Detecting COVID-19-infected persons in the surrounding areas using Armor Application

Saisindhu Jammula, Anil babu Tota, Anivi Reddy Yarabolu, Bharadwaj Vemulapally, Sahith Chowdary Narukulla, Sai Nikhil Reddy Tamma

Seidenberg School of Computer Science and Information Systems

Pace University, New York, NY, USA

***Abstract*-- Our product is focused on COVID-19 safety Protocols. It is designed to ensure that people are adhering to social distancing guidelines. Ultimately, it can help make people and the community safer from the virus. This is a tracking app that uses a smartphone's GPS and Bluetooth features to track COVID-19 cases. Bluetooth technology enables users to assess their risk of being exposed to the COVID-19 virus if they have been within six feet of a confirmed case. It is also able to detect specific locations that are deemed 'infected areas through location data analysis, and crowdsourcing. this data so it is easily accessible for decision-making purposes. The idea behind the app was to make people aware of their risk and give them the tools to protect themselves from contracting the virus. We developed a product that would be a cost-effective measure for protecting profits and human life. The idea for our product is based on the idea of "control" to prevent COVID-19 from spreading. It would be an app that uses Bluetooth and GPS capabilities to track COVID-19 cases and would also provide alerts in case it detects COVID-19 in an area. This product was designed to be inexpensive, and effective, providing information necessary for the public's awareness of where the virus is being spread. We designed this app as an incentive to prevent someone from spreading the virus as they can see how close they are getting to people who have contacted it.**

Keywords: depression, machine learning, natural language processing, facial expression, early detection.

I. Introduction

In Wuhan, China, in December, the 2019 Coronavirus disease (COVID-19) was discovered. Following that, the virus spread rapidly around the world. As of March 2022, records show that COVID-19 had infected about 520 million people and that more than 4 million people had died because of the illness. The COVID-19 respiratory syndrome is an acute respiratory illness characterized by a persistent cough, a high-grade fever, and breathlessness. The rapid spread of COVID-19 caused the World Health Organization (WHO) to classify it as a pandemic. Numerous healthcare systems throughout the world have been affected by and put to the test because of the enormous number of cases that want for medical attention.

Overall, the COVID-19 pandemic has posed several difficulties for patients and healthcare institutions. Since it was found that the virus is more active in social settings, public authorities and governments implemented lockdowns and quarantines to stop the infection's spread among the populace. To stop the infection from spreading, the quarantine, stay-at-home, and social isolation policies were put into place. According to research, many healthcare systems were susceptible to such a pandemic because it exacerbated already-existing problems such staff shortages, a lack of personal protective equipment (PPE), the capacity of intensive care units (ICUs), and hospital bed shortages. The current healthcare systems face new challenges as a result of this unique circumstance.

According to health officials, human interaction is the primary factor contributing to the pandemic's spread. To prevent the spreading, it is therefore necessary to maintain the necessary spacing with some effort. In addition, implementing a Contact Tracing technique that is automated and protects privacy might be helpful to stop the spread.

Modern cell phones are not only used for communication; they can also detect movement, proximity, direction, and other environmental cues. At the same time, nowadays practically everybody uses a smartphone. Considering these realities, we suggest a method for leveraging smartphones to foster social estrangement. We propose a strategy for using smartphones to promote social alienation considering these facts.

Bluetooth Low Energy, more commonly known as BLE, is a technology created as an Android application that is used for proximity detection. It will trade Bluetooth Low Energy (BLE) beacons with neighboring devices, calculate the separation between them, and alert the user if the closeness is too close. Using a pre-trained deep neural network, the direct distance between two devices is calculated based on Radio Signal Strength Indication values. The users' devices retain the proximity data as transformed information that cannot be used to immediately re-identify the contacts.

Safety Protocols for COVID-19 are the focus of our product. It is intended to make sure that people follow the

rules for social distancing. In the end, it may contribute to increased community and individual viral safety. This tracking application tracks the COVID-19 case using the GPS and Bluetooth capabilities of a smartphone. Users can evaluate their risk of contracting the COVID-19 virus if they have been within six feet of a confirmed case thanks to Bluetooth technology.

