

BASE STATION MONITORING APP

Mahakrishna Moorthy M^[1] , Dheenuka M^[2], Krishna Kumar E^[3]

[1]Final Year, Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu ,India- 638401

[2]Final Year,Information Science and Engineering,Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu ,India- 638401

[3]Final Year,Information Technology ,Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu ,India- 638401

[1] mahakrishnamoorthy.cs19@bitsathy.ac.in [2] dheenuka.ig19@bitsathy.ac.in,

[3] krishnakumar.it19@bitsathy.ac.in

ABSTRACT

The base station monitoring is a tool for monitoring and managing the performance of the base station in a telecommunication network. This software application enables the remote monitoring of base stations, which means that network operators can monitor the performance of their network from any location. The application utilizes information from different sources, for example, signal strength, organization traffic to give bits of knowledge about the internet signals. The app provides real-time information about the status of base stations, including their location, signal strength and connectivity and navigation to the base station signal. Clients can see information on a dashboard of the mobile application or guide connection point, the dashboard contains information about the analytics of the signal strength and have the mode of internet connection received in the application. Users can view the detailed report and performance of the individual base station and identify any potential issue that may affect the network performance. The app also provides features for alerts message and notification, allowing users to quickly respond to any network outages or other issues and it has the speciality to fetch and store the data locally and it will be helpful to the users to view the network availability areas when they are stuck in a poor network area. This mobile application has a user-friendly interface, making it easy for network operators to access and understand the it provides to the user and comprehensive monitoring capabilities, the base station monitoring application is an essential tool for network operators seeking to optimize the performance and reliability of their telecommunications networks. This app is a most useful tool for users to manage and view the telecommunication network. This mobile application is also bug free, we can achieve this with the help of automation. It will reduce the bugs and make the mobile app flexible and more essential to use. It also improves the usability of the user experience then to free better to use this mobile application.

Keyword: Base station,Monitoring system,Network management,Remote monitoring,Fault detection,Alarm notification,Performance analysis,Real-time monitoring,Data visualization,Machine learning,Artificial intelligence,Internet of Things (IoT),Wireless networks,Network operations center (NOC),Network automation,Infrastructure management,Radio frequency (RF),Operational efficiency,Service quality.

1. INTRODUCTION

We're planning to create a mobile application for base station monitoring. Based on a review of studies published up until 2011, the International Agency for Research on Cancer has classified Radio Frequency (RF) radiation as "possible carcinogenic to human" based on the limited evidence of a possible increase in risk for brain tumors among mobile phone users and inadequate evidence. Our software determines the user's present location while also measuring the internet speed. Following this procedure, the neighboring towers are displayed in map view to determine whether or not the user is within the dangerous range (300 meters).

This app provides a centralized platform for monitoring the performance status of base stations. It allows network operators to remotely monitor and manage their network from any location, providing real-time data on the status of the base station and enabling them to quickly and respond to issues. The aim of a base monitoring app can vary depending on the context, but generally it refers to an application that is designed to track and analyze the performance of a network infrastructure. To give current data on the functionality of base stations for cell phone networks, and also provide data on signal strength, conversation drop rates, and network traffic should be tracked and analyzed. Determine any possible problems that might affect the user experience.

The primary goal of such an app is to provide system administrators, travellers or IT professionals with real-time information about the health and status of their system or network. This information can include metrics such as memory usage, network traffic, and other performance indicators. By monitoring these metrics, a base monitoring app can help detect potential issues before they become critical, allowing system administrators to take proactive steps to resolve them. Additionally, the app can help identify trends and patterns in system behavior, which can be used to optimize performance and plan for future resource needs.

2. LITERATURE SURVEY

The deployment of 5G networks has brought about new challenges in the management and operation of wireless networks. One of the emerging solutions is the implementation of base station automation systems. To gain a comprehensive understanding of this topic, various research studies have been conducted, resulting in a wealth of information and insights. A survey by Ghazal and Qahwaji (2019) provides an overview of the state-of-the-art base station automation techniques, including network function virtualization, software-defined networking, and artificial intelligence-based methods. Another survey by Zhang et al. (2019) focuses on the challenges and opportunities of base station automation in 5G networks, and proposes a system architecture for a highly automated and intelligent network. In addition, Liu et al. (2020) provide a comprehensive survey of base station automation, discussing the technical aspects of automation such as network slicing, edge computing, and network orchestration. Other recent surveys by Zhang et al. (2020) and Zhao et al. (2021) provide valuable insights into the latest developments and trends in base station automation for 5G networks. Overall, these surveys offer a broad range of perspectives and valuable information for researchers, engineers, and network operators to further advance the development of base station automation systems.

