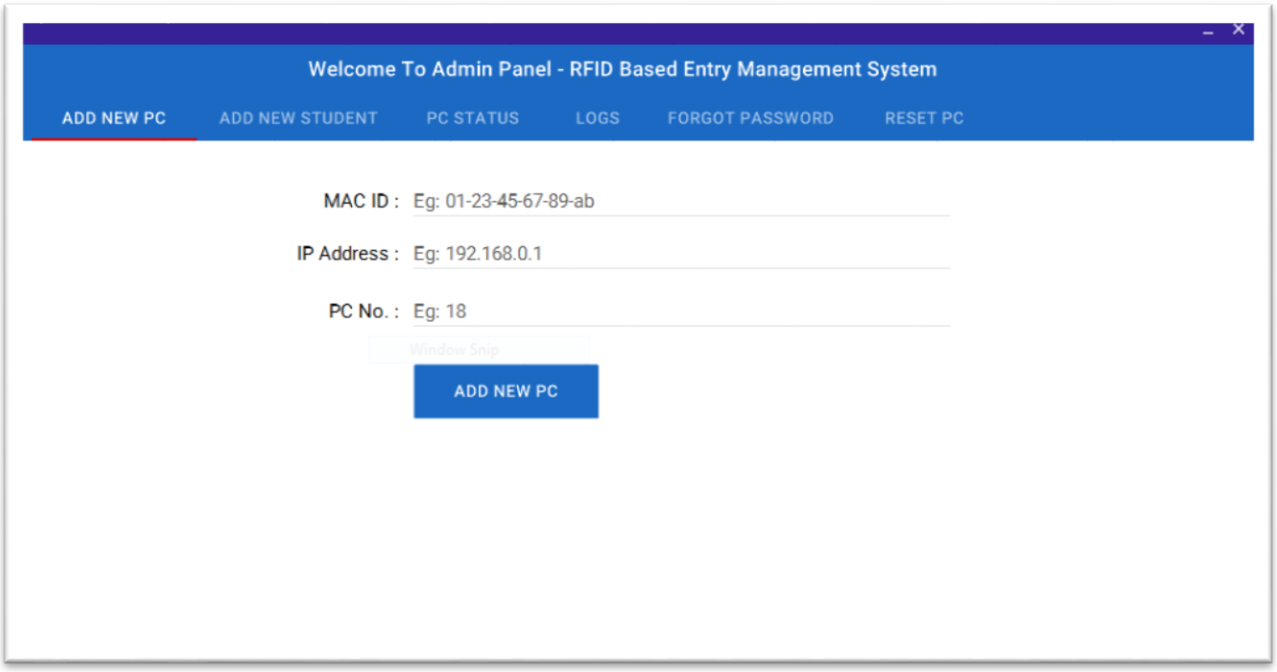
6. Prototype Testing:

➨ **Admin panel:** To test our prototype we have installed our admin panel to a local PC.