Through crowdsourcing and the analysis of location data, it is also able to identify areas that are infected areas. This information is readily available for use in making decisions.

The purpose of the app was to inform users of their risk and equip them with the means of preventing virus exposure. We created a device that would serve as an economical safeguard for both financial gain and human lives. Our product's concept is centered on the notion of "control" to stop the spread of COVID-19. It would be an app that tracks COVID-19 instances using Bluetooth and GPS technology and sends warnings if it finds COVID-19 in a certain location. The purpose of this device was to tell the public about the locations where the virus is spreading while being affordable and efficient.

Social Distancing has increased in frequency during the past two years as a result of the Covid-19 pandemic outbreak. Interpersonal communication is the main factor driving the pandemic's spread, said worldwide health officials. It is therefore vital to maintain the required spacing with some effort in order to prevent the spreading. To further stop infection, creating an automated Contact Detection phase with privacy protection would be helpful. Smartphones of today are used for more than simply communication; they are also capable of detecting a wide range of environmental cues (such as movement, proximity, direction, etc.). Additionally, almost everyone uses a smartphone these days. Think about these details. A mobile app that may alert users when a person is approaching from a closer distance than is recommended solves the problem.

II. LITERATURE REVIEW

Bluetooth Low Energy (BLE) stands as a standardized protocol formulated by the Bluetooth Special Interest Group (Bluetooth SIG) to facilitate wireless personal area networks. Its fundamental purpose revolves around the endorsement of Internet of Things (IoT) applications, leveraging cost-effective and energy-efficient devices. In contemporary scenarios, BLE finds extensive utility across various practical applications. For instance, AJ. Aljohani (2021) has developed a model dedicated to discerning COVID-19 risks within university interiors and classroom settings. Additionally, M.J. Keeling has conducted a study delving into the effectiveness of contact tracing for containing the 2019 coronavirus.

Presently, the widespread adoption of BLE-enabled medical and personal mobile devices is significantly contributing to enhanced interactions between healthcare institutions and their patients throughout the entire care process—from pre-visit, during the consultation, to post-visit follow-ups. Noteworthy statistics from a 2013 eClinicalWorks Survey highlight the consensus among healthcare professionals, with a staggering 93% expressing the belief that mobile applications have the potential to enhance both patient experience and overall outcomes in the realm of healthcare. This proliferation of BLE technology showcases its pivotal role in advancing connectivity and improving the healthcare landscape.

While Bluetooth Low Energy (BLE) has emerged as the predominant communication method in medical devices and sensors, the paramount significance of security and privacy in healthcare technologies cannot be overstated, particularly given their potential impact on morbidity. As the prevalence and utilization of medical devices and sensors utilizing Bluetooth Low Energy continue to rise, there is an escalating imperative to scrutinize the security and privacy facets of healthcare technology.

Consequently, there exists a growing necessity for a comprehensive and rigorous security analysis, specifically tailored to assess the security and privacy implications inherent in medical devices and sensors employing Bluetooth Low Energy. The heightened prevalence of these technologies underscores the urgency to implement a more robust evaluation framework that can thoroughly examine the intricate security and privacy dimensions associated with Bluetooth Low Energy-enabled healthcare devices and sensors. Such a nuanced analysis is vital to fortify the integrity of healthcare systems and instill confidence in users and stakeholders alike regarding the protection of sensitive health-related information.

III. PROJECT REQUIREMENTS

The primary goal of this project is to design and implement a mobile application named "Armor," which aims to detect potential COVID-19-infected individuals in the vicinity through Bluetooth coverage. The application subsequently notifies users of any potential risk in their surroundings.