3 PROBLEM DEFINITION

The problem that a base monitoring app is designed to address is the need for system administrators or IT professionals to have real-time visibility into the performance of their computer systems or network infrastructure. Without a monitoring app in place, it can be difficult or even impossible to quickly identify and troubleshoot issues that may arise within the system. For example, a sudden spike in network traffic may indicate a security breach, a

hardware failure, or a misconfigured application. Without real-time monitoring and alerts, system administrators may not be aware of these issues until they have already caused significant damage or downtime. A base monitoring app automates this process, collecting and analyzing performance data from across the system and presenting it in a clear and actionable format. Unquestionably, the development of the mobile network sector has had many advantages, including better connectivity, enhanced communication, and access to a variety of digital services. The possible health risks linked to electromagnetic radiation (EMR) exposure from base stations and mobile devices are one such drawback. Despite continuing studies into the effects of EMR, many people are still unaware of the dangers and are unsure of how to stay safe. Therefore, creating a mobile application to spread knowledge of these problems is a step in the right direction in solving the issue. The application contains a feature to measure internet traffic in addition to bringing EMR's potential risks to people's attention. Overall, the problem that a base monitoring app addresses is the need for real-time visibility into system performance in order to detect and resolve issues quickly and effectively, and to optimize the system for maximum efficiency and reliability.

4 OBJECTIVE

The fundamental goal of our task is to mindful individuals that their area is inside the unsafe scope of the base station or not. The main objective of our project is to design and implement an effective tool that provides real-time monitoring and analysis of network infrastructure. The objective can be broken down into several specific goals. Developing a clear understanding of the requirements and constraints for the monitoring app, based on the needs of the user and the specific context. Identifying the key metrics and performance indicators that need to be monitored in order to ensure the reliable and efficient operation of the system. Selecting appropriate monitoring tools and technologies that can collect and analyze the required data, and integrating them into the monitoring app. Designing and implementing a user-friendly interface that presents the monitoring data in a clear and actionable format. Developing an effective alerting system that can notify system administrators of any issues or anomalies in real-time, and provide guidance on how to resolve them. Testing and validating the monitoring app to ensure that it meets the requirements and is effective in detecting and resolving issues. Documenting the design, implementation, and testing processes in a comprehensive project that can be shared with stakeholders and other interested parties. The project should provide a detailed account of the design and implementation process, as well as the results and effectiveness of the monitoring app.

2.SYSTEM ANALYSIS

2.1 SYSTEM ANALYSIS

The main focus for a base station monitoring app typically involves a thorough evaluation of the requirements, constraints, and challenges associated with monitoring the performance of a computer system or network infrastructure. This analysis should be guided by the specific context of the project, including the type of system or network being monitored, the size and complexity of the system, and the goals and objectives of the monitoring app.

2.2 EXISTING SYSTEM

The current application's name connected with our venture are Cell Pinnacle Area, LTE Disclosure, Organization Cell Data Light, OpenSignal and so on. These applications are comparative in estimating the web speed of the client and finding the area of neighboring pinnacles with applicable data. One of these applications (LTE Disclosure) shows the best cell network inclusion in an unambiguous area. These applications don't show the distance between the pinnacle and the client. That is, to know regardless of whether the client is inside the destructive reach.

2.3 PROPOSED SYSTEM

The proposed system is a base station monitoring app that allows operators to remotely monitor and manage the performance and status of base stations. The app is designed to be easy to use and accessible to a wide range of operators, including those without specialized technical expertise.

2.3.1 Features:

Real-time monitoring: The app provides real-time monitoring of key metrics such as signal strength, network traffic, and performance.

Alerts and notifications: The app can be configured to send alerts and notifications to operators when certain metrics fall outside of predefined thresholds, allowing for quick response to issues.

Performance analysis: The app provides detailed performance analysis of the base station, allowing operators to identify areas for improvement and optimize network performance.

Data visualization: The app provides graphical representations of data and metrics, making it easier for operators to identify trends and patterns in the data.

User management: The app allows for multiple users to access the app with different levels of access rights and permissions.

Benefits:

Improved efficiency: The app allows operators to quickly and easily monitor and manage base stations, reducing the need for manual monitoring and troubleshooting.

Cost savings: The app can reduce the need for expensive hardware and specialized monitoring software, resulting in cost savings.

Improved network performance: The app provides detailed performance analysis of base stations, allowing operators to identify areas for improvement and optimize network performance.

Accessibility: The app is designed to be accessible to a wide range of operators, regardless of their technical expertise.

SYSTEM DESIGN

3.1 SYSTEM REQUIREMENTS (HARDWARE):

3.1.1 Hardware Requirements

Processor : Intel i5/i7(Laptop)

RAM : 8gb (min)

3.2 SYSTEM REQUIREMENTS (SOFTWARE):

3.2.1 Android Studio:

Android Studio is primarily used for developing Android apps. It provides a rich set of tools for building user interfaces, managing project resources, debugging, and testing. It also includes a powerful code editor with features such as auto-completion, refactoring, and code analysis. Emulator and device testing: Android Studio includes a built-in emulator that allows developers to test their apps on a virtual Android device. It also provides tools for connecting physical devices for testing, debugging, and deployment. Performance analysis and optimization: Android Studio includes tools for analyzing app performance, identifying bottlenecks, and optimizing code. This helps developers to ensure that their apps are fast, responsive, and efficient.

