

ANDROID BASED MOBILE PETITION FOR TRACKING COVID-19 VACCINATION CENTERS USING FIREBASE AND MONGODB

Dilip Chaudhary¹, Saurabh Sangunwedhe², Pranil Pandit³,
Harshal Gaikwad⁴, Ashish Yadav⁵

Department of Computer Engineering Bharat College of Engineering Badlapur (W), Thane-421503

¹*Ms. Sneha Dalvi*

Assistant professor, Bharat College of Engineering, Maharashtra, India

Submitted: 01-11-2021

Revised: 06-11-2021

Accepted: 09-11-2021

ABSTRACT:- Health, sickness and medical processes are vital to everyone. However, we realize that a lot of people are affected by the COVID-19 pandemic, we heard greater stories from social media and information TV, all of the states and countries round the sector are infected by COVID-19 disease. The virus is spread through the mouth, nose, or liquid of cough, sneeze, and breath. To reduce the impact of spreading, the authorities have announced the lockdown. People require vaccination in such circumstances. If they can't move out, how will they find the vaccine for themselves? To overcome infection with COVID, people want to be vaccinated. Our paper on the whole focuses on providing information to users via the use of an Android application that we designed to tell people about the Covid-19 vaccination location. The reason for the application is to deliver information regarding the vaccination center, date, charges, time, age limit, etc.

KEYWORDS:- Covid Vaccination System; Android Studio Application; Mongodb Database; Firebase Authentication system; Service-Oriented Architecture.

I. INTRODUCTION

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus. Most people infected with the virus will experience mild to moderate respiratory illness and recover without requiring special treatment. However, some will become seriously ill and require medical attention. Older people and those with underlying medical conditions like cardiovascular disease, diabetes, chronic respiratory

disease, or cancer are more likely to develop serious illness. Anyone can get sick with COVID-19 and become seriously ill or die at any age. The best way to prevent and slow down transmission is to be well informed about the disease and how the virus spreads. Protect yourself and others from infection by staying at least 1 metre apart from others, wearing a properly fitted mask, and washing your hands or using an alcohol-based rub frequently. Get vaccinated when it's your turn and follow local guidance[1].

The goal of the project is for people to use this application to find places without going out, so they can also protect themselves from COVID-19. The "COVID-19 Vaccination System" is developed within the Android app. The app was created in two sections, the first frontend and the second backend. The front end was used in the application design, and the second part is the back end. Android is an open-source portable working framework. Android is a complete set of software for mobile devices such as tablet computers, notebooks, smartphones, electronic book readers, set-top boxes etc [2].

The MongoDB is utilized as a backend server. It is a variant of Linux, hence providing extensive security, modularity and productivity at the mobile device level. Android applications are free to develop. There are no licensing or royalty fees to develop on the platform. No required membership, signing or certification fees. Android applications can be distributed and commercialized in a variety of ways [3]. It gives space for data to be stored in a database management system. It gives

the user a simple interface, and the user's data is kept safe in the database. For one-time-password (OTP) verification, we used Firebase software in this application. Using a simple user-friendly interface of the realm mongodb database, this project assists in administrative tasks such as looking for usernames and maintaining user record. The administration grants the user permission to log in to the application. It ensures that the database is completely secure. Using the area pin code number, the application provides a search mechanism for vaccination centers. It saves the user's information in the database. The objective of the covid vaccination system is to automate the existing manual system using android equipment and a full-featured android application, meeting their needs so that their valuable data/information may be stored for a longer period of time with easy access. This project aids in administrative tasks such as finding a user using a user id and storing the user's data in the form of a collection.

Before we begin working on our first app, I'd want to highlight some typical app development steps [4].

1. To begin, make an Android Studio project..
2. Configuring the app's Graphic User Interface (GUI).
3. Connecting the GUI components to the Java code, such as buttons, textboxes, and so on.
4. In the android studio, coding in Java is the actual programming component.
5. Building your project entails building the executable file that runs on the device or emulator. It is simple to complete all of the work with Android Studio with a single click.
6. Using an emulator to launch the application.
7. The application can be run on a real Android handset.
8. The application can also be published on the Google Play Store.

II. LITERATURE REVIEW

The goal of the Covid vaccination system is to automate the existing manual approach using computerised equipment and a full-featured Android application, supporting their needs so that their valuable data/information may be stored for a longer period of time with easy access. Using a basic user interface of the realm mongodb database, this project assists in administrative tasks such as searching for a username and maintaining user records. The administration grants the user permission to log in to the application. It ensures that the database is completely secure. Provides search capabilities depending on multiple of criteria. For example, the area's pin code number. It keeps track of the information that is used. In the

application, all fields are validated and do not accept invalid data. It keeps track of application usage in the database on a monthly, quarterly, or annual basis.

