IoT BASED STUDENT BIOMETRIC ATTENDANCE USING FINGERPRINT

A Thesis

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

In

COMPUTER SCIENCE AND ENGINEERING

By

SAJJA MOHAN KUMAR

(15001F0037)

Under the esteemed guidance of

Dr. P. Chenna Reddy

Professor of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY

COLLEGE OF ENGINEERING

(Autonomous)

ANANTHAPURAMU-515002

ANDHRA PRADESH-INDIA

JUNE, 2018

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY

COLLEGE OF ENGINEERING

(Autonomous)

ANANTHAPURAMU-515002

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"IoT Based Student Biometric Attendance Using Fingerprint"** is a bonafide work of **Mr. Sajja Mohan Kumar,** bearing Admn. No: 15001F0037, submitted to the faculty of Computer Science & Engineering, in partial fulfillment of the requirements for the award of degree of **MASTER OF COMPUTER APPLICATIONS** from Jawaharlal Nehru Technological University, Anantapuramu, College of Engineering (Autonomous), Anantapuramu.

Signature of the Supervisor	Signature of Head of the Dept
Dr. P. Chenna Reddy **M.Tech, Ph.D.**	**Dr. S. Vasundra** **M.Tech, Ph.D.**
Professor of C.S.E	**Professor & Head of CSE Dept.**
Dept. of Computer Science & Engg.	Dept. of Computer Science & Engg.
J.N.T. University Ananthapur	J.N.T.U.A. College of Engineering
Ananthapuramu-515002.	Ananthapuramu-515002.

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY

COLLEGE OF ENGINEERING

(Autonomous)

ANANTHAPURAMU-515002

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

I, **Sajja Mohan Kumar,** bearing admission no. **15001F0037,** hereby declare that the project report entitled **"IoT Based Student Biometric Attendance Using Fingerprint"** under the esteemed guidance of **Dr. P. CHENNA REDDY,** Professor & Director Skill Development Center & Incubation Center, JNTUA Anantapuramu is submitted in partial fulfillment of the requirements for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** in Computer Science & Engineering.

This is a record of bonafide work carried out by me and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

S Mohan Kumar

(15001F0037)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any tasks would be incomplete without the mention of the people who made it a possible, whose constant guidance and encouragement crowned our efforts with success. It is pleasant aspect that I have now the opportunity to express my gratitude for all of them.

I wish to thank **Prof. K. RAMANAIDU**, Principal of JNTUA College of Engineering (Autonomous), Ananthapuramu who has extended his support for the success of this project.

I wish to thank **Dr. S. VASUNDRA,** Professor and Head of Computer Science and Engineering Department, JNTUA college of Engineering (Autonomous), Anantapuramu. Her wide support, knowledge and enthusiastic encouragement have impressed me to better involvement into my project thesis and technical design also her ethical morals helped me to develop my personal and technical skills to deploy my project in success.

I would like to thank is my guide **Dr. P. CHENNA REDDY,** Professor of Computer Science & Engineering Department, Director Skill Development Center & Incubation Center of Jawaharlal Nehru Technological University Anantapur. His wide knowledge and logical way of thinking have made a deep impression on me. His understanding, encouragement and personal guidance have provided the basic for this thesis. His source of inspiration for innovative ideas and his kind support is well to all his students and colleagues.

I wish to thank teaching staff and non-teaching staff of Computer Science & Engineering Department, JNTUA College of Engineering Anantapuramu for giving support to completion of my project successfully.

Last but far from least, I also thank my family members and my friends and my juniors for their moral support and constant encouragement. I am very much thankful to one and all who helped me for the successful completion of the project.

Sajja Mohan Kumar

(15001F0037)

Abstract

This project uses a biometric concept to facilitate the attendance system in Educational institutes. It uses the most reliable way of uniquely identifying students through fingerprint analysis. A fingerprint-based biometric system is a good combination of low cost and high accuracy. Through this application, we can keep a systematic track of student's attendance. This project enables the easy way of maintaining class attendance with fewer efforts.

In the present system, the most common method to take student attendance in the classroom is by reading out the names or numbers of each student and mark the attendance manually in a register. This system requires a lot of paperwork. All calculations to generate report are done manually, so there is a greater chance of errors. The existing system is not user-friendly because the retrieval of data is very slow and data is not maintained efficiently. This approach, like manually taking and maintaining the attendance records is an inconvenient task.

The proposed system aims to automate the old and hectic process of manually taking and storing student attendance records. The records are securely stored and can be reliably retrieved whenever required by the teacher. This biometric student attendance system increases the efficiency of the process of taking student attendance. The presence of each student will be updated and the data will be passed to the server.

This project collects attendance in the form of an Internet of Things (IoT) based system that records the attendance using fingerprint-based biometric scanner and stores them securely. Here, Raspberry Pi is used to build a low-cost biometric system with the help of fingerprint scanner and python. Our module enrolls the student's as well as staffs fingerprints. This enrolling is a onetime process and their fingerprints will be stored in a database. Here person's authentication is done by updating time and attendance to a Web server.

CONTENTS

Chapter 1

INTRODUCTION

IoT based Student Biometric Attendance using Fingerprint is software developed for daily student attendance in schools, colleges and institutes**.** If facilitates to access the attendance information of a particular student in a particular class. The information is sorted by the operators, which will be provided by the teacher for a particular class.

The purpose of developing this system is to computerized the tradition way of taking attendance. Another purpose for developing this software is to generate the report automatically at the end of the session or in the between of the session

The scope of the project is the system on which the software is installed, i.e. the project is developed as a desktop application, and it will work for a particular institute. But later on the project can be modified to operate it online.

1.1 MOTIVITION

The world today is transforming to a Web Technology each and every functioning is happening with the small devices at our hands and PC. And it is the time for us to transform according to the requirements of the staff and student. This motivated us to develop an web application to student attendance.

1.2 OBJECTIVE OF THE PROJECT

The main objective of this system is to present an automated system for IoT based Biometric Attendance using Fingerprint. This web application is used for the information about the attendance of the student.

1.3 EXISTING SYSTEM

In the present system all work is done on paper. The whole session attendance is stored in register and at the end of the session the reports are generated. We are not interested in generating report in the middle of the session or as per the requirement because it takes more time in calculation. At the end of session the students who don't have 75% attendance get a notice.

Limitations Identified

• **Not User Friendly:** The existing system is not user friendly because the retrieval of data is very slow and data is not maintained efficiently.

• **Difficulty in report generating:** We require more calculations to generate the report so it is generated at the end of the session. And the student not gets a single chance to improve their attendance

• **Manual control**: All calculations to generate report are done manually so there is greater chance of errors.

• **Lots of paperwork**: Existing system requires lot of paper work. Even a single register/record led to difficult situation because all the papers are needed to generate the reports.

• **Time consuming**: Every work is done manually so we cannot generate report in the middle of the session or as per the requirement because it is very time consuming.

1.4 PROPOSED SYSTEM

• **User Friendly:-** The proposed system is user friendly because the retrieval and storing of data is fast and data is maintained efficiently. And also the graphical user interface is provided in the proposed system, which provides user to deal with the system very easily.

• **Reports are easily generated:** reports can be easily generated in the proposed system so user can generate the report as per the requirement (monthly) or in the middle of the session. User can give the notice to the students so he/she become regular.

• **Very less or no paper work:** The proposed system requires very less paper work. All the data is feted into the computer immediately and reports can be generated through computers.
And also work becomes very easy because there is no need to keep data on papers.

• **Computer operator control**: Computer operator control will be there so no chance of errors. Moreover storing and retrieving of information is easy. So work can be done speedily and in time.

1.5 SOFTWARE MODEL

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.
- ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.

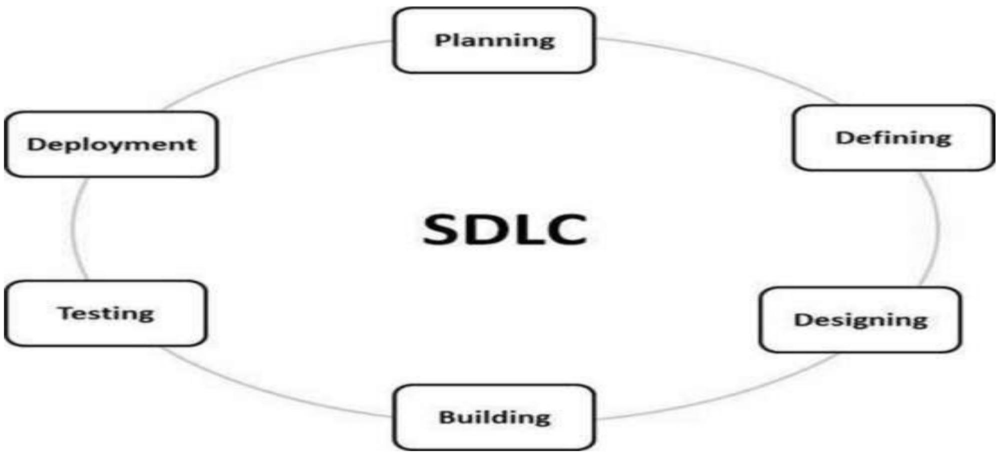


Fig:1.5.1 Various stages of SDLC

A typical Software Development Life Cycle consists of the following stages −

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry &miuns;

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Mode

SDLC - Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

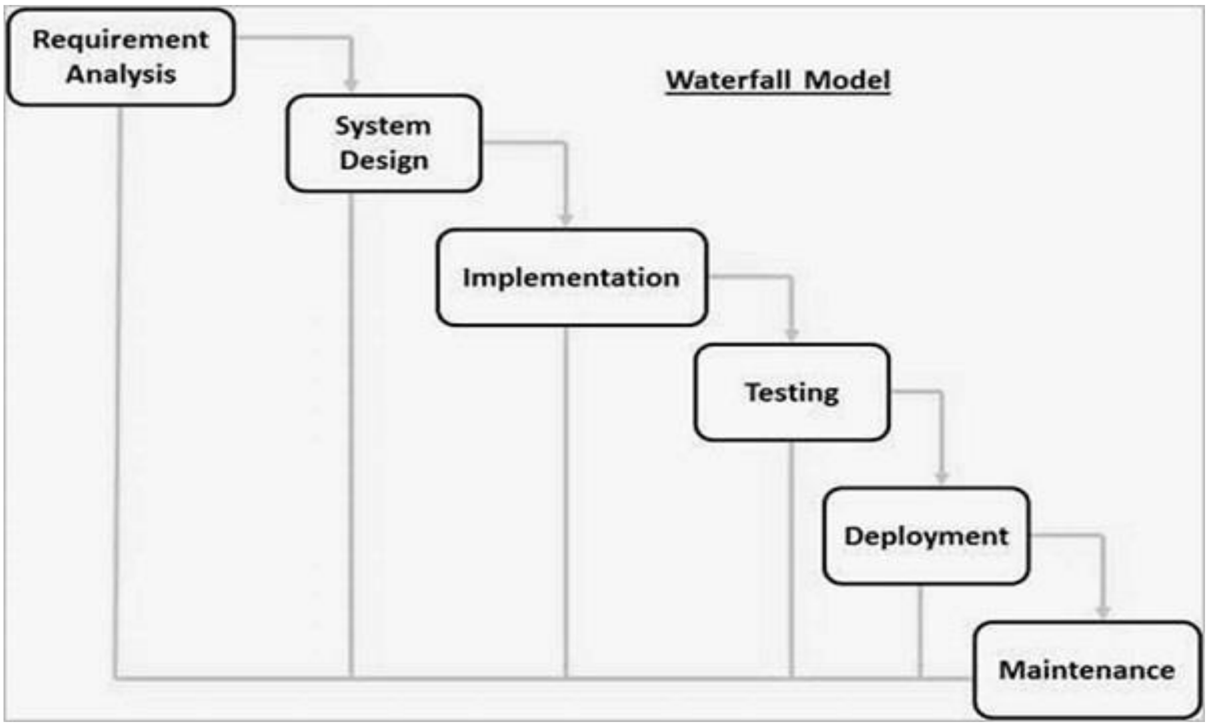


Fig: 1.5.2 Different phases in waterfall model

The sequential phases in Waterfall model are −

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows −

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.

- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

1.6 ORGANIZATION THESIS

The rest of the thesis is organized as fallows.

Chapter2: This chapter explains the requirement analysis and requirement specification of the system.

Chapter3: This chapter explains how to setup and install both software and hardware of the project.

Chapter4: This chapter presents the modules of the system.

Chapter5: This chapter covers the screens of the forms and reports generated in the project.

Chapter6: This chapter contains the conclusion.

Chapter 2
ANALYSIS

2.1 INTRODUCTION

The **analysis phase** defines the **requirements** of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the ``what" phase. The requirement document tries to capture the requirements from the customer's perspective by defining goals and interactions at a level removed from the implementation details.

2.2 REQUIREMENT ANALYSIS

SNO	**Requirement**	**Essential**	**Description**	**Remarks**
Rs1	Should have the login page	Essentia l	The Login Page should appear when the system is invoked	
Rs2	Add New Student	Essentia l	The logged in authorized user can Add New Student	In this admin can add only one Student
Rs3	Enroll Student Finger Print	Essentia l	The authorized user can enroll a student fingerprint to finger print scanner.	In this admin can enroll the student’s finger-print manually to finger-print scanner.
Rs4	Take Attendance	Essentia l	The device Raspberry Pi 3 connected with R305 fingerprint scanner will takes the Attendance by analyzing the fingerprint.	Always need internet.

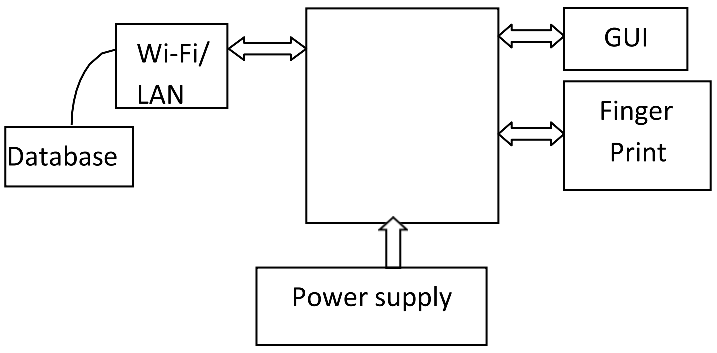
Rs5	Results	Essential	Automated Reports Generator gives automated attendance results.	

Table: 2.2.1 Requirement Analysis

2.3 REQUIREMENT SPECIFICATION

2.3.1 HARDWARE ARCHITECTURE:

This proposed system consists of raspberry pi, fingerprint module, LCD display and 4x4 Matrix Keypad. This system provides the features of real time authentication facilities



2.3.1. Proposed system block diagram

2.3.1.1. RASPBERRY PI 3:

Hardware is a physical device that can be touched or held, like a hard drive or a mobile phone. Software can be thought of as a program or a collection of programs that instruct a computer on what to do and how to do it. Below is an image of the Raspberry Pi which describes some of the components that make up the hardware.

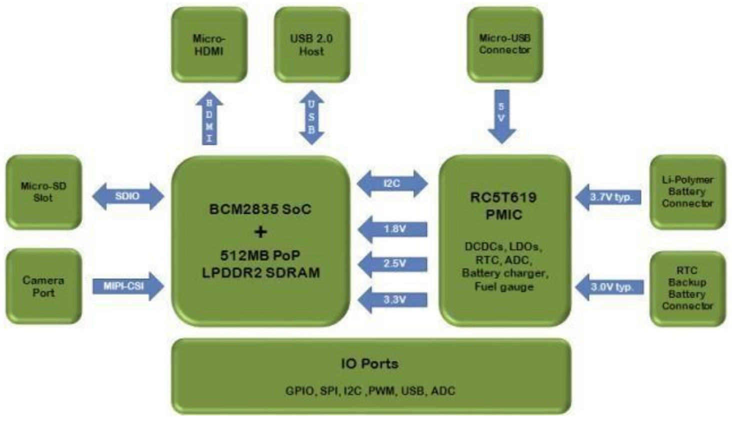

Micro-HDMI
USB 2.0 Host
Micro-USB Connector
HDMI
USB
5V
Micro-SD Slot
SDIO
BCM2835 SoC + 512MB PoP LPDDR2 SDRAM
I2C
1.8V
2.5V
3.3V
RC5T619 PMIC
DCDCs, LDOs, RTC, ADC, Battery charger, Fuel gauge
3.7V typ.
Li-Polymer Battery Connector
Camera Port
MIPI-CSI
3.0V typ.
RTC Backup Battery Connector
IO Ports
GPIO, SPI, I2C ,PWM, USB, ADC

Fig.2.3.2. Hardware Components of Raspberry Pi.

The shows the hardware description of raspberry pi. The real time vehicle tracking system uses the GPIO pin, Micro SD slot, USB port and Micro USB connector. The GPIO pins are used for serial communication for interfacing GSM and GPS. It uses 8GB SD for installing the Raspbian OS and for storage. The USB port is used for connecting keyboard, mouse, dongle and pen drive. The power supply is given through USB connector.

2.3.1.2. FINGERPRINT READER:

The R305 is the module which is used in the project. This module supports both windows and Linux based operating system. This self-contained module optically scans the fingerprint when the user touches the glowing window.

2.3.3. Finger Print Module

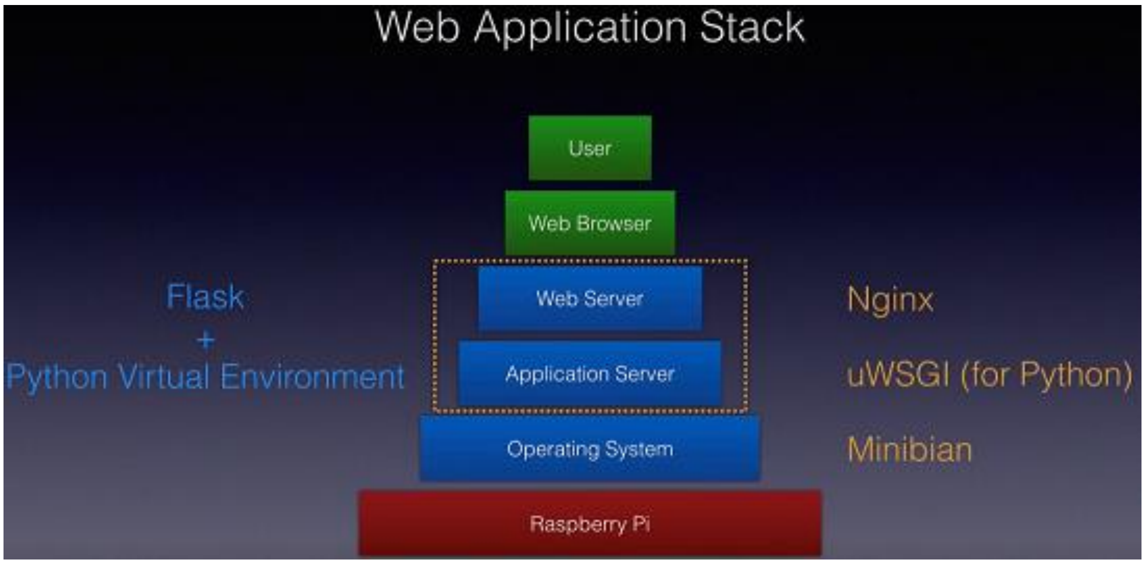
Optical technology gives the highest quality fingerprint templates and reliability along with the raspberry pi embedded computer.

The device can automatically control calibration, encryption, and data transfer through the USB interface.

This module is found to be more reliable for all kind of OS, even its easy to interface with raspberry pi. The R305 Module and digital Persona Fingerprint Recognition Engine have an unmatched ability to authenticate even the most difficult fingerprints accurately and rapidly.

2.3.2. SOFTWARE ARCHITECTURE:

For this project the used soft wares are like Python, SQLite, Flask (framework). The configuring procedures, saving and retrieval of fingerprint templates procedures are mentioned in the upcoming chapters.



2.3.2.1. PYTHON PROGRAMMING SOFTWARE:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

2.3.2.2 FLASK (Python Framework):

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins. In the case of Flask, its dependencies are:

- Werkzeug a WSGI utility library
- jinja2 which is its template engine

2.3.2.2 SQLite3:

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete

SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. Think of SQLite not as a replacement for Oracle but as a replacement for fopen()

SQLite is a compact library. With all features enabled, the library size can be less than 500KiB, depending on the target platform and compiler optimization settings. (64-bit code is larger. And some compiler optimizations such as aggressive function inlining and loop unrolling can cause the object code to be much larger.) There is a tradeoff between memory usage and speed. SQLite generally runs faster the more memory you give it. Nevertheless, performance is usually quite good even in low-memory environments. Depending on how it is used, SQLite can be faster than direct filesystem I/O.

SQLite is very carefully tested prior to every release and has a reputation for being very reliable. Most of the SQLite source code is devoted purely to testing and verification. An automated test suite runs millions and millions of test cases involving hundreds of millions of individual SQL statements and achieves 100% branch test coverage. SQLite responds gracefully to memory allocation failures and disk I/O errors. Transactions are ACID even if interrupted by system crashes or power failures. All of this is verified by the automated tests using special test harnesses which simulate system failures. Of course, even with all this testing, there are still bugs. But unlike some similar projects (especially commercial competitors) SQLite is open and honest about all bugs and provides bugs lists and minute-by-minute chronologies of code changes.

The SQLite code base is supported by an international team of developers who work on SQLite full-time. The developers continue to expand the capabilities of SQLite and enhance its reliability and performance while maintaining backwards compatibility with the published interface spec, SQL syntax, and database file format. The source code is absolutely free to anybody who wants it, but professional support is also available.

2.3.2.3 NGINX SERVER:

NGINX is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. NGINX is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

NGINX is one of a handful of servers written to address the C10K problem. Unlike traditional servers, NGINX doesn’t rely on threads to handle requests. Instead it uses a much more scalable event-driven (asynchronous) architecture. This architecture uses small, but more importantly, predictable amounts of memory under load. Even if you don’t expect to handle thousands of simultaneous requests, you can still benefit from NGINX’s high-performance and small memory footprint. NGINX scales in all directions: from the smallest VPS all the way up to large clusters of servers.

NGINX powers several high-visibility sites, such as Netflix, Hulu, Pinterest, CloudFlare, Airbnb, WordPress.com, GitHub, SoundCloud, Zynga, Eventbrite, Zappos, Media Temple, Heroku, RightScale, Engine Yard, MaxCDN and many others.

2.3.2.4 uWSGI:

uWSGI is a Python application server, which will execute our Python programs on behalf of the Nginx web server.

uWSGI is a software application that "aims at developing a full stack for building hosting services". It is named after the Web Server Gateway Interface (WSGI), which was the first plugin supported by the project.

uWSGI is often used for serving Python web applications in conjunction with web servers such as Cherokee and Nginx, which offer direct support for uWSGI's native uwsgi protocol.

Chapter 3

SETTING UP REQUIREMENTS

3.1 INSTALLATIONS

3.1.1 INSTALLING O.S.:

This resource explains how to install a Raspberry Pi operating system image on an SD card. You will need another computer with an SD card reader to install the image.

Download the image

Official images for recommended operating systems are available to download from the Raspberry Pi website Downloads page.

Alternative distributions are available from third-party vendors.

If you're not using Etcher (see below), you'll need to unzip .zip downloads to get the image file (.img) to write to your SD card.

