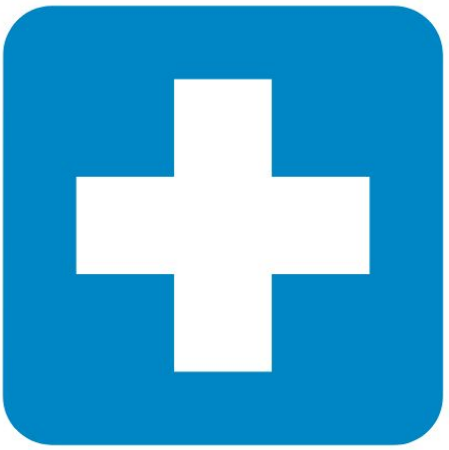


Handover Documentation

Disease Outbreak Manager

Group 2 - Trinity College Dublin.

NOTE: Read the entire document before carrying out any steps.

Backend instructions

Prerequisites:

- Firebase account and project needs to be set up at https://firebase.google.com/
- Node.js (https://nodejs.org/en/) and Yarn (https://yarnpkg.com/) installed

Set up required

1. Firebase service account file needs to be created as per https://firebase.google.com/support/guides/service-accounts
   Service account credentials file needs to be saved inside the backend folder.

2. Everything between
   ----------HANDOFF BEGIN----------
   and
   ----------HANDOFF END----------
   needs to be replaced with project specific information provided when setting up a Firebase account.

   Affected files:
   - backend/views/admin_main.hbs
   - backend/views/login.hbs

3. Modify the .env.sample file in the backend folder
   - PORT: port for server to run on
   - DB_URL: Firebase database URL
   - INITIAL_PASSWORD_LENGTH: Length of generated password for accounts
   - SETUP_KEY: Secret key to create admin account on first setup
   - SERVICE_ACCOUNT_FILE: Location of service account credentials file relative to the backend folder
   - EMAIL_EXTENSION: Email extension for admin and worker accounts (does not have to be real)

   Then rename .env.sample to .env

   Affected files:
   - backend/.env

4. While inside the backend folder, run the command: `yarn`
5. To start the server, run the command: `yarn run start:prod`

6. To create a new admin account, navigate to http://<YOUR_URL_HERE>/setup/new/<SETUP_KEY> in the browser, where YOUR_URL_HERE is the domain where the server is deployed and SETUP_KEY is the secret key from the .env file. The first line of the response is the username, the second line is the password. Delete the SETUP_KEY line from .env and restart the server. DO NOT leave the SETUP_KEY variable empty in the .env file instead of deleting the line.

   Affected files:
   - backend/.env

Android app instructions

Deployment

The app code is stored in frontend_app. A Firebase account and project needs to be set up. Step 1-3 of option 1 of the Android setup needs to be completed https://firebase.google.com/docs/android/setup. Some substeps of step 3 might have already been completed in the project.

The release APK of this application is stored at trocaire/frontend_app/app/release.
Here a signed APK is located and the following signed credentials are below for it:

- Key: key0
- Password: trocaireDOM

The key is located in the keystore at trocaire/frontend_app/Android_app_apk
These are required to upload the application to the play store.

To view the code - Android Studio is required - https://developer.android.com/studio

- Open the application up with android studio and this can be run locally through an emulator on the computer or exported to an android phone.

The following needs to be changed:

- Upload URL at line 120 frontend_app/app/src/main/java/com/example/trocaire_disease_outbreak_manager/View_patient_details.java needs to be changed to the URL where the backend server lives.
- auth_email_extension string at line 80 in frontend_app/app/src/main/res/values/strings.xml needs to be changed to the same value as EMAIL_EXTENSION in the backend .env file.

FAQ

How to add another language into the application?

Follow the outlines provided here - https://developer.android.com/guide/topics/resources/localization
This involves creating another Strings.xml file. This will be a mirror of the other Strings.xml files within this project, but with the new languages inserted into the xml slots.