3.2.2 Postman:

Postman is a popular tool for testing and debugging API requests. It provides a user-friendly interface for creating, testing, and documenting API requests and responses, making it easier for developers to work with APIs. Request building: Postman allows developers to build and send HTTP requests to APIs, including GET, POST, PUT, PATCH, DELETE, and others. It also supports authentication, headers, and other parameters that are commonly used in API requests.

The goal of web scraping is to extract data from websites that can be useful for various purposes, such as research, analysis, and automation.

3.2.3 Web Scraping:

Web scraping can be performed using various programming languages, such as Python, Java, and Ruby. Here are some of the main steps involved in web scraping:

Identifying the target website: The first step in web scraping is to identify the website or web page from which data needs to be extracted.

Storing the data: Once the data is extracted, it needs to be stored in a structured format such as CSV, JSON, or a database. This data can then be used for further analysis or automation.

3.2.4 Realm DB:

Realm is a mobile database solution that allows developers to build fast, reactive, and scalable applications with minimal overhead. It provides a local database solution that can be used to store and manage data on the device itself, which is especially useful for mobile applications that need to work offline or with limited connectivity.

3.2.5 Android SDK:

You must install the Android SDK in order to test Android applications. It is available for download on the website for Android developers. For the Android version you want to test, you will also need to install the proper Android Platform Tools and Build Tools.

3.2.6 Appium:

Appium works on portable application testing by giving a solitary system that upholds numerous stages, programming dialects, and test structures. It additionally empowers quicker testing, more straightforward upkeep, and further developed testing inclusion.

3.2.7 Testing framework:

For the testing framework we use TestNG. Adding the TestNG reliance to your task's pom.xml record or build.gradle record. You want to make test classes for every module or usefulness of the base station checking framework. To characterize test techniques for each experiment. Test techniques ought to incorporate declarations that check the normal way of behaving of the framework. Test suites permit you to run different experiments together and create merged test reports.

3.2.8 IntelliJ

Open IntelliJ Thought and select "Make New Task" from the home screen. In the following screen, select your programming language (e.g., Java), construct device (e.g., Expert or Gradle), and other venture settings. Click "Finish" to make the undertaking. Open the form document (e.g., pom.xml for Expert or build.gradle for Gradle) and add the Appium conditions to your undertaking. You can track down the most recent variant of the conditions on the Appium site. To navigate to the "src/test/java" envelope and select "New" -> "Java Class" to make another test class. Give it a name and snap "Alright".

3.3 SYSTEM DESIGN

For Data collection, several probes, and network interfaces are used to gather data from base stations. This data consists of important metrics including signal strength, network traffic, power consumption, temperature, and others. A database is used to store the gathered data for further analysis and processing. Scalable, resilient to faults, and secure databases are required. It must have the capacity to manage a sizable volume of data and offer quick access to the info. To identify any irregularities or problems with the base stations, the acquired data is immediately evaluated. Techniques for data visualisation, machine learning, and statistical analysis are all used in this investigation. To guarantee the data's confidentiality, integrity, and accessibility. This covers technologies for safe data transit, authentication, and access management.

