School of Engineering and Digital Sciences

Department of Mechanical and Aerospace Engineering

MMAE 603

Modern Control Engineering

Mobile COVID Warning App for NU

Group number: #6

Group members:

Aidana Tassanbi ID: 201593647

Daniyar Serikov ID: 201590091

Erkhan Sarsenov ID: 201594086

Salikh Omarov ID: 201696160

Date of Submission: 24.11.2021

Mobile COVID Warning App for NU

Salikh Omarov, Aidana Tassanbi., Daniyar Serikov, Erkhan Sarsenov.

Abstract— **This work presents the development of a GPS-tracker mobile application to determine the old locations of newly infected with COVID users with the additional features for maintaining public health. The other features of the app are to provide a guide for the covid patients, show X-ray results and survey for identification of possible symptoms. The app functions in 3 languages (English, Kazakh, Russian) and was designed by using the Android Studio software which focused mobile phones operating on Android operating system. The aim of the project is to grant the public with a reliable app, which can assist them to avoid the possible infections with coronavirus and to stay healthy.**

I. INTRODUCTION

The spread of the COVID-19 pandemic brought significant changes to the lives of people involving the ways they study, socialize and work. Up to this time, many regulations and policies such as distancing, restriction of gatherings as well as lockdowns and wearing masks to avoid the spread of the disease. In response to COVID-19, many countries put emphasis on the development of contact-tracing apps and other innovations. These apps are mobile platforms, which help to identify individuals who got into contact with the infected people, and collect data concerning the spreading of virus by these contacts [1]. Other than that, COVID-19 mobile applications are being designed for other different purposes like monitoring of quarantine and tracking of movement. The traditional manner of contact-tracing is done by using the personal mobile phones of people and installing the required apps, which use the modes like Global Positioning System GPS of the devices [4]. However, each of the mobile applications record the personal data of the users, involving location records, with or without the person's agreement, and keep the information on third-party servers, that could result in major cyberthreats and associated anxieties [26]. Therefore, it is required that the app requests consent for collection of information instead of operating without agreement. These kinds of apps can be useful in slowing the spread of the COVID-19. One of the main goals of this work is to develop an app, which is able to collect the data of location and time of users and to mark the saved data of a user if he/she was just infected and warned the app about that. By marking the data as “infected”, all other users who had the same data of location and time are warned by the app. For the purpose of the app development, the zone of contact was set to approximately 10 meters and time interval to 1 min. This work was performed to improve the GPS tracking for COVID-19 patients app, which was done in the previous year. Although it was used as a skeleton of the project, there have occurred issues with the implementation of this work. These issues and performed improvement are illustrated in the paper and discussed. The app is developed using the Android Studio for the mobile phones operating on Android.

II. LITERATURE REVIEW

In the paper by Birkhoff and Moriarty [2], several challenges of mobile apps were introduced. In detail, these challenges arise due to the complex requirements of patients and the big influence of the app on their health management. One of the mentioned challenges of creating mobile apps for health purposes was the lack of an understandable interface. Due to the big variety of users in region, age and population, the requirements of an app need some adaptations. In this paper, authors suggest several solutions in order to improve the design of the app for the purpose of usage, so that it has bigger engagement for users.
The work done by Chaturvedi [3] presents top 10 popular apps for mobile devices which focuses on tracking Covid-19. Obtained list of apps will be used to compare them to the one that will be created during this work. In addition, the several features of these programs were chosen as references in order to create the appropriate app for Kazakhstan.
In the work done by Cioffi et.al. [4], the issue of using contact-tracing strategy was discussed. There are several mentions about the risks of implementing mobile apps for taking safety measurements to prevent the spread of viruses. For example, there are some concerns such as the use of mobile apps that can gather the information on people who were in close contact with infected ones asking them to take isolation. This experience has raised several debates about the privacy rights of ordinary people. In addition, the data stored in the database can be leaked by cyber-attacks which also has potential hazard to the app users. Thus, these kinds of critics will be able to increase the quality of the mobile app by performing deeper work on the issues mentioned in the debate.
The observation made by Filipets [7] is focused on obtaining the statistics on different operating systems (OS) of mobiles. The important information provided by this source, will help to identify the most suitable mobile operating system on which the initial app will be written. One of the main finding of this work is providing the statistics of the most spread operation system among central Asia which is Android. This motivates us to create a mobile app based on this operating system. In addition, there is also detailed information about the users of iOS and Android which is sorted by their field of job, level of degree, etc. Lastly, it provides several suggestions on creating the mobile app for both operating systems.
In the article written by Holtz et.al. [10], the process of developing the app for adolescents with type 1 diabetes was described in detail. Taking this research as a reference example will help to develop the app which is focused on

medical purposes. In detail, the methodology section provides the information of the procedure which can be replicated in the current project but for the Covid-19 condition.

Garrett et al.[8] have worked on the tracking for COVID-19 in Australia. There are three methods of tracking illustrated in their work, which are tracking via Bluetooth, a government app 'COVIDSafe', and telecommunication network. According to the authors, such technologies are efficient in fighting COVID-19, only if everyone wants to use them. This study includes assessment of public attitude towards these technologies. Their results showed that, although people accepted the technologies, they did not actually use them. It is because of their concerns about security and privacy. In addition, the 'COVIDSafe' tracks people anonymously up to the point when the contact is infected. This work illustrates that apart from creating excellent apps, there is also a need for appropriate policies and commercials. Otherwise, all of the effort will be wasted.

Another work on GPS tracking was done by Moloo and Digumber [17]. Their work demonstrates a low-cost Geolocation system that makes use of GPS and GPRS on inexpensive cell phones. The suggested system lets users observe real-time location as well as recorded tracks using a cellphone that has been customized for the system. It uses a GPS Receiver that can be external or incorporated to identify the current device location and transmits information to a GSM network through GPRS, where it is transmitted to a primary database server. The authors created the web application with the help of PHP and MySQL, and the visualization of location was done with the Google Maps API's free version. This application sends a notification to people if they have crossed the restricted zones. Moreover, the authors state that their application is one of the cheapest GPS tracking systems in the world. This work can be very useful in creating similar apps for specific purposes.

Mallik and Bandyopadhyay [15] created an app for GPS tracking to track ambulances that carry people infected with COVID -19. According to them, it is critical to properly quarantine and monitor patients who have already been infected. When patients have to be transported to different sites by ambulances, a traffic police supervision of these vehicles can ensure distance and faster vehicle movement within the city. This paper describes the creation of a Real-time GPS tracking app for COVID-19 patients carrying ambulances, which will aid traffic cops in keeping the patients away from the general public. The authors state that this technology collects location data using Android cell phones, i.e., their GPS receivers. Their application is mainly designed to allow cab drivers and customers to communicate with one another. When customers want to hire a vehicle for themselves, they use the application to see what cabs are available in the area and then book one depending on their needs. The locations of both are tracked constantly, and this information is shared through an online database. This eliminates the burden of drivers and customers constantly contacting or texting each other to find out where they are. This paper will be very useful in developing a patient tracking app.

Manoharan [16] has also conducted a study on the GPS tracking system. The paper contains the implementation areas as well as the advantages and problems of GPS tracking. Additionally, it demonstrates a framework for mobile phones with a tracking system, which may be useful if the gadgets are lost, or for patients with dementia. This work also presents the way the location visualization is done and shows the problems with public privacy because of GPS tracking. The author also states that, if the monitoring and tracking are human-centric, there is always a significant aspect of ethics and privacy involved. As a result, privacy-conscious monitoring and tracking have become crucial. This paper can be used for creating a GPS tracking system.

The research of Hess et al. [9] examines the applicability of cases having fine-granularity criteria to modern cell phones in the setting of sailing sports. In their work, the cellphones are examined for their applicability to see if they can handle fine-granular tracking. Six modern cell phones operating on three separate platforms, such as IOS, Android, Windows were put under comparative investigation. The investigation involved the accuracy of GPS in meters, continuity, and integrity as well as battery life. Generally, their results showed that almost all modern mobile phones are capable of real-life sports tracking. Although this project was done for sports scenarios, it does not mean that the methodology and obtained results can be applied to other areas.

Paiva and Abreu [21] have created a GPS tracking system for the older population with Alzheimer's disease. As stated by the authors, the healthcare needs of the elderly are expanding as a result of the growth in life expectancy. Consequently, new applications based on developing technology are required to help ensure their safety and well-being. This research describes a low-cost GPS monitoring system for Alzheimer's patients that uses a mobile smartphone with GPS capability to track and track individuals in near real-time. Because the study focuses on Alzheimer's disease, particular preventive capabilities were considered, such as the flexibility for caretakers or healthcare experts to customize notifications. The main components of the proposed GPS tracking system are described in detail with diagrams and tables. The concept of their work can be used in making an app for COVID tracking.

The goal of the paper done by Josna et al. [12] is to use GPS tracking to track COVID patients. Their system is done by a combination of an Android and a Web app. The tracking of COVID patients is done in the web application, while the Android app is used by the patients. The main purpose of this system is to raise public awareness about COVID-19 and to monitor the locations of people to prevent the disease from spreading further. The authors designed the app in a way that asks for a picture of the user upon logging in, and the current location will be reported to the administrator. For tracking the patients must turn on GPS on their phones. This app is designed for only mobile phones that operate on

Android. This paper can be very helpful for the development of the COVID patients tracking applications.
Another source by Elkhodr, M. [6] is aimed to research and conduct analysis of COVID-19 tracing applications performance. This paper considers 13 apps launched in 10 different countries. Apps were classified depending on technologies used for COVID patients tracking and more than 30,000 reviews of apps in Apple App Store and Google Play were analyzed. In accordance with investigated reviews, it was found that more technical issues were reported for IOS Apps than for ones launched for Android phones. This research will be useful for troubleshooting our app issues and bugs, and to understand critical parts that are important to ease user interface for better convenience.
Aarogya Setu app is one of the most popular apps in COVID patient tracking. This app was launched in India for both IOS and Android users to control COVID-19 spread among the population. The aim of this paper [1] is to analyze the public interest in such a technology and point out the drawbacks. This source will be useful for our research as it considers and analyzes public interest in an app, created to warn users regarding risks and track infected patients. This paper also reveals some inaccuracies in identifying contact with an infected person, for instance it could assume that the contact happened between two people separated by a wall. Those drawbacks would also be applicable for our project, and this source will help to avoid making the same mistakes.
In contrast to the previous sources Nageshwaran, G., Harris, R. C., and Guerche-Seblain [19], C. E. consider only East and Southeast Asia countries, especially reviewing Hong Kong and Taiwan separately from China. The special attention in this study was paid to the big data search. Mostly, the data of user location is served for 14 days in the majority of apps, the same strategy will be applied for our app, to track the moment of infection and possible contact. Also, it is important to mention that in some countries not all people have stable connections or even smartphones, which will affect the accuracy of analysis and statistics. However, this is not applicable for our country, so the statistics derived by the app should be quite reliable.
Vladoescu, C., Tunea, M., and Stanciu, L. [28] analyze to what extent the creation and launch of mobile applications can affect the shortage of infected part of the population in different countries. They consider similar applications aimed to track infected patients and prevent spread of COVID-19. As we are planning, those applications will warn users regarding possible contact with infected patients and monitor compliance of recommendations. This source will be helpful for the project as it points out the main failures occurring during production of such an application, and moreover, reveal necessary functions that are required by users for better performance.
Rakshit, S., Islam, N., Mondal, S., & Paul, T. [25] consider the effect of Covid-19 pandemic on Small and Medium Enterprises (SME), the reasons of their struggling and uncertain situation. As a result, creation of such technologies could help to support SMEs and beneficially affect their performance. This research applies structural equation methods to identify the effects of Mobile apps on Indian SMEs and track down the strategies implemented by SME owners for better sustainability. This research will be a helpful resource for our research as we are planning to analyze the overall effect of the Covid tracking apps on Nur-Sultan city infrastructure.
Meanwhile another research considers the disputable situation arising from a user of a mobile application regarding the confidentiality of data. In fact, this is a very hot topic, as not all users would be agreeing to provide their location data to the app, since their personal information may be a drain of information to the state, which is undesirable for the majority. The authors state that only approximately 44% of users shared their health information with the professionals. This research is useful for user research as we need to carefully draw up a license agreement.
As the previous source, this paper by Zeinalipour-Yazti, D., and Claramunt, C. [24] also considers the privacy demolition which can occur during tracking. The conspiracy theorists are also considered in this research and the impact that they can cause to the trust of users, as well ase Ethical aspects of applying COVID-trackers. As it was mentioned above, personal data collection is an important and keen question that has to be considered during architecturing Mobile tracking apps.
In the paper of Liu et al [25] , the mobile application is developed for giving assistance to people, who have disease.The mobile application uses machine learning algorithm for disease detection.The machine learning algorithm is based on decision tree algorithm as self-diagnosis part of mobile application.In addition this software helps people find required medicine from nearby pharmacies.This paper can be used as assistance to find medication to cure COVID-19 close stores