1. Functional Requirements:

1. User Authentication:
   - The application facilitates users in creating accounts and logging in securely using their credentials, ensuring the protected storage of user data.
2. Services:
   - Utilizing BLE service and Cloud Firestore, the app manages Bluetooth beacons and securely stores users' data for efficient functionality.
3. Mobile OS:
   - The application offers compatibility with both Android and iOS mobile operating systems, ensuring widespread accessibility.
4. Users Input:

- Users are empowered to update their COVID-19-infected status within the application, contributing to a comprehensive database for risk assessment.

5. BLE Service:

- The app employs BLE service to exchange anonymous beacons containing unique keys. It stores local anonymous data about encountered individuals (contacts) and assesses the proximity of nearby devices using Bluetooth Low Energy (BLE) Received Signal Strength Indication (RSSI) values.

6. User Alerts:

- Armor diligently alerts users in the event of proximity to an infected person. The application regularly downloads the database of infected users to cross-reference and identify matching keys in the local database, ensuring timely and accurate risk notifications.

This set of functional requirements outlines the key features and capabilities that the Armor application must possess to fulfill its objective effectively. The emphasis on user authentication, secure data management, cross-platform compatibility, user input, BLE service utilization, and proactive user alerts ensures a robust and comprehensive approach towards the app's goal of identifying and notifying users about potential COVID-19 risks in their vicinity.

1) Non-functional Requirements

The non-functional requirements for the Armor app are as follows:

1. Performance:
- The app exhibits swift performance, ensuring that the initial screen loads within a maximum duration of 3 seconds, thereby providing users with a seamless and responsive experience.

2. Efficiency:
- Armor efficiently scans the nearby area for potential COVID-19-infected individuals, promptly alerting the user to ensure timely and effective risk awareness.

3. Security:
- Emphasizing robust security measures, all app data is meticulously secured and encrypted. This comprehensive protection extends to safeguarding against external threats and internal attacks, ensuring the integrity and confidentiality of user information.

4. Compatibility:
- The application is designed to be compatible with both Android and iOS devices, ensuring smooth functionality on the latest operating systems of each platform. This compatibility enhances the app's accessibility across a broad spectrum of devices.

5. Scalability:
- Armor possesses the ability to adapt to increased usage over time, demonstrating scalability in handling growing user numbers and expanding datasets. This ensures the app's continued effectiveness and performance as its user base and data load increase.

6. Screen Adaption:
- The application boasts screen adaptation capabilities, dynamically adjusting its layout to accommodate different screen sizes. Furthermore, it incorporates automatic adjustments for font size and image rendering, enhancing the user experience across various devices and screen dimensions.

These non-functional requirements establish essential benchmarks for the Armor app, encompassing swift performance, operational efficiency, robust security protocols, cross-platform compatibility, scalability to accommodate increased usage, and adaptive screen rendering for diverse devices. Adhering to these requirements ensures that the application not only meets its functional objectives but also delivers a secure, efficient, and user-friendly experience.

2) Hardware Requirements

1. IOS or Android devices to run the application.
2. Any OS that runs the browsers.

3) Software Requirements

1. Visual Studio, Microsoft Excel, Microsoft Word, Google Chrome, Safari, Microsoft PowerPoint, Git Hub, and JIRA software.

4) Constraints

The constraints for the Armor app are outlined as follows:

1. Performance Constraints:

- The application is expected to operate seamlessly on any device, ensuring a bug-free and crash-free experience. Simultaneously, it should exhibit optimized performance by consuming minimal memory and battery resources, enhancing user satisfaction and device efficiency.

2. Time Constraints:

- Adherence to a predetermined timeframe is imperative to meet project deadlines. The app must be developed, tested, and ready for deployment within the specified time frame, emphasizing the importance of efficient project management and timely execution.

3. Cost Constraint:

- The creation and maintenance costs of the Armor app must be kept to a minimum, aligning with the constraints of the given budget. This necessitates prudent resource allocation, cost-effective development practices, and ongoing financial considerations throughout the project's lifecycle.