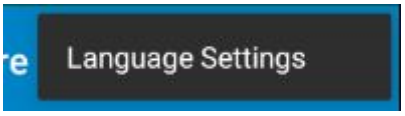
Once complete, you need to go to `/trocaire/frontend_app/app/src/main/java/com/example/trocaire_disease_outbreak_manager`.
Here another button needs to be inserted to allow the user to select the new language.
Within the OnCLick method add and extra case statement as such:

```
 case R.id.imageButtonNew:
                Toast.makeText(getApplicationContext(),
                        R.string.lang_selected, Toast.LENGTH_SHORT)
                        .show();
                settings =
getSharedPreferences("com.example.trocaire_disease_outbreak_manager",0);
                editor = settings.edit();
                lang = "ADD STRING.XML NAME HERE";
                editor.putString("lang",lang);
                editor.apply();

                break;
```

How to use the Android Application

The home screen is split into 3 sections, Adding, Updating and Saving patient details.
Clicking the three dots on the top right corner reveals the language settings



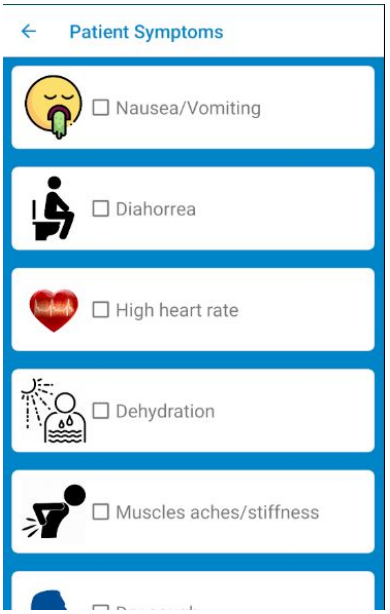
From here the user can select their language preference.

Adding a patient.

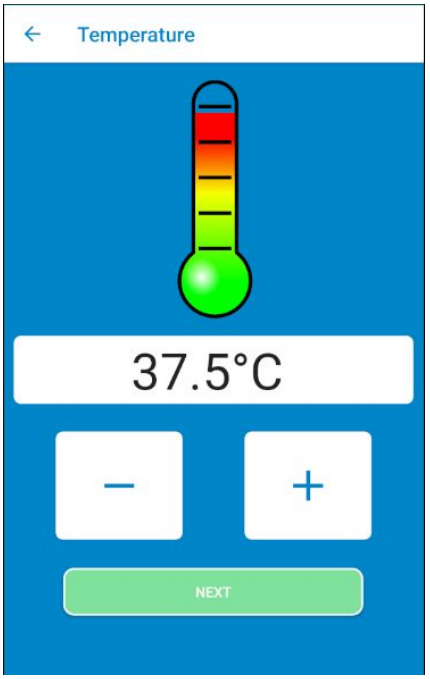
The user clicks the + icon on the add patient details page.

This will prompt a page where the user types in the basic details of the patient such as their name, Date of Birth, Village and Sex:

Next a scrolling list of common symptoms are shown. These have images to depict the words making it easier to pick what the patient has. The user click on the symptoms that tha patient has.

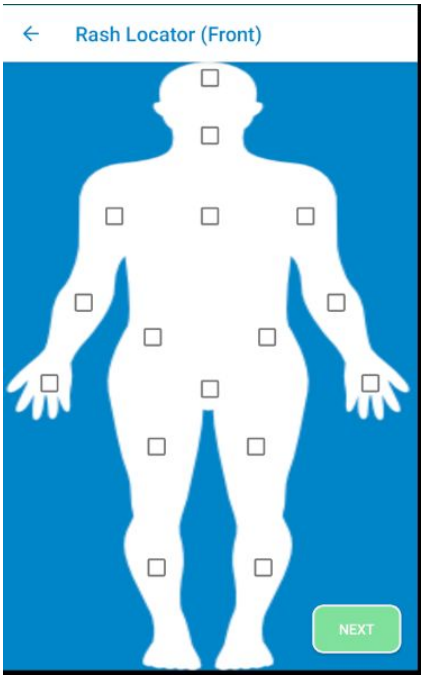


Next the patient's temperature is then recorded. These can be typed manually or by using the + and - buttons to move the temperature by 0.5 degree intervals.