EasyDetect Disease application[26] is developed to prevent disease spreading among childs under age of 5.This particular group of children are vulnerable to infections.The application helps to detect symptoms and illness at an early stage,provides information to caring mothers in form of photographs,video tutorials and text.This paper is beneficial for integration contact tracing application with information about COVID-19 development into new forms and what vaccine are available currently at the country.

Smart technology[27] is developed for patients with sickle cell disease.The patients are selected to interact with applications,but first they are asked to answer surveys on how comfortable they are with software.Then additional patients are selected to actually observe how patients can use this software to manage the disease effectively.The application assists patients by storing medicaments ,which patients were taken.In addition application warns ,when certain medication needs to be taken.The warning feature can be useful to implement in development of software for COVID-19 management.

The article of Neubeck et al [28] evaluates the effectiveness of mobile medical applications to prevent diseases,particularly cardiovascular disease and provides guidelines for development of such applications.This study reveals that applications should be personalized,rely on credible sources,use a system of reward and be simple and clear.This article is helpful because it set standards for successful development of application.

Mobile applications can be used as assistive technology for doctors to diagnose ptose disease.[29]The software initially takes the photo of the patient's eyes and then calculates parameters on the photo to effectively detect deflection from norm.Authors warns that application can only be used as assistance tool to doctors and cannot replace the doctor expertise.This approach can be beneficial dealing with COVID-19 patients to detect deflection from standard using photos and application as a tool.

The use of mobile applications can effectively analyse correlation between various activities to treat disease more efficiently. From the example in the paper,which studies ADHD [31],the mobile application can open new possibilities to treat COVID-19 effectively using data from sensors and photodiodes.

III. METHODOLOGY

As the work from the previous year was used as a foundation for this app, the used methods are nearly the same. The steps to follow are:

1. **Operating platform selection.** Since the previous work was done on the Android platform, this work was done on this platform as well. Furthermore, most of the programs and features of Android OS are more accessible, which makes it less problematic to create and test the app.

2. **Coding environment selection**. The project was done in the Android Studio software, which is specifically designed for app development. The language used for coding was Java.

3. **APIs selection.** API decrypts as Application Programming Interface. It is used for setting the connections between the Android tools like geolocation. For the creation of this app, API “FusedLocationProviderClient” was applied. It can establish the coordinates of the device by GPS as well as by Wi-Fi and Cell ID.

4. **Layout Creation.** It includes the app design such as placement of buttons and switches, representation of information, etc.

5. **Writing code**. It is the main part of the work. All the operations, actions are created only by coding.

6. **Database creation.** The database is required for the collection of the data like user accounts, location of the users. In Android Studio it can be done by using the Firebase online/offline-servers.

7. **Location data save format.** One of the aims of the project is to determine whether the current user was in contact with an infected user. In order to do this, the user's saved location data (time, lat, long) is compared with all the infected users saved location data. For that, the location time data is recorded in “yyyyMMddHHmm” format, and the longitude and latitude values are rounded up to 2 decimals. Despite having 2 decimals on the latitude and longitude, the mobile app was tested on a real phone where the accuracy of location reading made the final accuracy equal to 10 meters. Thus, if the current user was obtained to be within approximately 10 meters of an infected user at the same minute, the application displays a message about a possible infection.

8. **Translation Editor.** The translation editor was applied for translation of the strings, buttons, messages and Textviews.

Generally, these are the main steps used for the development of the app.

Previous work

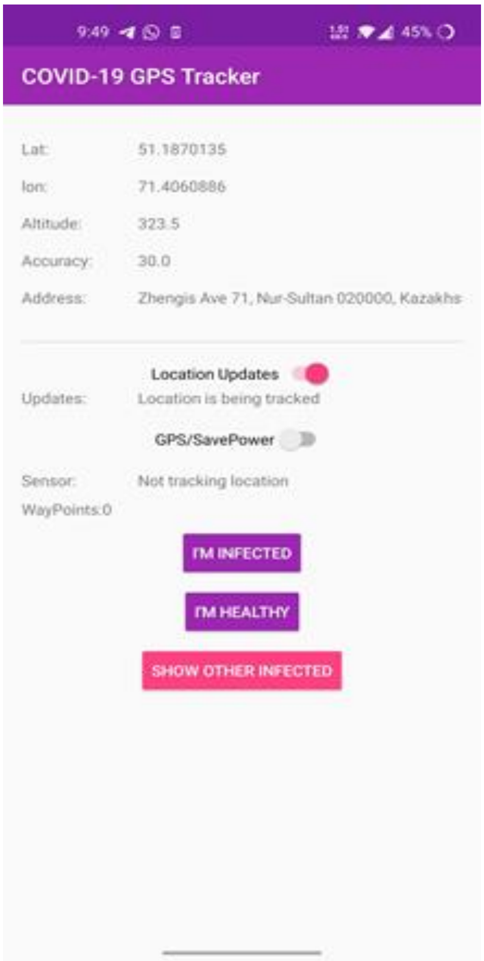


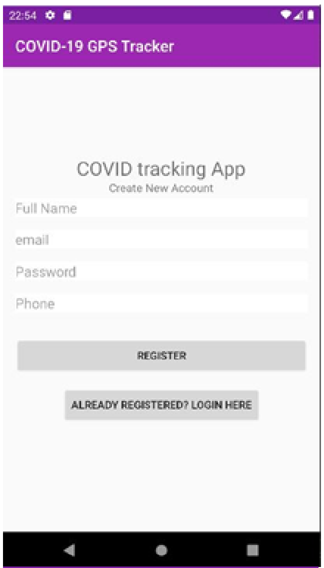
Figure 3.1: Home screen from the previous work..

In order to show the amount of work done it is necessary to represent the results of the previous work. The home screen of the app is shown on Figure 3.1. It can be seen that there is data of device location shown on the top of the screen with the values of latitude and longitude. It also shows the information regarding the altitude, accuracy of the location and the address. Below that, there are two switches called Location Updates" and GPS/SavePower. The first one is responsible for switching on and off the geolocation mode of the cell phone. While the second one is used for switching to data extraction by GPS satellites. Following that, there are buttons like “I'M INFECTED”, “I'M HEALTHY” and “SHOW OTHER INFECTED”. If the user is sick, the button “I'M INFECTED” is pressed to send the information about the user to the database and saves the location. Then, the user's location is shown on the map by a marker, which can be accessed by clicking the “SHOW OTHER INFECTED” button. Moreover, it shows the recorded locations of other infected. The markers show how many days have passed after the location was recorded and have a time limit of 14 days, since the sickness averagely lasts for 2 weeks. After that, the time is automatically set to zero. Finally, the button “I'M HEALTHY” is used if the user has recovered in less than 14 days to set the time to zero manually.

Although the previous work was given for further improvements, it was found that the app does not operate properly as shown in the paper. It can be due to the fact that it was designed one year ago and was not maintained as well as updated at all. In addition, the Android Studio had been updated several times during the year because of which some elements do not match and work. Thus, most of the program codes required to be reproduced.

Current work

Apart from fixing the errors, there were also several new features added to the app. The first one is the registration of the users. The Figures 3.2.1 and 3.2.1 illustrate the pages for registration and login, respectively. The app now looks more presentable and reliable. The created usernames and passwords are recorded in the Firebase servers.



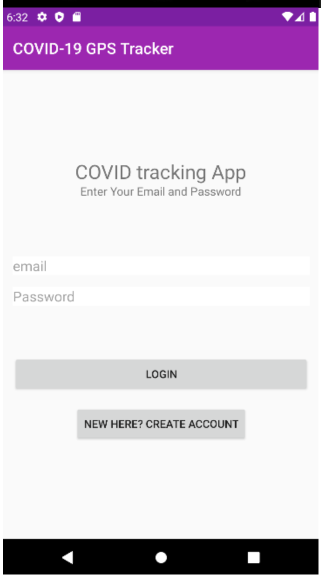


Figure 3.2.1 Registration Figure.3.2.2. Login.

Figure 3.3 shows the new home screen of the app. As it is seen, on the top-right of the software, there is a button "LOGOUT", which makes you log out of your account.

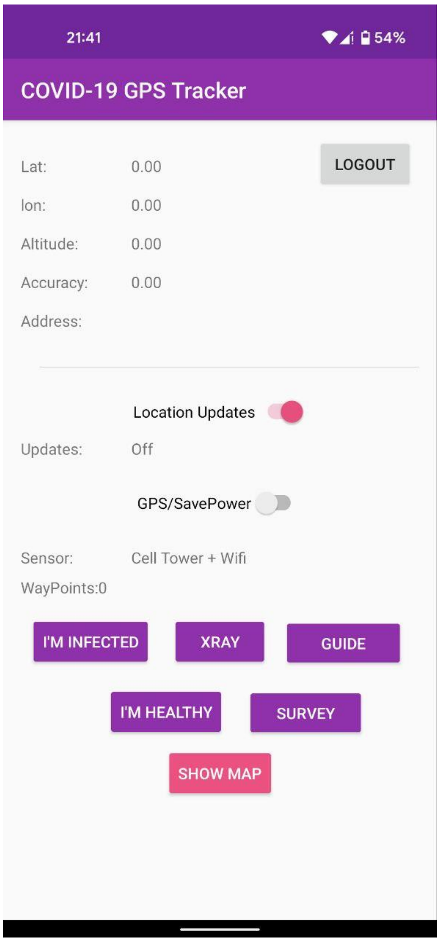


Figure 3.3: New home screen.

Following that, a guide for the users regarding the symptoms and actions that needs to be taken in case they get sick was added. Moreover, a link to a website of the World Health Organization was provided, where they can get more information regarding the COVID-19.

