INTELLIGENT RFID SYSTEM FOR MAINTENANCE MANAGEMENT

MIS 6308 Project Report

GROUP – 12

Ayush Baldwa
Richa Pandey
Khushi Shah
Sushmitha Madiraju
Taufeeq Samnani

Contents

1. Executive Summary

With the growth of industries and production in recent years, there has been a need to maintain and routinely monitor the machinery and heavy instruments in use. This is critical for the workers' safety as well as to avoid a large loss. Ignorance of the maintenance cycle may also result in licence cancellation or a heavy fee, which will not only cause major trust issues but also result in the loss of employment for personnel in that field. To solve this challenge, we devised a novel technique that involves RFID tags. For this, we used a Raspberry Pi and an Android application to build two systems. It will begin by launching the program, which will allow only authorised employees to log in with their login Id and password. The Raspberry Pi unit/Android application will scan RFID tags attached to machines and equipment, and the information will be stored on a server.

Following that, the employee will be directed to a website with numerous actions to complete. The employee's modifications or updates will be reflected on the server, resulting in one service cycle. This service cycle is unique to each piece of equipment and can be tailored to the needs of the client. The final step is to log out of the software, after which the progress will be kept in a database and a report mailed to the designated in charge/manager. Only the appointed in charge has access to the system, which includes adding employees, deleting services, adding, or upgrading services, and so on.

We not only make the data flow more efficient, but we also make it more secure and dependable by applying these methods. This will reduce the number of mistakes made when registering for equipment. All these factors combine to make it a one-stop shop for all industrial needs.

The objective of the Project By 2050, the world's population is predicted to exceed ten billion people. This will put a great strain on manufacturing companies, increasing the risk to those who work there.

Furthermore, it is already challenging to locate skilled and motivated personnel, as well as to maintain manufacturing security. To fill this critical gap, we devised a system that tracks the present state of machines.

The old approach is unreliable and authentic, and many environmental conditions such as dampness, paper deterioration, record misplacement, and so on can put the industry in jeopardy.

This error can result in severe fines, license cancellation, industrial distrust, and mishaps with persons working there, among other things.

Most appliances have a manual servicing record, which is prone to human errors such as untimed servicing, irregularly managing service records, and so on, which can lead to disaster.

So, a smart system is necessary to manage both timely servicing and retaining records to avoid these types of errors.

2. Problem Statement

Mismanagement and a lack of concern for the upkeep of industrial equipment and heavy gear could result in significant property damage and perhaps the death of workers in that industry.

The project must have key features such as data retrieval, data storage, reminders for forthcoming maintenance, missed maintenance, and authorized management.

3. Scope

Business Process Scope

• The business scope of the project is to eliminate human errors by developing an intelligent system that allows organizations to address customer services immediately upon request to avoid huge losses and ensure the safety of workers.

• The project aims in saving cost and time by implementing service cycles in the most unique, efficient, and reliable way.

Organization Scope

• Most organizations use a lot of guesswork to predict machinery failures due to inaccurate resources. Automation offers solutions to existing problems and provides the tools needed to track every machine. This helps organizations predict future outcomes and helps improve forecasting.

• The project also helps organizations to use real-time data with minimal human error to eliminate inefficiencies and repurpose resources.

• In any industry, Automation can initiate sales, attract new customers, new opportunities, and can further generate more profits.

• Greater visibility in organizations can lead stakeholders to make better decisions based on accurate data.

Location Scope

• This project can be applied to various industries, such as manufacturing, industrial, and medical organizations where the age and efficiency of machinery and its parts is imperative to the success of the business and its customers.

Application Scope

• Each data table will include its own unique name and ID to prevent misuse and protect data integrity.

• Users will have full administrative access to change the identifies and names of items (machinery/parts) to customize to business needs. Updates will automatically be represented in the SQL database.

• There will be cross-checks to ensure integrity within the data (no duplicates, etc.)

Data Scope

• Implementation of an SQL database to collect historical data for future alerts, processes, and analysis of specific machinery and parts within the organization.

• The main objective for this data will be to read and analyze whether servicing is required.

Technology Scope

• RFID Tags - ISO 14443A standard tags

• RFID Tag Reader - RFID RC522

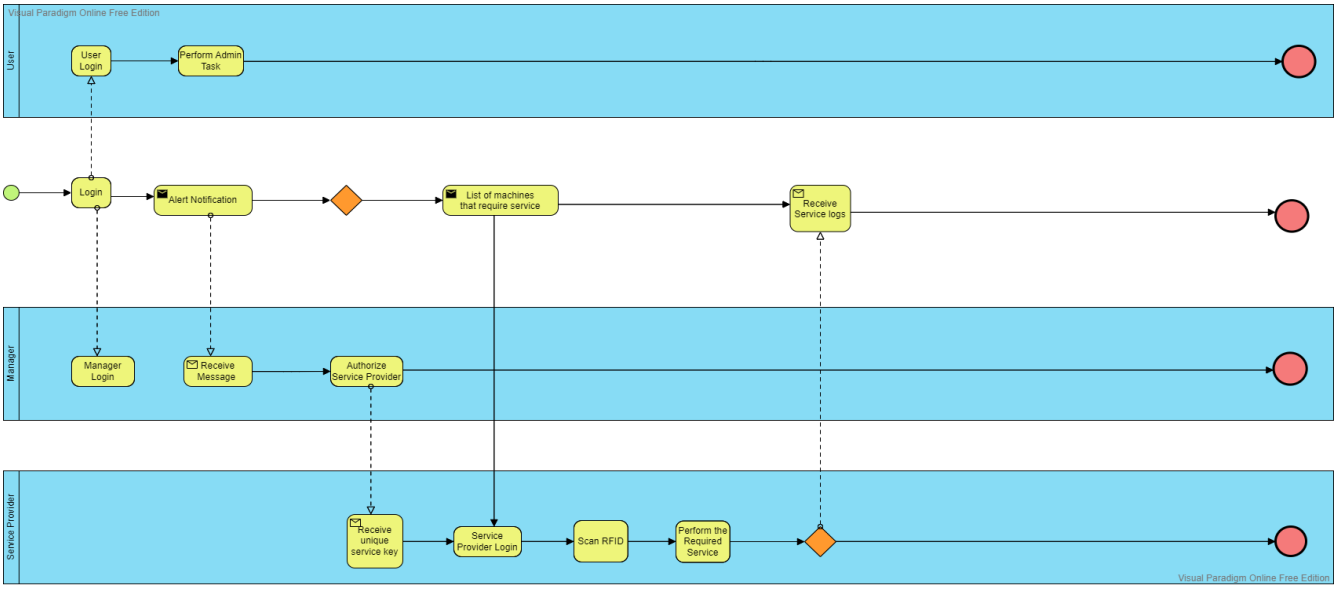
• Raspberry Pi - 3b+

• Android Phone for providing the front-end of the application -- This will provide a UI for Adding an employee, deleting a service, adding a service, updating a service, etc.
• Database System -- MySQL will be the main DBMS utilized.
• Python programming -- Should be easily usable and updatable to the newest version of Python v3.
• Android App Development -- Basic interface style in order to test processes. It will utilize native functions and libraries from Android Studio.

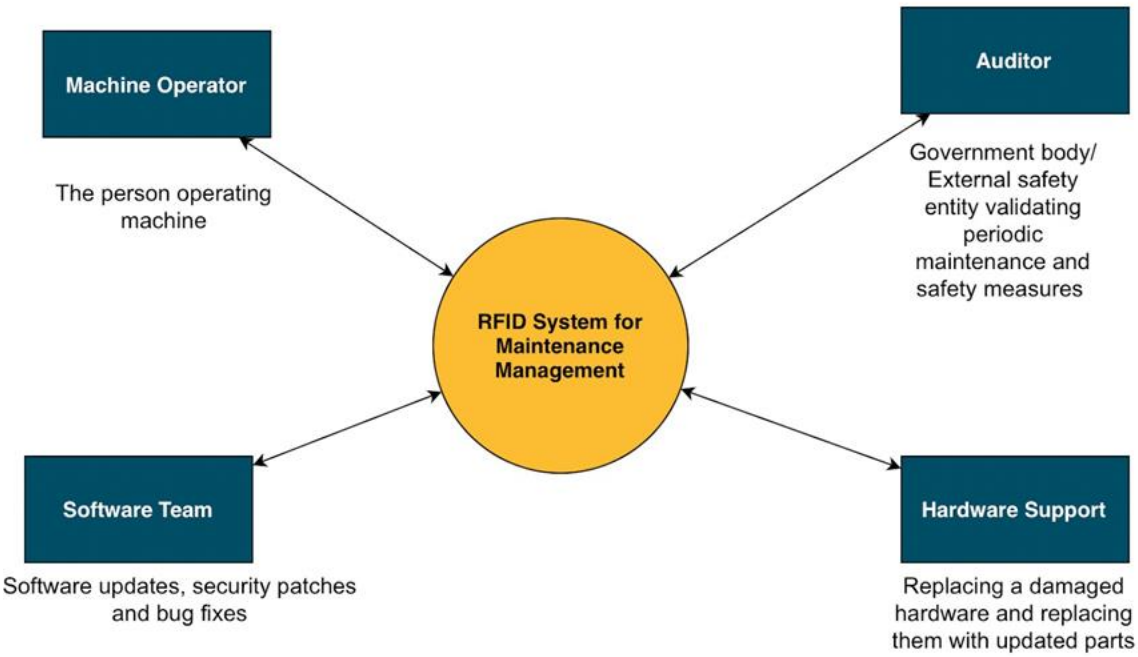
Out of Scope

• Implementation of a front-end framework for a seamless and user-friendly interface for the Android application (React, etc.)
• Trainings, tutorials, etc. will not be taught nor implemented at this stage for users.
• Manipulation of data past send and receive. For example, the ability to utilize API calls to send and set appointments.
• Authentication for security purposes will be considered later based on budget and user/audience. A simple SSO or two-factor authentication will most likely be implemented.
• Cross-platform with iOS is out of scope

