RADAR SOLUTIONS AND TECHNOLOGIES

1. ANDROID APPLICATION

This is main Radar Android application solution that users interact with and get to do self-assessment diagnosis and also the entry of the past 14day itinerary with the help of google search maps

Technologies used include:

- Android Studio v3.6.3 and above
- Java
- Firebase
- Realm 6.0.2
- Guava 28.0-android
- Min SDK version 19 (Android 4.1.x Jelly Bean)
- Android SDK
- Google APIs (Maps SDK for Android, Places API)

2. WEB DASHBOARD APPLICATION

This is the application that shows the data collected and some filtration feature to get specific insights a user might need

Technologies used include:

- Radar APIs
- VueJs and Bootstrap v4
- Laravel PHP Framework
- Vue JS
- MySQL

3. BACKEND

Both the Android and web dashboard applications receive and push data to the backend through REST APIs.
USSD on the other hand does not use the API's processing capability thus it has a normal backend developed in python.
The WhatsApp Bot backend also receives and pushes the information to and from the user interacting with it on the device.

Technologies used on the backend include:

- Python (Django Framework)
- Django-rest framework
- Database (PostgreSQL)

Renewable technologies:

- Cloud Server (AWS EC2 Instance)
- AWS SSL Certificate, Loadbalance and Route 53
- WSGI server (Gunicorn)
- Web Server (Nginx)

4. **WHATSAPP BOT**

This is another Radar solution that can be used for self-assessment and diagnosis and based the symptoms entry we use the APIs

Technologies use includes:

- Laravel PHP framework
- Twilio API
- MySQL
- HTML and CSS3 for the frontend
  NB: The Whatsapp bot is not yet approved by Twilio so it's not live.

5. **USSD APPLICATION**

This is a text input and display solution that enables quick and ease of interaction of users with the solution. They can trigger it and do a self-assessment and based on the symptoms entry they get recommendations on the best actions to take.

Technologies used include:

- The solution is integrated with Africastalking’s USSD solution as the service shortcode provider, they are responsible for pushing and handling of the user sessions
- Every entry during the session is pushed to our backend and based on the processed response the user can continue to interact with the system or the session can be terminated once the USSD flow has been completed