IV. Results

4.1. Registration and Sign in

The first part of the application was adopting the application to register new users and log in users that were registered before in this application. In order to do that a server that can store databases was required. "Google" offers a good option of using a database server. Google Firebase is a very suitable online and offline server which will be discussed in the next section. However, registration of new users is implemented by using Firebase Authentication which allows the creation of users in the Authentication database. The initial page of the application was shown in the figure.3.2.1.

The data such as full name, email address, password and phone number is registered under a new user which creates and stores this data under a unique key which is called UID. There are several interactions where programmed in order to provide appropriate registration. In the next figures some criteria for registration were shown.

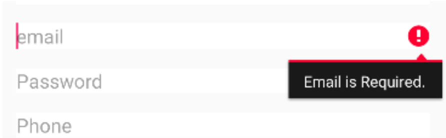


Figure 4.1.1. Requirement to enter email address in registration page.

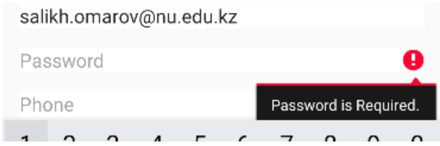


Figure 4.1.2. Requirement to enter password in registration page.

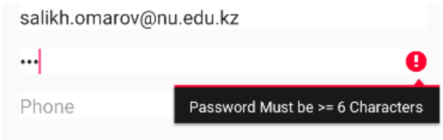


Figure 4.1.3. Requirement to enter password containing more than or equal six characters in registration page.

By storing email addresses and passwords users will be able to login into the application. In the figure 3.2.2, the layout of the login page was shown.

In the login page, the program reads the input values of entered email and password and checks the matching data. As in the previous page, there are also several requirements of filling email and password before processing data with the server. Firstly, email should not be empty. Otherwise the line will give out a message shown in figure 4.1.4. Secondly, the password must be entered as well as email (figure 4.1.5.). Thirdly, Since the registration requires the user to generate password with more than 5 characters, it requires the length of entered password to be more than 5 characters (figure 4.1.6.).

Figure 4.1.4. Requirement to enter email in the login page.

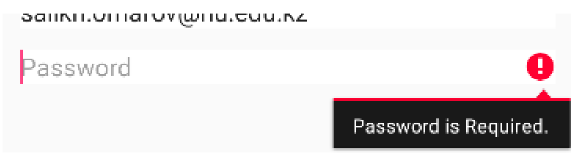


Figure 4.1.5. Requirement to enter password in login page.

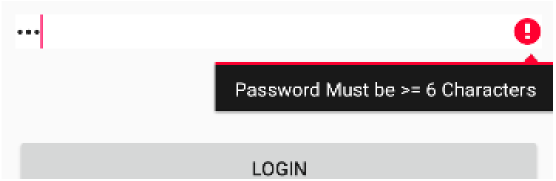


Figure 4.1.6. Requirement to enter password containing more than or equal six characters in login page.

If the data was entered, the program starts matching the data with the data located in the server. In the case when the data is not matched, the error message is displayed as in the figure below. Meanwhile, if the entered data is correct, the application page is changed to its main page.

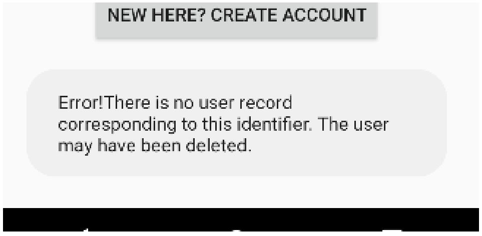


Figure 4.1.7. Error message of incorrect user email and password in login page.

In addition, paying attention to the minor detail, there is one extra button on each registration and login pages which shows “Already Registered? Login Here” and “New Here? Create Account” texts respectively. By pressing these buttons the page changes between registration and login pages. Furthermore, there is a progress bar which was added to show the user if there are background activities appearing after entering data (figure 4.1.8).

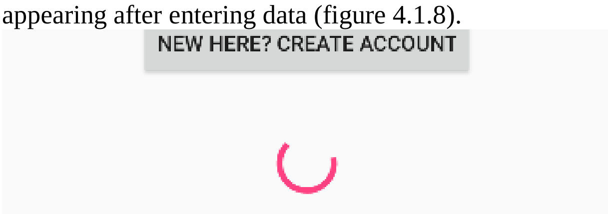
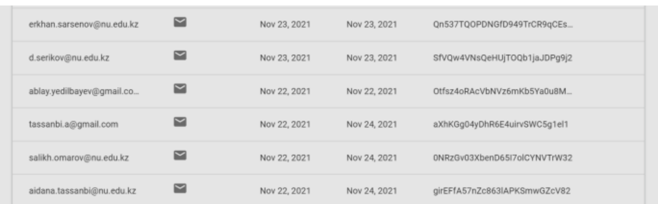


Figure 4.1.8. Progress bar.

4.2. Firebase

After registration, the user appears in the database and a unique User ID (UID) is assigned to him/her in Firebase Authentication server. The UID’s can be seen on the rightmost column on Fig.4.2.1.



erkhan.sarsenov@nu.edu.kz		Nov 23, 2021	Nov 23, 2021	Qn537TQOPDNGfD949TrCR9qCEs...
d.serikov@nu.edu.kz		Nov 23, 2021	Nov 23, 2021	SfVQw4VNsQeHUjTOQb1jaJDPg9j2
ablay.yedilbayev@gmail.co...		Nov 22, 2021	Nov 22, 2021	Otfsz4oRAcVbNVz6mKb5Ya0u8M...
tassanbi.a@gmail.com		Nov 22, 2021	Nov 24, 2021	aXhKGg04yDhR6E4uirvSWC5g1el1
salikh.omarov@nu.edu.kz		Nov 22, 2021	Nov 24, 2021	0NRzGv03XbenD65I7olCYNVTrW32
aidana.tassanbi@nu.edu.kz		Nov 22, 2021	Nov 24, 2021	girEFfA57nZc863lAPKSmwGZcV82

Figure 4.2.1: Firebase authentication page.

However, the main data is stored in the Firebase Realtime Database server. In general, the database consists of 3 parents, each of which contains user information in different formats. This was done to ease the information retrieval process and compare data.

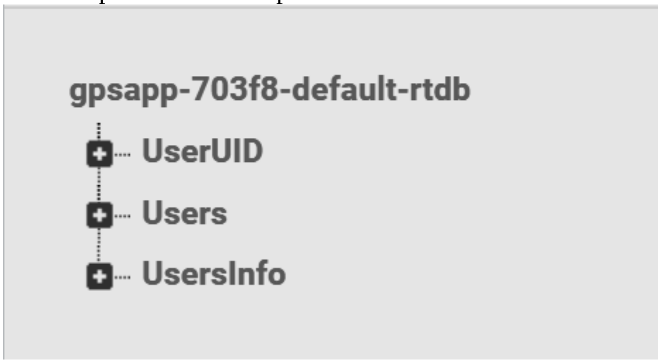


Figure 4.2.2: Firebase structure.

The “UserUID" parent contains the unique UID values of each registered user. As it can be seen from Figure 4.2.3, each UID has a parent named “UID” + counter. When the new user registers, the program identifies the number of registered users and saves the new UID to “UID” + (number of users).

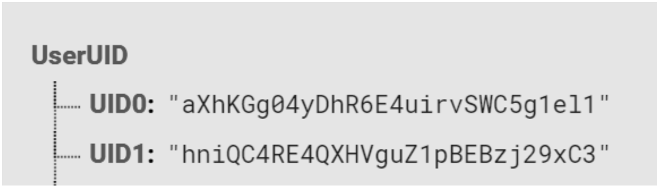


Figure 4.2.3: Firebase “UID” children.

The “Users” parent consists of childs named as users’ UID values, which in turn consist of other child information. On Fig.4.2.4 it can be seen that “0NRzGv03XbenD65I7olCYNVTrW32” contains status information and location data stored in “yyyyMMddHHmm, latitude, longitude” format.

Figure 4.2.4: Firebase “Users” children.

The “UserInfo” contains user location data, which updates within a setted interval (Fig. 4.2.5).

Figure 4.2.5: Firebase “UserInfo” children.

4.3. Comparison of user location data

This app is purposed to notify users regarding possible contact with an infected person. In order to do that, the saved location data should be compared. At first, the program retrieved the UserUID from Firebase as an Array list using Datasnapshot. This should be done since at compilation moment the program can retrieve data that belongs to the current user only. Then, the current user timesteps array is retrieved to a separate array. Using for loops the current user timesteps array is compared with an instantly created array of infected users timesteps. If the program finds that an infected user's timestep coincides with the current user timestep, it returns a toast with the message “You may get infected” at the bottom of the screen (Fig.4.3.1).

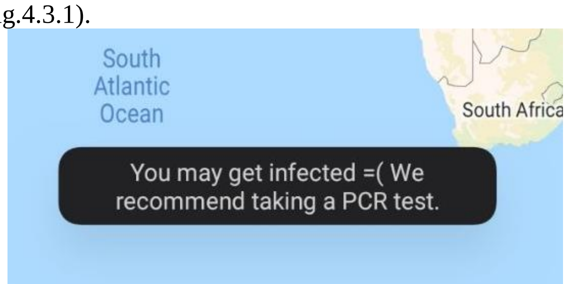


Figure 4.3.1: Toast message in case of coincidence.

Otherwise, it returns the message “No coincidences detected” (Fig.4.3.2).

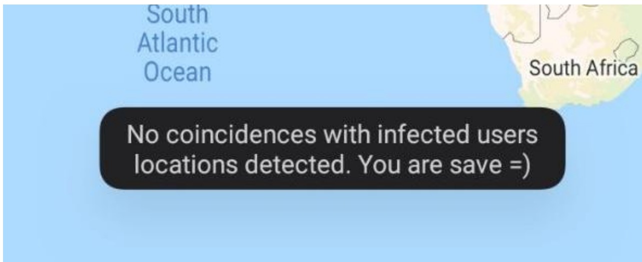


Figure 4.3.2: Toast message if no coincidences detected.

Figures 4.3.1 and 4.3.2 are made when smartphone dark mode is on. In the Appendix section (Fig. A.1) the dark-mode-off pictures can be viewed.

4.4. Guide

As it was mentioned before, the app can provide information about the COVID-19 symptoms and the instructions to follow if the user is already sick. It is convenient to have this information by hand, which can be easily accessed and help to identify the disease at an early stage. This page can be opened by clicking on the “GUIDE” button on the home screen. Since the text did not fit into the page, it was made scrollable by writing the code shown on Figure 4.4.1.

```
<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/textView">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:lineSpacingExtra="5sp"
        android:padding="10dp"
        android:text="The most common symptoms of COVID-19:  • Fever  • Dry c..."
        android:textAppearance="@android:style/TextAppearance.DeviceDefault.Medium"

        />

</ScrollView>
```

Figure 4.4.1: Code for scrollable page.

For the case when the user needs more information related to COVID-19, the link to the official website of WHO is provided. At first the link appeared just as plain text, then using the code in Figure 4.4.2, the link was made clickable.

```
    private void setupHyperlink() {
        TextView linkTextView = findViewById(R.id.textView2);
        linkTextView.setMovementMethod(LinkMovementMethod.getInstance());
        linkTextView.setLinkTextColor(Color.BLUE);
    }
}
```

Figure 4.4.2: Code for hyperlink.

