Introduction:

The topic of our Epics Project is an application based on Medical Analytics and Services. The application is named “niRog”. It is mainly focused on uplifting Health and Medical services to citizens of our nation. It’s a software tool that helps healthcare professionals and administrators collect and analyze data related to patient care and the assigned treatments. It is also focused on collecting insights and recommendations based on real-time data, prescriptions, etc.

Throughout our daily lives, we’ve learned about many incidents which have occurred and still occurring in many parts of our country where citizens face unnecessary encounters and feedback while they were trying to seek health facilities in government hospitals: be it standing in a queue, be it not being able to receive proper feedback, etc. Few of the problems faced by people are:

niRog application comes forward with the helping motive to provide citizens receive proper Healthcare. They will have their own freedom to book appointments from the software itself. Initally, the portal will lead them to register their accounts by filling the required details. Then it’d lead them towards their profile dashboard where they will have a clear overview of all their requested appointments and medicines. The portal will also have specialized corners for Doctors and Admins. With the given permission of the user, medicine and appointment details will be stored in a reliable database to provide better analytics and services.

Out of all the APIs which the application uses, niRog is constantly dependent on Google Location Services so that it can function its features properly. One of the features it includes has, to locate nearby Government Hospitals which are linked with niRog to work seamlessly. Users can choose any of those mentioned hospitals and then seek out the software’s services.

One of the main drawbacks regarding our project is that, it is not equipped to provide services during the time of emergencies as it is not logically accurate to book appointments and wait otherwise.

Scroll View – viewgroup in XML in android studio

A view group that allows the view hierarchy placed within it to be scrolled. Scroll view may have only one direct child placed within it. To add multiple views within the scroll view, make the direct child you add a view group, for example [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout), and place additional views within that LinearLayout.

Scroll view supports vertical scrolling only. For horizontal scrolling, use [HorizontalScrollView](https://developer.android.com/reference/android/widget/HorizontalScrollView) instead.

Relative Layout – viewgroup

[RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout) is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout) area (such as aligned to the bottom, left or center).

[RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout) is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements or in positions relative to the parent [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout) area.

A [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout) is a very powerful utility for designing a user interface because it can eliminate nested view groups and keep your layout hierarchy flat, which improves performance. If you find yourself using several nested [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout) groups, you may be able to replace them with a single [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout).

(For better performance and seamlessness in UI design of our pages, we could have adhered to use Constraint Layout rather than Relative Layout as CL is more flexible than RL in android studio environment).

Image Views, TextViews

To display Image resources and pictures, we use Image Views. For texts, we use Text Views.

Why we used AppCompatButtons instead of default Buttons in XML? Color placement

We used AppCompatButtons instead of default Buttons in XML in our Android Studio Project so that we can customize the themes and colors of our buttons on our own.

Regarding register page/activity:

We have tried to use SwipeRefresh Layout here because whenever the patient/user joins the activity, he or she will be able to refresh the contents of the view via a vertical swipe gesture. This will let the user stay updated.

Why use swiperefresh? The SwipeRefreshLayout should be **used whenever the user can refresh the contents of a view via a vertical swipe gesture**.

Why do we use linear layout?

Linear layout is a very basic Layout **to implement a UI for android application**. It has an orientation component which defines in which orientation you want all layout children to be aligned

Why do we use relative layout?

RelativeLayout **lets child views specify their position relative to the parent view or to each other** (specified by ID).

Here are the lists of the dependencies we are using:

implementation 'androidx.appcompat:appcompat:1.5.1'  
implementation 'com.google.android.material:material:1.7.0'  
implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
implementation 'com.google.firebase:firebase-auth:21.1.0'  
implementation 'com.google.firebase:firebase-database:20.1.0'  
implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0'

Exception Handling in Java while writing code and logic in Login Activity and other activities:

Here, we have taken our first step in working with Logic and Connections in our Main Activity, where we initialize our views and objects. They will be further recognized here from their submitted attribute IDs through the XML codebase by using R.findViewByID attribute where R denotes to the resources in our project.

Here, we have used a few onClickListeners on two of our Buttons: Login and Register Button so that users can freely choose whether they wish to register a new account or login to their original page every time they install the interface of our software. We have used intents here.

Here, moving on to the register activity page of our project, we have tried to introduce new views and objects such as the 4 kinds of edit texts which are based on details regarding the user(example- name, mobile number, etc.). We have also tried to implement a radio button in here for accepting the submissions of their gender. Then, we have introduced two important aspects of our Database registration, i.e. FirebaseAuth and FirebaseUser.

The Firebase user object represents **a user account that has signed up for our app in our project**

The Firebase Authentication SDK **provides methods to create and manage users that use their email addresses and passwords to sign in**. Firebase Authentication also handles sending password reset emails. iOS Android Web C++ Unity. Federated identity provider integration.

Here, we have used some important dependencies for our Project, such as Material Design and Firebase Database. We have also used AppCompat and a few other layout dependencies for the UI designs of our project.

While trying to work our way around our Login and Register Activity logic parts, we are in debt to face a lot of exceptions. To try to tackle a few of them, we have introduced try catch blocks to deal with the exceptions we may encounter while our user tries to register his/her account. Example – Weak Password exception where user’s password doesn’t cross more than 6 letters/numbers and also doesn’t have unique symbols. Another example is the Collision Exception which may occur if the user tries to register an account which is already registered with an email earlier.

Since these are the initial stages of our Project, we adhered to make attempts on starting out our Project on a full emphasis on all fields, both back-end and front-end. We have attempted to setup our Firebase Console in areas concerning topics such as Authentication, Storage and Realtime Database.

Maintaining future plans to develop the software further by improving the UI more and strengthening the back end part of it further.

The platform that we are adhering to develop is to be capable of providing more than just Medical Services to individuals who seek Healthcare Information, Support and access to care. It is also going to be a gateway for individuals to manage their health and well-being sufficiently, connect with other healthcare providers, and access medical services remotely. However, it is important to carefully consider the accuracy, reliability and security of the information of our users.

“Health is the greatest gift, contentment the greatest wealth, faithfulness the best relationship.”

Script part for Conclusion:

We need to take proper measures to the potential risks and limitations of our Project. It is also extremely important to have a serious note, that, our application cannot act as a substitute for professional medical advice, diagnosis or treatment and providing medicines without registered prescriptions and permissions. The software is mostly developed to avoid the unwanted problems faced by citizens of our country when they try to seek help from Government Hospitals.

Script for my slides from slides 16 to 19:

The idea that we gained from our experiential learning program:

On our visit to experiential learning in Pune

One day one of our friend fell ill in the afternoon so we rushed him to the nearest government hospital there we faced issues like as we mentioned in our problem identification so we got our motivation for our project

Impact:

From our Pune visit, we managed to spread the plethora of our ideas through the campaigns amongst students of our university. We reached out to them and explained about the problems that individuals face whenever they adhere to seek help from Government Hospitals and Services. We believe it is important for us to have a proper rundown regarding the kinds of tackling measures that we’d enforce to implement.

We need to take proper measures to the potential risks and limitations of our Project. It is also extremely important to have a serious note, that, our application cannot act as a substitute for professional medical advice, diagnosis or treatment and providing medicines without registered prescriptions and permissions. The software is mostly developed to avoid the unwanted problems faced by citizens of our country when they try to seek help from Government Hospitals.