Please make a copy of this document and include this in your GitHub repository for your submission, using the tag #AndroidDevChallenge

Tell us what your idea is.

Describe in 250 words what the feature or service will do and how you'll use Machine Learning to push the bar:

In our everyday life it is a common issue to be exposed to extreme stress. A handheld mobile **app for instant stress detection**, using a heart rate sensor of a smartwatch that continuously monitors physiological body response to stressors, can help us better understand our own life circumstances and, ideally, trigger behavior changes which will result in a healthier life.

Such an app using smartwatches for passive sensing is easier to use than the common medical devices. Users only need to give short reports on their perceptions of stress levels during their stressful life events so that the system can learn and provide personalized feedbacks.

Heart rate variability (HRV) is a good indication of how much stress the body is exposed to. It is, however, challenging to build a HRV based stress detection model:

- Depending on personal circumstances such as body weight, age, or experiences, each individual may have different physiological responses to stressors.

The **On-Device Machine Learning** with **TensorFlow Lite** may provide solid solutions to the aforementioned issue:

- During the device charging time an individual model for stress detection based on the past data samples can be downloaded from a ML service backend and be converted to a TensorFlow lite model for the future inference.

The realization of this suggested app may not only help gain insight into individual biofeedback such as stress, but also contribute to find the underlying relationship between body and mind. It helps push medical innovation through technology.

Tell us how you plan on bringing it to life.

Describe where your project is, how you could use Google's help in the endeavor, and how you plan on using On-

Device ML technology to bring the concept to life. The best submissions have a great idea combined with a concrete path of where you plan on going, which should include:

- *(1) any potential sample code you've already written,*
- *(2) a list of the ways you could use Google's help,*
- *(3) as well as the timeline on how you plan on bringing it to life by May 1, 2020.*

1 Status of Sample Codes:

This project has been carried out as a personal PhD research project so far. The current app implementation consisting of an **android handheld app** and a **Wear OS smartwatch companion app**, is deployed in **Play Store as Beta App** for testing and research purpose only (https://play.google.com/store/apps/details?id=lmu.pms.stila).

The following features are already implemented and function in the current app code template:

- Measuring heart rate through watchfaces with 1, 5, or 10 secs measuring interval
- Synchronization heart rate data between smartwatch app and handheld app
- Fetching heart rate data from Fitbit cloud if users prefer to use a Fitbit PurePulse Wristband instead of a Wear OS Smartwatch for passive sensing.
- Collecting user's perceived stress level with a self-report on the smartwatch app or the handheld app
- Displaying stress scores with computed stress measurement based on estimates of heart rate variability through a statical coded inference model (which can be improved through On-Device ML with TensorFlow Lite and Firebase ML custom model hosting)

2 Google can help me bring app to life with the following MoSCoW Priorization:

Must have:

- Help me reach out to more people with my app concept
- Assign me a mentor for weekly or bi-weekly coding and information exchange
- Support me with Cloud resources for building TF model and give me hints on how to load this model into Firebase via service API for every app user, if it is possible
- Give me hints on how to upload custom TF model into firebase through services API and convert custom TF model to TFLITE model in FireBase and deploy this model silently to my app for instant stress detection

- Give me advices on how to use work manager in android app to trigger background jobs
- Give me insides into Heart Rate Sensor API while using SensorManager with the fastest delay setting and the relating infos of the SensorEvent API to the underlying logic of Wear OS platform in order to valid whether heart rates can be collected fastly and accurately.

Should have:

- Help me better understand the Data Link Layer API of Wear OS
- Give me advice on Android Profiling and Testing using Dependency Injection with Dagger1/2
- Help me with the Watch Face optimization
- Give me advice on App packaging using Gradle with multi Flavour or SourceSets settings
- Arrange a meeting with NLP researchers having knowledges on building personlized MLP or RNN models for personalized language recognition models, so that I can develop a generalized MLP or RNN stress detection model with personalization bits in the input vector

Could have:

- Give me early access to Wear OS Tiles API
- Give me advice on how to deploy A/B testing for Play Store using Google Play console
- Give me hints on how to use offline Google Assistant to call an android deep link

Won't have:

- Help me train a ML custom model directly on users' handheld devices during the charging time that would not require me to host the .tflite mode on Firebase
- Help me build a generalized model for stress detection using federated learning
- Help me build a Google Assistant Custom Action using offline Google Assistant to give users further hints on their current stress levels and provide health related suggestions through voice interaction

3 Project Time Lines:

now – 15.Dec.2019	Refactoring the current sample source code for further extention and enabling the fetching of users' demographical data
16. – 21.Dec.2019	On-Device ML concept evaluation via implementing simple testing projects (grateful for possible guidance from a google technology mentor via email or skype)

22.Dec.2019 – 8.Jan.2020	No activity during chrismas holidays (for work-life balance, via phone call available)
9.Jan – early.Feb.2020	Sample source code improvement
Early.Feb – 15.Mar.2020	Building a TF model based on my personal training data with the google tensorflow cloud api, and transfer the TF model for stress detection to Firebase ML service. Finally deploy the TFlite file in the testing app for inference to replace the current statical hard coded model. Alpha Deployment in Playstore.
16.Mar. – 31.Mar.2020	Finding a solution with Google Experts for On-Device ML and with Researchers a way to generalize the TF custom model with personalization bits in the input or to build personalized TF custom model for each user. (to be discussed due to technical possibility)
01.Apri-15.April.2020	Performance Testing and Testing Capacity of upscaling to large user base
15.April.-25.April.2020	Bug fixing of „Must have“ features or further planning and implementation of „should have“ features. Play Store Deployment
25.April.-01.May.2020	Bug fixing or Further Idea development of the app. (Ready for show down)

Tell us about you.

A great idea is just one part of the equation; we also want to learn a bit more about you. Share with us some of your other projects so we can get an idea of how we can assist you with your project.

I started my journey as an Android developer in 2011 and currently I am a PhD student and research associate of computer science at Ludwig Maximilian University of Munich Germany. My research interests are time series data analysis, persuasive and pervasive systems, human computation, information retrieval, affective computing, and behavior analysis. More about me is available on LinkedIn.

I have been working on various different projects (solely responsible or in cooperation with partners):

- Research cooperator in the topic “Characterizing the mind-body system for enhancing neurofeedback with a HRV biofeedback” with CANLAB, Department of Psychiatry and Psychotherapy University of Tuebingen Germany
- Main developer and idea holder of project Stila for mental stress detection and behavior recommendation
- Building a vertical search engine for MOOC (Massive Open Online Courses) recommendation with TensorFlow, Angular2+, Play Framework, MongoDB and Flask

- Research Supervisor of project CodeKoan: a source code pattern search engine extracting programming patterns from Stack Overflow
- Winner of the 1st Prize in Hack Roboy Hackathon 2016 using Affectiva SDK to recognize user emotion through facial expression and replicate the same emotion on the face of Roboy robot to simulate empathy
- Developer of Alexa skill Tierlaute for imitating an animal sound

Next steps.

- Be sure to include this cover letter in your GitHub repository
- Your GitHub repository should be tagged #AndroidDevChallenge
- Don’t forget to include other items in your GitHub repository to help us evaluate your submission; you can include prior projects you've worked on, sample code you've already built for this project, or anything else you think could be helpful in evaluating your concept and your ability to build it
- **The final step is to fill out this form to officially submit your proposal.**