The final view of the Guide For Patients is shown in the figure below. The text taken from the WHO website and was organized by putting special signs such as “/n*” and “<b>” for spacing and making bold text, respectively. The link is located at the bottom of the page.

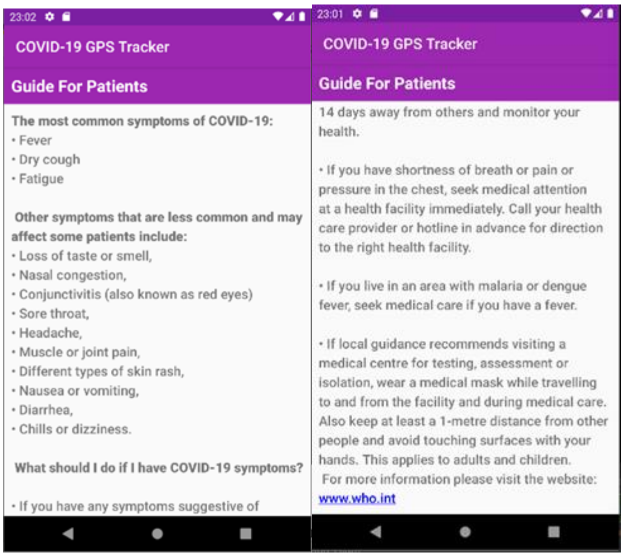


Figure 4.4.3: Guide page.

4.5. Webview (XRAY)

The idea is to use a website ,which contains the neural network that detects from XRAY chest images the pneumonia.

The website https://mlmed.org/tools/xray/ is selected to provide users with information about the condition of the user's chest.

This website,which contain the neural network,which assess the condition of human chest is developed by group of scientists [32]

Webview is selected to access the https://mlmed.org/tools/xray/ through application on Android.

The neural network is downloaded inside the web browser and images of Xray chest are uploaded to the website.

Webview is basically the browser which can access the website and has functionality of the website.

The 3 main challenges occurred during the process of accessing the website:

1.Creating the way of accessing the website by clicking the button

2.Downloading the neural network to from the website to the smartphone via application

3.Uploading the image of Xray chest to the website via application

4.Adding the zoom controls to the webview.

1.Accesing the website :

To tackle the first challenge the webview class is programmed in android studio

```
private WebView webView;
```

Figure 4.5.1: Webview implementation

The loadUrl and JavaScriptEnabled methods allow the application to directly open the website instead of the standard browser on a smartphone.

```
webView.getSettings().setJavaScriptEnabled(true);
webView.loadUrl("https://mlmed.org/tools/xray/");
```

Figure 4.5.2: Loading website through application.

In the manifest file the access to the internet from smartphone is programmed

```
<uses-permission android:name="android.permission.INTERNET" />
```

Figure 4.5.3: Permission access to the Internet

The webview button is programmed to access the website by tapping on it

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:backgroundTint="@color/colorPrimary"
    android:text="XRay"
    android:textColor="#FFF"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tv_labelCrumpCounter" />
```

Figure 4.5.4.X Ray button definition

2.Function to download the files in the Webview:

In order to download any file in the webview it is required to ask for permission to download the files on android storage in the manifest file

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Figure 4.5.5. Permission to the External storage

Notifies the user that download should start

```java
webView.setDownloadListener(new DownloadListener() {
    @RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
    @Override
    public void onDownloadStart(String url, String userAgent, String contentDisposition, String mimetype, long contentLength) {
        String filename = URLUtil.guessFileName(url,contentDisposition,getFileType(url));
        sFileName = filename;
        sUrl = url;
        sUserAgent = userAgent;
        if (Build.VERSION.SDK_INT>= Build.VERSION_CODES.M){
            if (ContextCompat.checkSelfPermission( context: Main2Activity.this, Manifest.permission.WRITE_EXTERNAL_STORAGE)
                    == PackageManager.PERMISSION_GRANTED){
                downloadFile(filename,url,userAgent);
            }else {
                requestPermissions(new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, requestCode: 1001);
            }
        }else {
            downloadFile(filename,url,userAgent);
        }
    }
});
}
```

Figure 4.5.6. onDownloadListener method

The Download manager is used to download the file.In our case the neural network model.Basically the method download file with certain name,and from a certain website asking the permission from manifest.

```java
@RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN)
private void downloadFile(String filename, String url, String userAgent) {
    try {
        String cookie = CookieManager.getInstance().getCookie(url);
        DownloadManager downloadManager = (DownloadManager)getSystemService(DOWNLOAD_SERVICE
        DownloadManager.Request request = new DownloadManager.Request(Uri.parse(url));
        request.setTitle(filename)
                .setDescription("Downloading...")
                .addRequestHeader( header: "cookie",cookie)
                .addRequestHeader( header: "User-Agent",userAgent)
                .setMimeType(getFileType(url))
                .setAllowedOverMetered(true)
                .setAllowedOverRoaming(true)
                .setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE|
                        DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
        downloadManager.enqueue(request);
        Toast.makeText( context: this,  text: "Download Started", Toast.LENGTH_SHORT).show();

        sUrl = "";
        sFileName = "";
        sUserAgent = "";
    }catch (Exception ignored){
        Toast.makeText( context: this, ignored.toString(), Toast.LENGTH_SHORT).show();
    }
}
```

Figure 4.5.7. DownloadManager function

3. Uploading the file to smartphone through application

Similarly in the case of downloading the file, permission to access the camera and file storage on the smartphone is required.

```xml
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.CAMERA" />
```

Figure 4.5.8. Permission to Read External storage and access to camera

The storage is accessed by the command onShowFileChooser:

```java
// for Lollipop, all in one
public boolean onShowFileChooser(
        WebView webView, ValueCallback<Uri[]> filePathCallback,
        WebChromeClient.FileChooserParams fileChooserParams) {
    if (mFilePathCallback != null) {
        mFilePathCallback.onReceiveValue(null);
    }
    mFilePathCallback = filePathCallback;

    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {

        // create the file where the photo should go
        File photoFile = null;
        try {
            photoFile = createImageFile();
            takePictureIntent.putExtra( name: "PhotoPath", mCameraPhotoPath);
        } catch (IOException ex) {
            // Error occurred while creating the File
            Log.e(TAG,  msg: "Unable to create Image File", ex);
        }
```

Figure 4.5.9 .OnShowFileChooser method

On activityResult method is used for selection of the file either from the camera or file storage

```java
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {

    // code for all versions except of Lollipop
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.LOLLIPOP) {

        if (requestCode == FILECHOOSER_RESULTCODE) {
            if (null == this.mUploadMessage) {
                return;
            }

            Uri result = null;

            try {
                if (resultCode != RESULT_OK) {
                    result = null;
                } else {
                    // retrieve from the private variable if the intent is null
                    result = data == null ? mCapturedImageURI : data.getData();
                }
            } catch (Exception e) {
                Toast.makeText(getApplicationContext(),  text: "activity :" + e, Toast.LENGTH_LONG).show();
            }