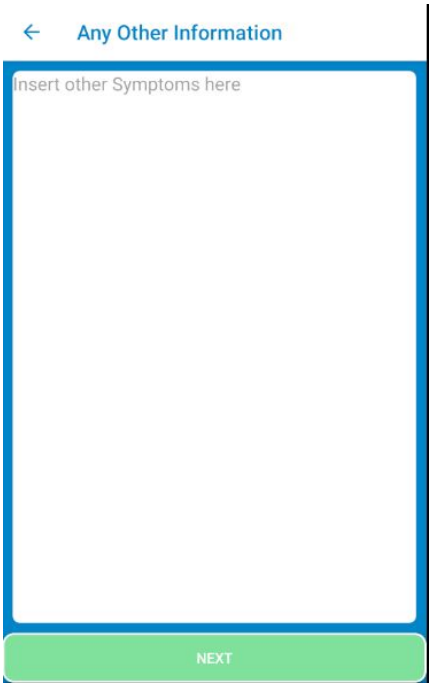
If the user selected a that the patient has a rash from the symptoms page before, an identification page of the rashes is shown to the user. This allows the user to identify what type of rash the patient has.

Next a body is shown. This is to locate the rashes on the patients box. The user clicks where the rashes are located on the front and back of the patient.



Next the pain locator and pain level page is shown. Similar to the rash location page, the user clicks where the pain is located. At the bottom of this page, A scale can be moved to how the patient is feeling from low to a high pain level.

Finally, if the user has any more information about the patient they can insert it into a textbox which will be included into the patients report.



Once completed, the user may be asked to provide location information, Select yes to these requirements as it allows for more accurate statistics to be gathered for the web application.

The user is then prompted with a screen displaying all their inputted information about the patient. This information is stored locally on the phone until the user offloads them in a wifi hotspot area.

Updating patient details

The middle tab on the home screen is for updating the saved patient details. Here the user can select the patient to be updated or removed from the system. The user is then brought through the same pages as adding a patient but all the fields are populated with that patients information.

Offloading patient details

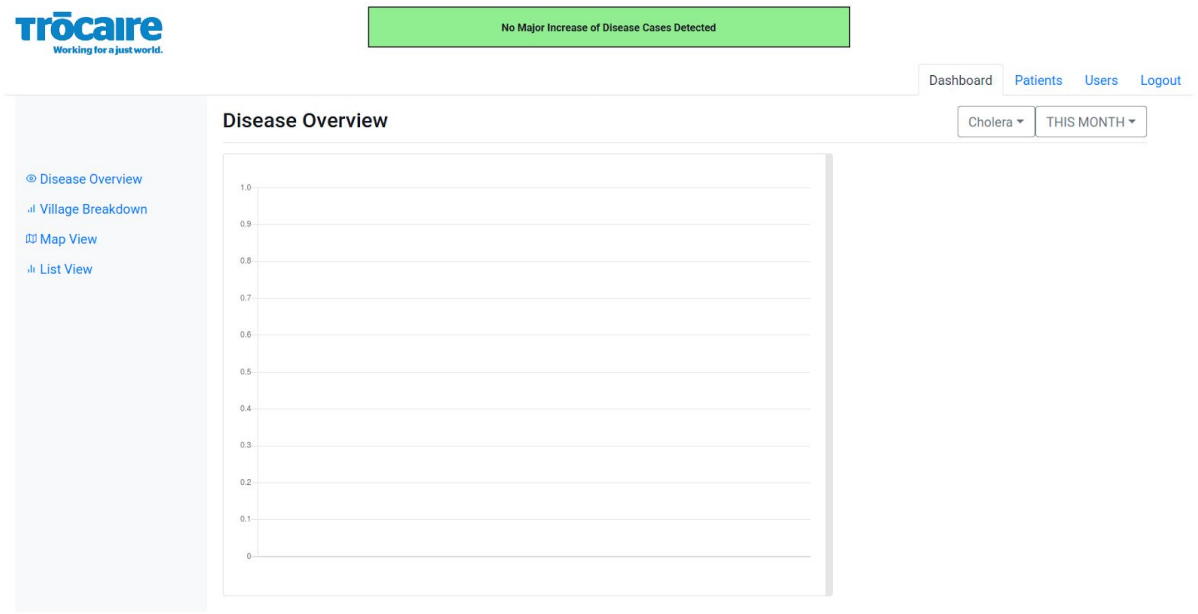
Once the user is in a wifi hotspot, they can offload the patient details to the database. This is located on the third tab of the home screen. Once selected, the user is prompted to sign in with their user id and password. Once successfully logged in, the user can view the patient details before they are set. This allows for a final check of details before sending them. Once sent, the current patient data is then cleared from the phone.