Note: the Raspbian with Raspberry Pi Desktop image contained in the ZIP archive is over 4GB in size and uses the ZIP64 format. To uncompress the archive, a unzip tool that supports ZIP64 is required. The following zip tools support ZIP64:

- 7-Zip (Windows)
- The Unarchiver (Mac)
- Unzip (Linux)

Writing an image to the SD card

You will need to use an image writing tool to install the image you have downloaded on your SD card.

Etcher is a graphical SD card writing tool that works on Mac OS, Linux and Windows, and is the easiest option for most users. Etcher also supports writing images directly from the zip file, without any unzipping required. To write your image with Etcher:

- Download Etcher and install it.

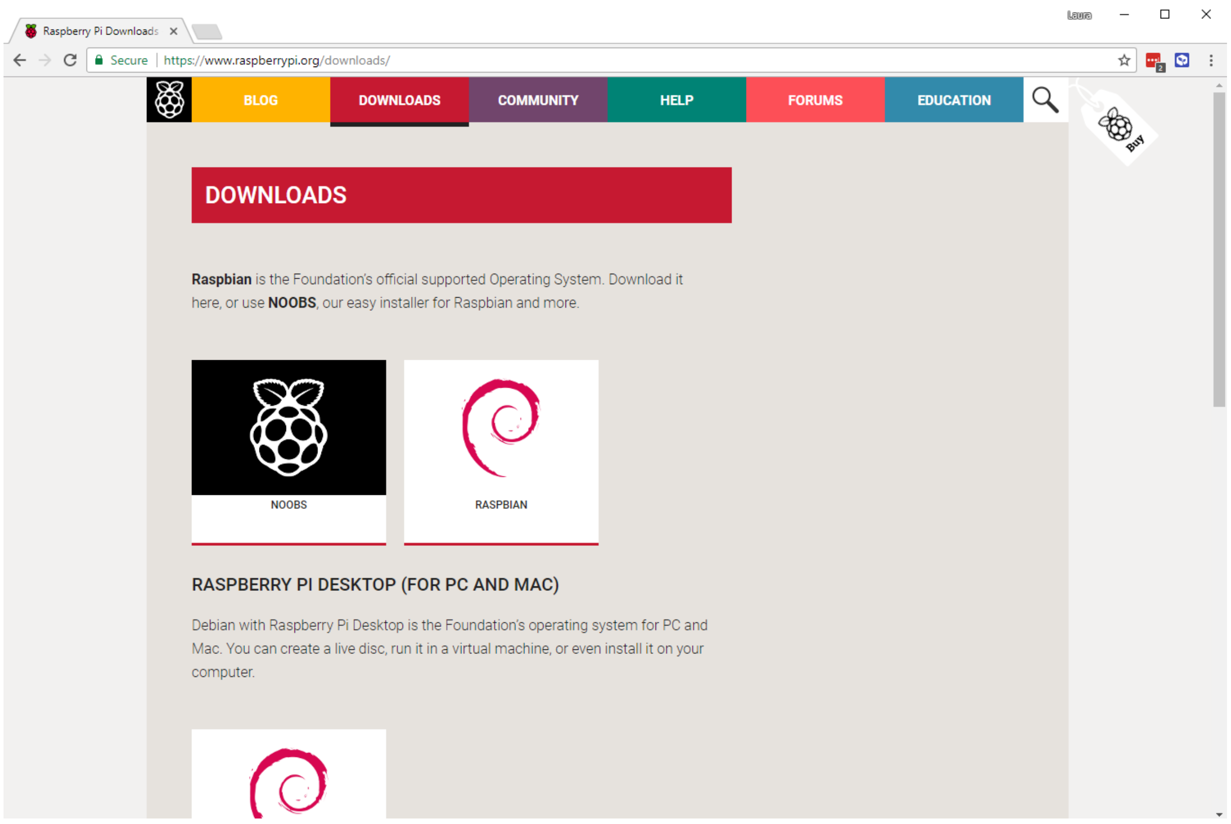
- Connect an SD card reader with the SD card inside.
- Open Etcher and select from your hard drive the Raspberry Pi .img or .zip file you wish to write to the SD card.
- Select the SD card you wish to write your image to.
- Review your selections and click 'Flash!' to begin writing data to the SD card.

There is another way to install OS by downloading NOOBS.

Downloading NOOBS

Using NOOBS is the easiest way to install Raspbian on your SD card. To get hold of a copy of NOOBS:

- Visit www.raspberrypi.org/downloads/



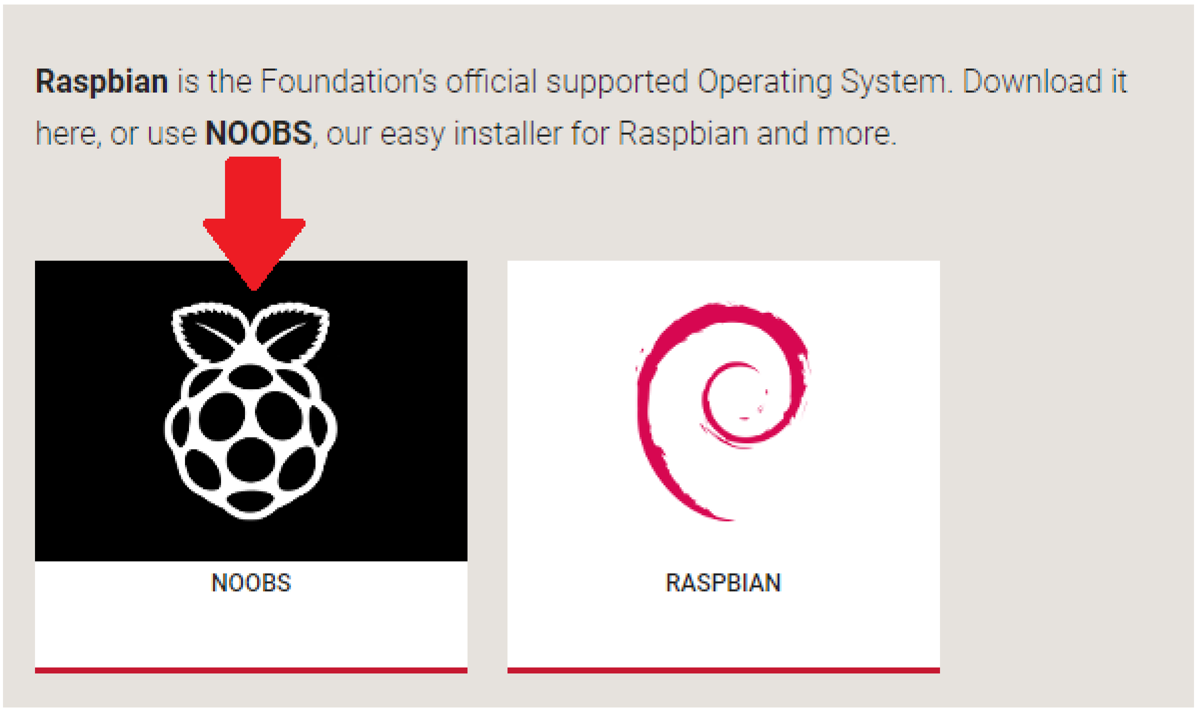
DOWNLOADS

Raspbian is the Foundation's official supported Operating System. Download it here, or use **NOOBS**, our easy installer for Raspbian and more.

RASPBERRY PI DESKTOP (FOR PC AND MAC)

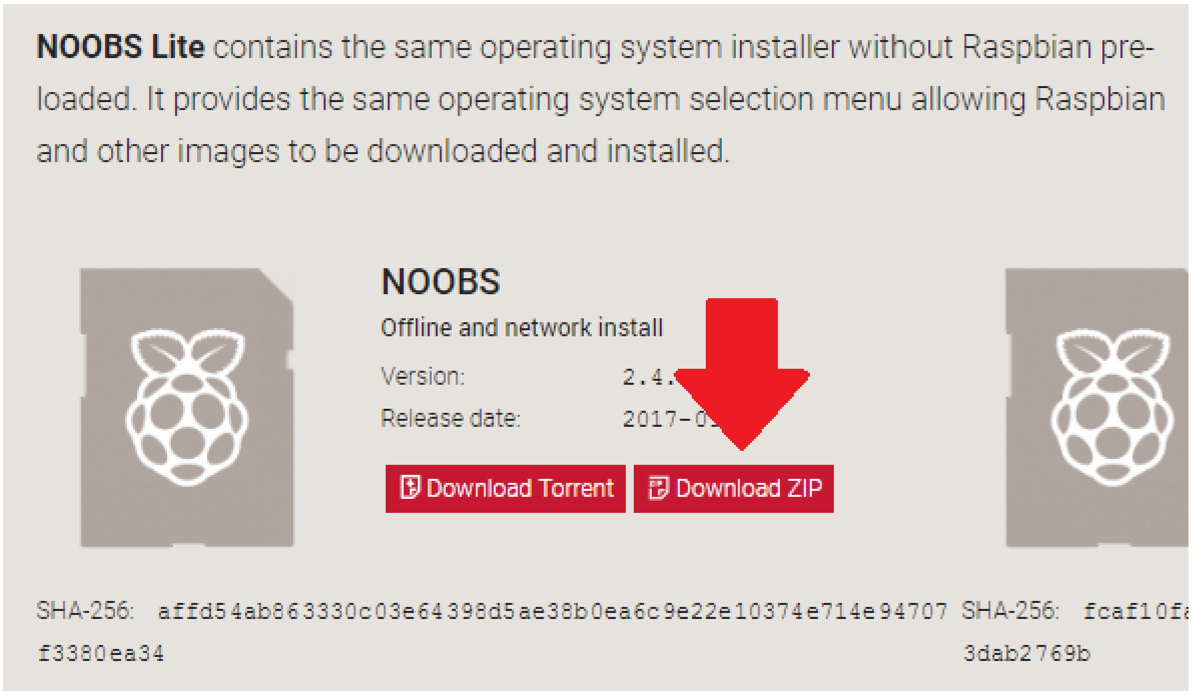
Debian with Raspberry Pi Desktop is the Foundation's operating system for PC and Mac. You can create a live disc, run it in a virtual machine, or even install it on your computer.

- You should see a box with a link to the NOOBS files. Click on the link.



Raspbian is the Foundation's official supported Operating System. Download it here, or use **NOOBS**, our easy installer for Raspbian and more.

- The simplest option is to download the zip archive of the files.



NOOBS Lite contains the same operating system installer without Raspbian pre-loaded. It provides the same operating system selection menu allowing Raspbian and other images to be downloaded and installed.

NOOBS
Offline and network install
Version: 2.4.
Release date: 2017-0

SHA-256: affd54ab863330c03e64398d5ae38b0ea6c9e22e10374e714e94707f3380ea34
SHA-256: fcaf10f
3dab2769b

Formatting the SD Card

If the SD card on which you wish to install Raspbian currently has an older version of Raspbian on it, you may wish to back up the files from the card first, as they will be overwritten during this process.

- Visit the SD Association’s website and download SD Formatter 4.0 for Windows or Mac.
- Follow the instructions to install the software.
- Insert your SD card into the computer or laptop’s SD card reader and make a note of the drive letter allocated to it, e.g. F:/.
- In SD Formatter, select the drive letter for your SD card, and format it.