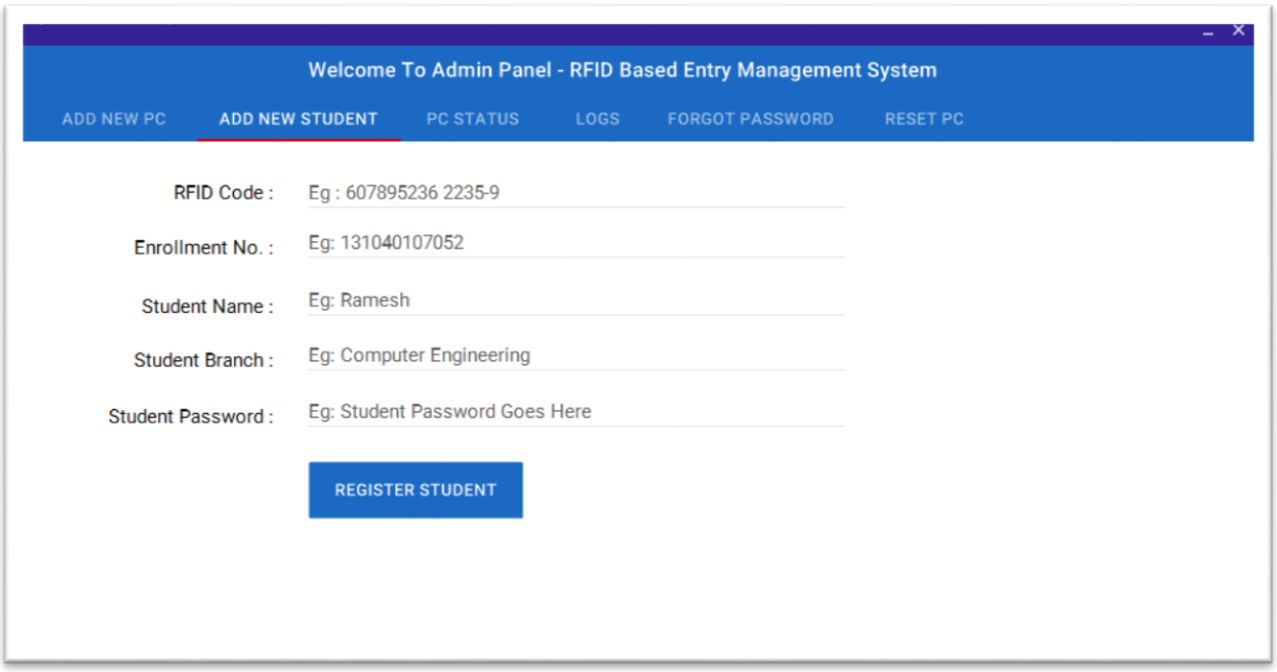
The images listed below are the images of our admin panel.