There is always a requirement for an application that will provide users with an easier way to become vaccinated. This database will reduce the amount of time it takes to manually manage all of the register user records and save each user's information with a unique identifier. The most crucial aspect of database storage is the system's connection. The realm cluster provides a one-of-a-kind code for connecting the application to the database.

III. WORKING OF APPLICATION

Android studio, firebase, and mongodb were used to construct the application. The user can use the app to find a vaccination center and obtain the vaccine they need. The user must follow the procedures outlined below in order to utilize this program. Once the application was installed on the user's device, he was ready to use it. He must input his phone for the one-time-password verification process when he opens it. The verification screen is the screen where the user receives the one-time password generated by the firebase authentication mechanism.

The user is directed to the login screen after being confirmed. For the first time user, he or she must first register by pressing the register button. The user is taken to the register page after clicking the register button, where he must first enter his username, email id, password, and confirm password to complete his registration procedure. The user can log in using his or her registered email id and password after finishing the registration process. The user is directed to the vaccination search screen after successfully logging in. On this screen, the user is given a text field in which to enter the area's pincode in order to locate the vaccination center. After entering the pincode when the user clicks on the search button, the dialog box will appear. When the user selects the search button after entering the pincode, a dialog box will appear. The user must select a date for his or her vaccinations in that dialog box. When the user types the pincode, it displays the vaccination locations that correspond to the specified pincode.

IV. FRONTEND AND BACKEND CONNECTION

The front-end languages are used as Android studio with java and kotlin. Firebase is a mobile platform that helps you quickly develop high-quality apps, grow your user base, and earn more money. Firebase is made up of

complementary features that you can mix-and-match to fit your needs, with Google Analytics for Firebase at the core [5].

A. ANDROID STUDIO AND FIREBASE CONNECTION

To make a connection between the android studio and the firebase, we have to perform these steps:

- 1) First, go to the tools and then SDK Manager.
- 2) Click the SDK tools.
- 3) Search for the Google repository checkbox, and then click on OK.
- 4) Please click OK to install the repository.
- 5) Please click the background to complete the installation or wait for the installation to finish.
- 6) Once you finish the installation, you can open the assistant window after that click on the tools and go to the firebase to open the assistant window.
- 7) Click to connect the firebase too, make a connection between the android studio and firebase.

B. ANDROID STUDIO AND MONGODB REALM CONNECTION

The Realm includes an embedded, object-oriented database that lets you build real-time, offline-first applications. You can also use Realm's hosted application services as a secure backend that can sync data between devices, authenticate and manage users, and run server less JavaScript functions. With the MongoDB Realm Android SDK, you can access objects stored in a local instance of Realm Database [6].

With the MongoDB Realm Android SDK, you can access objects stored in a local instance of Realm Database. To build the connection between them we have to add In `buildscript.repositories` that is `jcenter()`.

In `allprojects.repositories`, add the following:
`mavenCentral()`

Paste the Realm classpath into the `buildscript.dependencies` block: `classpath "io.realm:realm-gradle-plugin:10.8.0"`[7].

To make a connection between the MongoDB and the android studio, we have to follow these steps:

- 1) We have to first import some libraries and modules in a gradle build file to use the realm with android studio.
- 2) After importing and adding the libraries, we have to create the mongodb atlas cluster account.
- 3) To create the new realm app, open the realm tab.
- 4) Click on the Create a New App button to start without a template.
- 5) Configure your app and give the application name.
- 6) Click to create a new realm application.

V. ANALYSIS & DESIGN

A. FLOWCHART DIAGRAM

A flowchart represents an application's workflow. A parallelogram, arrows, and diamonds are used to show numerous processes[8]. The administrator has access to all user accounts and their data in the flow chart. The administrator has the ability to create new users, modify existing users' information, and delete existing users' information. The user has only read and write access to their own data. The user began with the OTP verification stage in the image above, and then had to login to the application using a valid email id and password. If there is a new user, he or she must first register; after that, he or she can proceed to the next display, where the vaccine center can be found. The user data is stored in the MongoDB database as a collection.