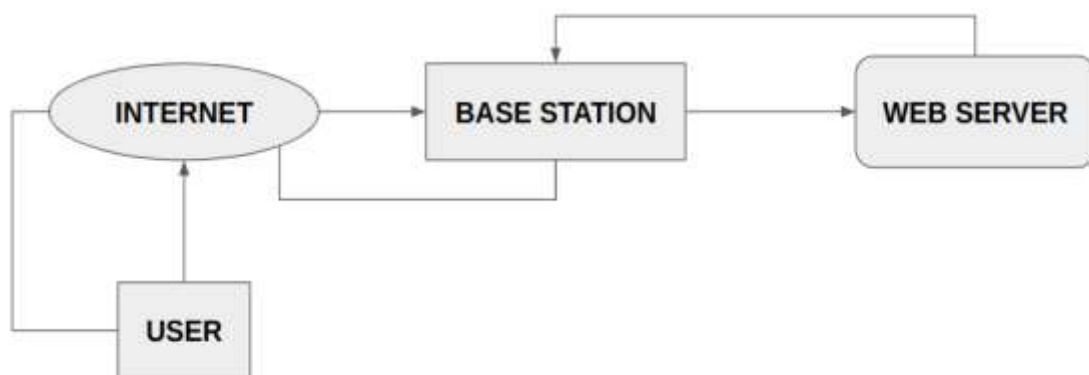
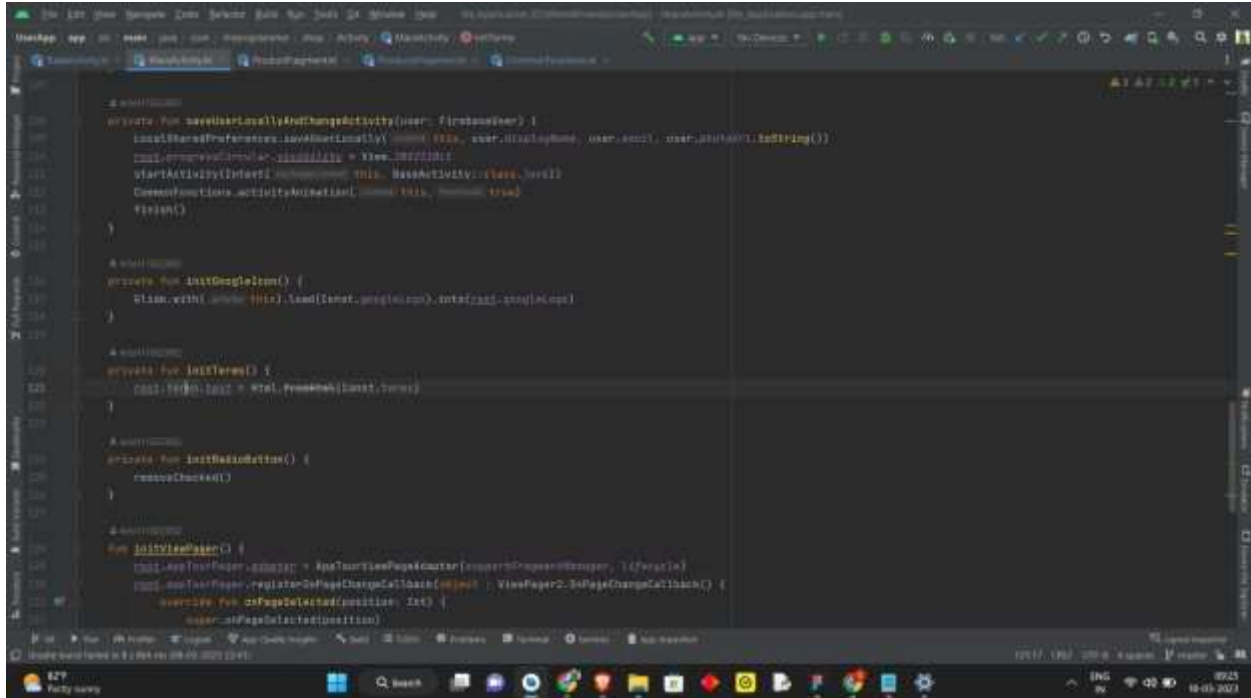


Fig:1

DIAGRAMS

The Unified Modeling Language is a standard language for business modeling, non-software systems, and defining, visualizing, building, and documenting the artifacts of software systems. The UML is an amalgamation of the top engineering techniques that have been effective in simulating big, complex systems.



```
private void saveUserLocallyAndChangeActivity(user: FirebaseUser) {
    LocalSharedPreferences.saveAndCommit(this, user.displayName, user.email, user.photoURL.toString());
    LocalSharedPreferences.putString("username", username);
    startActivity(Intent(this, MainActivity.class));
    ContextCompat.startActivity(this, Intent(this, MainActivity.class), null);
}

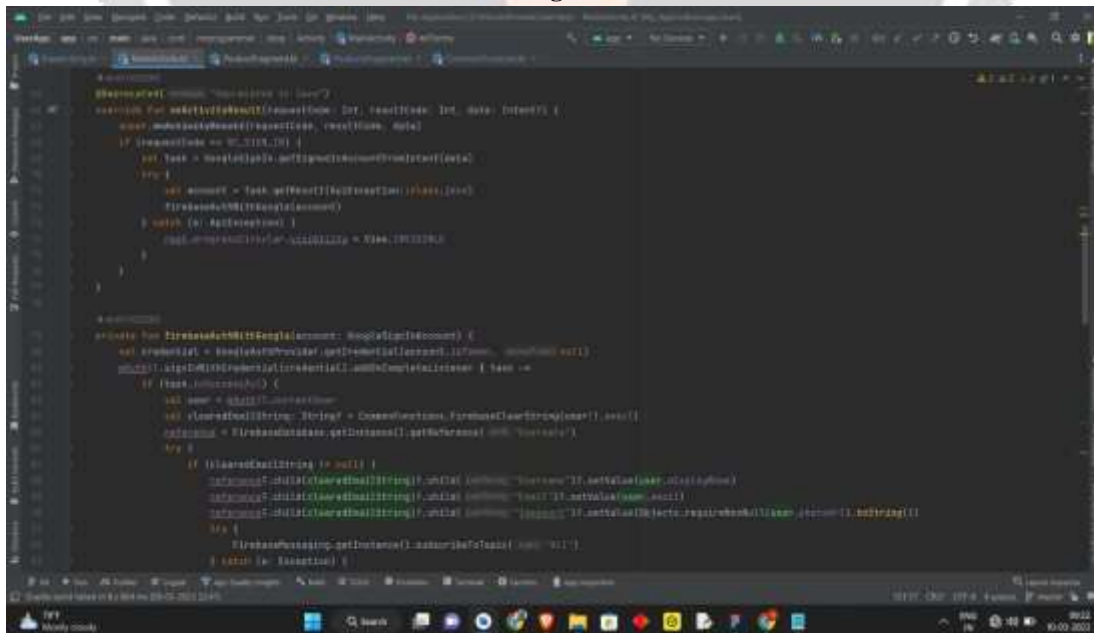
private void loginGoogleItem() {
    signInWith(provider: this).load(isnot.google.com).startActivityForResult()
}

private void loginTerms() {
    signInWith(provider: this, Provider.LoginTerms)
}

private void loginRadioButtons() {
    radioButtons()
}

private void loginViewPager() {
    viewPager.setAdapter(adapter: new ViewPager2Adapter(this, viewPager));
    viewPager.registerOnPageChangeListener(new ViewPager2.OnPageChangeListener() {
        override fun onPageSelected(position: Int) {
            viewPager.setCurrentItem(position)
        }
    });
}
```