We have tested the various features that are provided in admin panel like

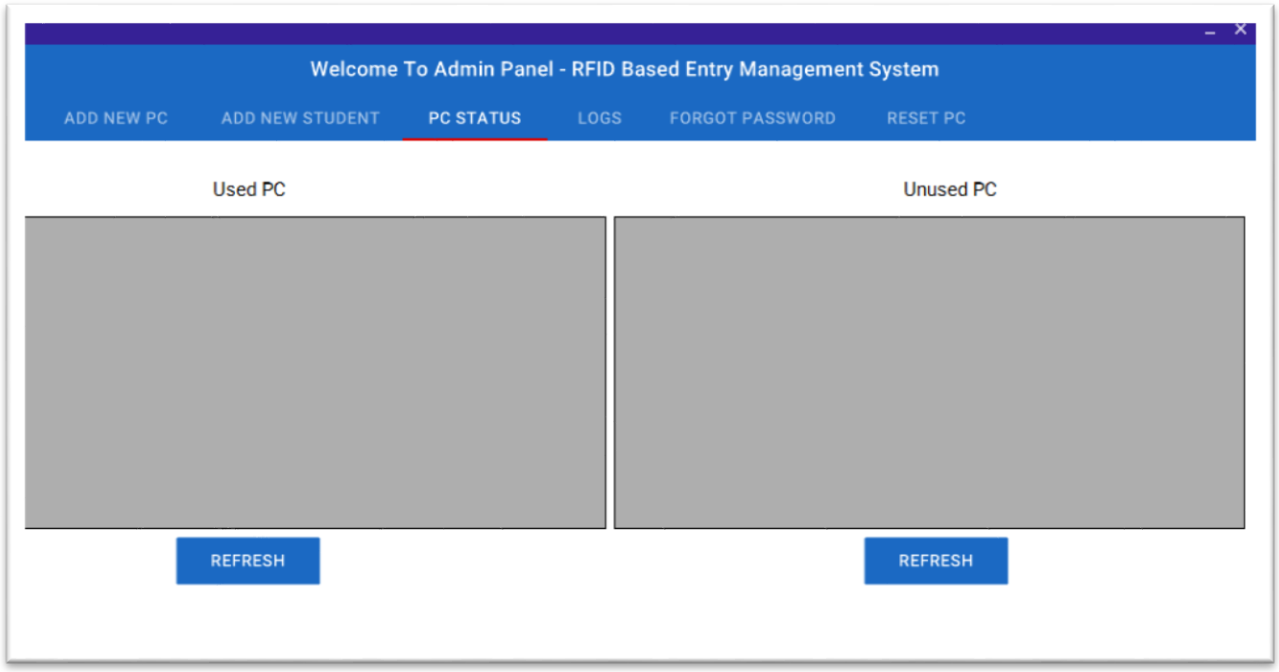
- ADD NEW PC
- ADD NEW STUDENT
- PC STATUS
- LOGS
- FORGOT PASSWORD
- RESET PC



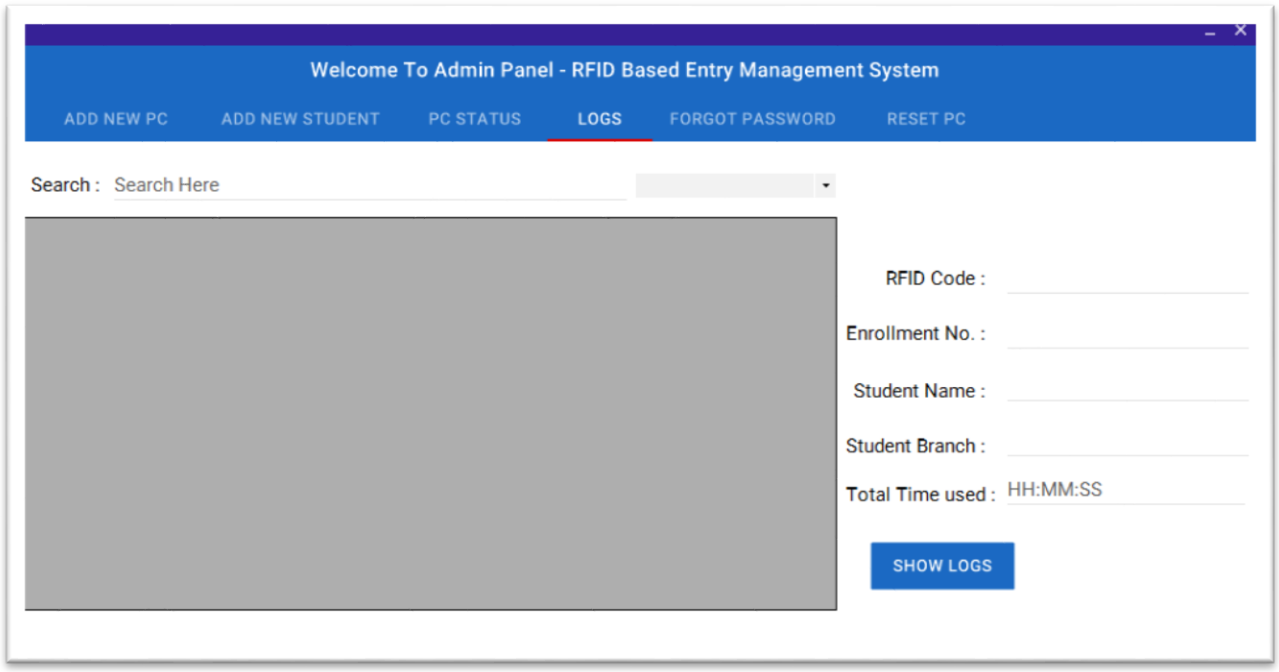
(Fig 6.1: New PC can be added)