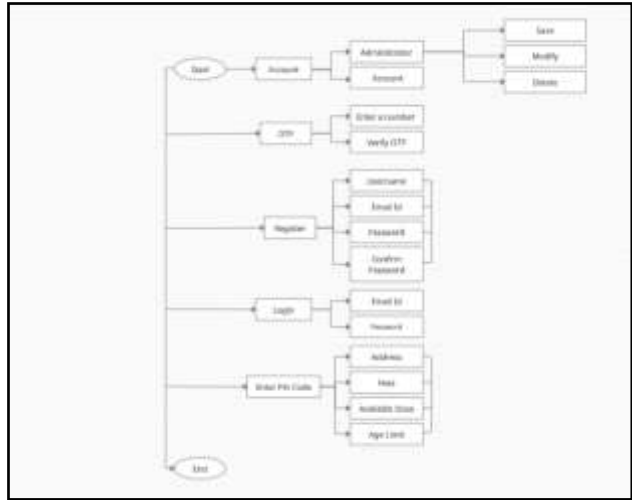


Fig. 1. Flow chart of “Covid Vaccination Syatem”

B. ER (ENTITY RELATIONSHIP) DIAGRAM

The ER diagram is known as the Entity Relationship Diagram, it is also known as ERD. An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. Also, known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to show entities, relationships and their attributes[9]. They mirror grammatical structure, with entities as nouns and relationships as verbs. ER diagrams

are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model) [10]. It is a diagram that shows the relationship of an entity stored in mongodb database. The ER diagram is created based on the entities, attributes and relationships.

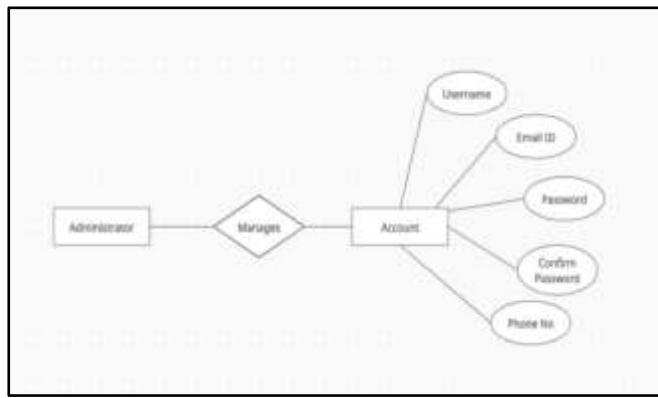


Fig. 2 ER Diagram of Covid Vaccination System

C. USECASE DIAGRAM

An actor and their relationship are used to represent a use case diagram. The use case diagram depicts the user's relationship with the program.

The use cases are the functions that the user and the administrator perform. Use cases are prepared in an easy-to-understand use case template format.



Fig. 3 Usecase Diagram

VI. APPENDIX

We'll look at various displays from our app in this section. There will be seven screens on which various activities will be carried out.

Enter Number :- In Fig. 4.1, we've made a title bar and given it the name Enter Number. We've also implemented the OTP Verification log for a better user interface. We put some messages below the OTP verification text as we will send you a one-time password to the given mobile number. Then

we added a textbox for entering the number, and finally, a button to get the OTP.

OTP Verification :- We've added a title bar to this Fig. 4.2 and used the log for OTP verification. To add some text, we used the same text message as in the previous display. We made a small six box over here to enter the number that came from the firebase authentication, then we created a message to resend the OTP below. Finally, we included a submit button to validate the OTP.

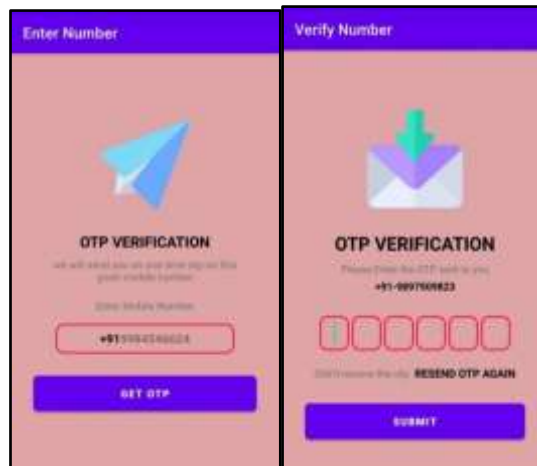


Fig. 4.1 Enter Number Screen

Fig. 4.2 Verify Number Screen

Login Form:- Figure 4.3 This is our login form screen. The user must login with the correct email id and password. On the display, we've included a textbox with appropriate validation and two buttons: one for login and another for directing users who haven't yet registered an account in the application.

Signup Form:-We've constructed four textboxes and applied validation to them, as well as a register button, in this signup form (Fig. 4.4). When a user enters his information and clicks on the register button, the information is saved in the mongodb realm cluster database, and the user receives a message indicating that he has successfully register.