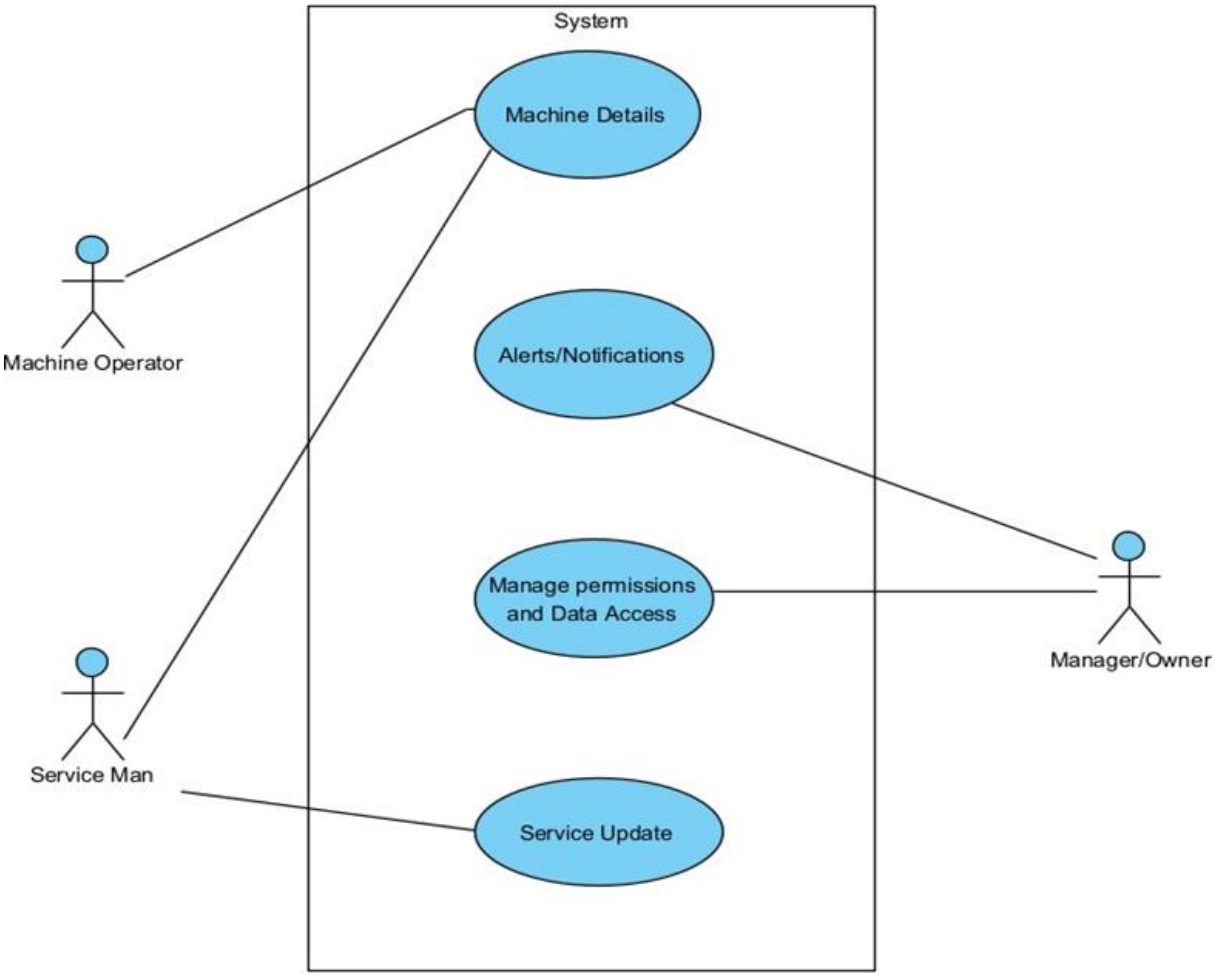
4. Business Process Model Notation



5. Context Diagram



6. Use Case Diagram



7. Use Case Descriptions

Use Case Name	Authorized User/Machine Operator Login
Primary Actor	Machine Operator
Stakeholder	System
Brief Description	When a user wants to login in the system to view information he can log in
Trigger	When the machine operator enters the credentials and clicks on login button
Normal Flow of Events	1. User clicks on Log-In button. 2. User enters the User ID and Password 3. Retrieve User Info from User database file 4. If credentials match, then authorize Log-In
Exception Flow	If user enters invalid details, then display “Login failed”

Use Case Name	Scan RFID Tags/QR
Primary Actor	System
Stakeholder	Equipment, RFID Tags
Brief Description	RFID tags/ OR code needs to be scanned when the machine needs to be operated
Trigger	When the scan button is clicked
Normal Flow of Events	1. User clicks on Scan. 2. Scans the QR Code or RFID tag attached to the machine 3. Retrieve User Info from User database file 4. If successful match, then authorize 5. Store the data log on the server
Exception Flow	When the system doesn’t recognize the QR Code or RFID tag then show “Unable to find the information”

Use Case Name	Alerts/Notifications
Primary Actor	System
Stakeholder	Manager
Brief Description	Manager receives the notification from system when a machine needs service
Trigger	When a machine needs to be serviced
Normal Flow of Events	1. System detects machine that needs service 2. Sends a notification to the manager 3. Store the data log on the server
Exception Flow	When the service is not done in 10 days, it sends a similar notification or alert to the manager

Use Case Name	Authorize Service Provider
Primary Actor	Manager
Stakeholder	System, Service Provider
Brief Description	The Service provider needs to be authorised in order to perform any operation on the machine
Trigger	When a manager wants to send the list of machines to be serviced to the service provider
Normal Flow of Events	1. Manager send the list of machines to the service provider 2. Manager authorizes the service provider to perform the service 3. Manager sends a unique service code for login 4. Details are stored in the database
Exception Flow	When the service provider doesn’t show up in a week the manager sends the report again and if it happens twice he looks for other service provider

Use Case Name	Service Provider Login
Primary Actor	Service Provider
Stakeholder	System, Machine
Brief Description	Service Provider logs in with the credentials to service the machine
Trigger	When clicked on “Service Login” Button
Normal Flow of Events	1. Service Provider clicks on “Service Login” Button . 2. Enters the unique service code provided by the manager 3. Retrieve Info from database 4. If successful match, then authorize 5. Store the data log on the server
Exception Flow	If the service provider enters invalid information then the systems shows “Details not Found”

Use Case Name	Machine Service
Primary Actor	Service Provider
Stakeholder	Machine
Brief Description	Service provider performs certain operations provided in the list for proper functioning of the machine
Trigger	After the service operator logs in the system
Normal Flow of Events	1. Service Provider logs in the application/system 2. He performs the tasks assigned for servicing 3. Service operator logs out of the system 4. Operations performed are stored in the database server 5. Service Provider send out detailed report to manager
Exception Flow	If the service is not finished in one attempt - then service provider needs to send the report and inform the manager about the same to receive another unique code

8. Data Dictionary

RFIDMaintenanceSystem = AdminLogin + UserLogin + UserID + AdminID

AdminLogin = AdminID + AddUserDetails + UpdateUserDetails + AddMachineDetails + UpdateMachineDetails

UserLogin = UserID + ScanMachineDetails [RFIDScanner | AndroidApp] + ServiceRequired [Yes | No]

ServiceRequired = ServiceID + AddServiceDetails + SendNotifications + SendServiceReport

MachineOperator = OperatorID + OperatorName + MachineID

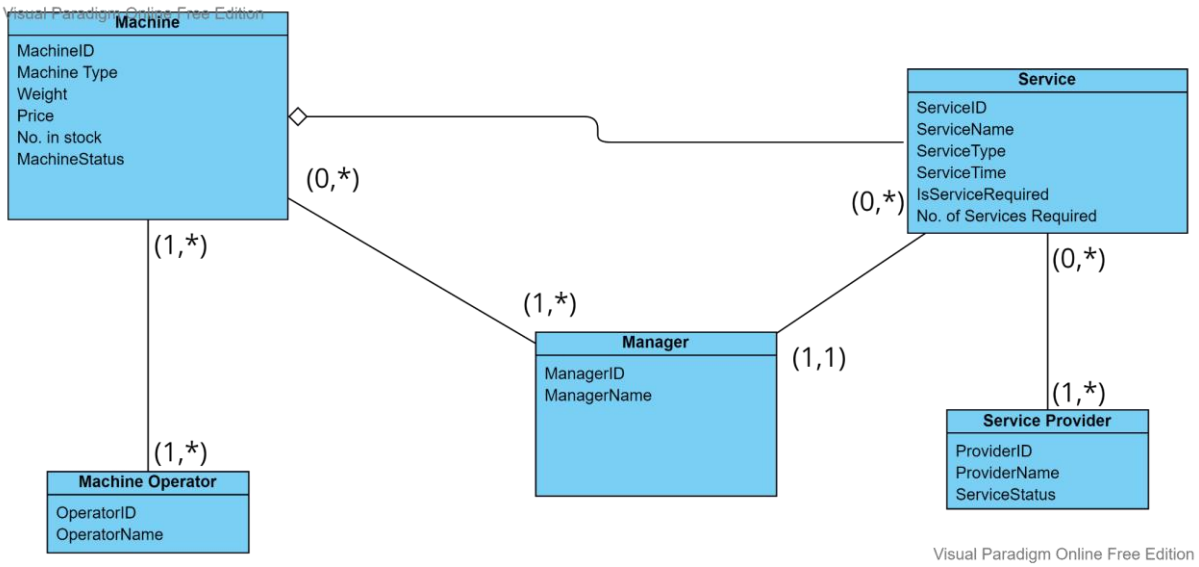
Machine = MachineID + MachineType + Weight + Price + NoInStock + MachineStatus + ServiceID + ManagerID

Service = ServiceID + ServiceName + ServiceType + ServiceTime + isServiceRequired + NoOfServiceRequired + LoginID

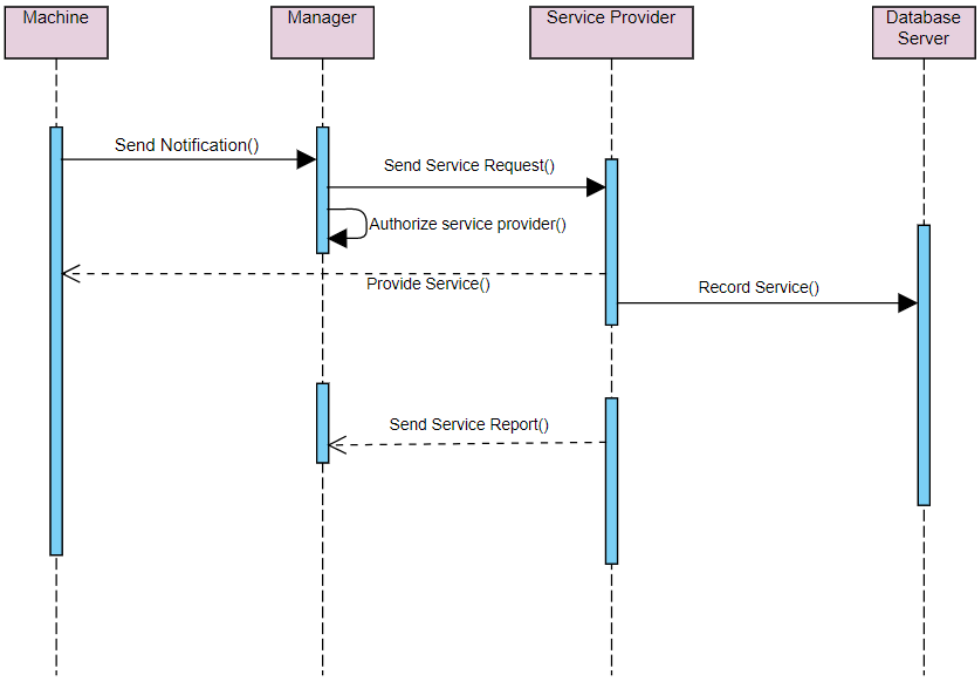
ServiceProvider = ProviderID + ProviderName + ServiceStatus + ServiceID + LoginID