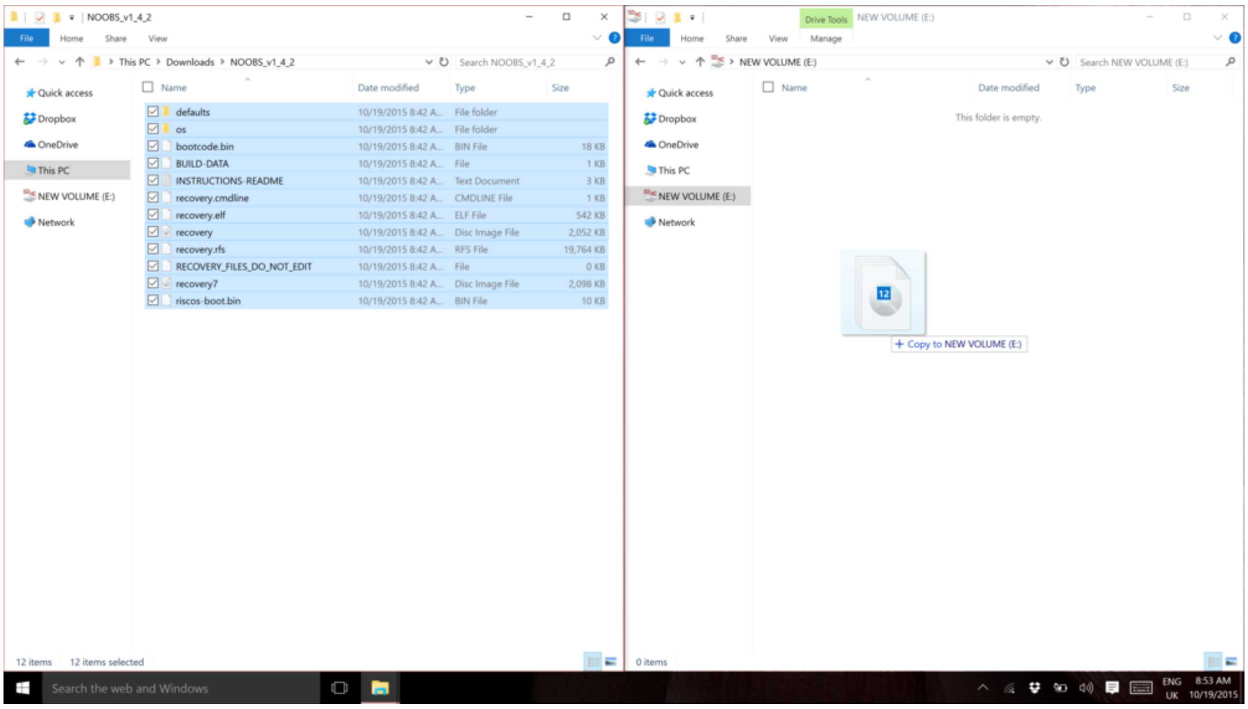
Extracting NOOBS from the zip archive

Next, you will need to extract the files from the NOOBS zip archive you downloaded from the Raspberry Pi website.

- Go to your *Downloads* folder and find the zip file you downloaded.
- Extract the files and keep the resulting Explorer/Finder window open.

Copying the files

- Now open another Explorer/Finder window and navigate to the SD card. It’s best to position the two windows side by side.
- Select all the files from the *NOOBS* folder and drag them onto the SD card.



Name	Date modified	Type	Size
defaults	10/19/2015 8:42 A...	File folder	
os	10/19/2015 8:42 A...	File folder	
bootcode.bin	10/19/2015 8:42 A...	BIN File	18 KB
BUILD-DATA	10/19/2015 8:42 A...	File	1 KB
INSTRUCTIONS-README	10/19/2015 8:42 A...	Text Document	3 KB
recovery.cmdline	10/19/2015 8:42 A...	CMDLINE File	1 KB
recovery.elf	10/19/2015 8:42 A...	ELF File	542 KB
recovery	10/19/2015 8:42 A...	Disc Image File	2,052 KB
recovery.rfs	10/19/2015 8:42 A...	RFS File	19,764 KB
RECOVERY_FILES_DO_NOT_EDIT	10/19/2015 8:42 A...	File	0 KB
recovery7	10/19/2015 8:42 A...	Disc Image File	2,098 KB
riscos-boot.bin	10/19/2015 8:42 A...	BIN File	10 KB

12 items 12 items selected

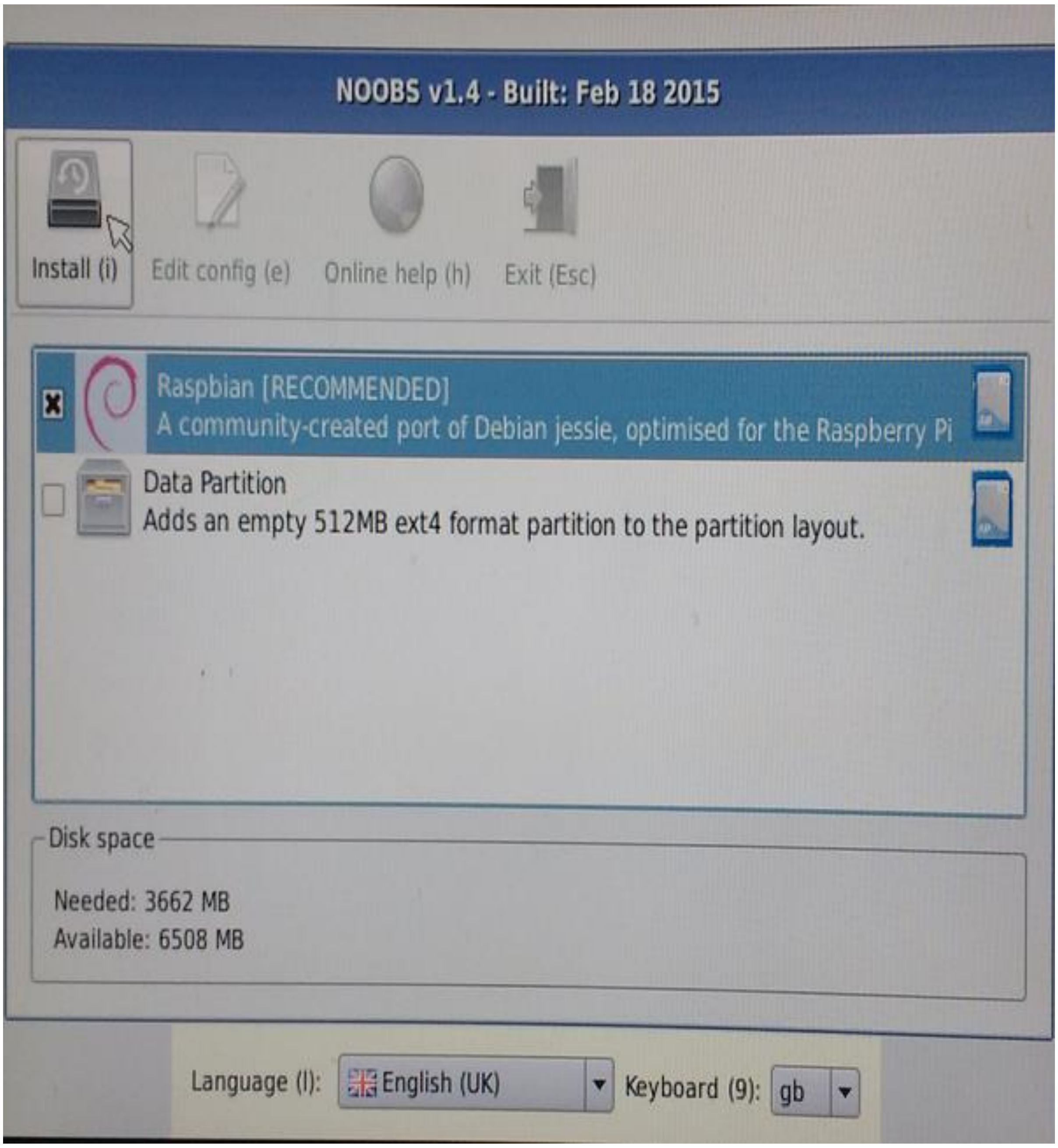
This folder is empty.

0 items

- Eject the SD card.

Booting from NOOBS

- Once the files have been copied over, insert the micro SD Card into your Raspberry Pi, and plug the Pi into a power source.
- You will be offered a choice when the installer has loaded. You should check the box for Raspbian, and then click Install.



NOOBS v1.4 - Built: Feb 18 2015

[✓] Raspbian [RECOMMENDED]
A community-created port of Debian jessie, optimised for the Raspberry Pi

[ ] Data Partition
Adds an empty 512MB ext4 format partition to the partition layout.

Disk space
Needed: 3662 MB
Available: 6508 MB

Language (l): English (UK) Keyboard (9): gb

- Click Yes at the warning dialog, and then sit back and relax. It will take a while, but Raspbian will install.

3.1.2 INSTALLING PYTHON:

Raspbian comes with a preinstalled Python. But if it not present, follow the below.

INSTALLING PYTHON PACKAGES

APT

Some Python packages can be found in the Raspbian archives and can be installed using APT. For example:

```
sudo apt-get update

sudo apt-get install python3-picamera
```

This is the preferred method of installing software, as it means that the modules you install can be kept up to date easily with the usual sudo apt-get update and sudo apt-get upgrade commands.

Python packages in Raspbian which are compatible with Python 2.x will always have a python- prefix. So, the picamera package for Python 2.x is named python-picamera (as shown in the example above). Python 3 packages always have a python3- prefix. So, to install picamera for Python 3 you would use:

```
sudo apt-get install python3-picamera
```

Uninstalling packages installed via APT can be accomplished as follows:

```
sudo apt-get remove python3-picamera
```

They can be completely removed with purge:

```
sudo apt-get purge python3-picamera
```

pip

Not all Python packages are available in the Raspbian archives, and those that are can sometimes be out-of-date. If you can't find a suitable version in the Raspbian archives, you can install packages from the Python Package Index (PyPI). To do so, use the pip tool.

pip is installed by default in Raspbian Jessie (but not Raspbian Wheezy or Jessie Lite). You can install it with apt:

```
sudo apt-get install python3-pip
```

To get the Python 2 version:

```
sudo apt-get install python-pip
```

pip3 installs modules for Python 3, and pip installs modules for Python 2.

For example, the following command installs the Unicorn HAT library for Python 3:

```
pip3 install unicornhat
```

The following command installs the Unicorn HAT library for Python 2:

```
pip install unicornhat
```

3.1.3 INSTALL FLASK:

Use the following command to install Flask:

```
pip install Flask
```

3.1.4 INSTALL NGINX:

First install the nginx package by typing the following command in to the Terminal:

```
sudo apt-get install nginx
```

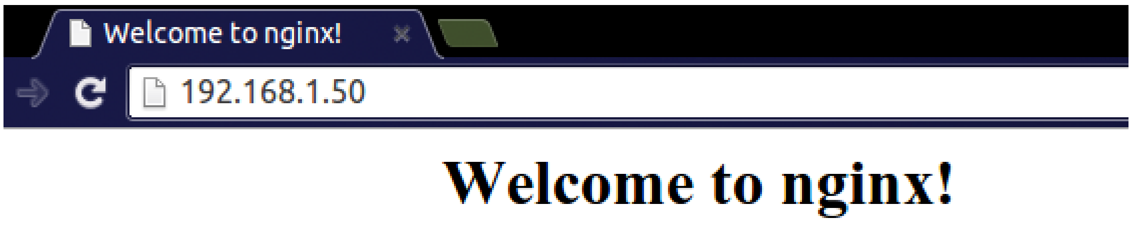
and start the server with:

```
sudo /etc/init.d/nginx start
```

Test the web server

By default, NGINX puts a test HTML file in the web folder. This default web page is served when you browse to http://localhost/ on the Pi itself, or http://192.168.1.10 (whatever the Pi's IP address is) from another computer on the network. To find the Pi's IP address, type hostname -I at the command line.

Browse to the default web page either on the Pi or from another computer on the network and you should see the following:


Welcome to nginx!

192.168.1.50

Welcome to nginx!

And finally, dump all the project source code into Raspbian Home page.