Fig:2



```
private void registerUser() {
    FirebaseAuth.getInstance().createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) {
            if (task.isSuccessful) {
                startActivity(Intent(this, MainActivity.class));
            } else {
                Toast.makeText(this, "Registration failed", Toast.LENGTH_SHORT).show();
            }
        }
}

private void loginGoogleItem() {
    FirebaseAuth.getInstance().signInWithCredential(credential)
        .addOnCompleteListener(this) {
            if (task.isSuccessful) {
                startActivity(Intent(this, MainActivity.class));
            } else {
                Toast.makeText(this, "Login failed", Toast.LENGTH_SHORT).show();
            }
        }
}
```

Fig:3

3.5 Use Case Diagram:

A use case diagram depicts the different actors that interact with the system and the various use cases or scenarios that the app supports.

The use case include:

Screen Base Station: Permits the framework to screen the situation with a base station and its sensors.

Screen Sensor: Permits the framework to screen the readings and status of a sensor on a base station.

Ready Client: Permits the framework to caution a client or head when a base station or sensor has an issue.

View Verifiable Information: Permits the client or head to see verifiable information on the readings and status of a base station or sensor.

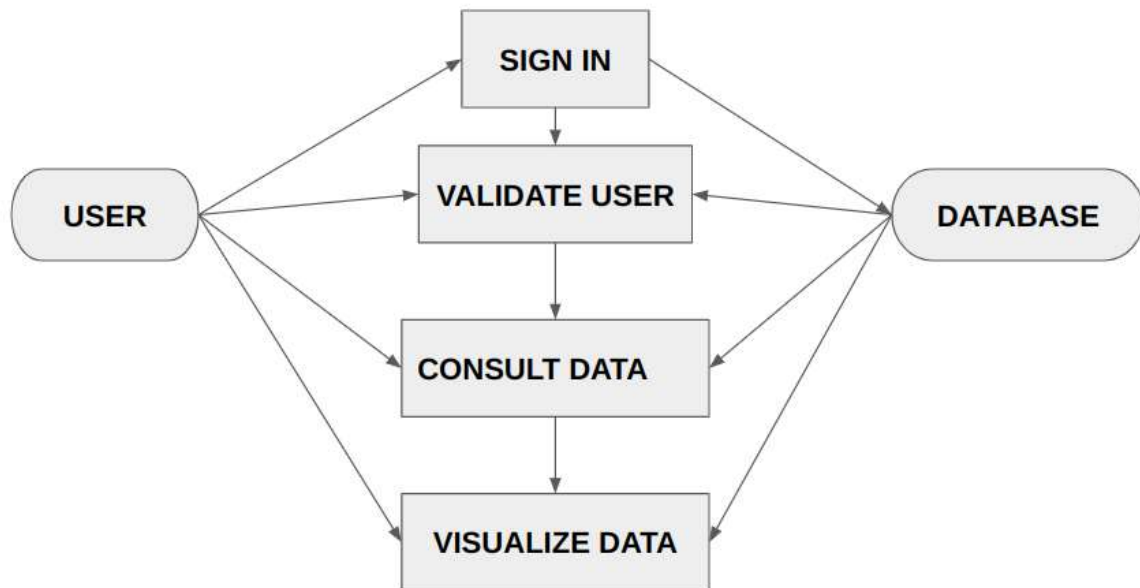


Fig:4

Snippet:

- ```

| Base Station App |

|-View Base Station |
|-View Sensor |
|-Update Sensor Reading |
|-Update Sensor Status |
|-Update Base Station |
|-Add Base Station |
|-Remove Base Station |
|-Add Sensor to Base Station|

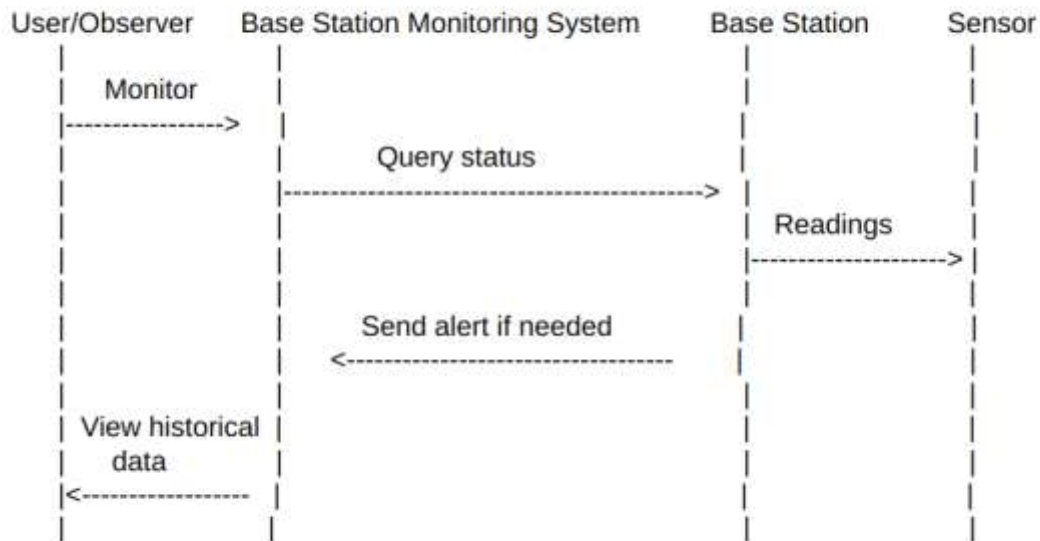
```



|-Remove Sensor from Base Station|  