Manager = ManagerID + ManagerName + ServiceID

9. Class Diagram



10. Sequence Diagram



11. Functional Specifications for the Proposed System

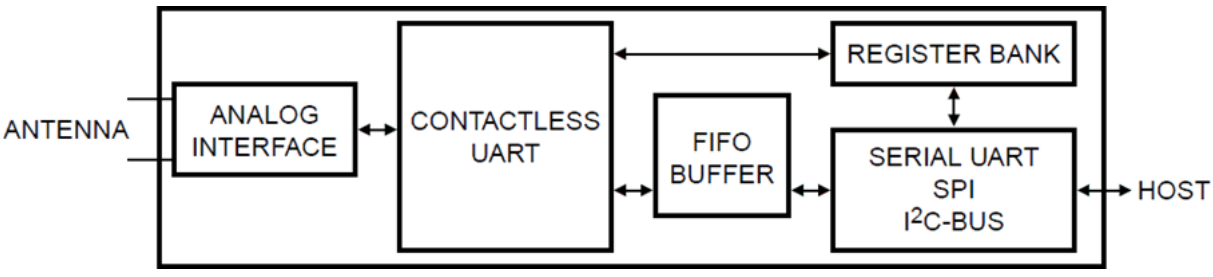
11.1 Components Used

11.1.1 RFID RC522

The NXP RC522 RFID module, which is based on the MFRC522 IC, is one of the most relatively affordable RFID alternatives. The RC522 RFID Reader module creates a 13.56MHz electromagnetic field in order to connect with RFID tags (ISO 14443A standard tags). The reader's interface with a microcontroller is a 4-pin Serial Peripheral Interface (SPI) with a top data rate of 10Mbps. The I2C and UART protocols are also supported. An interrupt pin is provided with the module. Instead of repeatedly asking the RFID module, "Is there a card in view yet?" it will notify us when a tag approaches.
The module's operational voltage ranges from 2.5 to 3.3V, but the good news is that the logic pins are 5-volt tolerant, allowing us to connect it directly to an Arduino or other 5V logic microcontroller without the need for a logic level converter.

RC522 Features
• 13.56MHz RFID module
• Operating voltage: 2.5V to 3.3V
• Communication: SPI, I2C protocol, UART
• Maximum Data Rate: 10Mbps • Read Range: 5cm
• Current Consumption: 13-26mA
• Power-down mode consumption: 10uA (min)

Block Diagram
• The Analog interface handles the modulation and demodulation of the analog signals.
• The contactless UART handles the protocol requirements for the communication schemes in co-operation with the host. The comfortable FIFO buffer allows a fast and convenient data transfer from the host to the contactless UART and vice versa.
• Various host interfaces are implemented to fulfil different customer requirements.



Block Diagram of RC522 Reader

Image of RFID RC522 Reader

11.1.2 Raspberry Pi 3B+

The Raspberry Pi 3 Model B is the Raspberry Pi's third iteration. This powerful credit-card-sized singleboard computer succeeds the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B and can be utilized for a variety of applications. The Raspberry Pi 3 Model B maintains the popular board format while providing a 10x faster processor than the original generation Raspberry Pi. It also includes wireless LAN and Bluetooth connectivity, making it an excellent choice for linked designs. The RASPBERRY PI 3 is a PI series development board. It can be thought of as a single-board computer that runs the LINUX operating system.

The board not only offers a lot of functionality but also has a fast processor, making it ideal for complex applications. The PI board was created with hobbyists and engineers in mind who are interested in LINUX systems and IoT (Internet of Things).

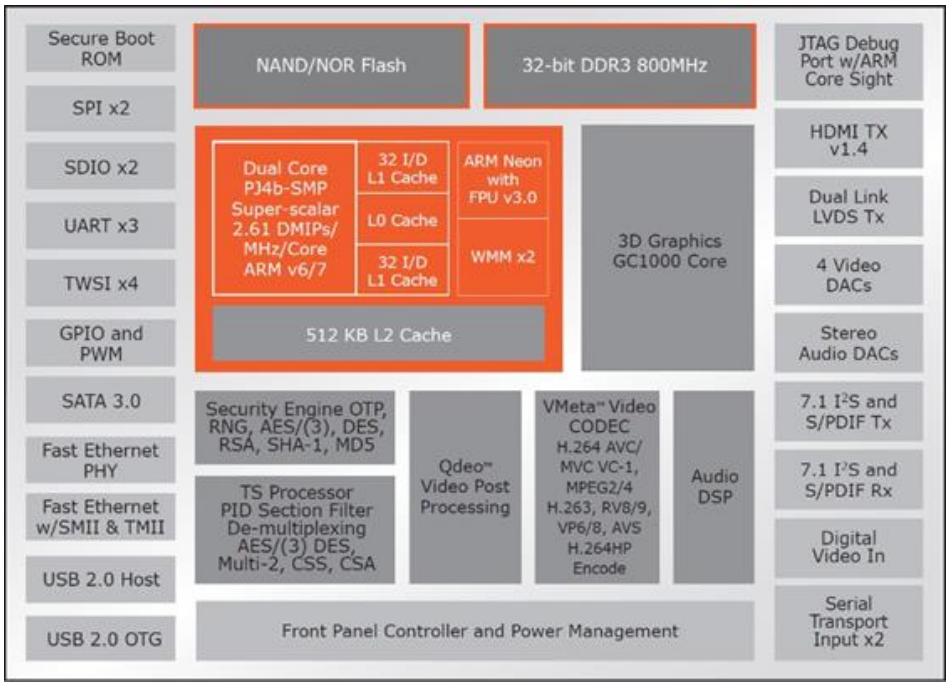
The Raspberry Pi 3 Model B+ is the most recent addition to the Raspberry Pi 3 family, with a 64-bit quad-core processor running at 1.4GHz, dual-band 2.4GHz, and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE functionality through a separate PoE HAT.

The dual-band wireless LAN has modular compliance certification, which allows the board to be incorporated into end devices with much reduced wireless LAN compliance testing, reducing both cost and time to market.

Both the Raspberry Pi 2 Model B and the Raspberry Pi 3 Model B have an identical mechanical footprint.

Raspberry Pi 3B+ Features

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)



Block Diagram of Raspberry Pi 3B+

Images of Raspberry Pi 3B+

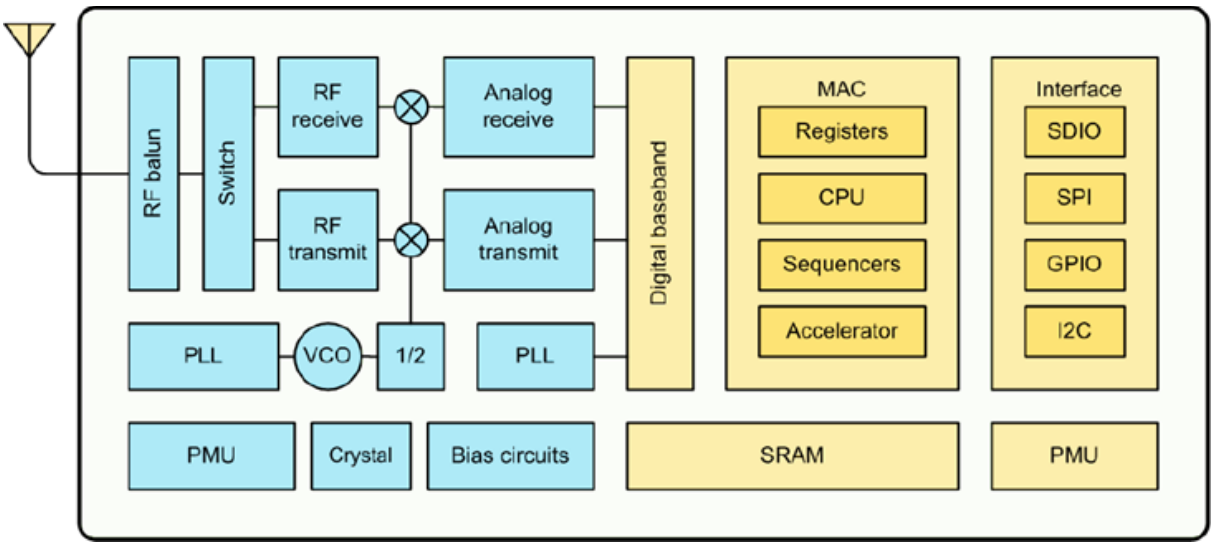
11.1.3 NODEMCU

The ESP32 chip, which is scalable and adaptive, lies at the heart of this module. Individual control of two CPU cores is possible. The clock frequency can be adjusted from 80 to 240 MHz, and RTOS is supported. It's a Wi-Fi+BT+BLE MCU module for general use. ESP-WROOM-32s Traditional Bluetooth, Bluetooth low energy, and Wi-Fi are all included in the module. Various applications: Wi-Fi allows users to connect to a mobile phone or broadcast a BLE Beacon for signal detection; Bluetooth allows users to connect to a mobile phone or broadcast a BLE Beacon for signal detection.

For maximum wireless connectivity, the module offers data rates of up to 150 Mbps with a 20 dBm antenna output power. As a result, this module meets industry-leading criteria in terms of high integration, wireless transmission distance, power consumption, and network connectivity.