            mUploadMessage.onReceiveValue(result);
            mUploadMessage = null;
        }
```

Figure 4.5.10. onActivityResults method

The function of create_image is used to store the image which is uploaded to the website through application.

After programming the webview to the website.

The webview is checked on workability:

Firstly the application is opened:

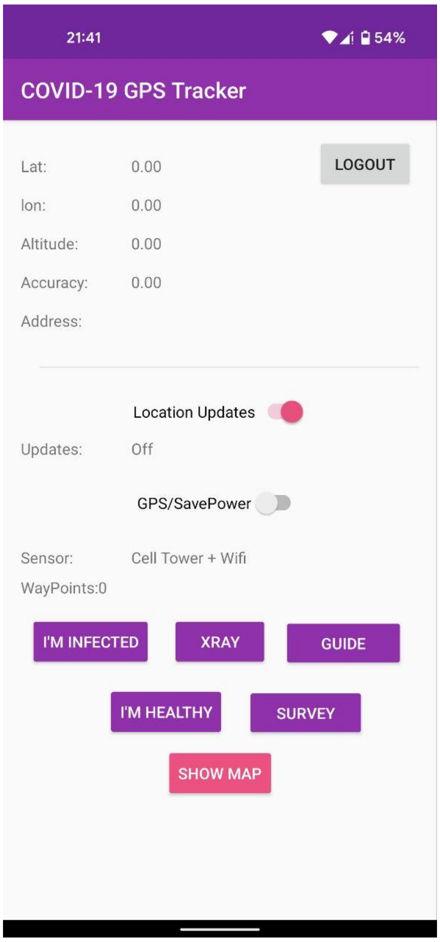


Figure 4.5.11.Application view after signing up

Then the “XRAY” button is clicked

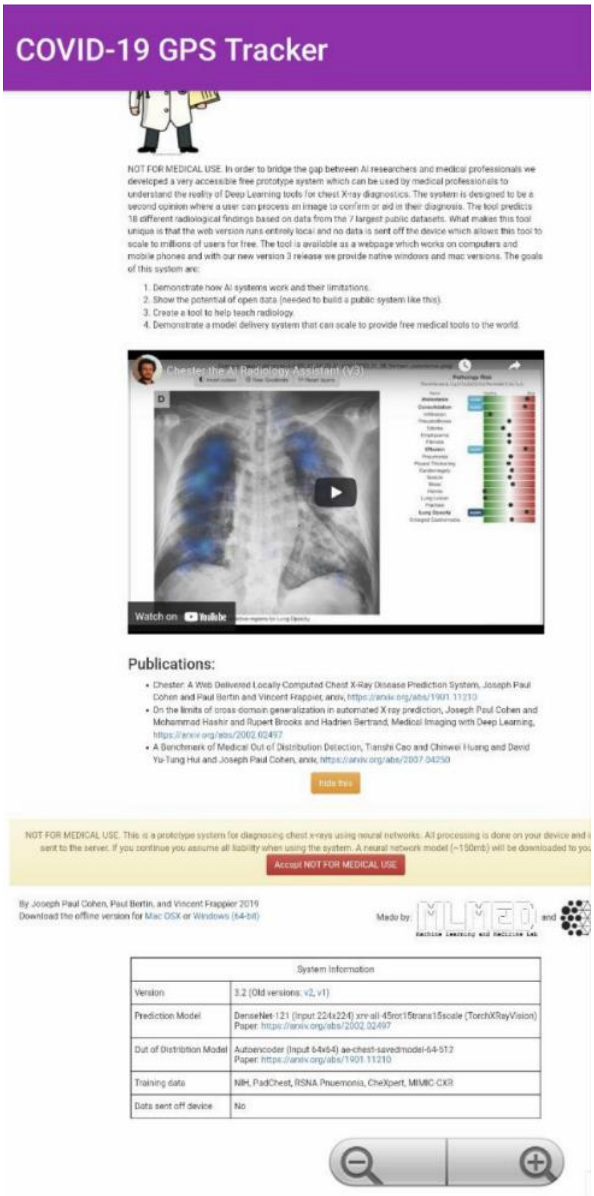


Figure 4.5.12.Access to the website through app

The red button”Accept NOT FOR MEDICAL USE” is tapped to accept the terms of using the neural network.

The reason for the agreement is that users should not believe the data that a neural network gives and should check with the professional data.Neural network can be inaccurate and data that is received should be checked by a professional

Then choose files button is clicked on the site and image is uploaded

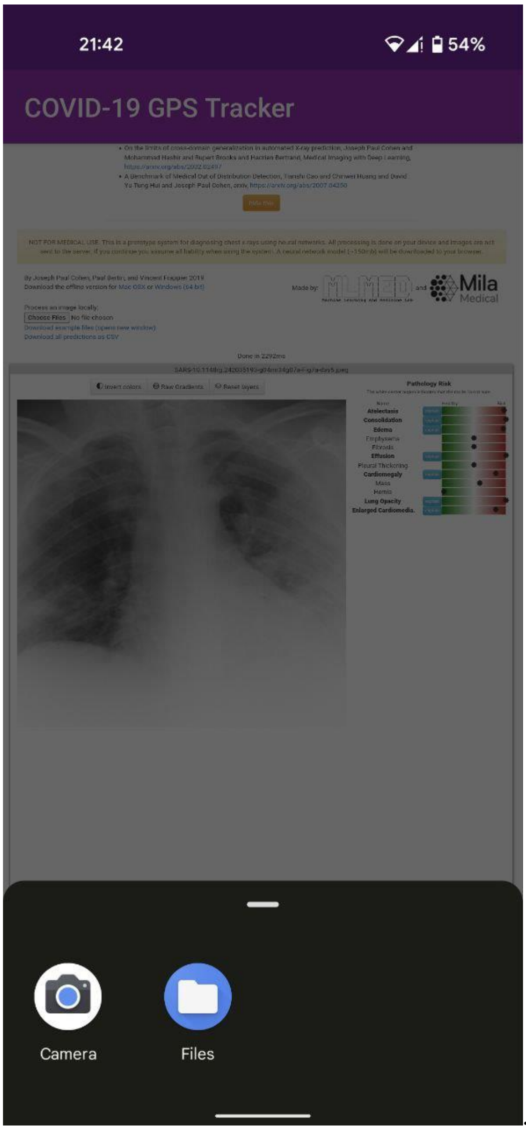


Figure 4.5.13.The upload process

The results are given below:

The Xray images are downloaded from this link:

https://github.com/ieee8023/covid-chestxray-dataset

The Xray chest of the patient with COVID:

From the figure below it can be seen that **lung opacity** is at risk (red zone), which can be the first sign of pneumonia.

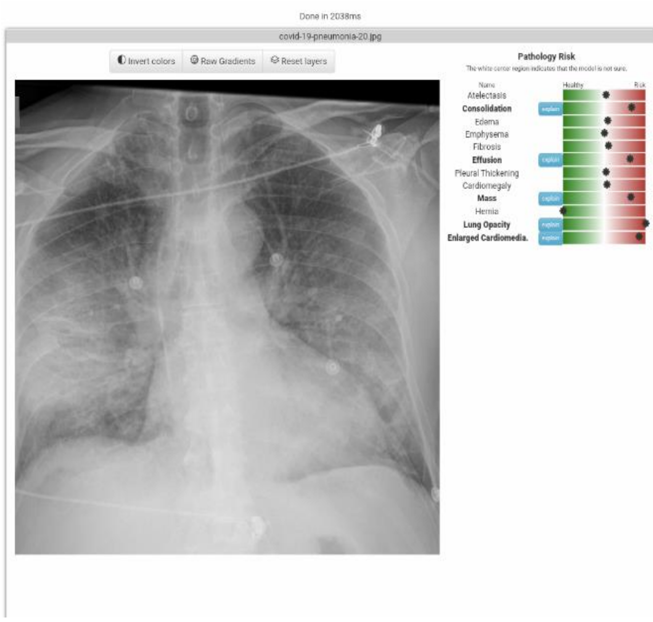


Figure 4.5.14.Xray chest image of healthy human

The Xray chest of the healthy person:

From the image below,it can be seen that lung opacity is low and patients have lower chances of pneumonia or COVID.

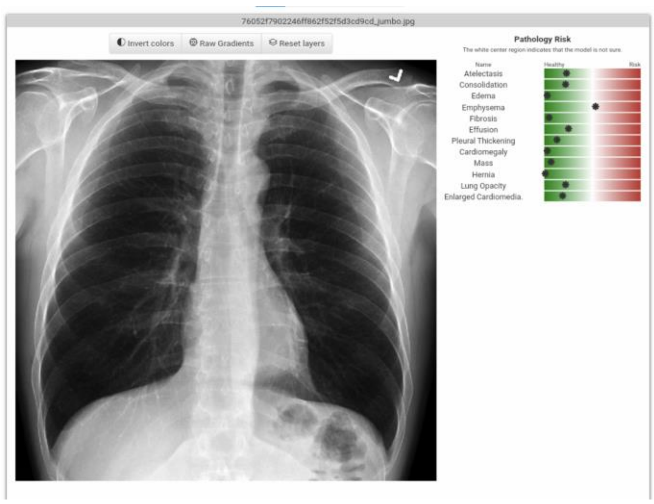


Figure 4.5.15.Xray chest image of Covid patient

4.6 Survey

The survey is created to assess the condition of the user in terms of having the symptoms of the COVID-19.

The question are taken from [33]

As inspiration the code from https://github.com/skooltch84/TrueOrFalse1/blob/master/a pp/src/main/java/com/blogspot/skooltchdev/trueorfalse1/ResultsActivity.java [34]

is taken to make a survey to help detect the symptoms of COVID.The survey is not accurate,and users should consult with a professional doctor.However the survey can be helpful to raise awareness and urge to call the doctor for help with symptoms.

Firstly the arrays of questions,images and choices for the right answers are created.

```java
package com.example.gps_tracking_demo;

public class QuizBook {
    public static String[] questions = new String [] {
            "Do you feel a sore throat",
            "Do you feel cough and sputum production",
            "Do you feel fatigue",
            "Do you feel short of breath or difficulty breathing",
            "Do you feel fever>37.8.",
            "Have you had fever for more than three days>37.8",
            "Have you had any contact with anyone who has been diagnosed or suspected of the new coronavirus",
            "Have you had contact with someone who has been diagnosed with CoVId-19",

    };

    public static int[] images = new int [] {
            R.drawable.img_0512, R.drawable.img_0516, R.drawable.img_0521, R.drawable.img_0522,
            R.drawable.img_0523, R.drawable.img_0525, R.drawable.img_0526, R.drawable.img_0527,

    };

    public static boolean[] answers = new boolean[]{
            true, true, true, true, true, true, true, true
    };


}
```

Figure 4.6.1.The list of questions,images and answers

Then the two methods are created to update the question and score.The method allows to update the image and score for every new question.

```java
private void updateQuestion() {

    mImageView.setImageResource(QuizBook.images[mQuestionNumber]);
    mQuestion.setText(QuizBook.questions[mQuestionNumber]);
    mAnswer = QuizBook.answers[mQuestionNumber];
    mQuestionNumber++;
}

private void updateScore(int point) { mScoreView.setText("" + mScore); }

public void clickExit(View view) { askToClose(); }
```

Figure 4.6.2.updateQuestion method

Then the logic is programmed if true button is tapped:

If the user taps on a button and mscore coincides with the right answer in the list of right answers (yes or no) the score is increased by one. In addition the method check if user of the last question the program jumps into results window (Results Activity)

```java
mTrueButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(mAnswer == true) {
            mScore++;
            updateScore(mScore);

            //perform check before you update the question
            if (mQuestionNumber == QuizBook.questions.length) {
                Intent i = new Intent( packageContext: QuizActivity.this, ResultsActivity.class);
                Bundle bundle = new Bundle();
                bundle.putInt("finalScore", mScore);
                i.putExtras(bundle);
                QuizActivity.this.finish();
                startActivity(i);
            } else {
                updateQuestion();
            }
        }
        else {
            if (mQuestionNumber == QuizBook.questions.length) {
                Intent i = new Intent( packageContext: QuizActivity.this, ResultsActivity.class);
                Bundle bundle = new Bundle();
                bundle.putInt("finalScore", mScore);
                i.putExtras(bundle);
                QuizActivity.this.finish();
                startActivity(i);
            } else {
                updateQuestion();
            }
        }
    }
```

Figure 4.6.3:True (YES) button method

Similar logic is programmed in the false button:

If answer in coincides with boolean (true or false)

list of right answers the score is increased.The check of the last question is also programmed.

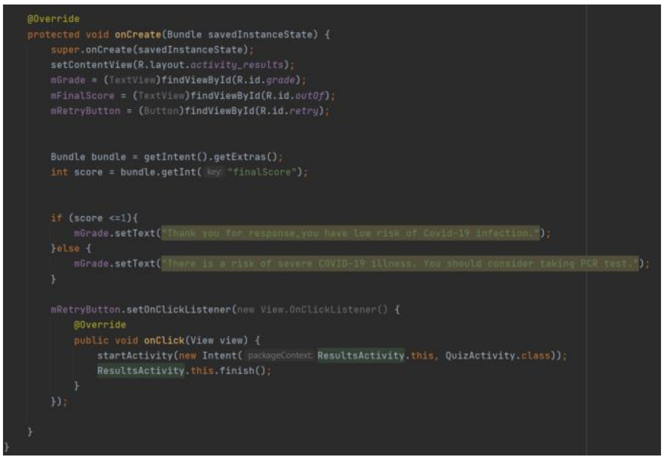
The bundle function is used to pass the variable score to the next activity (window) ResultsActivity.

```java
mFalseButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(mAnswer == false) {
            mScore++;
            updateScore(mScore);

            //perform check before you update the question
            if (mQuestionNumber == QuizBook.questions.length) {
                Intent i = new Intent( packageContext: QuizActivity.this, ResultsActivity.class);
                Bundle bundle = new Bundle();
                bundle.putInt("finalScore", mScore);
                i.putExtras(bundle);
                QuizActivity.this.finish();
                startActivity(i);
            } else {
                updateQuestion();
            }
        }
        else {
            if (mQuestionNumber == QuizBook.questions.length) {
                Intent i = new Intent( packageContext: QuizActivity.this, ResultsActivity.class);
                Bundle bundle = new Bundle();
                bundle.putInt("finalScore", mScore);
                i.putExtras(bundle);
                QuizActivity.this.finish();
                startActivity(i);
            } else {
                updateQuestion();
            }
        }
    }
});
```

Figure 4.6.4:False(NO) button method

Finally in the ResultsActivity the logic ,which selects which message to display depends on the amount of questions the user answered with right answers.If the user answered yes on two or more questions there is risk that the user has COVID and he should call the doctor for advice.