3.2 MODIFICATIONS

To Work with Serial Ports:

There is yet another wrinkle in that in the latest Jessie releases (as of May 2016) the GPIO serial port is disabled by default. In order to enable it, edit config.txt:

```
$ sudo nano /boot/config.txt
```

and add the line (at the bottom):

```
enable_uart=1
```

As of May 2016 this will also lock the cpu core frequency for you so there's nothing else you need to do (If you aren't convinced and you really like to belt and braces it the command is: core_freq=250 which you add to the same file as well).

Reboot for the changes to take effect.

This should get you good serial communications for most uses.

Serial Aliases

On the Raspberry Pi 3 the second serial port is called /dev/ttyS0 and is by default mapped to the GPIO pins 14 and 15. So immediately, if you have code that references /dev/ttyAMA0 you're going to have problems and things aren't going to work.

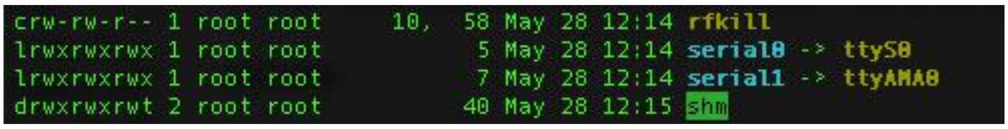
You could go through your code and replace ttyAMA0 with ttyS0 and that should work. However, if you find yourself use the same SD card on a Raspberry Pi other than a rpi3 your code won't work again.

In order to try and get around this the Foundation have introduced a serial port alias (as of May 2016 – 2016-05-10). Thus you have serial ports: serial0 and serial1 (rpi3). The Raspberry Pi kernel sorts out where these point to depending on which Raspberry Pi you are on. Thus on a

Raspberry Pi 3 serial0 will point to GPIO pins 14 and 15 and use the “mini-uart” aka /dev/ttyS0. On other Raspberry Pi’s it will point to the hardware UART and /dev/ttyAMA0.

To find out where it is pointing you can use the command:

```
ls -l /dev
```



```
crw-rw-r-- 1 root root     10,  58 May 28 12:14 rfkill
lrwxrwxrwx 1 root root           5 May 28 12:14 serial0 -> ttyS0
lrwxrwxrwx 1 root root           7 May 28 12:14 serial1 -> ttyAMA0
drwxrwxrwt 2 root root          40 May 28 12:15 shm
```

Default Raspberry PI 3 serial port aliases

So where possible refer to the serial port via it’s alias of “serial0” and your code should work on both Raspberry Pi 3 and other Raspberry Pi’s.

Disabling the Console

If you are using the serial port for anything other than the console you need to disable it. This will be slightly different depending on whether you are running a Raspberry Pi 3 or not.

For Raspberry Pi 3’s the command is /dev/ttyS0:

```
sudo systemctl stop serial-getty@ttyS0.service

sudo systemctl disable serial-getty@ttyS0.service
```

You also need to remove the console from the cmdline.txt. If you edit this with:

```
sudo nano /boot/cmdline.txt
```

You will see something like:

```
dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes root wait
```

remove the line: console=serial0,115200 and save and reboot for changes to take effect.

Swapping the Serial Ports on Raspberry Pi 3

What if you don’t want to use the Bluetooth and you want that high performance /dev/ttyAMA0 back on the GPIO? Well you can do this and the way you do this is via a device overlay called “pi3-miniuart-bt” i.e. use the mini-uart (/dev/ttyS0) for Bluetooth (you may get some loss of performance on your Bluetooth though).

You can also just disable the Bluetooth all together by using another overlay “pi3-disable-bt”. In both cases if you can find out more of what they do here: /boot/overlays/README

To use add the following line to the /boot/config.txt

```
sudo nano /boot/config.txt
```

and add:

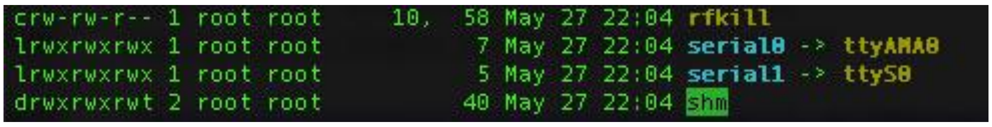
```
dtoverlay=pi3-miniuart-bt
```

Save and reboot for changes to take effect.

You can check that it has worked by:

```
ls -l /dev
```

and you'll see something like this:



```
crw-rw-r-- 1 root root     10,  58 May 27 22:04 rfkill
lrwxrwxrwx 1 root root           7 May 27 22:04 serial0 -> ttyAMA0
lrwxrwxrwx 1 root root           5 May 27 22:04 serial1 -> ttyS0
drwxrwxrwt 2 root root          40 May 27 22:04 shm
```

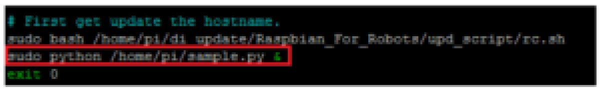
Swapped Raspberry PI 3 serial port aliases

Run a Program At Startup:

On your Pi, edit the file /etc/rc.local using the editor of your choice. You must edit it with root permissions.

```
sudo nano /etc/rc.local
```

Add commands to execute the python program, preferably using absolute referencing of the file location (complete file path are preferred). Be sure to leave the line exit 0 at the end, then save the file and exit. In nano, to exit, type Ctrl-x, and then Y.



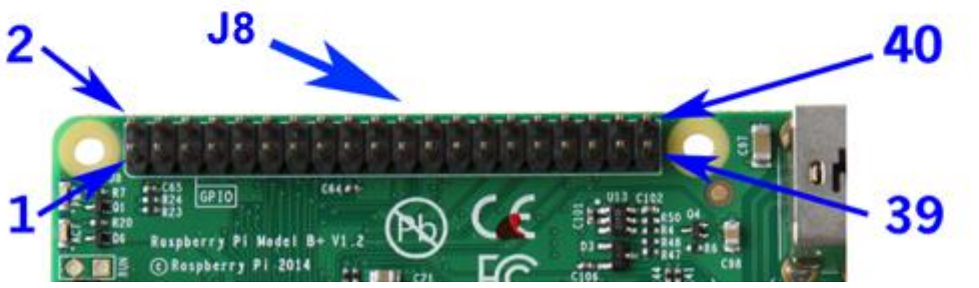
```
# First get update the hostname.
sudo bash /home/pi/di_update/Raspbian_For_Robots/upd_script/rc.sh
sudo python /home/pi/sample.py &
exit 0
```

If your program runs continuously (runs an infinite loop) or is likely not to exit, you must be sure to fork the process by adding an ampersand ("&") to the end of the command, like: sudo python /home/pi/sample.py &

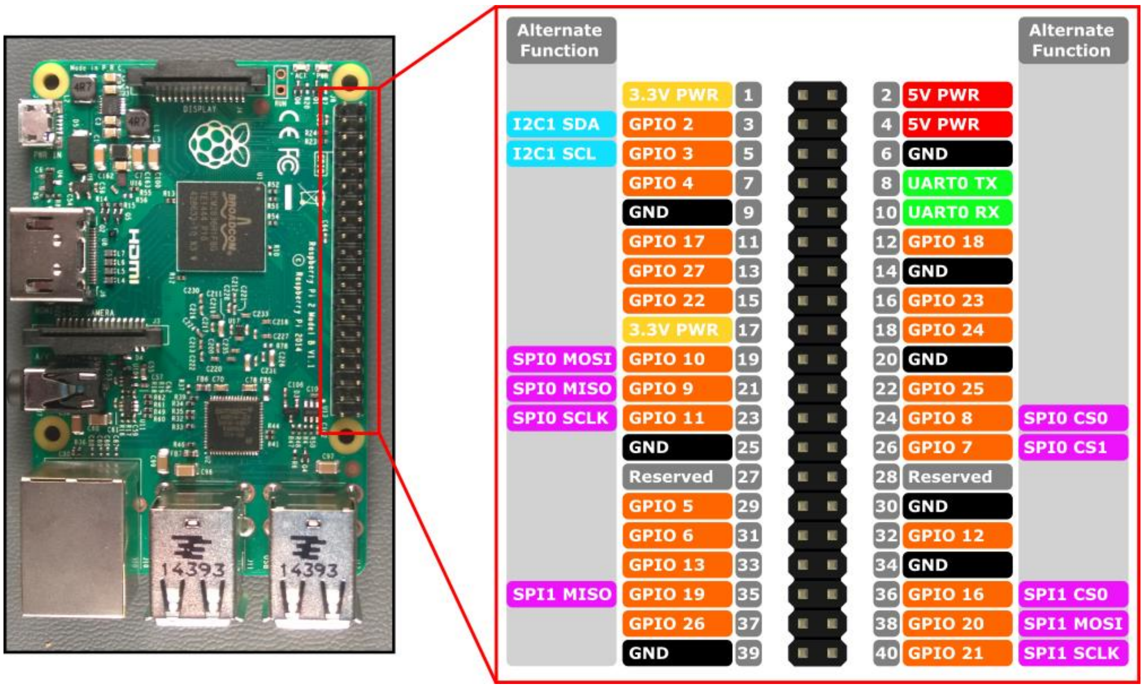
The Pi will run this program at bootup, and befor e other services are started. If you don't include the ampersand and if your program runs continuously, the Pi will not complete its boot process. The ampersand allows the command to run in a separate process and continue booting with the main process running. Now reboot the Pi to test it.

3.3 Hardware Interface

The Raspberry Pi 3 Model B board contains a single 40-pin expansion header labeled as 'J8' providing access to 28 GPIO pins. (Pins 1, 2, 39 & 40 are also labeled below.)

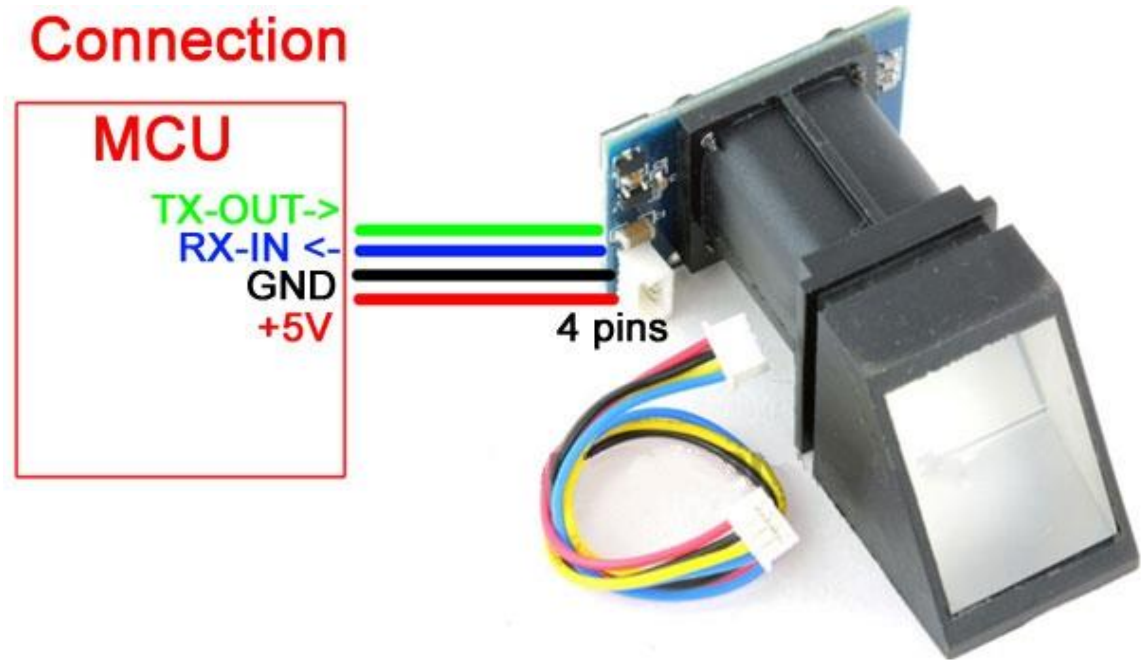


The diagram below illustrates the GPIO pinout using the Pi4J/WiringPi GPIO numbering scheme.