NODEMCU Features

- 802.11b/g/n （802.11n，speed up to 150Mbps）
- WIFI Frequency Range 2.4GHz ~ 2.5GHz
- Clock frequency adjustment range from 80 MHz to 240 MHz, support for RTOS
- Built-in 2-channel 12-bit high-precision ADC with up to 18 channels
- Support UART/GPIO/ADC/DAC/SDIO/SD card/PWM/I2C/I2S interface
- Support multiple sleep modes, ESP32 chip sleep current is less than 5 μA
- Embedded Lwip protocol stack
- Supports STA/AP/STA + AP operation mode
- Supports remote firmware upgrade (FOTA)
- General AT commands can be used quickly
- Support secondary development, integrated Windows, Linux development environment



Block Diagram of NODEMCU

Images of NODEMCU

11.2 Softwares Used

11.2.1 Arduino IDE

Arduino is a free and open-source electronics platform with simple hardware and software. Arduino boards can detect inputs such as light on a sensor, a finger on a button, or a Twitter post and convert them to outputs such as turning on an LED, triggering a motor, or publishing anything online. By providing a set of instructions to the board's microcontroller, you may tell it what to do. The Arduino programming language (based on Wiring) and the Arduino Software (IDE) (based on Processing) are used to accomplishing this.

Thousands of projects have used Arduino throughout the years, ranging from simple household items to complicated scientific apparatus. This open-source platform has united a global community of makers - students, amateurs, artists, programmers, and professionals - whose contributions have added up to an enormous quantity of accessible knowledge that may be of tremendous benefit to novices and specialists

alike. Arduino has been used in millions of projects and applications due to its simple and accessible user interface. Beginners will find the Arduino software simple to use, but advanced users will find it adaptable.

It is available for Mac, Windows, and Linux. It is used by teachers and students to build low-cost scientific equipment, to demonstrate chemistry and physics principles, and to begin learning programming and robotics. Designers and architects create interactive prototypes, while musicians and artists utilize it to create installations and try out new instruments. Makers, for example, utilize it to construct many of the items on display at the Maker Faire. Arduino is an essential tool for learning new skills. Anyone - youngsters, amateurs, artists, programmers - may get started tinkering by following the step-by-step instructions in a kit or sharing ideas with other Arduino members online.

For physical computing, there are numerous additional microcontrollers and microcontroller platforms. Similar capability is available in Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many other programs. All of these tools condense the complicated nuances of microcontroller programming into a user-friendly package. Although Arduino simplifies the process of working with microcontrollers, it has one advantage over other systems for teachers, students, and curious amateurs:

• **Low-cost -** Compared to other microcontroller platforms, Arduino boards are affordable. The Arduino module's cheapest version can be assembled by hand, and even pre-made Arduino modules cost less than $50.

• **Cross-platform -** The Arduino Software (IDE) is compatible with Windows, Macintosh OSX, and Linux. The majority of microcontroller systems are only compatible with Windows.

• **Easy-to-use programming environment -** The Arduino Software (IDE) is simple to use for novices while yet being flexible enough for advanced users. It's built on the Processing programming environment, which is useful for teachers because students learning to program in that environment will be familiar with the Arduino IDE.

• **Open source and extendable software -** The Arduino software is accessible as open source tools that experienced programmers can extend. C++ libraries can be used to extend the language, and those interested in learning more about the technical aspects can switch from Arduino to the AVR C programming language. If you desire, you can also include AVR-C code directly in your Arduino applications.

• **Open source and extendable hardware -** The Arduino boards' plans are made available under a Creative Commons license, allowing expert circuit designers to create their own version of the module, extending and upgrading it. The breadboard version of the module can be built by even inexperienced users to learn how it works and save money.

11.2.2 Android Studio

Built on JetBrains' IntelliJ IDEA software and designed exclusively for Android development, Android Studio is the official integrated development environment (IDE) for Google's Android operating system. It's available for Windows, macOS, and Linux users to download. It takes the position of the Eclipse Android Development Tools (ADT) as the primary IDE for developing native Android apps.

At the Google I/O conference on May 16, 2013, Android Studio was revealed. It was in an early access preview stage starting with version 0.1 in May 2013 and then moved into beta stage with version 0.8 in June 2014. Starting with version 1.0, the first stable build was released in December 2014.

Kotlin overtook Java as Google's recommended language for Android app development on May 7, 2019. Java and C++ are still supported.

Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version," as well as "all Java 7 language features and a subset of Java 8 language features that vary by platform version." Some Java 9 features are backported by external projects. Although IntelliJ claims that Android Studio supports all available Java versions, including Java 12, it is unclear to what extent Android Studio supports Java versions up to Java 12. (the documentation mentions partial Java 8 support).

Android supports at least some new language features up to Java 12.

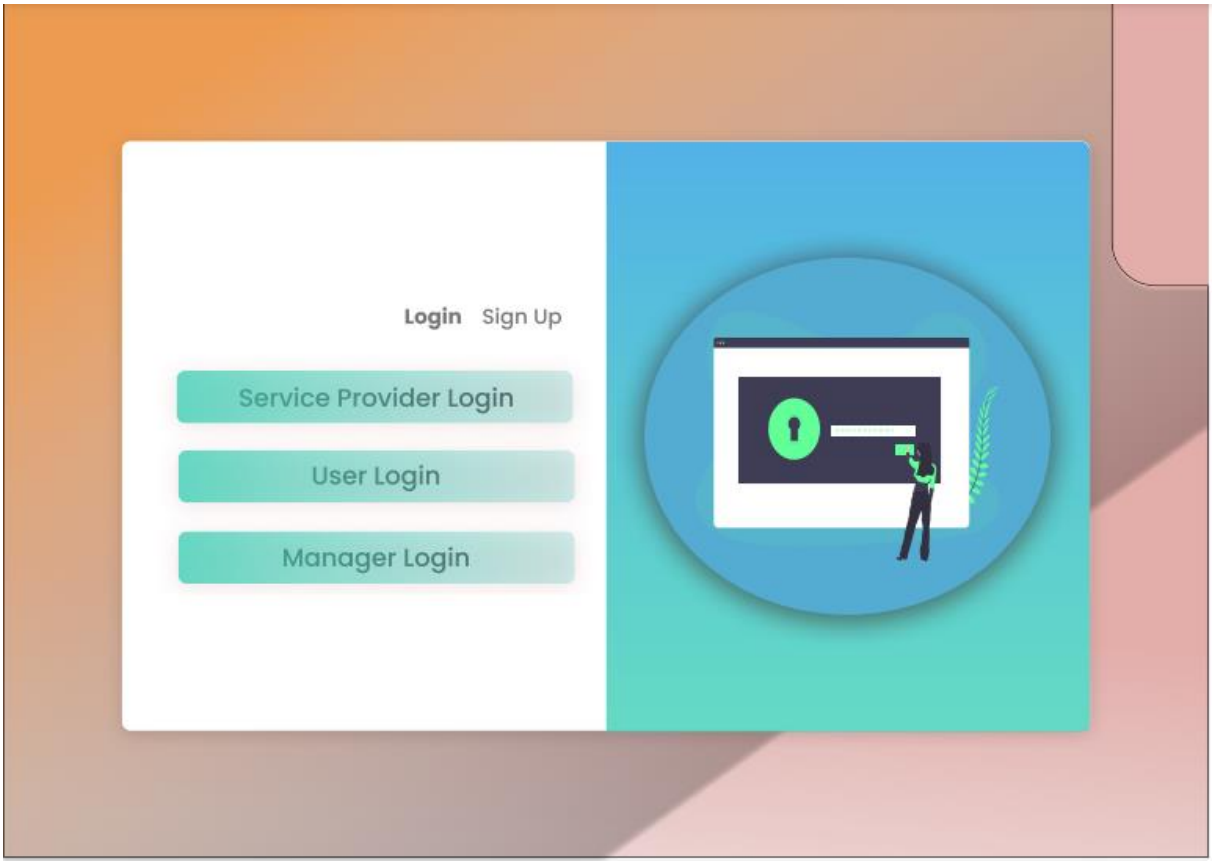
Android Studio has vector graphics-enabled features that allow you to work with vector graphics, which results in better graphics and eliminates pixel distortion in the user interface. It also has the flutter plugin, which allows you to create apps for both Android and iOS with a single code using Google's Dart programming language.