```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_results);
    mGrade = (TextView)findViewById(R.id.grade);
    mFinalScore = (TextView)findViewById(R.id.outOf);
    mRetryButton = (Button)findViewById(R.id.retry);


    Bundle bundle = getIntent().getExtras();
    int score = bundle.getInt( key: "finalScore");


    if (score <=1){
        mGrade.setText("Thank you for response,you have low risk of Covid-19 infection.");
    }else {
        mGrade.setText("There is a risk of severe COVID-19 illness. You should consider taking PCR test.");
    }

    mRetryButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent( packageContext: ResultsActivity.this, QuizActivity.class));
            ResultsActivity.this.finish();
        }
    });

}
}
```

Figure 4.6.5.Activity results.

The survey is checked for the suitable working conditions:

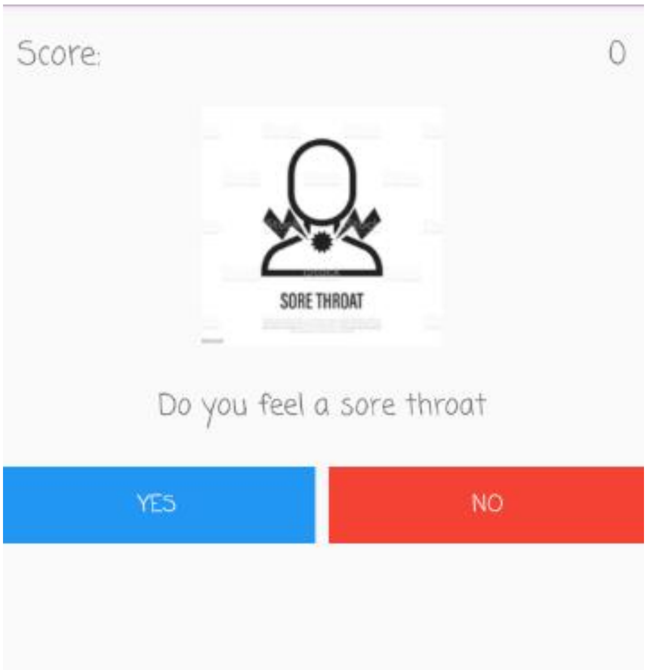


Figure 4.6.6:First question

Figure 4.6.7:Second question

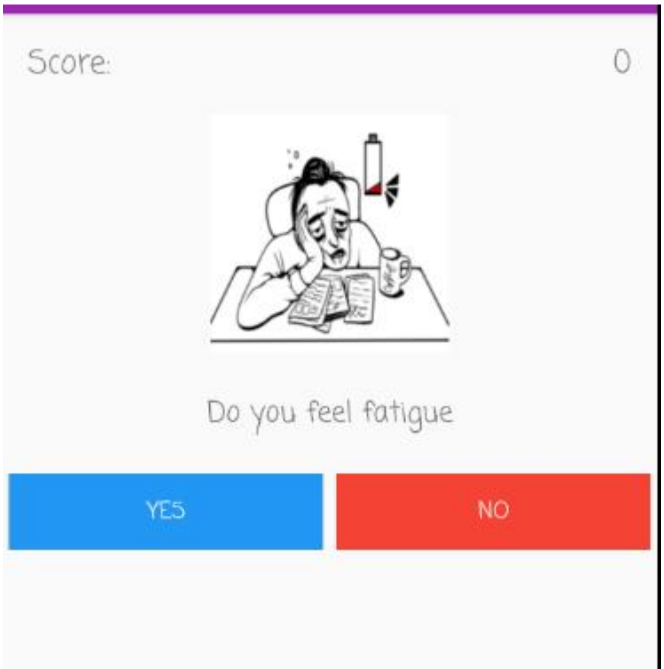


Figure 4.6.8:Third question

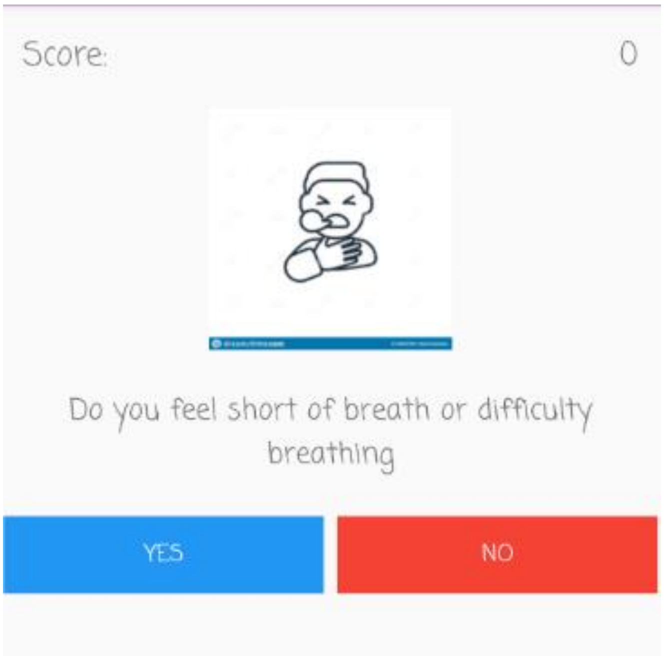


Figure 4.6.9:Fourth question

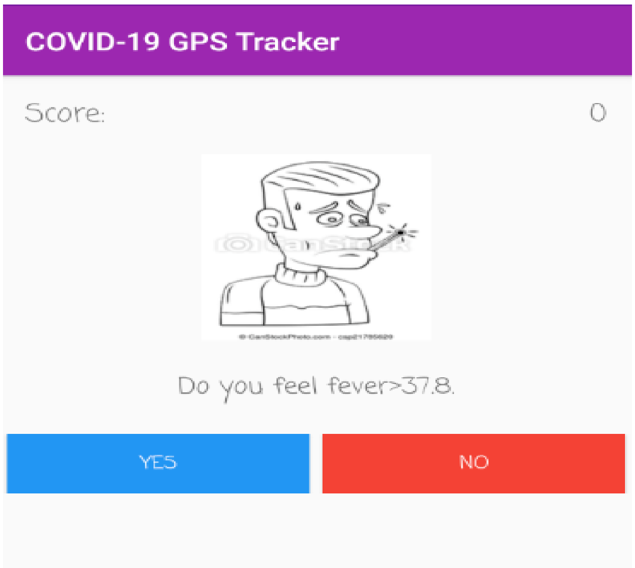


Figure 4.6.10:Fifth question

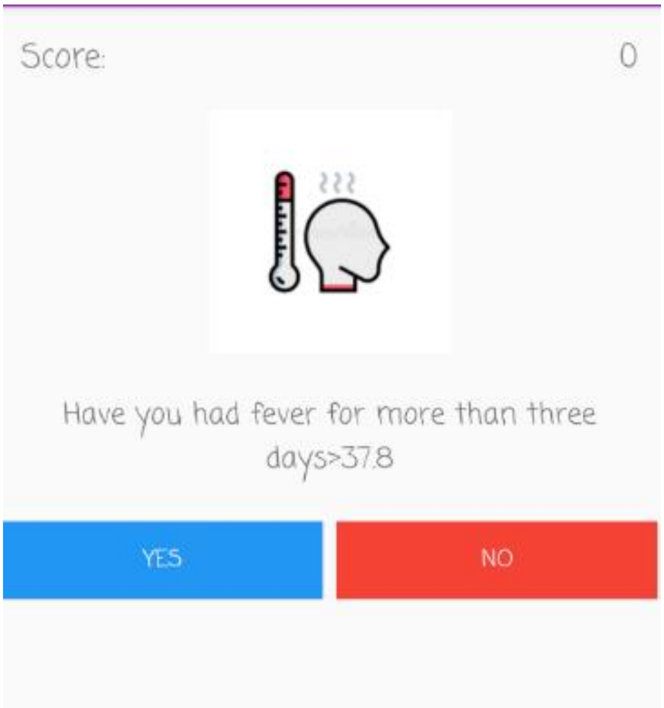


Figure 4.6.11.Sixth question

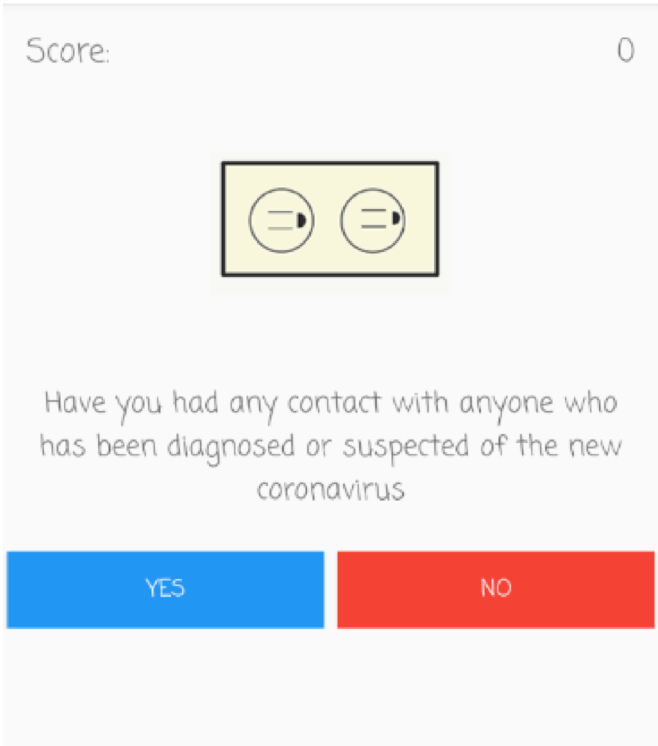


Figure 4.6.12.Seventh question

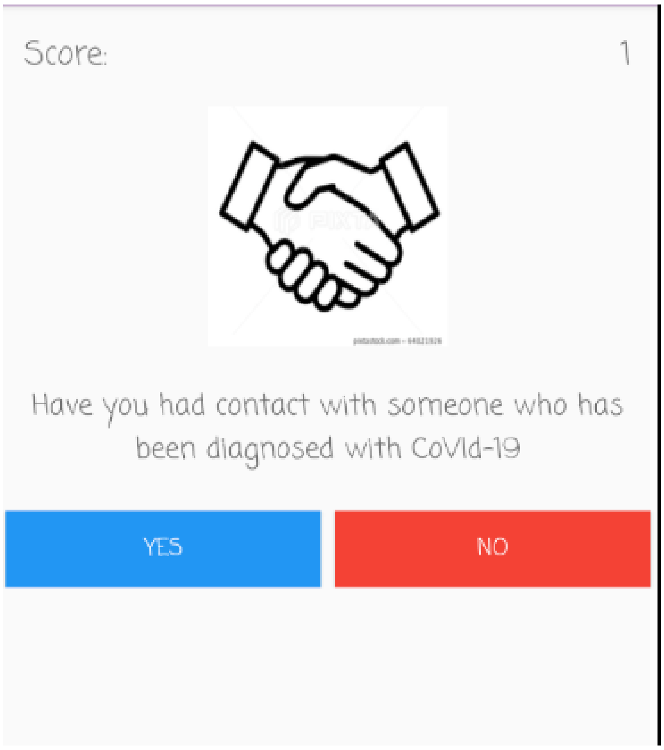


Figure 4.6.13.Eight question

Depending on the outcome there are two version of answers:

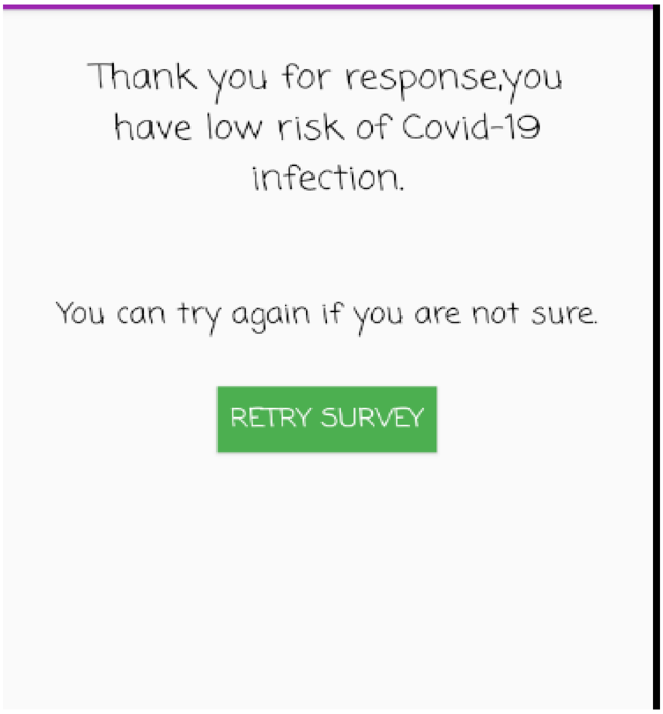


Figure 4.6.14.Final window first outcome

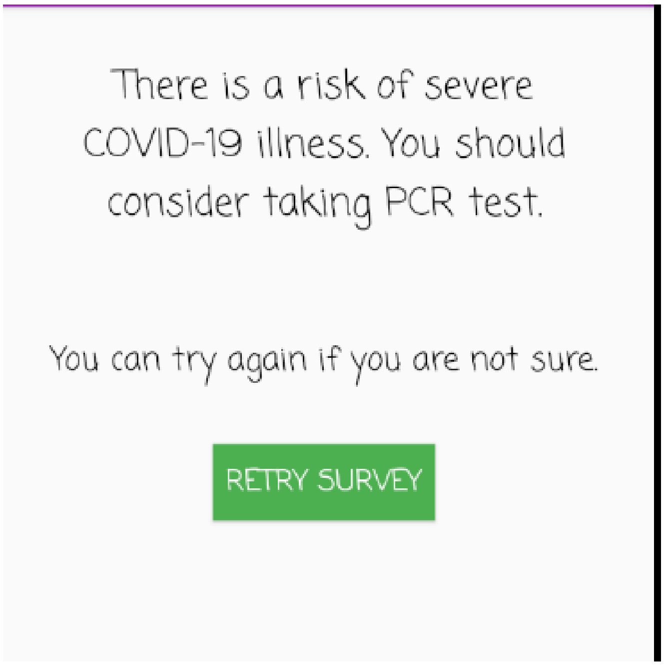


Figure 4.6.15:Final window second outcome

4.7. Languages

The application was localized for the language in which the user's smartphone is configured.

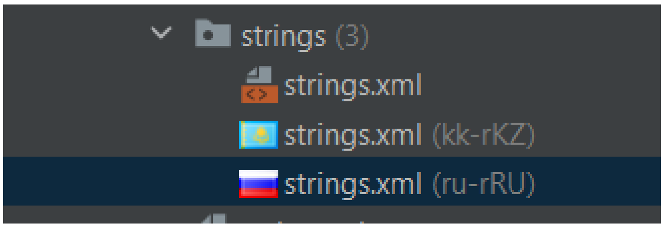
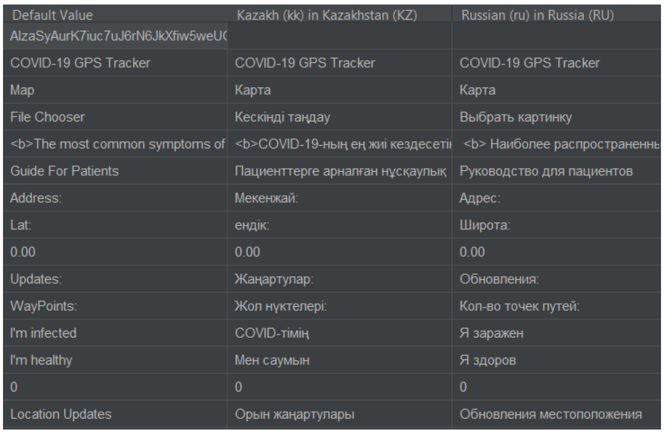


Figure 4.7.1: Strings.xml files created for Kazakh and Russian translation.

Translation of the application was performed using Translator Editor, which contains all the strings created in strings.xml (Fig. 4.7.1). The translation process runs in the Translation Editor tab (Fig. 4.7.2), where the string which needs to be translated can be viewed. In the case if the string is too long it is possible to open extended view (Fig.4.7.3)



Default Value	Kazakh (kk) in Kazakhstan (KZ)	Russian (ru) in Russia (RU)
AlzaSyAurK7iuc7uJ6rN6JkXfiw5weU(		
COVID-19 GPS Tracker	COVID-19 GPS Tracker	COVID-19 GPS Tracker
Map	Карта	Карта
File Chooser	Кескінді таңдау	Выбрать картинку
<b>The most common symptoms of	<b>COVID-19-ның ең жиі кездесеті	<b> Наиболее распространенны
Guide For Patients	Пациенттерге арналған нұсқаулық	Руководство для пациентов
Address:	Мекенжай:	Адрес:
Lat:	ендік:	Широта:
0.00	0.00	0.00
Updates:	Жаңартулар:	Обновления:
WayPoints:	Жол нүктелері:	Кол-во точек путей:
I'm infected	COVID-тімін	Я заражен
I'm healthy	Мен саумын	Я здоров
0	0	0
Location Updates	Орын жаңартулары	Обновления местоположения

Figure 4.7.2: Strings.xml files created for Kazakh and Russian translation.

V. Discussion and Analysis

The final result of the application showed some results. Tracking users for being infected with COVID-19 is useful in preventing it from further spread of the virus. In fact, nowadays implementing different methods for reducing it from the further spread currently is very viral. That is why this application can find its place in the current implementation.

However, due to the experimental conditions, most of the parameters were decreased in scale due to several reasons. In fact, the capacity of users was set to be no more than 20 people. In addition, the demo version of the application is able to store the locations 20 times with a maximum of 1 min of interval. As well, the location accuracy was set to be approximately 10 m. The decreased amount of data flow was set to be minimum giving main attention to obtain the code which is able to extract desired results. In addition to that, the free version of Firebase Database is limited in data storage capacity while the extended version charges a price. Despite that, by implementing the application in wider usage, the mentioned parameters of the code can be easily adopted for bigger data.

Also there is another issue related to the dataflow. In fact, there are several servers that can perform some algorithms to work with the data it stores. For example, the data can be filtered when it is no longer useful as well as it can find intersections between an infected and healthy users without utilizing the mobile app. However, these servers are non-free. Therefore all the algorithms were programmed in the app. So, due to the lack of these possibilities, the flow of the data is increased which consequently consumes the productivity of the application. In detail, the current app passes the data of the user to the database, and then downloads the data for all users for further purposes. So by setting a small time interval for processing the data, the workload on the server as well as the application significantly increases.