Alternate Function					Alternate Function
	3.3V PWR	1	2	5V PWR	
I2C1 SDA	GPIO 2	3	4	5V PWR	
I2C1 SCL	GPIO 3	5	6	GND	
	GPIO 4	7	8	UART0 TX	
	GND	9	10	UART0 RX	
	GPIO 17	11	12	GPIO 18	
	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
	3.3V PWR	17	18	GPIO 24	
SPI0 MOSI	GPIO 10	19	20	GND	
SPI0 MISO	GPIO 9	21	22	GPIO 25	
SPI0 SCLK	GPIO 11	23	24	GPIO 8	SPI0 CS0
	GND	25	26	GPIO 7	SPI0 CS1
	Reserved	27	28	Reserved	
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
SPI1 MISO	GPIO 19	35	36	GPIO 16	SPI1 CS0
	GPIO 26	37	38	GPIO 20	SPI1 MOSI
	GND	39	40	GPIO 21	SPI1 SCLK

3.3.1 Connecting with Fingerprint Scanner:



Name	Type	Function Description
+5V	in	Power input
GND	•	Signal ground. Connected to power ground (color: black
Tx	in	Data Output. TTL logic level
Rx	Out	Data input. TTL logic level

3.3.2 Connecting An I2c Enabled LCD:

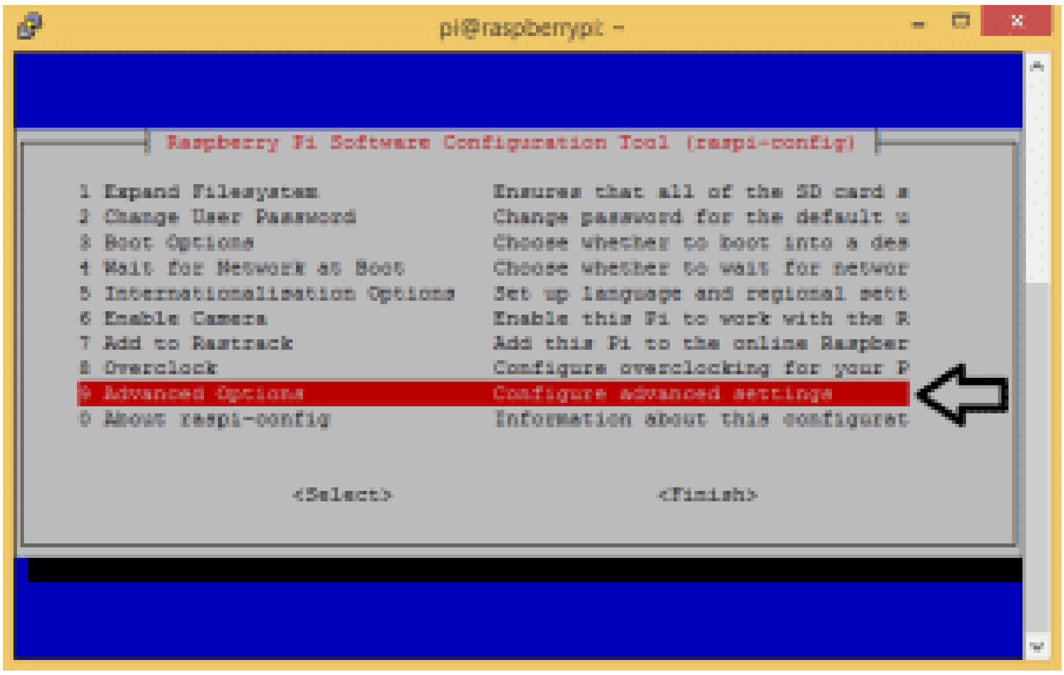
Connecting an LCD with an I2C backpack is pretty self-explanatory. Connect the SDA pin on the Pi to the SDA pin on the LCD, and the SCL pin on the Pi to the SCL pin on the LCD. The ground and Vcc pins will also need to be connected. Most LCDs can operate with 3.3V, but they're meant to be run on 5V, so connect it to the 5V pin of the Pi if possible.

ENABLE I2C ON THE PI

Before we get into the programming, we need to make sure the I2C module is enabled on the Pi and install a couple tools that will make it easier to use I2C.

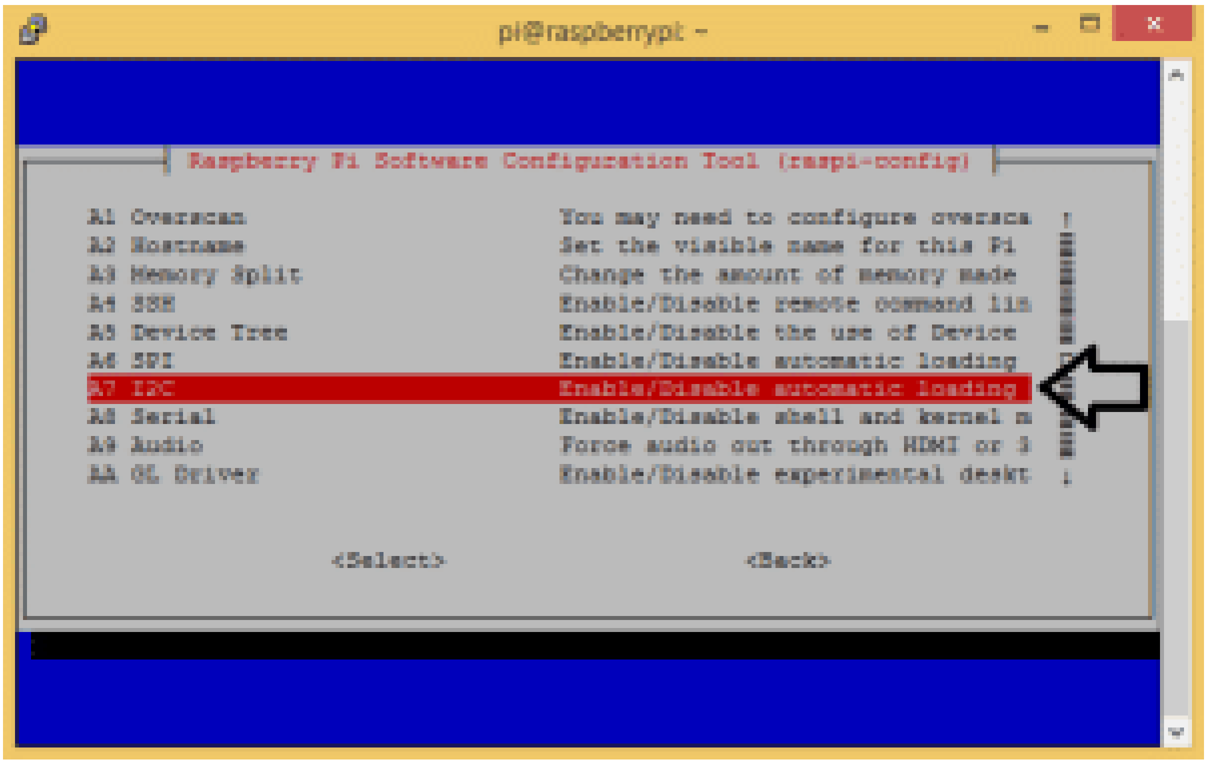
ENABLE I2C IN RASPI-CONFIG

First, log in to your Pi and enter sudo raspi-config to access the configuration menu. Then arrow down and select "Advanced Settings":



```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem              Ensures that all of the SD card s
2 Change User Password           Change password for the default u
3 Boot Options                   Choose whether to boot into a des
4 Wait for Network at Boot       Choose whether to wait for networ
5 Internationalisation Options   Set up language and regional sett
6 Enable Camera                  Enable this Pi to work with the R
7 Add to Rastrack                Add this Pi to the online Raspber
8 Overclock                      Configure overclocking for your P
9 Advanced Options               Configure advanced settings
0 About raspi-config             Information about this configurat

            <Select>                      <Finish>
```

Now arrow down and select "I2C Enable/Disable automatic loading":

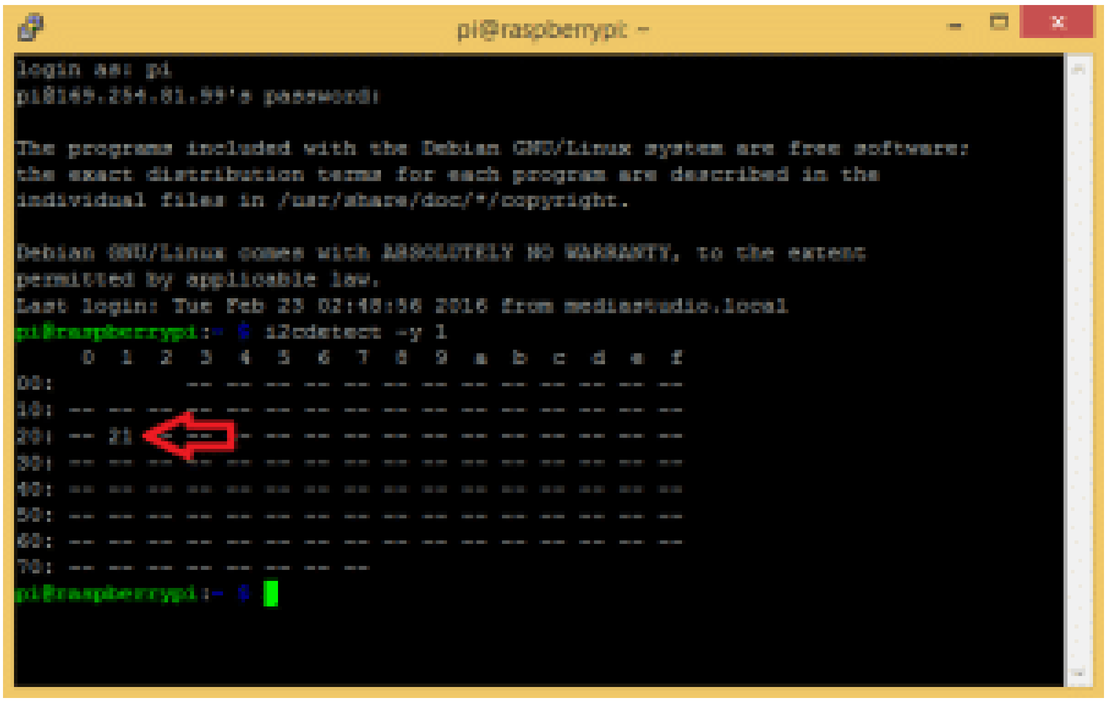


```
pi@raspberrypi: ~

      Raspberry Pi Software Configuration Tool (raspi-config)

  A1 Overscan                        You may need to configure oversca
  A2 Hostname                        Set the visible name for this Pi
  A3 Memory Split                    Change the amount of memory made
  A4 SSH                             Enable/Disable remote command lin
  A5 Device Tree                     Enable/Disable the use of Device
  A6 SPI                             Enable/Disable automatic loading
  A7 I2C                             Enable/Disable automatic loading
  A8 Serial                          Enable/Disable shell and kernel m
  A9 Audio                           Force audio out through HDMI or 3
  AA GL Driver                       Enable/Disable experimental deskt

                   <Select>                      <Back>
```

Choose “Yes” at the next prompt, exit the configuration menu, and reboot the Pi to activate the settings.