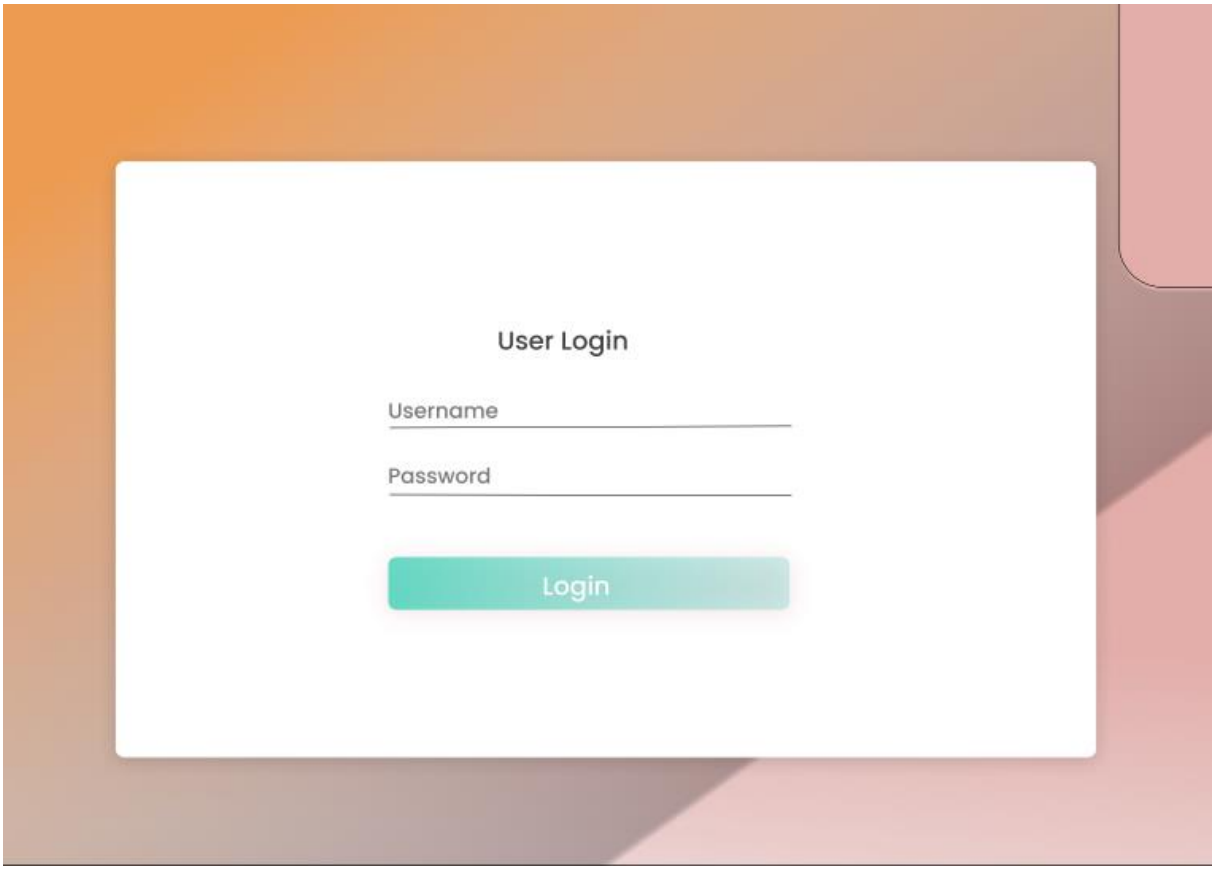
Features

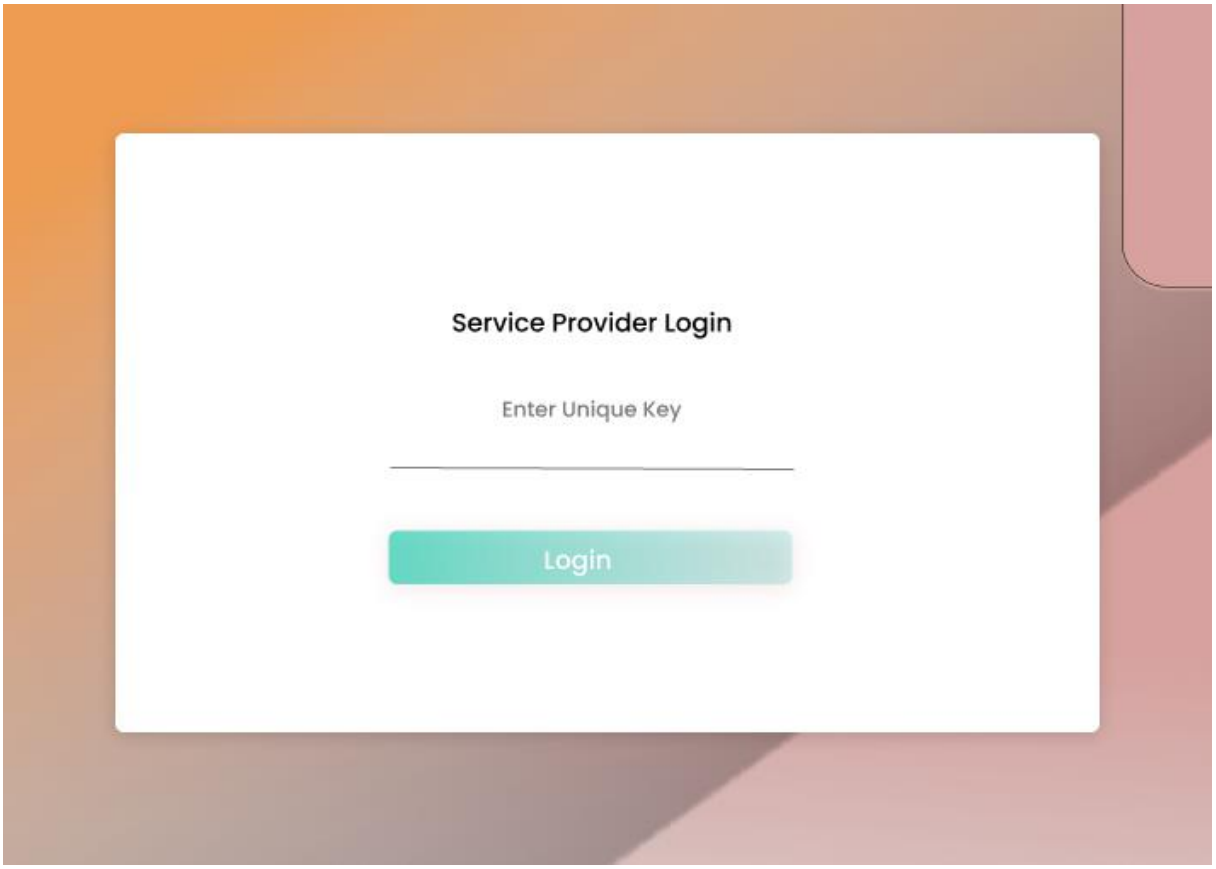
The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine[18]
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