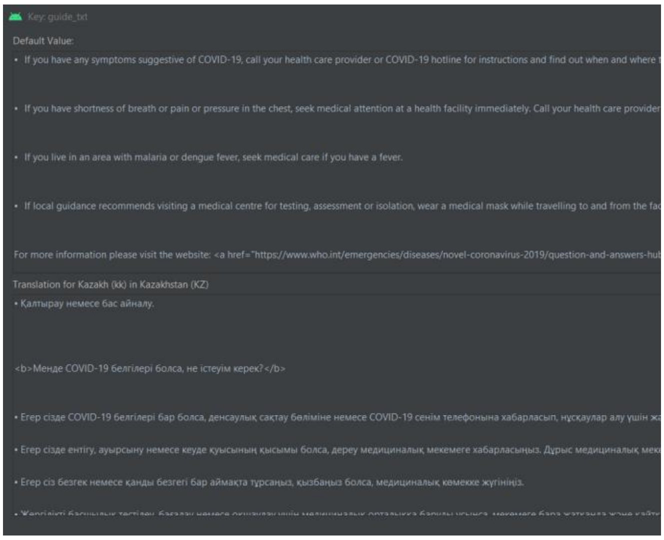


Figure 4.7.3: Extended view of string translation.

VI. Conclusion

In conclusion, the mobile application for warning users on COVID-19 was created. At the moment it is capable to:

- register user
- assign the user a unique user ID
- conduct singing in process
- show the route on the map
- show the user location on the map
- save user data in database
- retrieve data and compare it
- notify user if current user’s location data equals to the location data of an infected user
- provide guidance on the measures that should be take to prevent COVID infection, and on the measures that should be taken if the user is infected already
- open X-Ray analysing website, for indication of possibility of pneumonia
- run in Kazakh, Russian and English languages languages

The tests that were conducted have shown that the application works normally, without errors. This proves that it can be employed in life if some parts would be

Despite the mobile app contains the code on warning users of posibile infection, there are still several works that can be done in the future.

- add new features
- extent the capacity
- optimize the codes
- implement in global use

APPENDIX

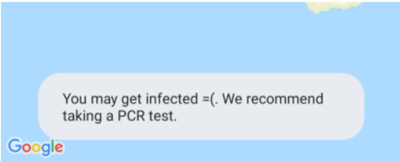
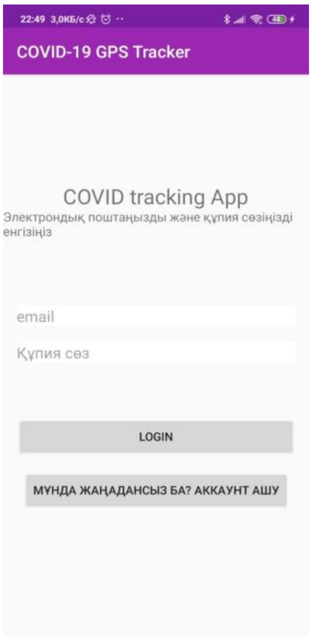


Figure A1. Dark-mode-off toast message.



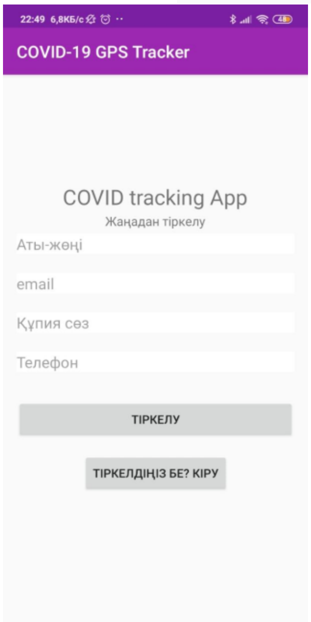
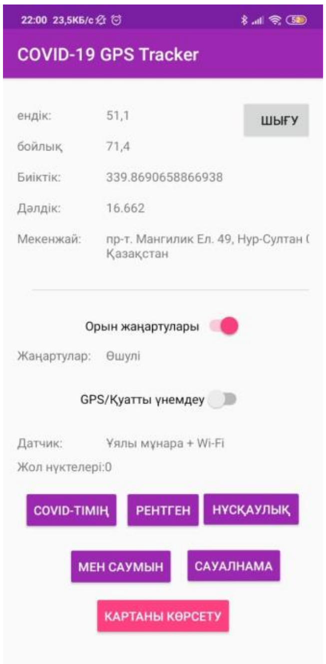


Figure A2. Kazakh translation of Registration and Sign in pages.



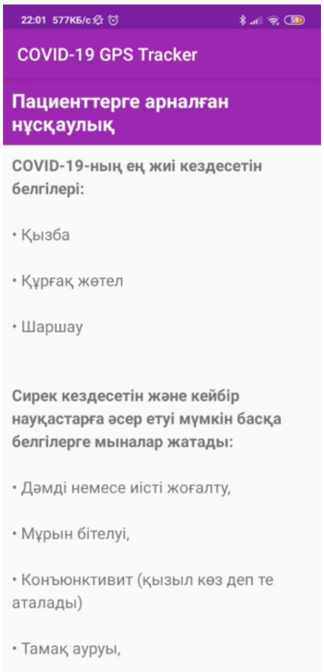


Figure A3. Kazakh translation of the Main page and Guide.

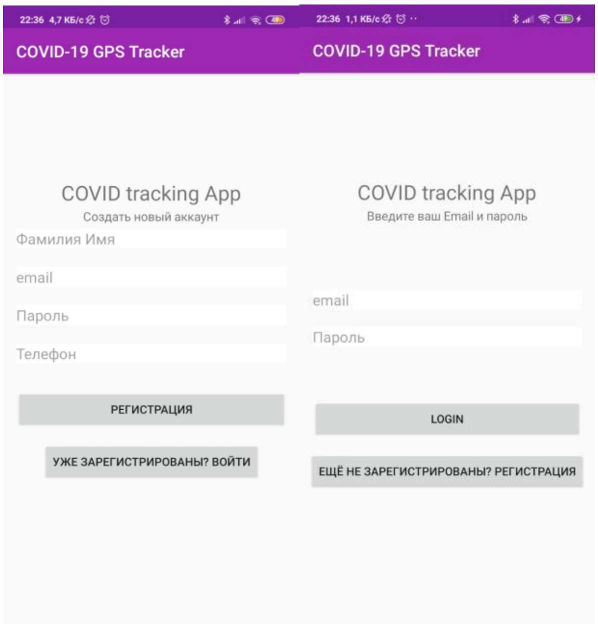


Figure A3. Russian translation of Registration and Sign in pages.

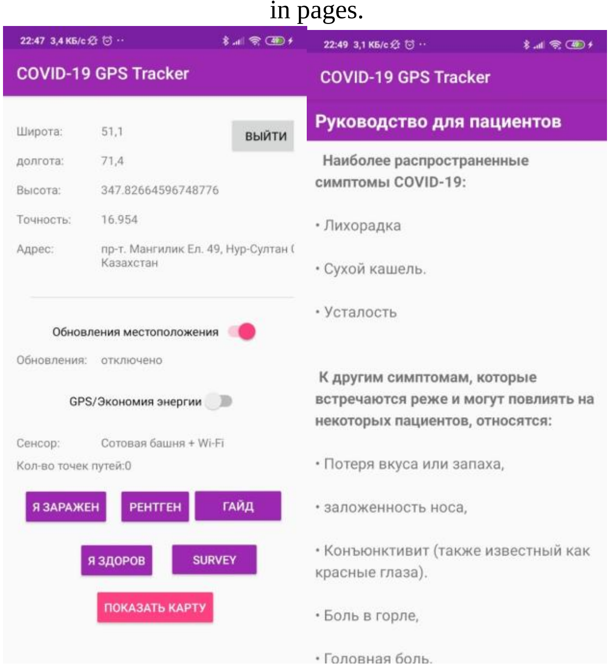


Figure A5. Kazakh translation of the Main page and Guide.

REFERENCES

[1] Arevalo, F. N. (2020, November). "Decoding the Public interest of Aarogya Setu, contact tracing app for managing the COVID19 pandemic in India". In *2020 IEEE International Symposium on Technology and Society (ISTAS)*, pp. 508-512. IEEE.

[2] Birkhoff, S., and Moriarty, H. (2020). "Challenges in mobile health app research: Strategies for interprofessional researchers". *Journal of Interprofessional Education & Practice, 19*, 100325. doi: 10.1016/j.xjep.2020.100325

[3] Chaturvedi, A. (2020). *Top 10 popular smartphone apps to track Covid-19*. Geospatial World. Retrieved from https://www.geospatialworld.net/blogs/popular-apps-covid-19/.

[4] Cioffi, A., Lugi, C., and Cecannecchia, C. (2020). "Apps for COVID-19 contact-tracing: Too many questions and few answers". In *Ethics, Medicine and Public Health*, 15, 100575. doi: 10.1016/j.jemep.2020.100575 https://www.sciencedirect.com/science/article/pii/S2352552520301134

[5] Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, *13*(3), 319. doi: 10.2307/249008

[6] Elkhodr, M., Mubin, O., Iftikhar, Z., Masood, M., Alsinglawi, B., Shahid, S., & Alnajjar, F. (2021). Technology, Privacy, and User Opinions of COVID-19 Mobile Apps for Contact Tracing: Systematic Search and Content Analysis. Journal of Medical Internet Research, 23(2), e23467.

[7] Filipets, V. (2020). *Android vs iOS Development - What and Why is Better? [2020 UPDATE]*. Theappsolutions.com. Retrieved 6 November 2020, from https://theappsolutions.com/blog/development/ios-vs-android/.

[8] Garrett, P. M., White, J. P., Lewandowsky, S., Kashima, Y., Perfors, A., Little, D. R., ... & Dennis, S. (2021). The acceptability and uptake of smartphone tracking for COVID-19 in Australia. *Plos one*, *16*(1), e0244827.

[9] Hess, B., Farahani, A. Z., Tschirschnitz, F., & von Reischach, F. (2012, November). Evaluation of fine-granular GPS tracking on smartphones. In Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems (pp. 33-40).

[10] Holtz, B., Murray, K., Hershey, D., Dunneback, J., Cotten, S., & Holmstrom, A. et al. (2017). Developing a Patient-Centered mHealth App: A Tool for Adolescents with Type 1 Diabetes and Their Parents. *JMIR Mhealth And Uhealth*, *5*(4), e53. doi: 10.2196/mhealth.6654

[11] Ilchenko, V. (2020). Most Popular Medical Apps for Doctors and Patients. *ByteAnt*. https://www.byteant.com/blog/most-popular-medical-apps-for-doctors-and-patients/

[12] Josna, R. S., Karkera, S., Sagar, D., & Pawar, S. COVID Tracker–Using GPS Tracking & Image Capture.