INSTALL I2C-TOOLS AND SMBUS

Now we need to install a program called I2C-tools, which will tell us the I2C address of the LCD when it’s connected to the Pi. So at the command prompt, enter sudo apt-get install i2c-tools.
Next we need to install SMBUS, which gives the Python library we’re going to use access to the I2C bus on the Pi. At the command prompt, enter sudo apt-get install python-smbus.
Now reboot the Pi and log in again. With your LCD connected, enter i2cdetect -y 1 at the command prompt. This will show you a table of addresses for each I2C device connected to your Pi:



```
login as: pi
pi@169.254.81.55's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Feb 23 02:48:56 2016 from mediastudio.local
pi@raspberrypi:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- 21 -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

The I2C address of my LCD is 21. Take note of this number, we'll need it later.

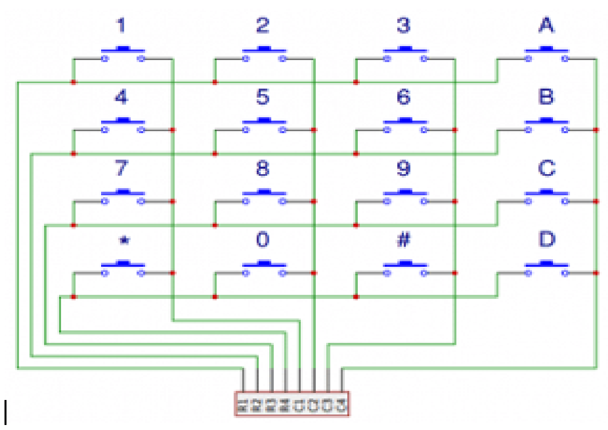
3.3.2 Connect 4x4 Keypad to Raspberry pi 3:

There are many libraries which are available to interface a keypad with a RasberryPi, I chose this one pad4pi was very handy and easy to start.

Installation of pad4pi