4. Scope Constraints:

- To ensure the efficient development of the app, continuous monitoring of the project scope is essential. Any changes to the scope must be meticulously documented to maintain clarity, manage expectations, and facilitate effective project communication. This helps in controlling project scope creep and ensuring alignment with project objectives.

5. Device Constraints:

- The app should be developed using a responsive design approach, allowing it to run seamlessly on any device, irrespective of screen sizes and pixels. This adaptability enhances the user experience by ensuring consistent performance and usability across a diverse range of devices.

These constraints underscore the critical considerations and limitations that must be acknowledged and managed during the development and implementation of the Armor app. Balancing optimal performance, adherence to project timelines, cost-effectiveness, scope management, and device compatibility are pivotal in achieving the project's success within the defined constraints.

IV. SYSTEM DIAGRAM

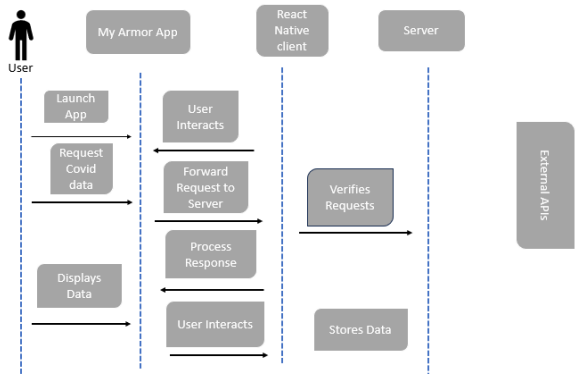

Figure 4.1 Sequence Diagram

A sequence diagram is one which illustrates the interactions between the objects in a sequence in which they occur. Here it shows the interaction between the user and the armor app. The user interacts with the application by using the User Interface. The main service manages the flow of the application and coordinates with other services. BLE services manage beacon advertising and scans for nearby devices. Machine Learning Service classifies the proximity of nearby devices using RSSI signal strength.

Notification service is designed to send out alerts or notifications to users or administrators, and the type of notification sent depends on specific events or occurrences within the system. For example, if there is an error, a new message, or an important update, the notification service will generate and deliver the appropriate alert to the relevant recipients. Database service is responsible for collecting and storing data that is generated or sent by devices when they connect to the system. Here the data is stored anonymously. Anonymizing the data helps protect user privacy and security. Background tasks, which are operations that run in the background of your application without requiring user interaction, are managed using Firebase Work Manager. Firebase Work Manager is a tool or service provided by Google that helps schedule and execute these background tasks efficiently. These tasks can include things like data synchronization, updates, or other automated processes that don't need to be in the foreground.
When certain events or occurrences impact users, their anonymized data is stored in Firebase Cloud Storage. Firebase Cloud Storage is a cloud-based storage service provided by Google. It's commonly used to store various types of data, such as user-generated content or application data. The term "anonymized data" implies that any data stored here has been stripped of information that could identify individual users, ensuring their privacy while still allowing for analysis or other purposes.

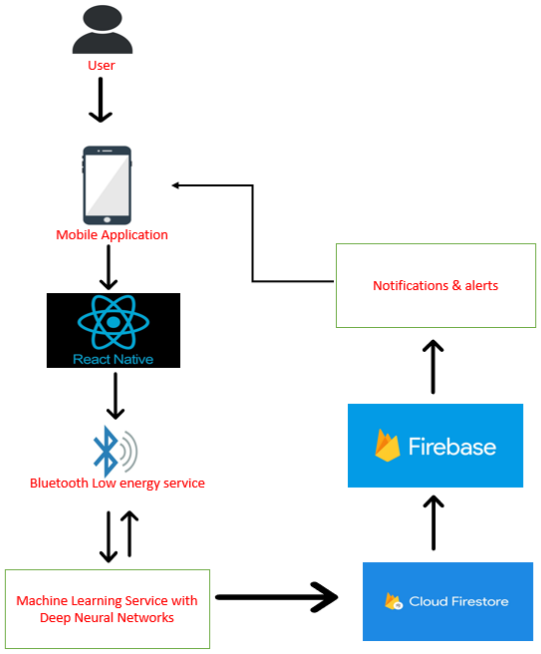

Figure 4.2 Conceptual Architecture Diagram

The high-level conceptual design of a system or software application is represented by the Conceptual Architecture Diagram above. It gives a general overview of the main parts

of the system and shows how they work together to accomplish the required functionality.