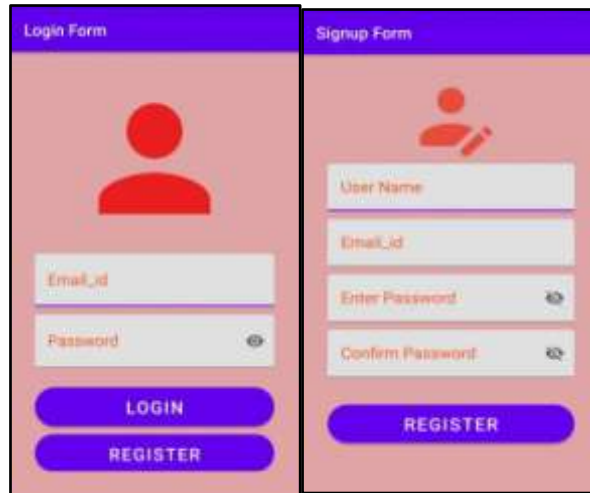


Fig. 4.3 Login Form

Fig. 4.4 Signup Form

Enter Pincode:- The screen where you enter your pin code. This screen shown in Figure 4.5 is where we can look for our vaccination center. To make this page, we included a text box where the user may enter the pin code and a search button on the right side of the text field to search the results.

Select Date:-The select date for vaccination screen is shown in Fig. 4.6. We built a date picker dialog box on this screen so that the user may select a date for a vaccine and then search for the vaccination's details.

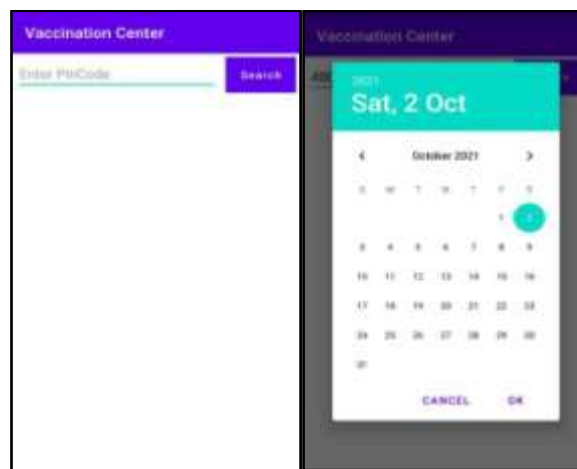


Fig. 4.5 Enter Pincode

Fig. 4.6 Select Date Of Vaccination

Vaccination Center:- We can see the result of the pin code search in Fig. 4.7 vaccination center screen. We used the vector notation for a better user interface and designed a textbox for the vaccine

name, age limit, price, vaccination center, and timing. If the user enters an invalid pin code, a message instructing them to enter the proper pin code.



Fig. 4.7 Vaccination Center

VII. SOFTWARE & HARDWARE SPECIFICATIONS

➤ SOFTWARE / LANGUAGES REQUIREMENT

1. Android Studio
2. MongoDB Realm
3. Firebase
4. Java
5. Kotlin

➤ HARDWARE REQUIREMENT

1. CPU: Intel i5 (2.3GHZ or Higher)
2. Processor: Preferably 1.80 GHz or Greater.
3. RAM: 8 GB or Greater.
4. HDD: 2GB or Above.
5. Other configurations as required.

VIII. CONCLUSION

For a better and easier interface, the project "Covid Vaccination System" is designed utilizing Android Studio as a front end and MongoDB as a back end. This project covers the Android Studio GUI as well as a number of fundamental Java and Kotlin functions. We learn a lot in this application and obtain a lot of knowledge about the development sector. The results show that connectivity provides an empirically valid basis for clustering data models, which closely matches human perceptions of importance and "chunking" behavior [11]. For rendering visuals, we used the java and kotlin libraries. We'll look at how to make a button widget. All of the essential elements of the vaccination system are included in this android-based application. The future implementation will assist the administrator in

adding and managing new security features, as well as the appointment booking functionality.

REFERENCE

- [1] <https://www.who.int/health-topics/coronavirus>
- [2] <https://www.javatpoint.com/android-tutorial>
- [3] <https://docs.mongodb.com/realm/mongodb/insert-documents/#insert-a-single-document>
- [4] <https://developer.android.com/training/basics>
- [5] <https://developer.android.com/studio/write/firebase>
- [6] <https://docs.mongodb.com/realm/sdk/android/>
- [7] <https://docs.mongodb.com/realm/sdk/android/>
- [8] Elmasri, R. and Navathe, S.B., Fundamentals of Database Systems, 3rd ed., Addison-Wesley, Reading, MA, 2000.
- [9] <https://www.lucidchart.com/pages/er-diagrams>
- [10] <https://docs.mongodb.com/realm/sdk/android/quick-start-local/>
- [11] Goldstein, R.C. and Storey, V.C., "Some Findings on the Intuitiveness of Entity Relationship Constructs," in Lochovsky, F.H., Ed., Entity-Relationship Approach to Database Design and Querying, Elsevier Science, New York, 1990.