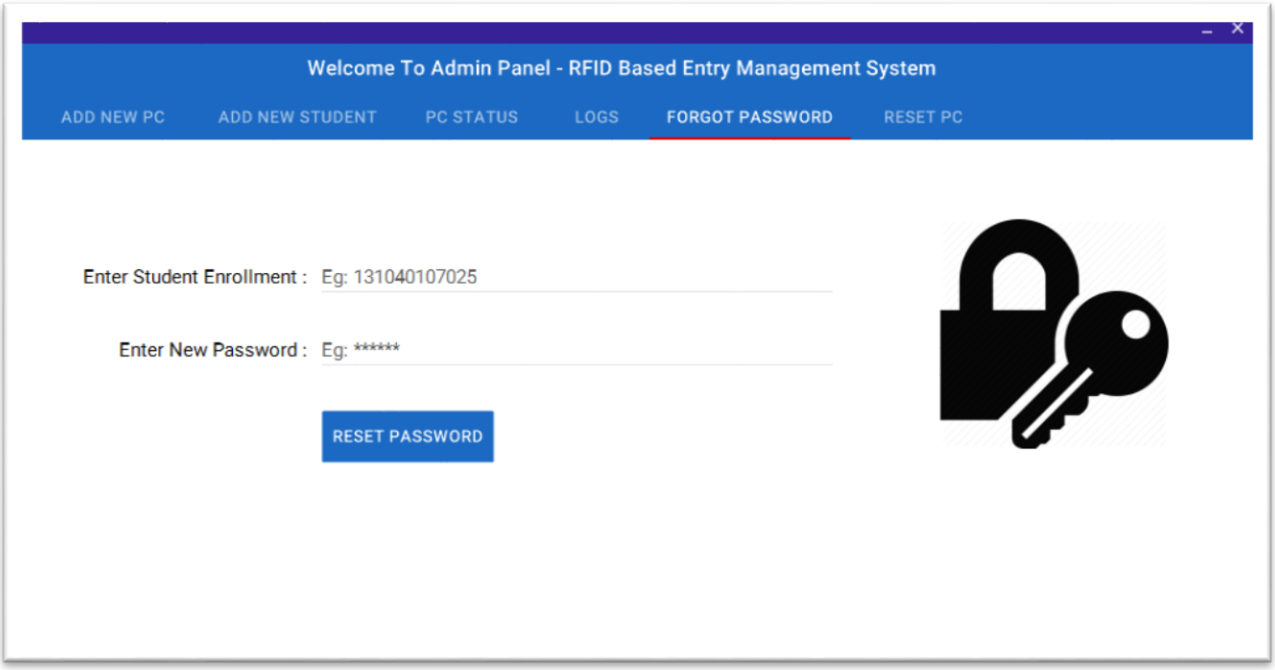
(Fig 6.2: Information of new student can be added by filling the details)