Users install the Armor mobile application, which is being built using the React native framework. We are using Bluetooth Low energy service (BLE) which helps to scan the nearby devices which have unique keys. It scans the devices that will be in the range. These keys are exchanged between the BLE and the Machine Learning Service with Deep Neural Networks which are further processed and stored in the database.

Whole data, like who is infected with the virus, and the people the user met, is stored in the Cloud Fire store and the Firebase is the local database from which the system frequently retrieves data. According to the situation, all the notifications and alerts will be sent to the mobile application and the user can follow the steps accordingly.

V. PRODUCT RESULTS

The Bluetooth technology, a pivotal element in the application's operation, has experienced remarkable enhancements. Significant strides have been made to streamline the account creation process, ensuring user convenience by offering straightforward sign-up options through email addresses. Upholding the utmost regard for user privacy, all user details are treated with the highest confidentiality standards.

Moreover, an enhanced user feedback mechanism has been seamlessly integrated to augment the application's user-friendliness. A notable development is the empowerment of users to manage the Bluetooth function directly within the application. This newfound capability grants users the flexibility to effortlessly enable or disable Bluetooth according to their specific needs, contributing to an enhanced user experience.

Furthermore, the application has now instituted timely notifications for COVID-19 alerts and updates, providing users with access to relevant information promptly. This strategic integration ensures that users stay well-informed about crucial developments related to the ongoing pandemic. Recognizing the paramount importance of privacy, users are now granted the option to render their devices invisible to others, thereby bolstering their control over gadget visibility and enhancing overall privacy measures.

In essence, the application has undergone substantial refinements, not only in its technological core, such as Bluetooth functionality but also in user-centric aspects, including simplified account creation, robust privacy measures, and an enriched user feedback mechanism. These improvements collectively contribute to a more sophisticated and user-friendly application experience.