```
pip install pad4pi
```

Pin Mapping:



Key pad = Rpi pin
Col 1 = GPIO 6
Col 2 = GPIO 13
Col 3 = GPIO 19
Col 4 = GPIO 26
Row 1 = GPIO 16
Row 2 = GPIO 20
Row 3 = GPIO 21
Row 4 = GPIO 5

Chapter 4

SYSTEM TESTING

4.1 INTRODUCTION

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

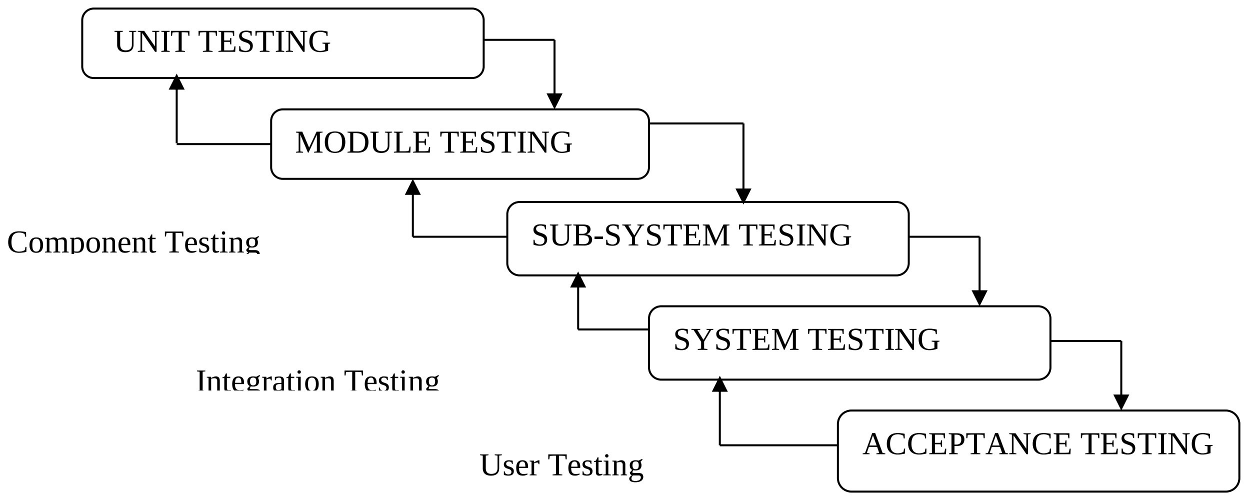


Fig: 4.1.1 Various Phases in Testing

4.2 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

4.2.1 WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

4.2.2 BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

V(G)=E-N+2 or

V(G)=P+1 or

V(G)=Number Of Regions

Where V(G) is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

4.2.3 CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

4.2.4 DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

4.2.5 LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops. All the loops were tested at their limits, just above them and just below them. All the loops were skipped at least once. For nested loops test the inner most loop first and then work outwards. For concatenated loops the values of dependent loops were set with the help of connected loop. Unstructured loops were resolved into nested loops or concatenated loops and tested as above. Each unit has been separately tested by the development team itself and all the input have been validated.

4.3 DESIGN OF THE TESTCASES

Test Cases: Login Form

Test Case-1	
Field	Username
Validation	Is Null
Error Message	User Name must be entered

Test Case-2	
Field	Password
Validation	Is Null
Error Message	Password must be entered

Test Case-3	
Field	Username/Password
Validation	Is not exist
Error Message	Invalid User

Test Cases: Add Student:

Test Case-1	
Field	First Name
Validation	Is Null
Error Message	First Name must be entered

Test Case-2	
Field	Last Name
Validation	Is Null
Error Message	Last Name must be entered

Test Case-3	
Field	Email
Validation	Is Null
Error Message	Email must be entered

Test Case-4	
Field	Roll Number
Validation	Is Null
Error Message	Roll number must be entered

Test Case-5	
Field	Mobile
Validation	Is Null
Error Message	MobileNo must be entered

Test Case-6	
Field	Department
Validation	Is Null
Error Message	Dept Must be selected

Chapter 5

RESULTS

The results of our system are explained below using screen shots

5.1 Login page for all users:

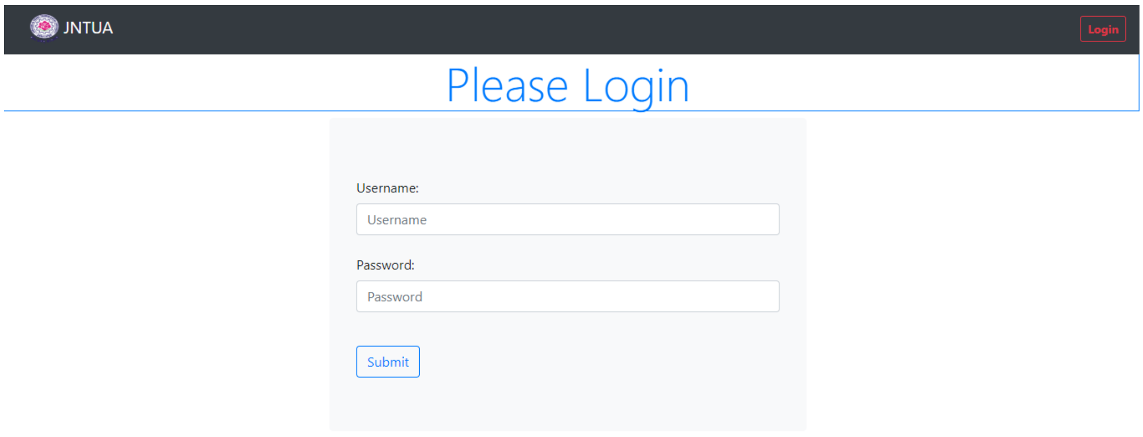
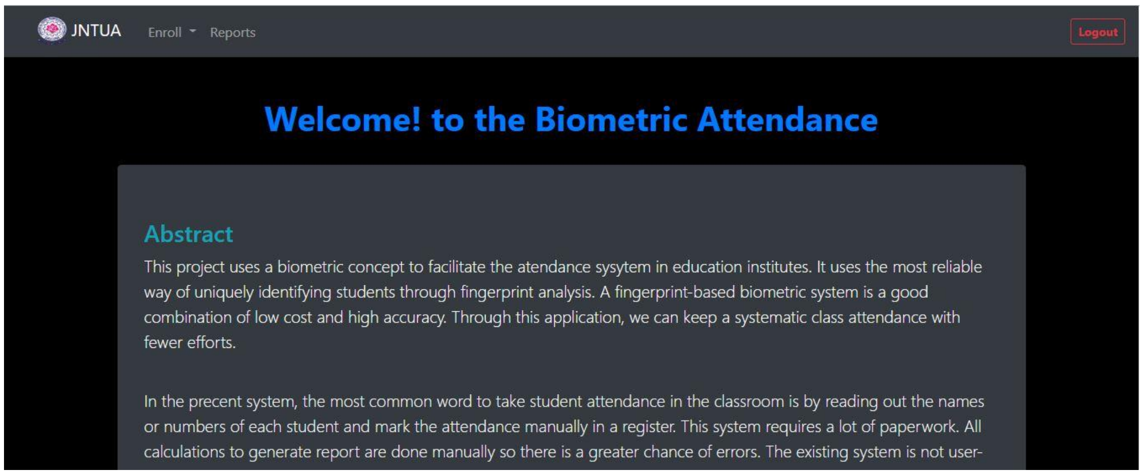


Fig: 5.1 Login Page

5.2 Home page:

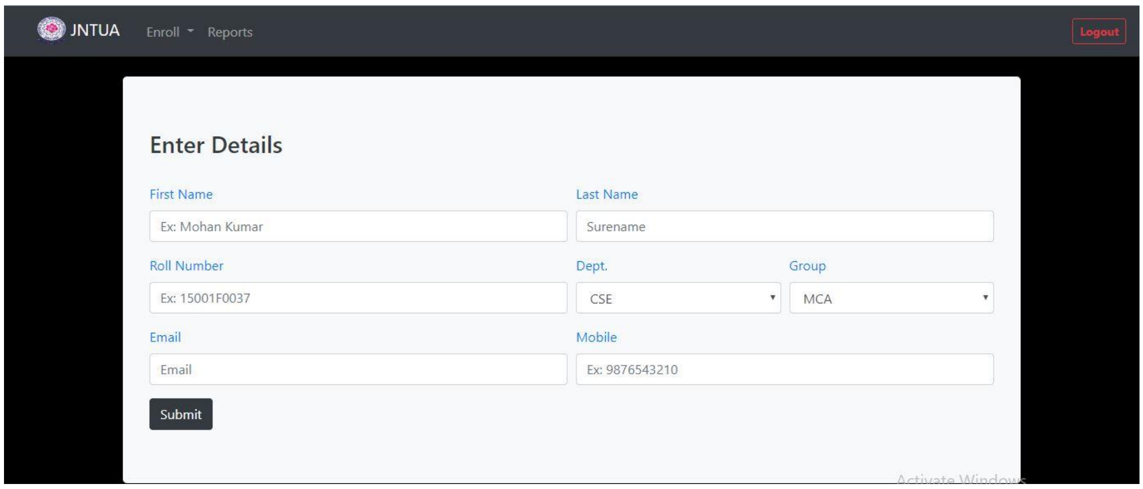


This project uses a biometric concept to facilitate the atendance sysytem in education institutes. It uses the most reliable way of uniquely identifying students through fingerprint analysis. A fingerprint-based biometric system is a good combination of low cost and high accuracy. Through this application, we can keep a systematic class attendance with fewer efforts.

In the precent system, the most common word to take student attendance in the classroom is by reading out the names or numbers of each student and mark the attendance manually in a register. This system requires a lot of paperwork. All calculations to generate report are done manually so there is a greater chance of errors. The existing system is not user-

Fig: 5.2 Project Home Page

5.2.1 Enroll New Student

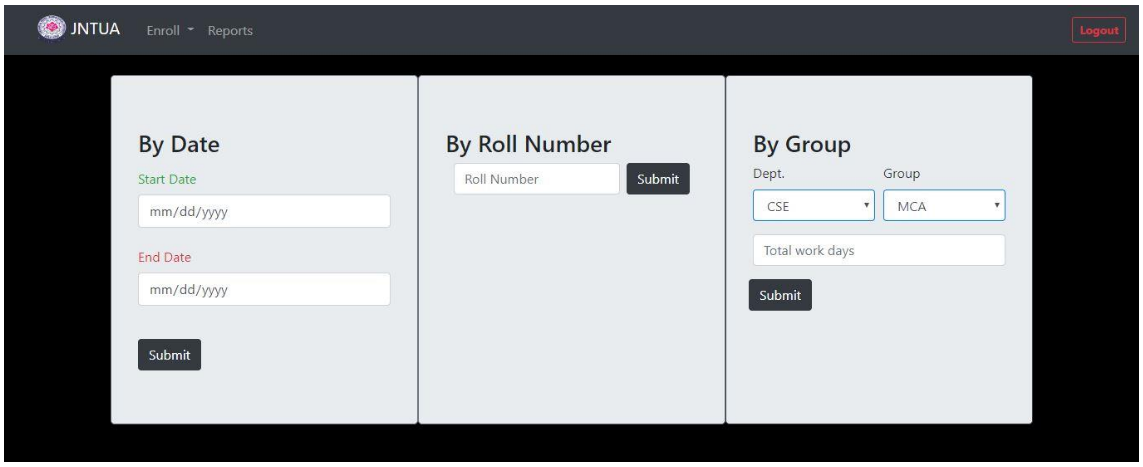


Enter Details

First Name: Ex: Mohan Kumar
Last Name: Surename
Roll Number: Ex: 15001F0037
Dept.: CSE
Group: MCA
Email: Email
Mobile: Ex: 9876543210

Fig: 5.2.1 Enroll Student

5.2.2 Reports Page:



By Date

Start Date: mm/dd/yyyy
End Date: mm/dd/yyyy

By Roll Number

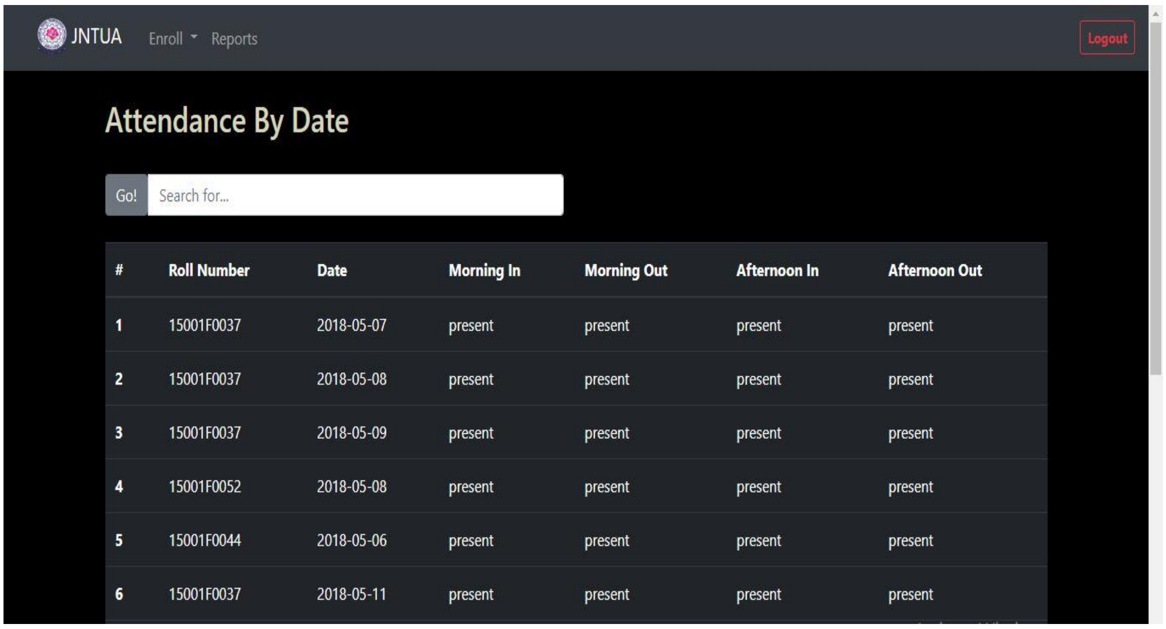
Roll Number

By Group

Dept.: CSE
Group: MCA
Total work days

Fig: 5.2.2 Types of Reports

5.2.3 Generate Reports by Date:

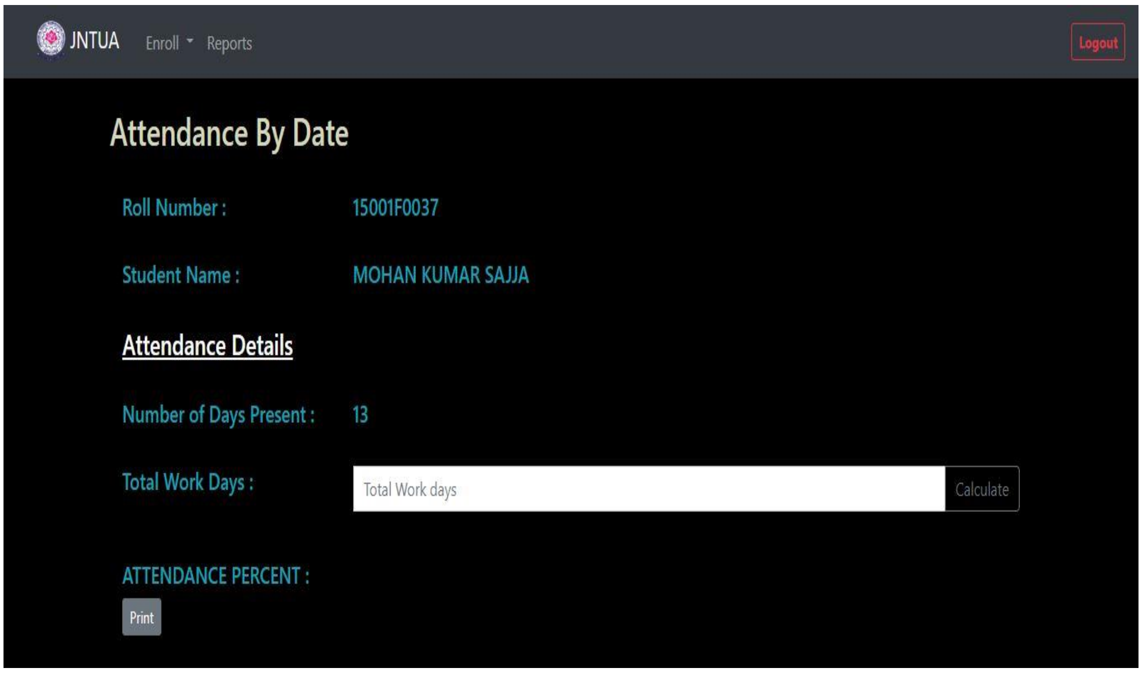


Attendance By Date

#	Roll Number	Date	Morning In	Morning Out	Afternoon In	Afternoon Out
1	15001F0037	2018-05-07	present	present	present	present
2	15001F0037	2018-05-08	present	present	present	present
3	15001F0037	2018-05-09	present	present	present	present
4	15001F0052	2018-05-08	present	present	present	present
5	15001F0044	2018-05-06	present	present	present	present
6	15001F0037	2018-05-11	present	present	present	present

Fig: 5.2.3 Attendance reports by date

5.2.4 Generate Reports by Roll Number:



Attendance By Date

Roll Number : 15001F0037

Student Name : MOHAN KUMAR SAJJA

Attendance Details

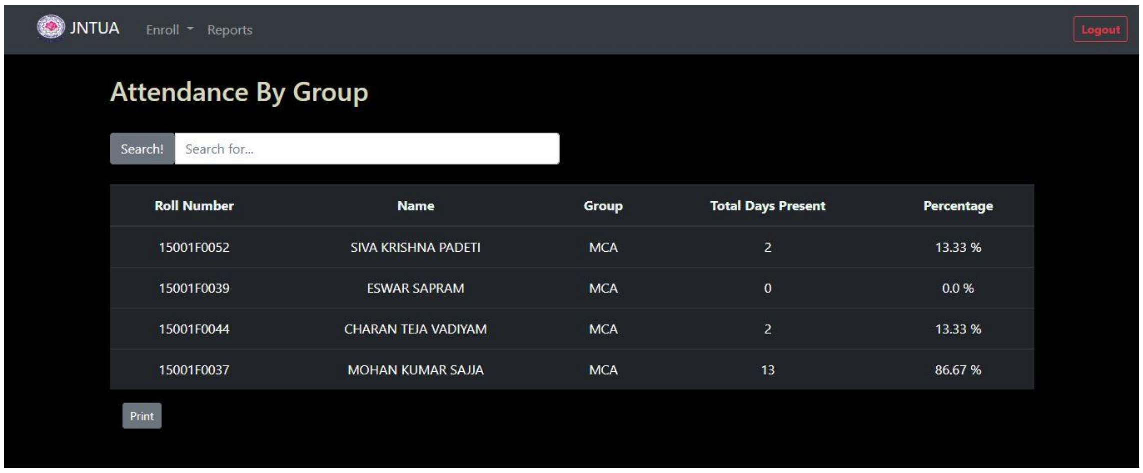
Number of Days Present : 13

Total Work Days :

ATTENDANCE PERCENT :

Fig: 5.2.4 Attendance reports by ID

5.2.5 Generate Reports by Branch:



Attendance By Group

Roll Number	Name	Group	Total Days Present	Percentage
15001F0052	SIVA KRISHNA PADETI	MCA	2	13.33 %
15001F0039	ESWAR SAPRAM	MCA	0	0.0 %
15001F0044	CHARAN TEJA VADIYAM	MCA	2	13.33 %
15001F0037	MOHAN KUMAR SAJJA	MCA	13	86.67 %

Fig: 5.2.5 Attendance reports by branch/group

Chapter 6
CONCLUSION

Regularity of student attendance is a key concern in educational institutions. Their overall academic performance depends on attendance to a great extent. This system is fully automated technology, highly flexible and reliable to use. It gives accurate results and reduces paper based work, thus reducing environmental impact

The developed system is very helpful in saving valuable time of students and lecturers, by generating a report at required time. Also, it reduces most of the administrative jobs and minimizes human errors, avoids proxy punching, eliminates time-related disputes and helps to update and maintain attendance records.

The IoT based Student Biometric Attendance using Fingerprint is developed using Raspberry Pi3, Python and Flask that fully meets the objectives of the system which it has been developed. The system has reached a steady state where all bugs have been eliminated. The system is operated at a high level of efficiency and all the teachers and user associated with the system understands its advantage. It was intended to solve the requirement specification.

Scope:

- This project can be extended to college staff also.
- We can send e-mails to the students after attendance is taken.
- Delete fingerprints and details of students need to be automated.
- By using Camera, we can also store guest’s fingerprints and their face images.

References:

1. Le Hoang Thai and Ha Nhat Tam [2010]," Fingerprint recognition using Standardized M fingerprint model" IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, PP 1116.

2. Mukesh Kumar Thakur, Ravi Shankar Kumar, Mohit Kumar, Raju Kumar [2013],"Wireless Fingerprint Based Security System Using Zigbee Technology" International Journal of Inventive Engineering and Sciences (IJIES),ISSN:2319– 9598, Volume-1, Issue-5, PP 14-17.

3. Arun, Emmanuel, Diwakar & Rajeswari Automated attendance system using biometrics with Embedded web server‖ Graduate Research in Engineering and Technology (GRET): An International Journal, page No-(54to57).

4. Karthik Vignesh E, Shanmuganathan S, A.Sumithra S.Kishore and P. Karthikeyan ―A Foolproof Biometric Attendance Management System‖ International Journal of Information and ComputationTechnology.ISSN:0974-2239.