 |-Login |  
 |-Logout |  
 -----

**3.6 Sequence Diagram:**

A sequence diagram depicts the sequence of events or messages exchanged between different objects or components in the system. In the case of a base station monitoring app, a sequence diagram could depict the sequence of events involved in triggering an alarm.



**Fig:5**

**3.7 Activity Diagram:**

An activity diagram depicts the different activities or tasks that need to be performed to achieve a specific goal or objective. In the case of a base station monitoring app, an activity diagram could depict the different tasks involved in setting up the app or adding a new sensor to the system.

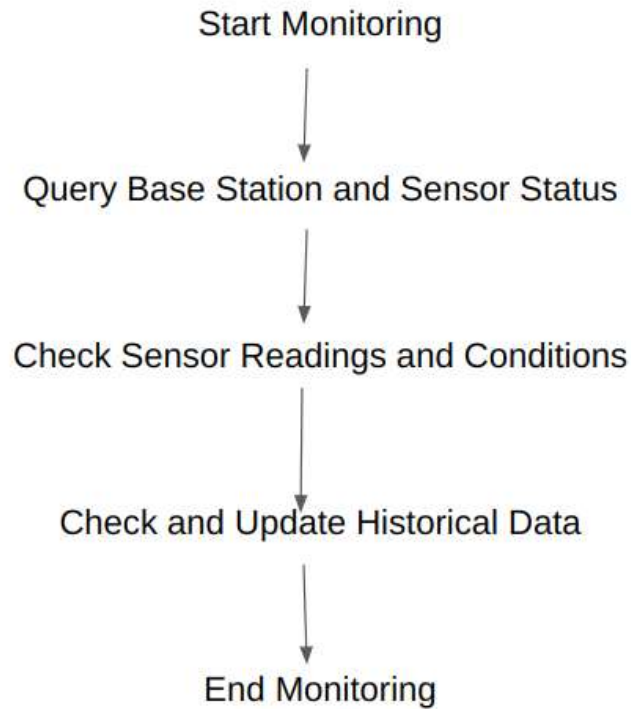


Fig: 6

## 4.IMPLEMENTATION

### 4.1 Planning and design:

The implementation is to plan and design the app. This involves identifying the key features and requirements of the app, selecting the appropriate technology tools and frameworks, and creating a detailed project plan.

### 4.2 Use cases:

Based on the requirements identified, use cases can be defined to outline the key scenarios in which the app will be used. This helps to ensure that the app meets the needs of its intended users.

**Login and Verification:** A client signs into the application utilizing their certifications or through web-based entertainment confirmation.

**View BaseStation:** A client can see a rundown of all base stations in the framework. The client can channel base stations by area, status, or different measures.

**View BaseStation Details:** A client can see subtleties of a particular base station, including its area, status, and sensor information. The client can see verifiable information for the base station and its sensors.