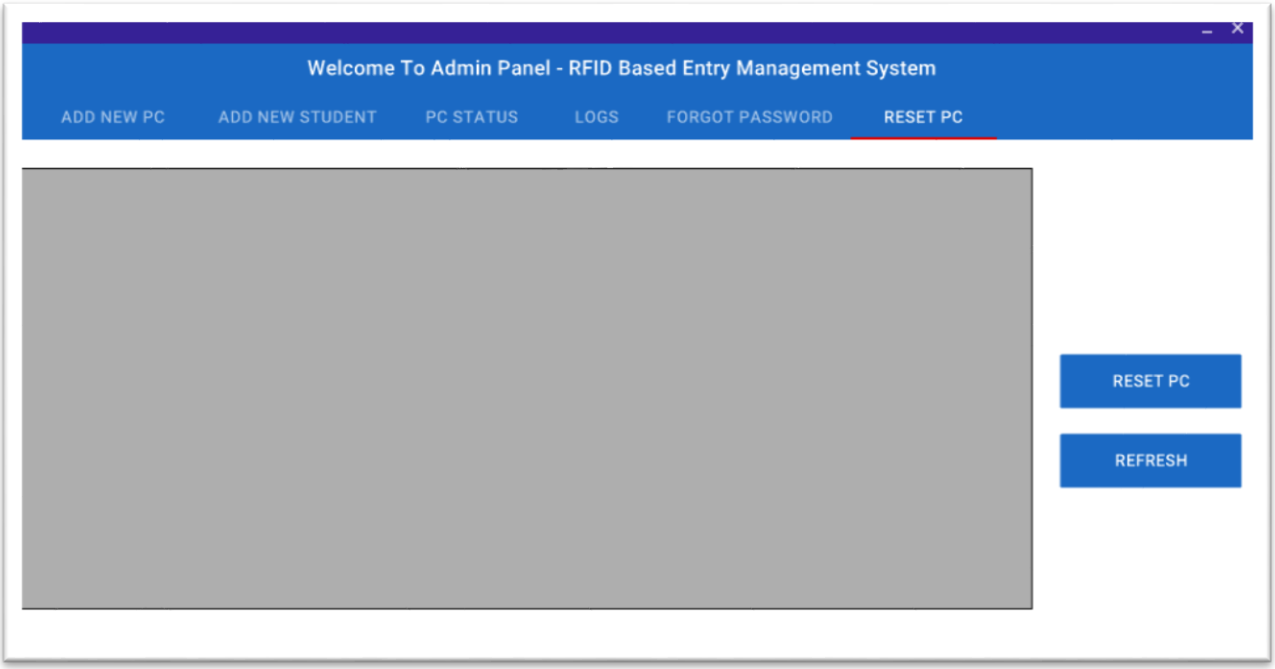
(Fig 6.3: PC STATUS shows used and unused PC)



(Fig 6.4: Searching option in Admin panel)

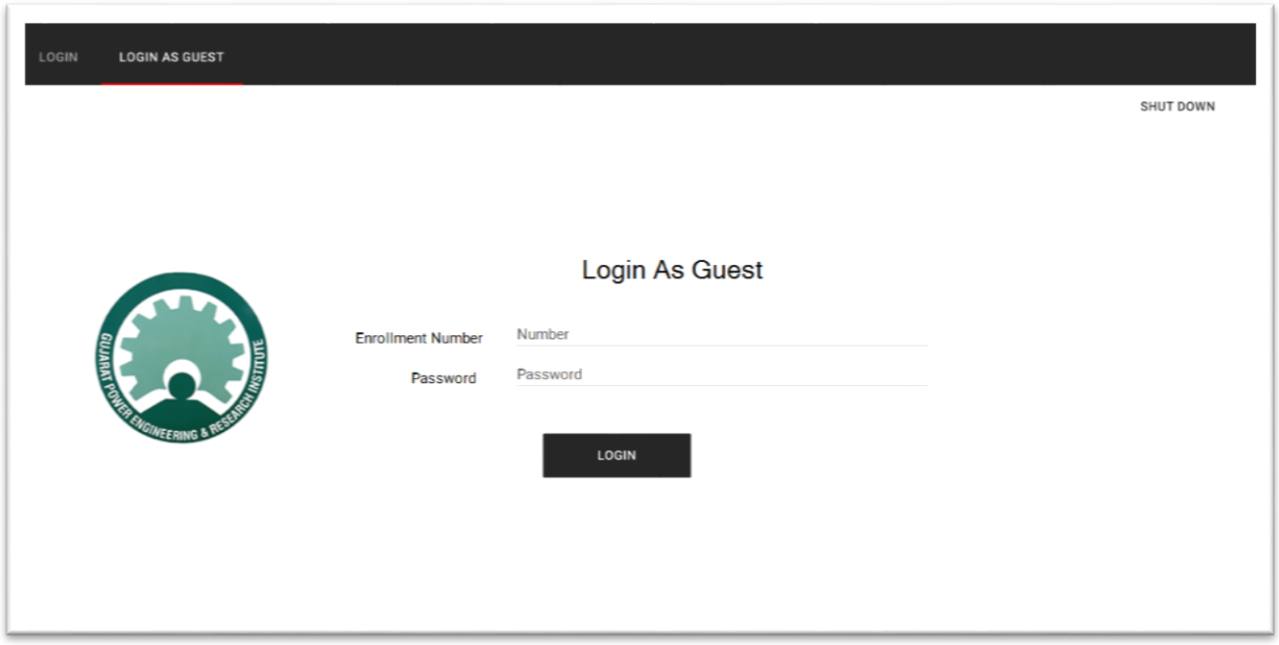


(Fig 6.5:Admin can manually reset the password)