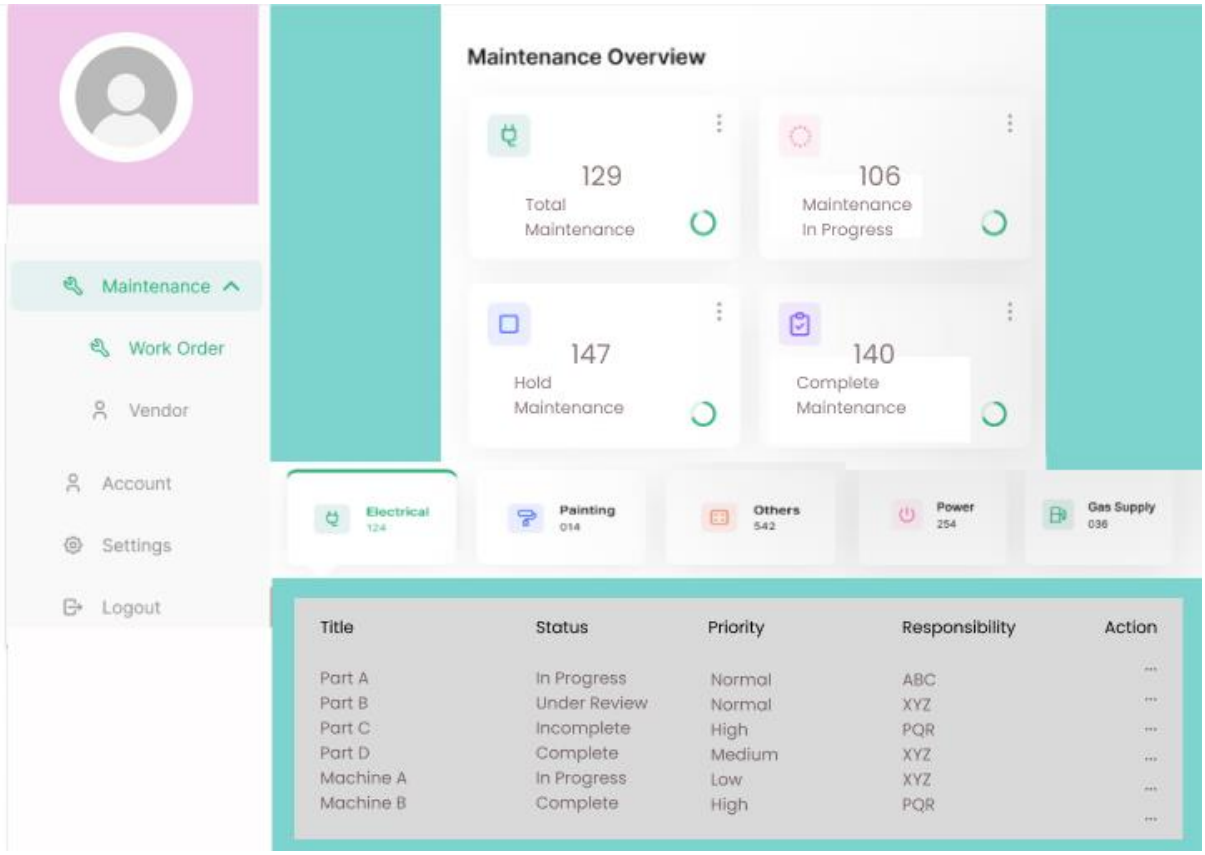
12. Interface Design



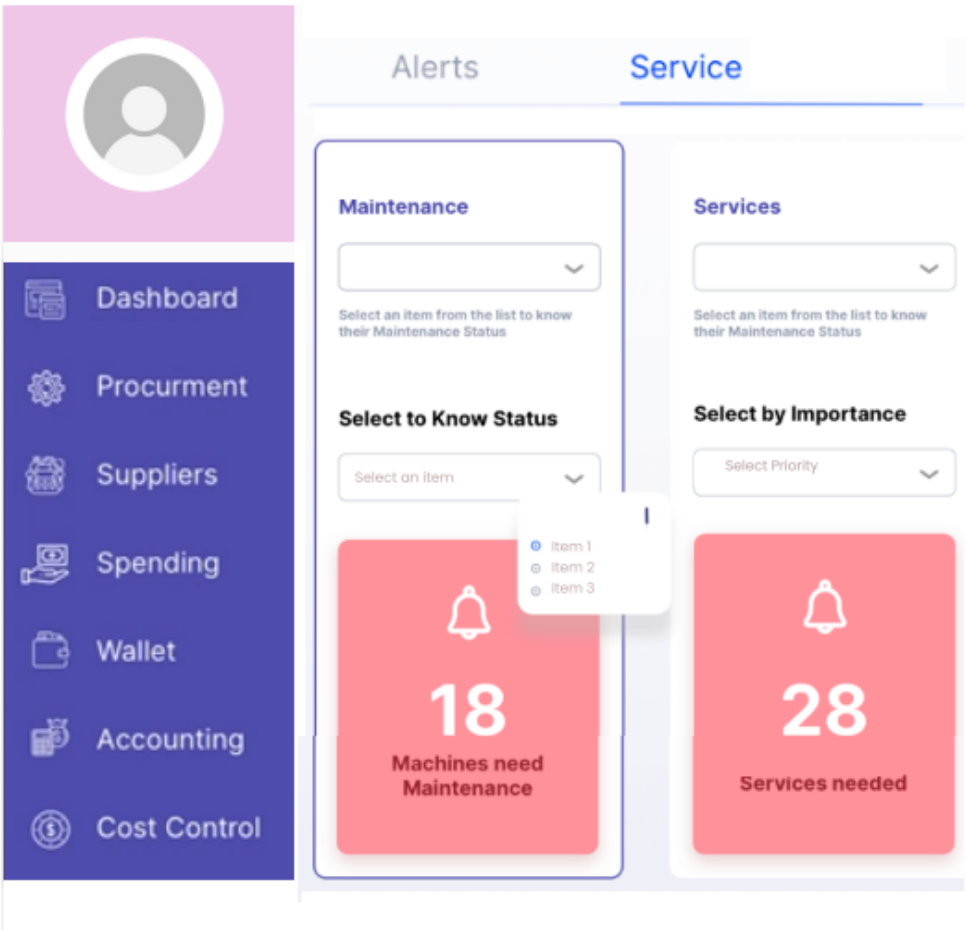




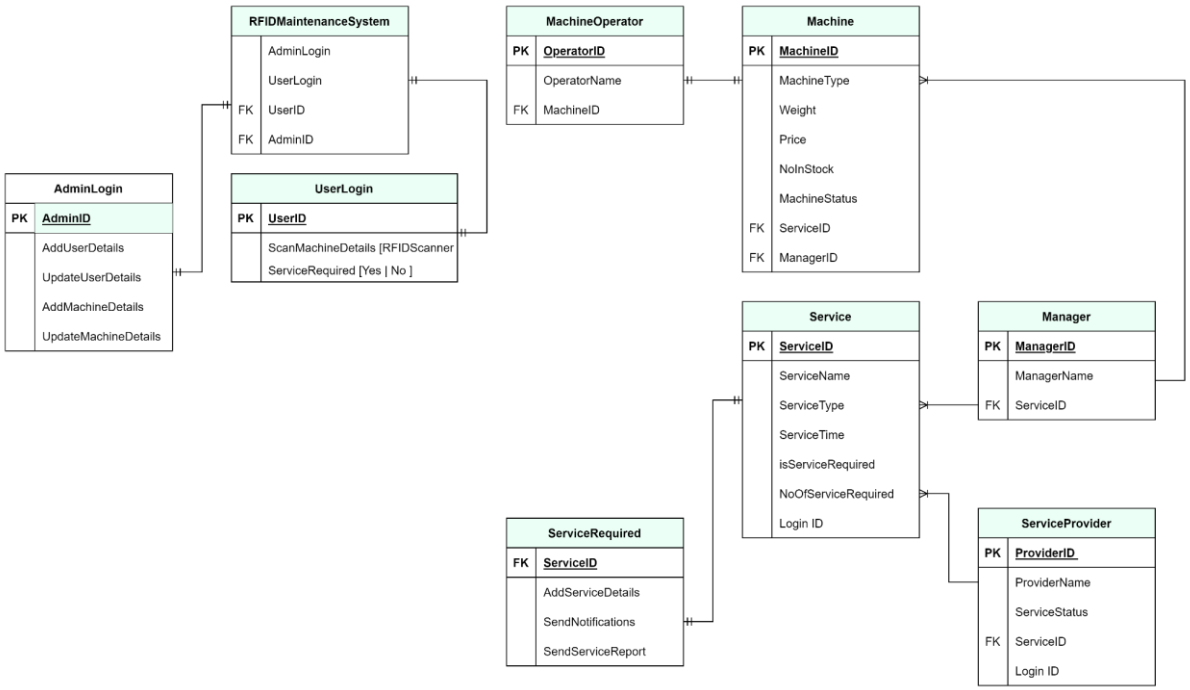
Service Provider Profile



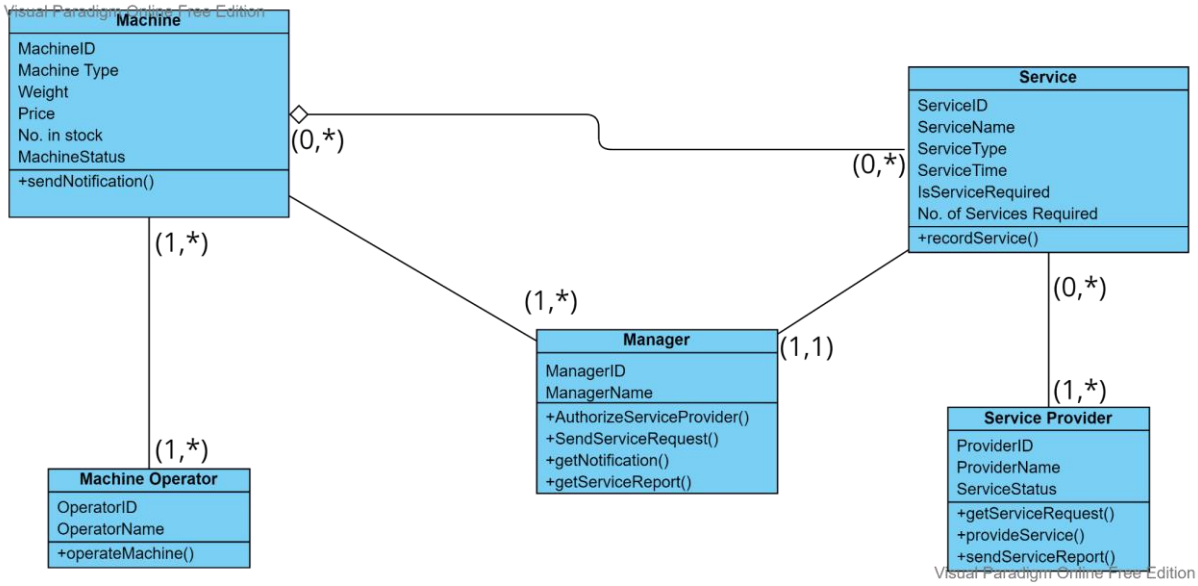
User Profile



13. Database Design



14. Complete Class Diagram



15. Software Requirements

Method 1

Contract
Method Name: sendNotification()
Class Name: Machine
ID: 0001
Clients: Manager
Description of Responsibilities: A Machine will use sendNotification() to make the Manager aware of the Machine requiring service
Arguments Received: MachineID, Machine Status
Type of Values Returned: String
Pre-conditions: Machine has a service requirement
Post-conditions: Machine is waiting to get servic

Logic
```
IF MachineStatus == 1
        sendNotification()
```

Method 2

Contract
Method Name: SendServiceRequest()
Class Name: Manager
ID: 0003
Clients: Managers, Service Provider
Description of Responsibilities: A manager should be able to send a request to the Service Provider for servicing a machine.
Arguments Received: Machine ID
Type of Values Returned: Null
Pre-conditions: Manager should have the permission to send a request. The IsServiceRequired is TRUE.
Post-conditions: Service Provider should be able to view the Service Request

Logic
```
IF IsServiceRequired = TRUE
        SendServiceRequest()
```

Method 3

Contract
Method Name: recordService()
Class Name: Service
ID: 0007
Clients: Managers, Service Provider
Description of Responsibilities: If a service is being provided, this method should record the details of the service
Arguments Received: ServiceID, Service Name
Type of Values Returned: String
Pre-conditions: IsServiceRequired should be TRUE. Manager should have authorized the Service
Post-conditions: Service should be done to the Machine

Logic
```
FOR all MachineID
IF IsServiceRequired = TRUE
        provideService()
            recordService()
```

Method 4

Contract
Method Name: sendServiceReport()
Class Name: Service Provider
ID: 0008
Clients: Service Provider, Manager
Description of Responsibilities: The method should be able to send a Service Report once the Machine is serviced.
Arguments Received: ProviderID
Type of Values Returned: Int
Pre-conditions: A service should have been done
Post-conditions: The similar service shouldn’t be done on the same machine until another request is raised

Logic
```
IF provideService() = TRUE
    sendServiceReport()
```

Method 5

Contract
Method Name: operateMachine()
Class Name: Machine Operator
ID: 0013
Clients: Manager, Machine Operator
Description of Responsibilities: This method should allow the Machine Operator to check if the Machine is up to date after servicing
Arguments Received: OperatorID
Type of Values Returned: Int
Pre-conditions: A service should have been done
Post-conditions: The Machine should be working properly

Logic
```
IF getServiceReport() = TRUE
        operateMachine()
```

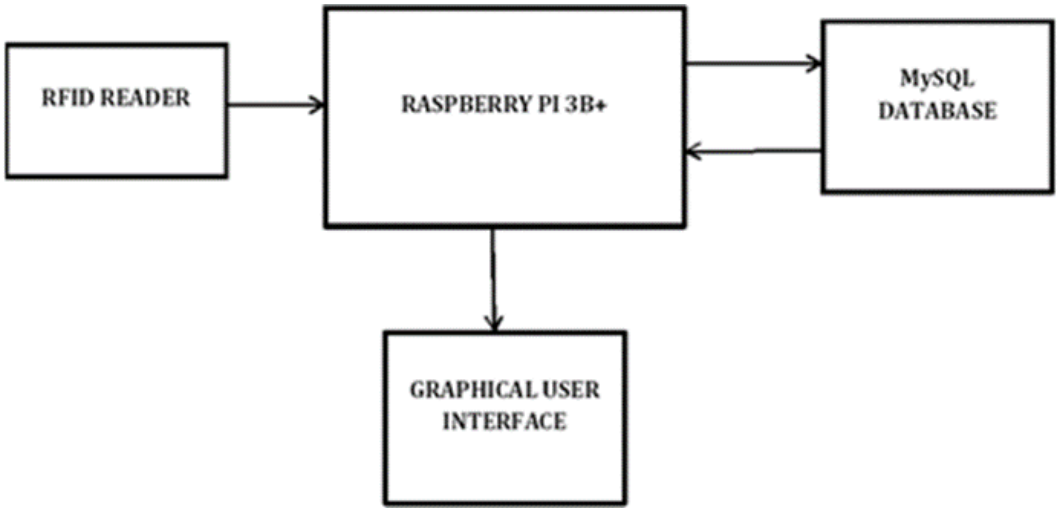
16. Project Management Deliverables

This project will cover the following activities as a part of successful project completion

16.1 Design

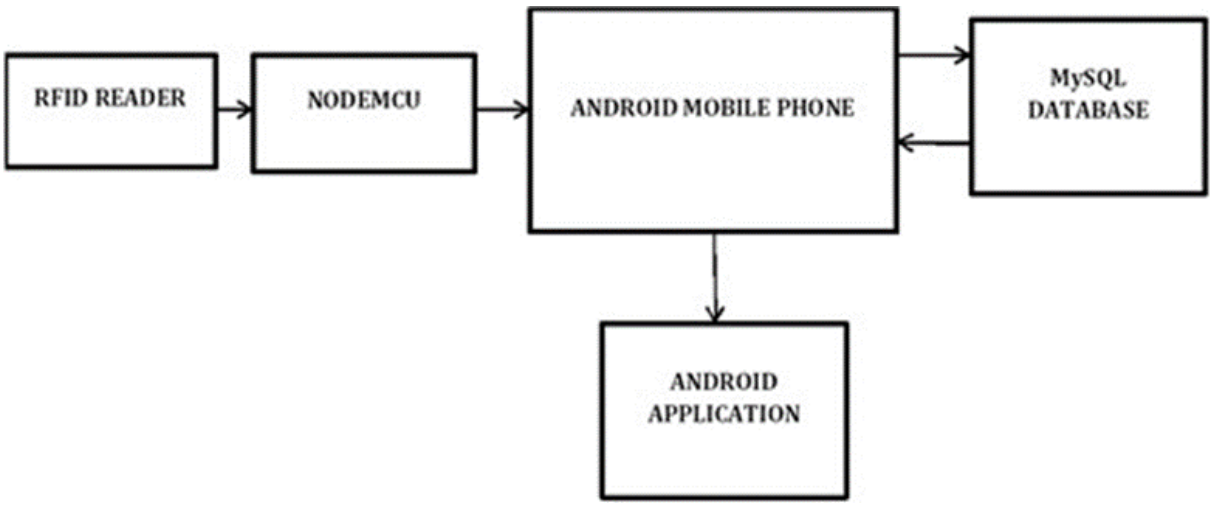
Since the system consists of two parts, our project design will consist of two separate blocks

Part I – Block Diagram for Raspberry Pi Model



The RFID reader will be connected to Raspberry Pi 3B+ using jumper wires. The GUI will be created using python programming language. MySQL Database will be stored on the server which can be accessed using the internet to store the data.