**For testing:** Use Cucumber BDD for testing the use case and tested in IntelliJ idea

Feature: Base Station Monitoring

Scenario: Login and View Base Station List

Given I am on the login page

When I enter valid login credentials

Then I should be redirected to the dashboard

And I should see a list of base stations

Scenario: View Base Station Details

Given I am on the dashboard

When I click on a base station in the list

Then I should see the details of the base station

And I should see the real-time sensor data for the base station

Fig:7

#### 4.3 Frameworks:

**React:** A JavaScript library for building UIs. Respond can be utilized with different libraries and structures to construct a full-stack web application.

**Node.js:** A server-side JavaScript runtime that can be utilized to fabricate versatile and elite execution web applications. Node.js is frequently utilized with the Express system to assemble web APIs.

#### 4.4 User interface designs:

The user interface (UI) design of the app is critical in ensuring that it is user-friendly and easy to navigate. The UI design should be based on the requirements and use cases identified in the previous steps.

**Clear Navigation:** The app should have a clear navigation menu that allows users to quickly access the different sections of the app.

**Dashboard:** A dashboard can give clients an outline of the base station's status, including measurements like sign strength, information use, and gadget temperature.

**Alerts and Notifications:** The application ought to give continuous cautions and notices to clients when a base station goes disconnected or encounters an issue.

**Map View:** A guide view can be utilized to show the area of the base stations and their status. This can be helpful for checking countless base stations.

**Data Visualization:** The application ought to utilize information perception methods to show information in an unmistakable and natural manner. Outlines, charts, and intensity guides can be utilized to show information like sign strength, information utilization, and gadget temperature.

## 5. TESTING

### 5.1 Mobile testing:

Mobile phones are becoming the most important part of daily life. In today's world nearly 5.48 billion mobile users are there. The web traffic by mobile users has slightly increased in the past decade and 60% of web page views from mobile devices is high compared to desktop. So, mobile phone usage will increase more in the future. And, there are many mobile applications that are developed in a short span. Nowadays people are using mobile phones for browsing and opening web pages and handling mobile applications on mobiles rather than computers. This creates a demand among mobile application developers who want to develop the best application across different platforms quickly. Growth always brings challenges. Due to this enormous growth, a lot of mobile application industries try to deliver their product in good quality and easy to user's. Mobile apps have a lot of challenges. One of them is supporting multiple platforms and devices. If it's not supported in all devices or platforms it will fail in user experience and the tough challenge is testing the mobile application. Testing is an important part in software development. It will improve the quality, performance, UI and reliability of the application. In testing there are two forms. First, manual testing is the tester acts as an end user and test the application with the help of test cases on hands without any automation tools and it is difficult to identify the edge cases and time-consuming. Another one is automated testing. This technique uses some automation testing tools to execute test cases and compare the actual and expected results. Automation testing is becoming the new trend to develop a high quality application in a short time. The major benefit of automation is reducing the testing effort and less time-consuming. The software development companies use automation to improve their testing efficiency, quality and delivery.

### **5.2 Software testing vs Manual testing:**

Software testing can be classified into two categories mainly as manual testing, and automated testing: In manual testing, no programming is done to write use cases for which give hidden errors and points to missing information and edge cases while in automation testing testers can program the test script to bring out hidden errors and points. Manual testing requires huge human resources and it takes a long time to test the application. It is also possible to make human errors in manual testing. This type of testing becomes slow and time consuming. With the arrival of an increasing number of mobile applications, Automation provides us a solution, by performing the same operation for multiple alliterations and they run the same action for each time and these eliminating risk of errors as this process makes use of scripts and allowing us to test more number of possibilities to reduce further errors. Both approaches are good together. In automated testing used to prevent new errors in the already tested working modules and it can test a large number of use cases in minimum time, whereas manual testing the tester use knowledge to target testing of the parts of the system that are assumed to be more edge error cases/error-prone and is better to finding new and unexpected errors.