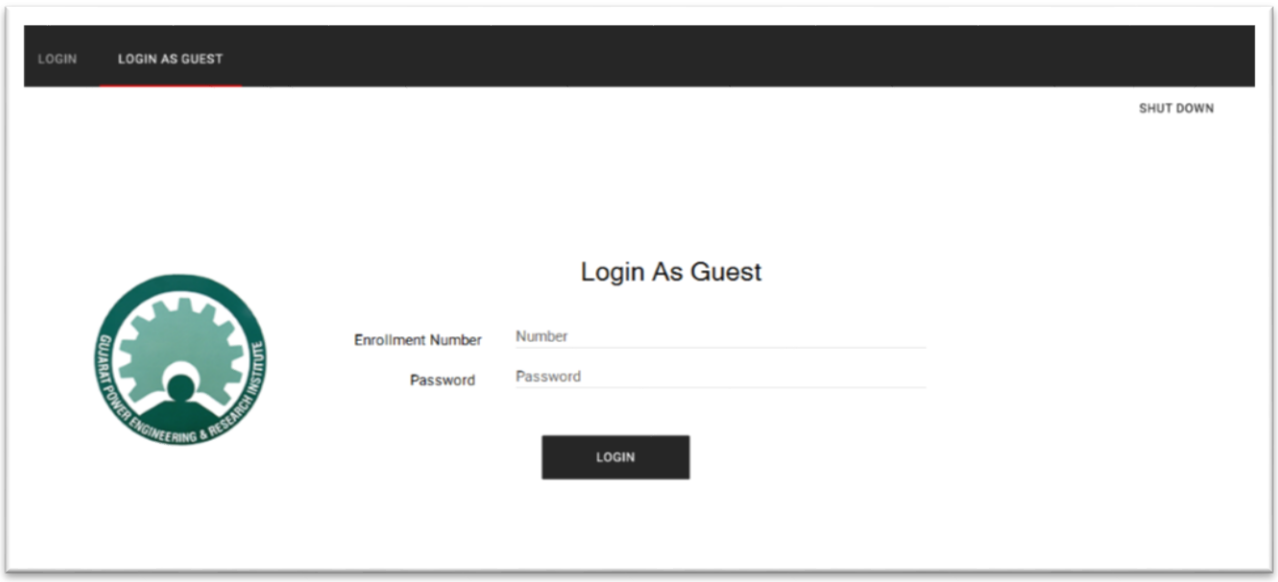


(Fig 6.6:PC reset option in Admin panel)

- We have installed client software in local PC for testing purpose, the screenshots of the client side software are as following.
    - **Login:** when the system allocate the PC to the student the System will automatically starts and the login page as following is displayed.
    - **Login as Guest:** when a user is guest then the login form should be displayed as fig 6.8.