Part II – Block Diagram for Android App Model



The RFID reader will be connected to the NodeMCU which will contain the program to read the RFID tags using the RFID reader. NodeMCU will be connected to the android phone using USB OTG. The phone will be running the android application and will also be connected to MySQL database to store the data.

16.2 Build

The project will be developed using different IDEs

16.2.1 Arduino IDE

Arduino is an open-source electronics platform which is used for programming Arduino boards. These boards are microcontrollers which will be used to detect RFID tags and trigger events to update the database accordingly. Arduino is a good match for our project as it is inexpensive, cross-platform and open-source.

16.2.2 Android Studio IDE

Android Studio is the official integrated development environment (IDE) for Android development. It will be used for developing our android application, which will provide a GUI for the user to scan RFID and update the data in the database.

16.3 Deploy

All the components of this project will be deployed on localhost and on personal android devices

16.4 Deliverables

As a part of project completion, we will be able to deliver the following –

i) Raspberry PI tool for RFID scanning – The Raspberry Pi RFID scanner will be able to read demo RFIDs from machineries. It will also be able to trigger maintenance reminder emails.

ii) Maintenance Application in Android – This will be an android application that allows the user to update the maintenance information. It will also be used by Administrators to manage user information if required.

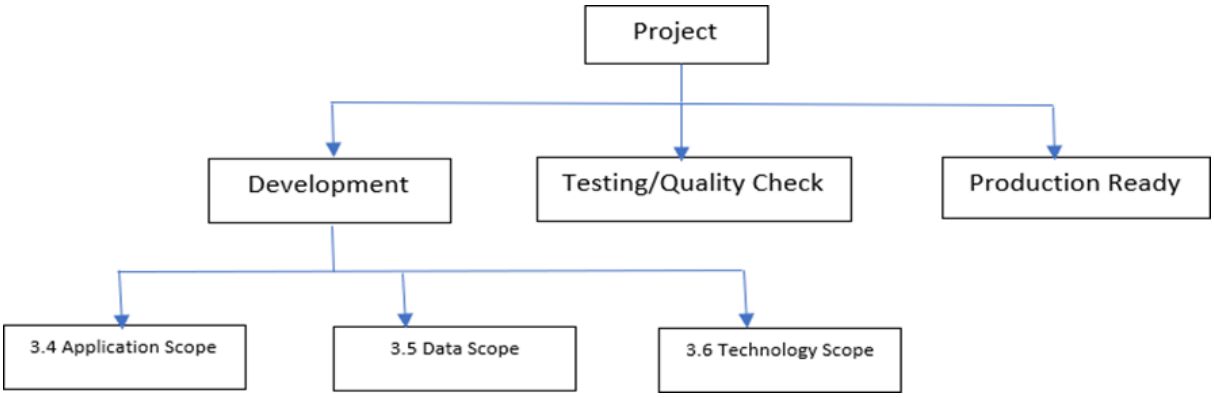
iii) MySQL Database – This database will store all user information, machine information, and maintenance data. This information will be used by the application to trigger emails to users if a service was missed for a machine.

iv) Project Report – Finally we will be submitting a project report which will contain all details about the project including different system diagrams, data flow diagrams, use cases, etc. This report will be useful for future maintenance of the application.

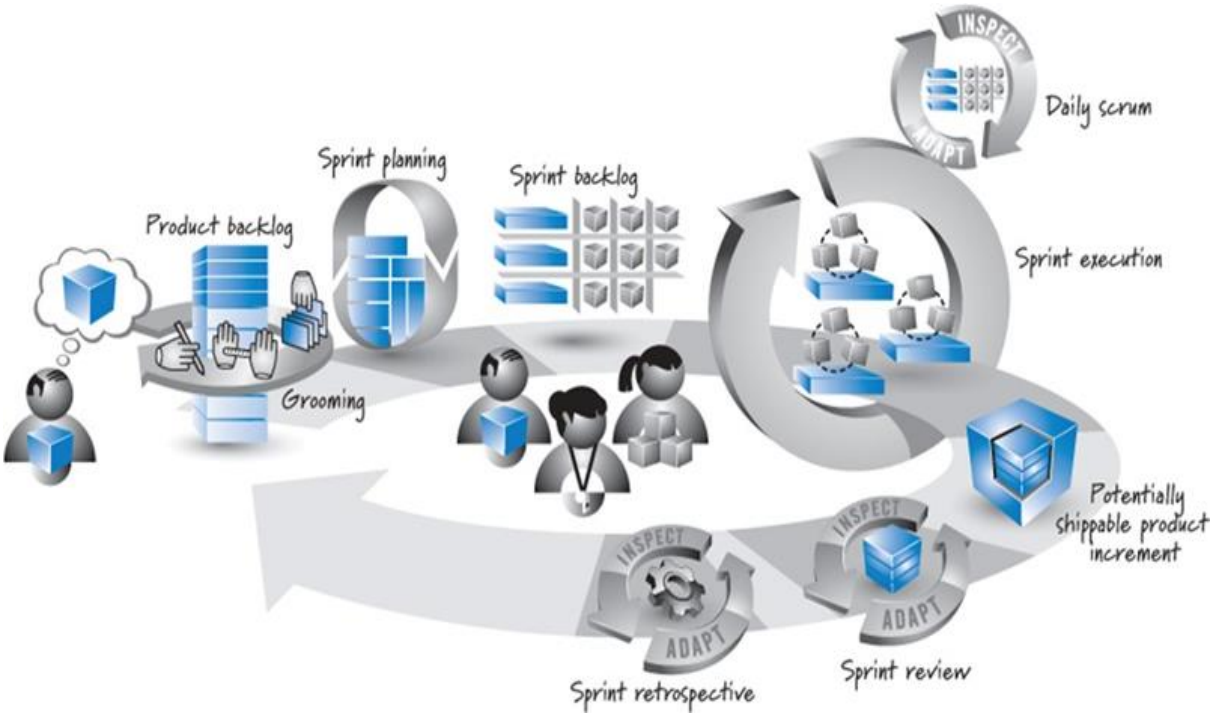
16.5 Project Management Approach

16.5.1 Agreed upon Baseline

According to the agreed baseline, below is the Work Baseline Structure (WBS).



16.5.2 Process to Monitor Progress



Project Planning: Project is monitored using Project Management tool “JIRA”. All the tasks in the project are created as JIRA stories. All the user stories are sub divided into categories Testing, Development, DevOps to differentiate the stories.

Development: Before the Start of Sprint Planning, Product Owner/ Project Manager grooms the stories according to the iterations defined (refer 4.1). After the sprint Backlog, stories are assigned to the developers. Scrum Master is the process Authority here coaching all the team members in the sprint.

QA Testing: At the end of sprint all the stories are moved to testing environments and tested before moving to production environment. User Acceptance test reports are taken to analyze any discrepancies in the output.

Deployment: Once all the testing is done with no errors, everything will be moved to production and made it live to use. These deployments are taken care by DevOps Engineers to maintain isolation between environments.

Scrum Review & Retrospective: Development team, Project Owner, Project Manager and Scrum Master come together to discuss what is working and what is not working and how to improve sprints.

Scrum Daily Calls: Daily Standups are needed to give updates to everyone in the team. This brings transparency in the assigned work.

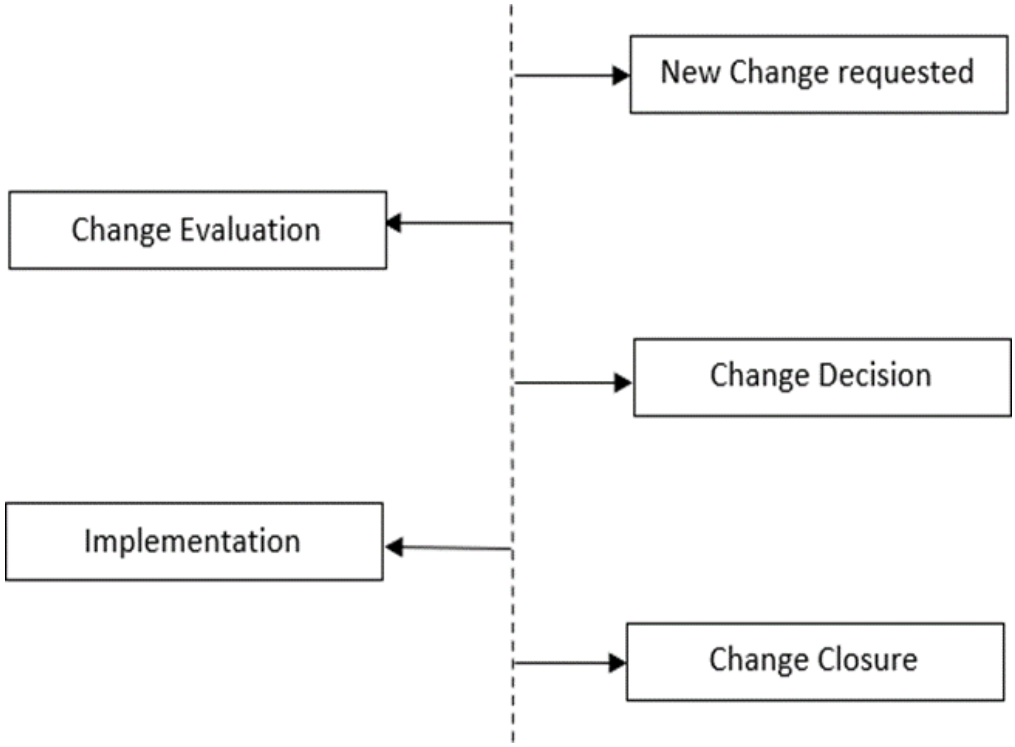
16.5.3 Means of Communication

- Regular Stand-up calls helps team to get updates on daily basis. Post standup calls, team maintains Minutes of Meeting (MoM) which tracks all the discussions.
- Communication is a key if any person is not able to make up for the meeting. An email is needed to convey the absence before the call of meeting.
- For individual dependency works one-to-one calls are sufficient but for team dependency works email communication should be preferred before scheduling any call.