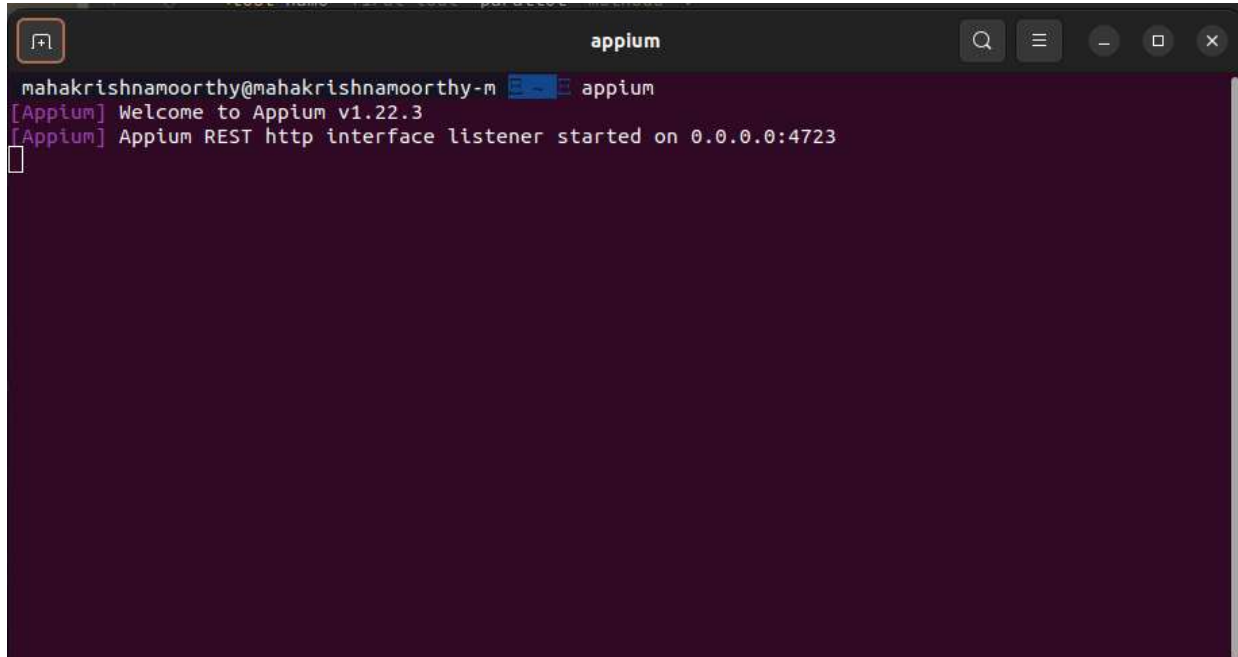
### **5.3 Appium:**

Appium is an open source testing tool used to automate the mobile applications that helps to deliver powerful features. It allows native, web and hybrid application testing and supports automation tests on emulator and simulator as well as both emulator and simulator. We can easily test multiple devices in parallel in automation. It has no dependency on real mobile devices because there is a framework that translates Selenium webdriver commands into UIAutomation(Android) commands depending on device type. It offers cross-platform mobile automation testing, which means that it allows the same test to be run on multiple platforms. Several devices can be easily tested by Appium in parallel. It supports automation using different programming languages like C, JAVA, Python and PHP, etc. So, that user can use any programming language to write automation scripts. It is comfortable for users.

### **5.4 Appium works on android:**

Appium Client libraries convert the user written commands to the API requests. These API requests are sent to the Appium Server using Mobile JSON Wire Protocol. The Appium server forwards these requests to the required

android device(virtual device). These commands are interpreted by bootstrap.jar which converts them into a simple format that mobile devices can understand in UIAutomator format. The UIAutomator commands are now performed on the virtual devices. Device then reverts the outcome of the performed command to the Appium server via bootstrap.jar. The server forwards this response to the client.



```

mahakrishnamoorthy@mahakrishnamoorthy-m appium
[Appium] Welcome to Appium v1.22.3
[Appium] Appium REST http interface listener started on 0.0.0.0:4723

```

Fig: 8

## 6. CONCLUSION

base station monitoring app provides a modern and efficient approach to monitoring and managing base stations. The proposed system offers a range of features, including real-time monitoring, alerts and notifications, performance analysis, remote access, and data visualization, all of which can help operators to quickly identify and resolve issues. By using modern technology tools and frameworks such as Android Studio, Realm database, Retrofit library, and Postman tool, the app can be developed efficiently and deployed on a cloud-based platform for optimal performance. Moreover, the app is designed to be accessible to a wide range of operators, regardless of their technical expertise. Overall, a base station monitoring app can help to improve the efficiency and performance of base stations, resulting in cost savings and improved network performance for organizations. As such, it represents an important and valuable tool for modern telecommunications operations.

## 7. REFERENCES

- [1] Alizadeh, M., & Bahrami, H. (2019). Base station automation: a comprehensive survey. *IEEE Access*, 7, 25805-25826.
- [2] Nitsche, T., & Schellenberg, R. (2020). Intelligent base station automation for 5G networks. *IEEE Transactions on Network and Service Management*, 17(3), 1216-1230.



[3] Xu, Y., Zhang, Y., Li, X., & Wu, Q. (2019). An intelligent base station automation scheme for 5G networks. *IEEE Wireless Communications*, 26(1), 20-27.

[4] Zhang, L., Han, T., & Jiang, M. (2019). A survey on base station automation for 5G networks. *IEEE Communications Magazine*, 57(6), 86-92.

[5] Zhang, K., Zhu, M., & Yang, Z. (2020). A survey on the development of base station automation in 5G networks. *IEEE Transactions on Industrial Informatics*, 16(6), 3956-3967.