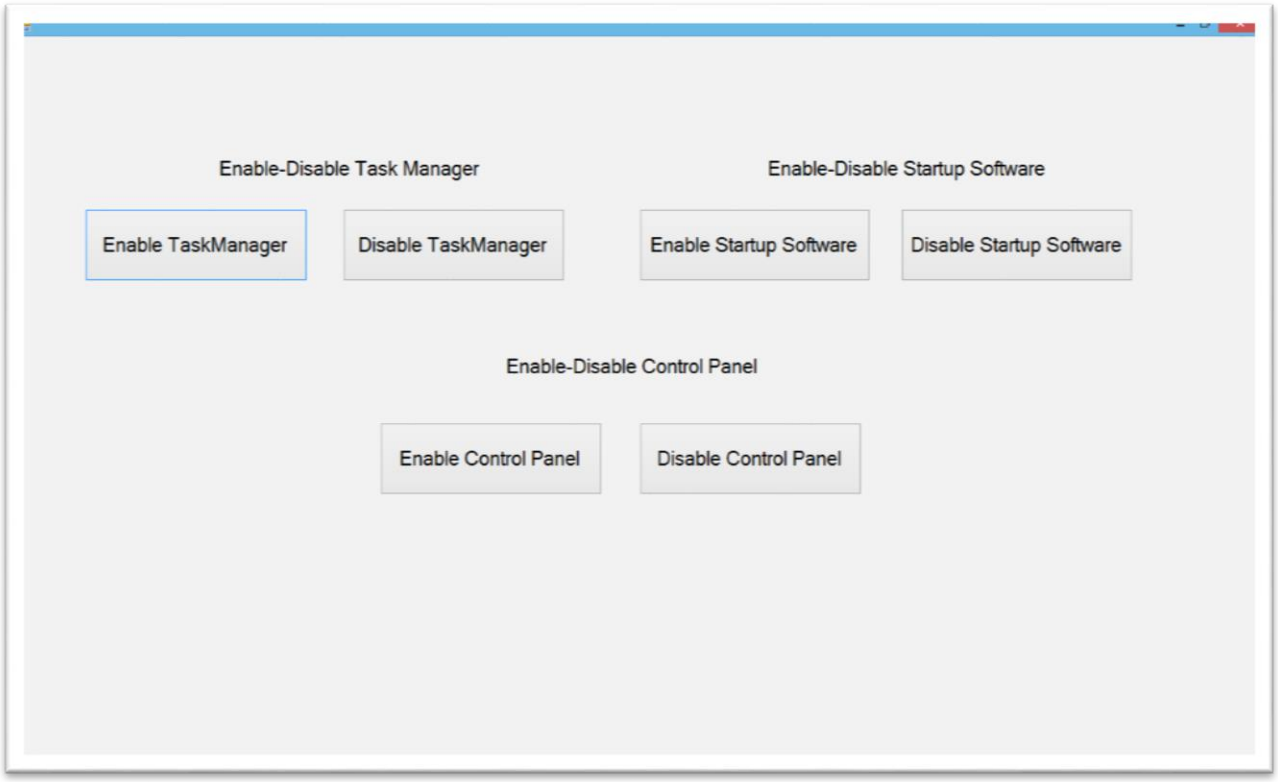
(Fig 6.7: Login page in client side software)



(Fig 6.8: Login as Guest in client side software)

- To prevent the user from unauthorized access of the task manager, startup software etc, we have provided the Enable/disable directory feature to the admin in admin panel.

  The screenshot of enable/disable registry is as following.

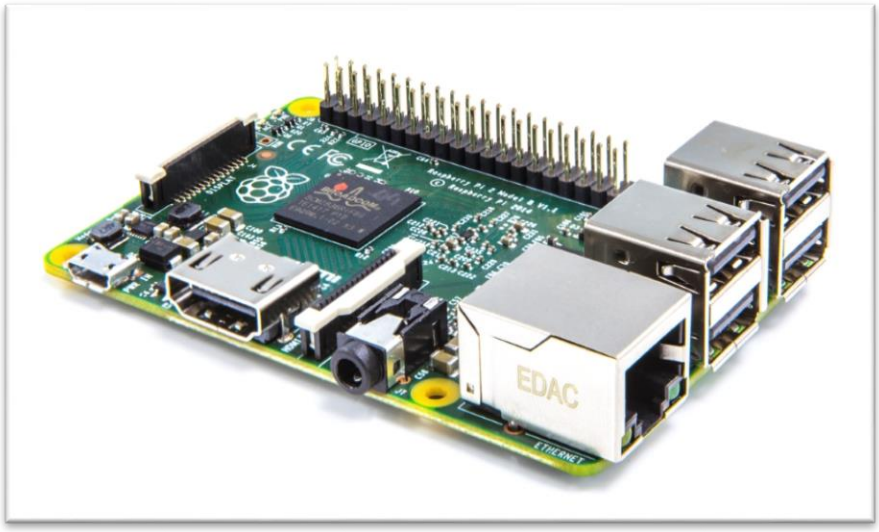


(Fig 6.9:Enable/disable registry in Admin panel)