16.5.4 Approach for Dealing with Issues

- Whenever there is a new issue, key aspects like the scenario & use case must be noted.
- After this, next phase will be finding the root cause of the issue.
- Documentation must be maintained as sometimes we can find answers to these issues.
- If the issue is a small correction in the current sprint, it is fine to be included, else it would be appropriate to move this issue into the next spring planning.

16.5.5 Change Control Procedure



A New Change timeline consists of stages ranging from Change request till the close of the change.

- **Change Request**: Whenever a new use case is encountered, the issue is noted and the scenario is also re-tested. If the issue still exists, then a new change is requested.
- **Change Evaluation**: Once the change is requested Product Owner, Project Manager, Scrum Master takes up a meeting and checks the use case and finalizes the request.
- **Change Decision**: A Decision is given whether the change is needed or its not required for the current MVP.
- **Implementation**: Once the change is included, it is moved to the Product Backlog with all the existing stories.
- **Closure**: As the stories are being completed in the sprints, all of them are closed after successful testing on production and test environments.

16.5.6 Risk Management

Risks can come from various sources including uncertainty in threats from equipment failures (at any phase in design, development, production, or sustaining of life-cycles), accidents, natural causes and disasters, deliberate attack from an adversary,

or events of uncertain or unpredictable root-cause. So, for the maintenance of industrial equipment, risk management can be handled by following a 5-step strategy.

1) Identify potential risks
2) Measure frequency and severity
3) Examine alternative solutions
4) Decide which solution to use and implement
5) Monitor results

16.5.7 Acceptance of Deliverables

The ultimate goal of deliverables acceptance management is to ensure successful maintenance of all the equipment in a specified time period, within budget and within scope. For achieving this goal, there are some key tasks that the project manager must accomplish.

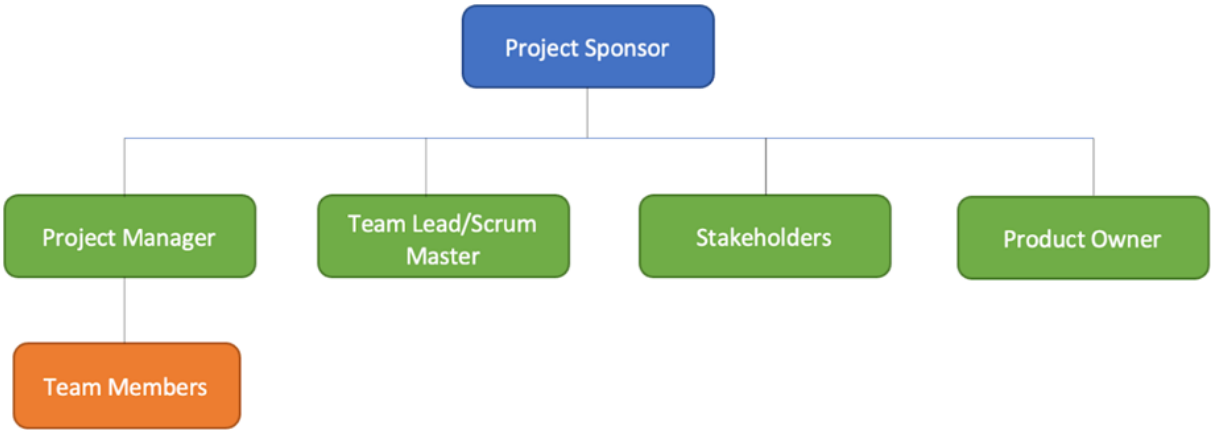
These tasks are:

- Determine and assign a person who will review the deliverables and assure their adherence to the criteria listed in the plan (as it's been said, normally the project manager accomplishes this task but there can be exceptions).
- Set up any time considerations that may be required for managing the acceptance of project deliverables and adherence to customer expectations.
- Make a deliverables acceptance document report that represents all the deliverables actually accepted and approved.

16.5.8 Project Completion Criteria

Completion criteria can be a simple checklist or a more comprehensive set of standards or protocols. Whatever will be used to determine if the task was successfully completed should be defined in advance, in the completion criteria for the task. Often, the task owner or the subject matter expert is the best person to create these completion criteria.

16.6 Project Organization



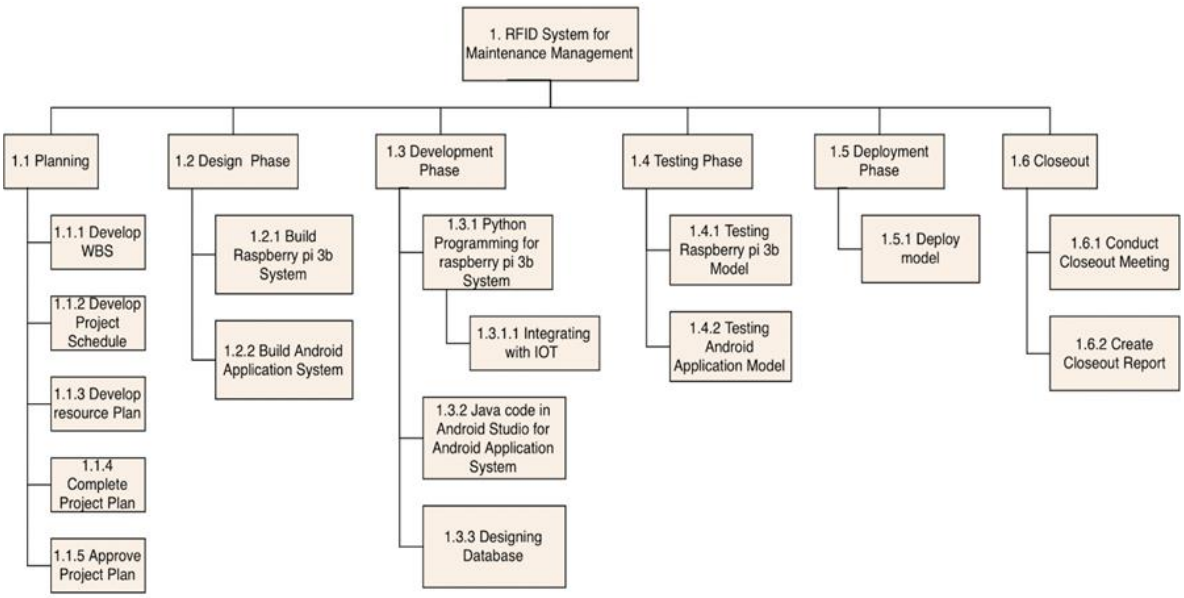
16.7 Roles and Responsibilities

Note: Apart from the roles mentioned below, everyone's work includes software development which includes collaborating with management, departments and customers to identify end-user requirements and specifications and producing efficient and elegant code based on requirements.

Sno	Name	Role	Responsibilities
1	Ayush Baldwa	DB Administrator	• Find ways to store, organize, and manage your data using database software. • Ensuring that databases meet user requirements. • DB design, Use Cases, Class diagrams
2	Khushi Shah	DevOps Engineer	• Planning the team structure, activities, and involvement in project management activities. • Implementing various development, testing, automation tools, and IT infrastructure. • Software design • Context Digaram
3	Richa Ashutosh Pandey	Project Manager	• Plan,Develop the project Idea. • Monitor Project Progress and set deadlines. • UI Design • Project Management deliverables • Class Diagram
4	Sushmitha Madiraju	IOT Engineer	• Designing, coding and testing features of IoT devices • Developing software that allows

			IoT devices to function and connect to other devices • BPMN and Functional Specifications
5	Taufeeq Mahendi Samnani	Test Engineer	• Writing test plans and creating test cases for the product. • Troubleshooting any errors and streamlining the testing procedures. • Sequence diagram, Data Dictionary, Executive Summary

16.8 Project Work Breakdown Structure



16.9 Planned and Execution Timeline

Weekly Schedule	Tasks
Aug 29th 2022 - Sep 4th 2022	Forming a Team
Sep 5th 2022 - Sep 11th 2022	Team Introduction
Sep 12th 2022 - Sep 18th 2022	Project Idea Brain storming
Sep 19th 2022 - Sep 25th 2022	Meet with Professor and discuss the project idea to gather suggestions
Sep 26th 2022 - Sep Oct 2nd 2022	Problem Solution, Requirements verification, BPMN
Oct 3rd 2022 - Oct 9th 2022	Context Digram, Use Case Digram and Descriptions, Data Dictionary
Oct 10th 2022 - Oct 16th 2022	Class Diagram, Sequence Diagram
Oct 17th 2022 - Oct 23rd 2022	Complete Class Diagram
Oct 24th 2022 - Oct 30th 2022	Functional Specifications
Oct 31st 2022 - Nov 6th 2022	Interface Design
Nov 7th 2022 - Nov 13th 2022	Database Design
Nov 14th 2022 - Nov 20th 2022	Software Design
Nov 21st 2022 - Nov 27th 2022	Executive Summary, Project Management Deliverables Documentation
Nov 28th 2022 - Dec 4th 2022	Document Formatting and Proof-Reading
Dec 5th 2022 - Dec 10th 2022	Submit the Project Report

16.10 Meeting Minutes

Date & Time	Team Members Present	Topics Discussed
Sep 15th 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Project Idea Brain storming
Sep 22nd 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Finalized the features of the project after meeting with professor, made the problem statement, assigned tasks
Sep 29th 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Finalized BPMN Model and context diagram. Discussed use cases
Sep 6th 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Reviewed the use cases and made adjustments to it. Built the data dictionary too.
Oct 20th 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Discussed the class diagrams built, figured out relationships/cardinalities along with the methods. Verified the sequence diagram
Oct 27th 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Made sure all the functional specifications have been documented correctly and the methods have been added in a new class diagram
Nov 10th 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Added elements to the UI design being built and added DB design
Nov 17th 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Software Design Enhancements
Dec 1st 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Documented the Project Management Deliverables correctly and reviewed UI, DB design and software design
Dec 5th 2022 - 10:00PM	Ayush, Khushi, Richa, Sushmitha, Taufeeq	Proof-read everything and ensured that the document is ready to submit