[13] L. Neubeck, N. Lowres, E. J. Benjamin, S. B. Freedman, G. Coorey, and J. Redfern, "The mobile revolution—using smartphone apps to prevent cardiovascular disease," *Nature Reviews Cardiology*, vol. 12, no. 6, pp. 350–360, 2015. Doi: 10.1038/nrcardio.2015.34

[14] M. Ponum, O. Hasan, and S. Khan, "EasyDetectDisease: An android app for early symptom detection and prevention of childhood infectious diseases," *Interactive Journal of Medical Research*, vol. 8, no. 2, 2019. Doi: 10.2196/12664

[15] Mallik, R., Sing, D., & Bandyopadhyay, R. (2020). GPS tracking app for police to track ambulances carrying COVID-19 patients for ensuring safe distancing. *Transactions of the Indian National Academy of Engineering*, *5*, 181-185.

[16] Manoharan, S. (2009, April). On GPS tracking of mobile devices. In *2009 Fifth International Conference on Networking and Services* (pp. 415-418). IEEE.

[17] Moloo, R. K., & Digumber, V. K. (2011, July). Low-cost mobile GPS tracking solution. In *2011 International Conference on Business Computing and Global Informatization* (pp. 516-519). IEEE.

[18] N. Shah, J. Jonassaint, and L. De Castro, "Patients welcome the sickle cell disease mobile application to record symptoms via technology (smart)," *Hemoglobin*, vol. 38, no. 2, pp. 99–103, 2014. Doi: 10.3109/03630269.2014.880716

[19] Nageshwaran, G., Harris, R. C., & Guerche-Seblain, C. E. (2021). Review of the role of big data and digital technologies in controlling COVID-19 in Asia: Public health interest vs. privacy. Digital Health, 7, 20552076211002953.

[20] O'Neill, P., Ryan-Mosley, T., & Johnson, B. (2020). A flood of coronavirus apps are tracking us. Now it's time to keep track of them. *MIT Technology Review*. Retrieved from https://www.technologyreview.com/2020/05/07/1000961/launching-mittr-covid-tracing-tracker/

[21] Paiva, S., & Abreu, C. (2012). Low cost GPS tracking for the elderly and Alzheimer patients. *Procedia Technology*, *5*, 793-802.

[22] Pandria, N., Spachos, D., & Bamidis, P. D. (2015). The future of mobile health ADHD applications. 2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL). doi:10.1109/imctl.2015.7359603

[23] Patel, R. (2014). *Design for Use and Acceptance of Tracking Tools in Healthcare* (Doctoral dissertation). University of Washington, Seattle, WA. http://hdl.handle.net/1773/25153

[24] Privacyinternational. (2020). GPS tracking and COVID-19: A tech primer. Retrieved from https://privacyinternational.org/explainer/3753/gps-tracking-and-covid-19-tech-primer

[25] Rakshit, S., Islam, N., Mondal, S., & Paul, T. (2021). Mobile apps for SME business sustainability during COVID-19 and onwards. Journal of Business Research, 135, 28-39

[26] Tran, C. D., & Nguyen, T. T. (2021). Health vs. privacy? The risk-risk tradeoff in using COVID-19 contact-tracing apps. Technology in Society, 67, 101755.

[27] Verma, S. (2020). *Aarogya Setu now world's most downloaded covid-19 tracking app*. India Today. Retrieved from https://www.indiatoday.in/technology/news/story/aarogya-setu-now-world-s-most-downloaded-covid-19-tracking-app-1701273-2020-07-16.

[28] Vladoescu, C., Tunea, M., & Stanciu, L. (2020, October). Benefits of Using Mobile Applications to Keep Under Control, Detect, Attenuate and Monitor COVID-19 Pandemic. In 2020 International Conference on e-Health and Bioengineering (EHB) (pp. 1-4). IEEE.

[29] W. Kimpan and P. Sirivimonsattaya, "Using mobile application as an instrument for ptosis diagnosis," *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2017. Doi: 10.1109/iciibms.2017.8279739

[30] W. Liu, L. Yuhang, P. Lingling, and C. Na, "Design and implementation of family doctor app on android platform," *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, 2018. Doi: 10.1109/itme.2018.00038

[31] Zeinalipour-Yazti, D., & Claramunt, C. (2020, June). Covid-19 mobile contact tracing apps (mcta): A digital vaccine or a privacy demolition?. In 2020 21st IEEE International Conference on Mobile Data Management (MDM) (pp. 1-4). IEEE.

[32] J. P. Cohen, P. Bertin, and V. Frappier, "Chester: A Web Delivered Locally Computed Chest X-Ray Disease Prediction System," *arXiv:1901.11210 [cs, q-bio]*, Feb. 2020, Accessed: Nov. 24, 2021. [Online]. Available: http://arxiv.org/abs/1901.11210.

[33] L. de Oliveira Neto, V. D. de Oliveira Tavares, F. B. Schuch, and K. C. Lima, "Coronavirus Pandemic (SARS-COV-2): Pre-Exercise Screening Questionnaire (PESQ) for Telepresential Exercise," Frontiers in Public Health, vol. 8, p. 146, 2020, doi: 10.3389/fpubh.2020.00146.

[34] skooltch84, *TrueOrFalse1*. 2019. Accessed: Nov. 24, 2021. [Online]. Available: https://github.com/skooltch84/TrueOrFalse1/blob/78104aa730dad07fd20ba2bf7edfd5cdbc07993f/app/src/main/java/com/blogspot/skooltchdev/trueorfalse1/ResultsActivity.java