Fig 5.1 Home page of the Armor application

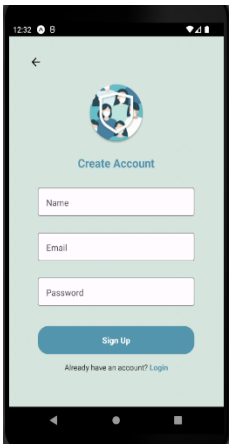

Fig 5.2 Registration page of Armor application

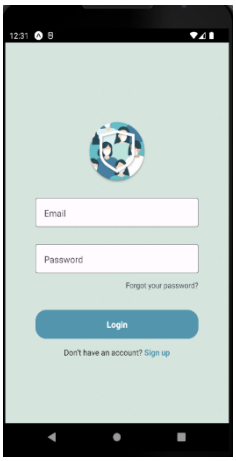

Fig 5.3 Login Page of the Armor application

Fig 5.4 Welcome page of the Armor application

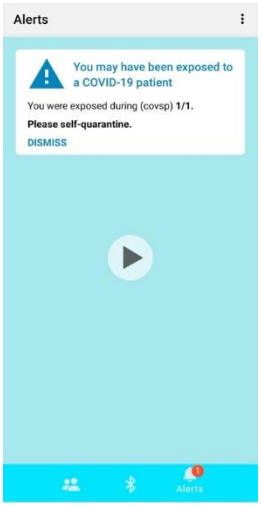

Fig 5.5 Alerts Page of the Armor application

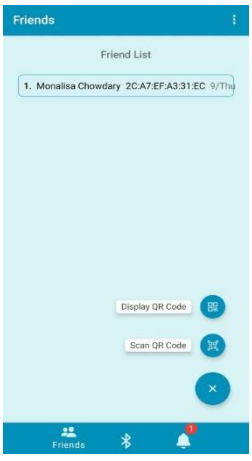

Fig 5.6 Friend List Page of the Armor application

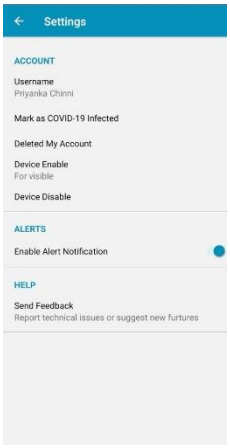

Fig 5.7 Settings page of Armor application

Fig 5.8 Profile edit page of Armor application.

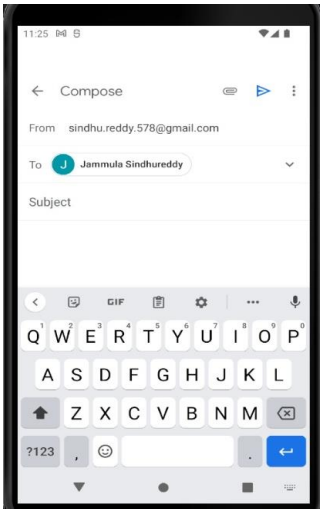

Fig 5.9 Feedback Page

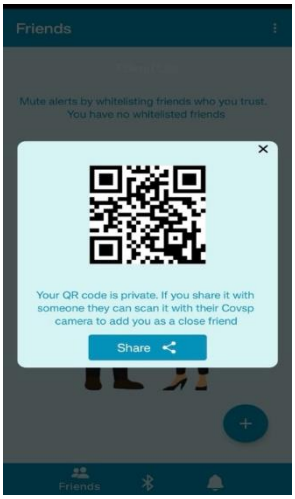

Fig 5.10 Display QR Code page

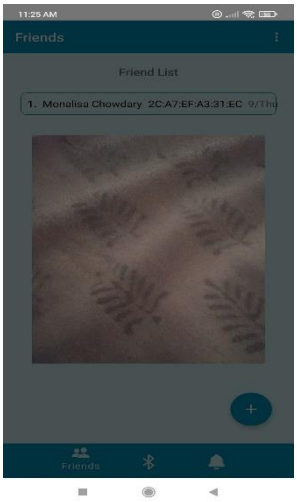

Fig 5.11 QR Code scanner page

Fig 5.12 Bluetooth permission page

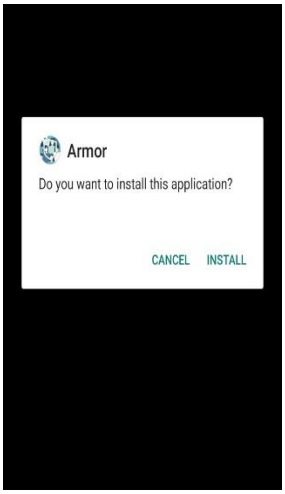

Fig 5.13 Installation Page

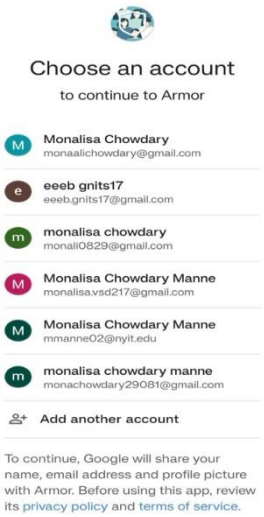

Fig 5.14 Gmail login Page

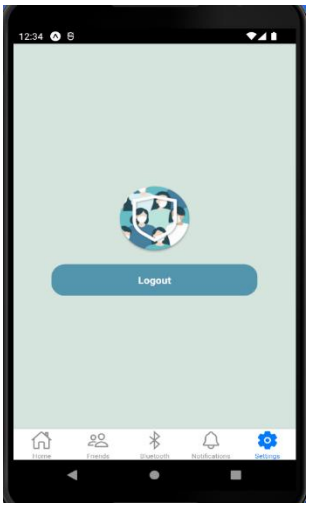

Fig 5.15 Logout page

VI. Conclusion

Armor Application uses Bluetooth low-energy signals and tokens to identify COVID-19-infected persons in the surrounding areas. Social distancing notifications and digital contact tracing are two aspects of a smartphone application related to COVID-19.