Web app instructions

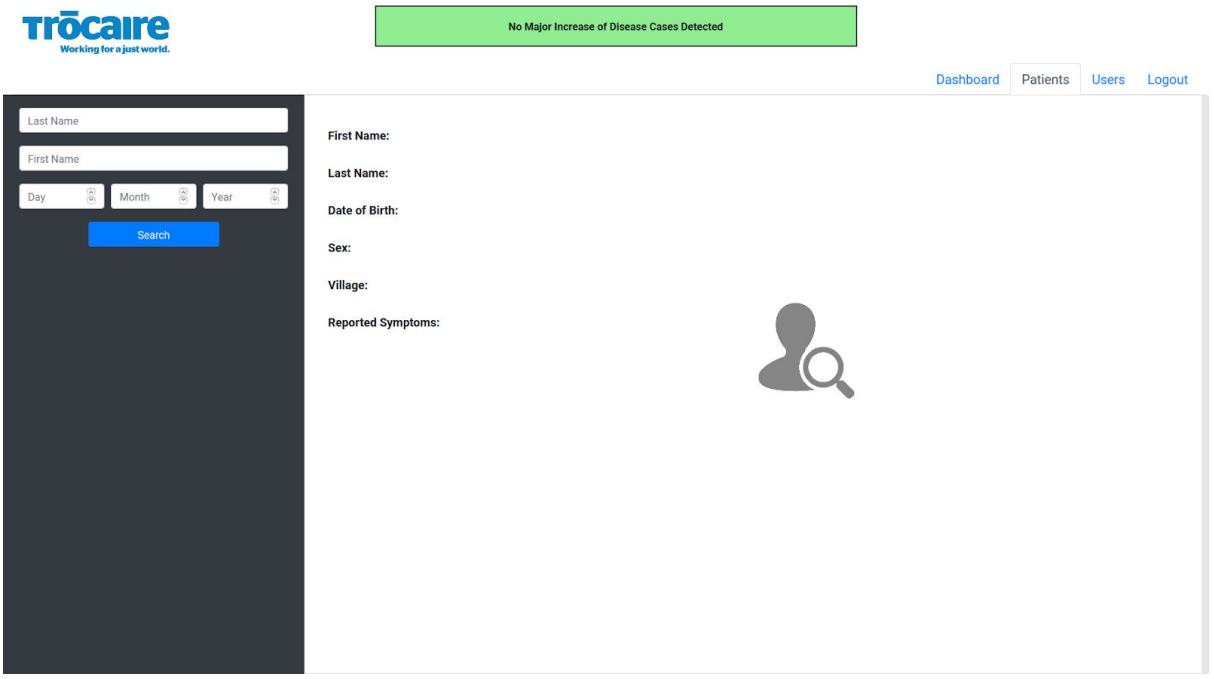
The admin website code is stored in the ‘static’ and ‘views’ subfolders of the ‘backend’ folder. The html files (with .hbs file extension) are found in the ‘views’ folder. The ‘admin_main.hbs’ file contains the main parts of the website, where everything is added to it.

The username and password that are used to test the website are as follows:

- Username: 00123
- Password: 000123



The above web page is the main part of the website. It gives analysis and insights into any incoming data from the Community Healthcare Workers.



The above web page searches for patients in the system’s database. A warning message will appear if no patient is found in the database.

The above web page allows admins to create a new admin and health worker account with automatically-generated passwords.

The box, that is currently green, which appears in all the webpages analyse any incoming live data from the healthcare workers to give quick warnings to the admins. Such warnings include: green box with message 'No Major Increase of Disease Cases Detected', orange box with message 'There is an Increase of ____ Cases Detected. Please Check Dashboard', and the red box with the message 'There is a High Outbreak of _____ Detected. Please Check Dashboard'.