

ReadMe for FHIRWearks: GOSH-FHIRworks 2020 hackathon submission

Alex Tcherdakoff

Contents

1	Summary and Architecture	1
2	Features	1
3	Demos	1
4	Installation Guide	2
5	Future Plans	2
6	Acknowledgements	2

1 Summary and Architecture

FHIRWearks is a Samsung Watch application inspired by the FHIRFLI package that sends patient heartrate data to FHIR.

It is built in JavaScript, HTML, and CSS, and relies on a specially written Python Flask API (backend) running on a Microsoft Azure Linux Virtual Machine Server.

It was built and tested using the Samsung Tizen IDE (based on Eclipse) which is proprietary to Samsung Gear development, and the json-to-fhir toolkit.

2 Features

The application starts with a menu that allows the user to choose between "Send to FHIR", and "Settings".

If they choose to "Send to FHIR" before choosing "Settings", the page will prompt them to input their personal info in the "User Info" page of "Settings".

The default sending mode is manual.

If the user info is input incorrectly (if there was a typo or the name and date of birth input don't match FHIR records) or the heartrate could not be measured (like if the user is trying to use the app while charging, or is holding the watch away from their wrist), the app will inform them of this.

This page gets the Patient ID from FHIR and posts an Observation with their heartrate to FHIR.

In "Settings/Send Mode", the user can leave the mode as manual, or set it to automatic hourly, daily, or weekly.

The latter three will prompt the app to send push notifications regularly to remind the user to send their heartrate to FHIR.

Checking any of the options also returns the user to the Settings page.

The next page is "Settings/User Info", which allows the user to input and submit their first name, last name, and date of birth. This will allow the app to search FHIR for their Patient ID.

The last page is the "Settings/App Info" page: it summarizes app functionalities, explains certain specific features, and lays out the plan for future work.

Each page has a "Home" or "back" button, and the Samsung Watch "Back" button can also be used everywhere.

3 Demos

You will find three demos for this app:

30-second functionality and proof:

<https://youtu.be/53w2R1J4xg4>

Full features demo:

<https://youtu.be/0u6D5ViQoE4>

Setup and environment, and architecture:

https://youtu.be/dFxG_mPa3Q4

4 Installation Guide

You can download Tizen and open the workspace to run the app in the emulator, or if you have a Samsung Watch, simply switch the device to debug mode and make sure it is connected to the same WiFi network as your computer. You may have to make sure you have the correct certificates to launch (use the certificate generator to create one if not).

Then, open the remote device manager, and search for your watch's IP address. You can now run the app on your watch!

Otherwise, you can also load the app via SDB in the command line, or use an emulator, or even test remotely on Remote Test Lab: find more details here:

<https://developer.samsung.com/galaxy-watch-develop/testing-your-app-on-galaxy-watch.html>

Note: this project is compatible with devices running Tizen 4.0 and up: this means the Galaxy Watches and Galaxy Watch Actives.

5 Future Plans

This particular watch was chosen because it looks like it will be the first smartwatch to calculate blood pressure: you can already measure your blood pressure with the Samsung Watch Actives and a Samsung Galaxy phone past S9 with the MyBP app if you are above 18 and would like to participate in the study. This feature will be useful as Blood Pressure is one of the only Observations in FHIR that can be written to.

I'd also like to add a companion Android App to turn off automatic push notifications if a user loses their watch (i.e. left it at home, lent their watch out, their watch was stolen...)

In actual practise, these changes to FHIR would have to be approved by a medical professional before they were to be written permanently, so ideally I'd also write an app for a medical professional to sign off on the observations being sent.

6 Acknowledgements

I'd like to thank Ethan Wood for his FHIR-parser

(<https://fhir-parser.readthedocs.io/en/latest/getting-started.html>),

Dao Hang Liu for his FHIR Endpoint

(<https://fhir.compositegrid.com:5001/api/Patient>),

and Rajesh Goyal for his observation posting algorithm

(<https://github.com/GCHeroes1/GOSH-FHIRworks2020-SQL>).

I'd also like to thank Rikaz Rameez and Zak Morgan for helping me film, debug, being excellent rubber ducks, and beacons of common sense at times.