The application uses Bluetooth signals to detect other users when they are within a set distance typically 6 feet or 2 meters and informs the user to maintain their distance. The goal is to help users stay away from close contact and prevent the transmission of COVID-19.

Users may freely update the app with information about their COVID-19 status, which can then be used to alert other users who could have encountered the infected individual. To stop COVID-19 from spreading, Bluetooth signals are used, which can establish whether two users are sufficiently close to one another for a certain period.

All the registered users now can mark themselves as COVID-19-infected persons. Users will be able to update their COVID-19 status whenever they are affected by the virus. This feature enhances and encourages responsible behavior. The registered users will also be able to share their application QR code with friends which enables seamless integration into their friend lists. With this user can easily connect with their friends on this platform which in turn promotes social engagement. As per our commitment to user privacy and security, we also implemented a feature that allows users to whitelist their trusted friends. With this, the users can enhance their control over who can access their data and interact with them. In the same way, users can also be able to remove friends from their whitelist which provides the flexibility to adjust their trusted connections over time.

Users can also delete their accounts by visiting the settings page, ensuring that they have the autonomy to manage their presence on this platform. This feature is strictly incorporated for the user's convenience and control.

Making it simpler to locate and isolate possible instances is the aim. It is critical to keep in mind that user uptake and engagement, Bluetooth signal accuracy, and the notification system may all influence how well consumers will react to these features.

VII. History of Updates

We updated our conceptual architecture diagram as per the professor’s feedback from the previous sprint. conceptual architecture diagram gives a general overview of the main parts of the system and shows how they work together to accomplish the required functionality.
The Armor mobile application, which was created with the React Native framework, gets installed by users. The Armor mobile application, which was created with the React Native framework, gets installed by users. To scan neighboring devices with unique keys, we are employing Bluetooth Low Energy (BLE). The devices that will be in the range are scanned. These keys are sent back and forth between the BLE and the Deep Neural Network-based Machine Learning Service before being further analyzed and saved in the database.
The Cloud Fire store houses all the data, including who is infected with the virus and the individuals the user met, and Firebase is the local database from which the system routinely requests information. All notifications and alerts will be delivered to the mobile application in accordance with the scenario, and the user may then take the appropriate action.

VIII. References

[1] J. Hopkins, “Digital Contact Tracing for Pandemic Response,” doi: 10.1353/book.75831.

[2] “Using Bluetooth RSSI Values to Classify Proximity with Deep Neural Networks for COVID-19 Contact Tracing – Young Scientists Journal.” https://ysjournal.com/using-Bluetooth-RSSI-values-to-classify-proximity-with-deep-neural-networks-for-covid-19-contact- tracing/ (accessed Feb. 07, 2022)A. Ashraf, T. S. Gunawan, F. D. A. Rahman, M. Kartiwi, N. Ismail and Ulfiah, "A Summarization of the Visual Depression Databases for Depression Detection," *2020 6th International Conference on Wireless and Telematics (ICWT)*, Yogyakarta, Indonesia, 2020, pp. 1-6, doi: 10.1109/ICWT50448.2020.9243625.\

[3] Evaluating the dynamics of Bluetooth low energy based COVID-19 risk estimation for educational institutes AJ Aljohani, J Shuja, W Alasmary, A Alashaikh - Sensors, 2021 - mdpi.com

[4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9320375/

[5] https://www.oreilly.com/library/view/getting-started-with/9781491900550/

[6] Efficacy of contact tracing for the containment of the 2019 novel coronavirus (COVID-19) Matt J Keeling, corresponding author1 T Deirdre Hollingsworth,2 and Jonathan M Read3,4