7. Measuring Instruments/ techniques - knowledge and use, manufacturing/fabrication process, electronic circuit/boards, open source tool

7.1 Introduction to Single Board Computer:

- A **single-board computer** (**SBC**) is a complete computer built on a **single circuit board**, with **microprocessor(s)**, **memory**, **input/output (I/O)** and other features required of a functional computer for use as **embedded computer controllers**.



The Raspberry Pi 2 Model B is the second generation Raspberry Pi. It replaced the original Raspberry Pi 1 Model B+ in February 2015. Compared to the Raspberry Pi 1 it has:

- A 900MHz quad-core ARM Cortex-A7 CPU
- 1GB RAM

Like the (Pi 1) Model B+, it also has:

- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core

7.2 RFID Module Used in this Project:

(RFID Rader/Writer Module with Tags)

7. Comparison of existing materials, methods, tools and equipment for your project and justify your selection of materials, methods, tools and equipment etc.

We have reviewed 10 different IEEE Research papers in which the RFID had used from the research papers we got the basic idea that how RFID can be useful in our Entry Management System.

After deciding to use RFIDs as a unique identifier, we have to choose the server for our system. We have to find the device which can serve 24*7 and we also have to maintain the cost of the system low plus we also have to mention that the system should also be mobile.

We selected the single board computer (raspberry pi) as a server which is very cost effective, efficient and we can easily integrate it with our system.

The after the selection of the components we have to choose the platform on which we can develop our client as well as server side software.

We have selected Microsoft visual studio for making the client side software. The reason behind choosing visual studio is we want our client side software to access the sql database the visual studio provides the flexibility to make an efficient software.

9. Conclusion and Future Work:-

- Feature like, SMS/Email alert.
- We will wide our definition of the project form rather than just focusing on the computer system entry management, we will do different project like library management, toll both management, etc.
- We will develop our own software for the system which will include all the feature as per the requirement of the firm.
- Finalize a EMS Product & Patent it.
- Reset Password , Access Logs, Etc. facilities from Android App.
- SMS/Email Alert.
- Integrating & Centralizing the EMS.
- Integrating IoT Modules to improve the Efficiency.
- Integrating EMS with Ticket Management System.

8. References:

[1] RFID Technology Based Attendance Management System,Sumita Nainan, Romin Parekh, Tanvi Shah,NMIMS University Mumbai, Maharashtra 400 056. INDIA

[2] RFID Based Students Attendance Management System , Arulogun O. T., Olatunbosun, A., Fakolujo O. A., and Olaniyi, O. M. , Department of Computer Science and Engineering , Ladoke Akintola University of Technology, Ogbomoso, Nigeria

[3] RFID Based Security System , K.Srinivasa Ravi, G.H.Varun, T.Vamsi, P.Pratyusha

[4] RFID Technology Based Security and Access Control System , Umar Farooq, Mahmood ul Hasan, Muhammad Amar, Athar Hanif, and Muhammad Usman Asad

[5] RFID Technology Based Attendance Management System Parvathy A, Venkata Rohit Raj,Venumadhav Reddy M, Manikanta Chaitanya.G , ECE/SEE SASTRA University , Thanjavur ,Tamil Nadu, India

[6] RFID Technology Based Library Management System , Priyanka Grover and Anshul Ahuja Information Technology Engg. Lingaya's Institute of Mngt. & Tech, India

[7]. WIKIPEDIA

[8]. IEEE JOURNAL

[9]. Adeept.com RFID with Raspberry pi Integration Article.