```
/*
 * COPPE / UFRJ - Universidade Federal do Rio de Janeiro
 * PESC - Programa de Engenharia de Sistemas e Computacao
 *
 * Disciplina: CPS730 - Internet das Coisas
 * Professor: Claudio Miceli
 *
 * Trabalho Final - Emergency Sound Detection
 * Data: 21/AGO/2022
 *
 * Nome: Luiz Marcio Faria de Aquino Viana, M.Sc.
 * Formacao: Engenheiro Eletricista com Enfase em Engenharia de Sistemas e Computação
 * DRE: 120048833
 * CPF: 024.723.347-10
 * RG: 08855128-8 IFP-RJ
 * Registro: 2000103581 CREA-RJ
 * E-mail: luiz.marcio.viana@gmail.com
 *         lmarcio@cos.ufrj.br
 * Tel.: +55-21-99983-7207
 *
 * 002-EmergencySound_Components.txt
 */
```

I. Emergency Sound Detection Mobile Application
--

#1. INITIALIZATION

- Load initial parameters
- Connect to the server using the MAC ADDRESS
- Request the sound classification models
- Save the updated sound classification models localy

- Create the sound classification model table

+ Enviroment Sound Detection Models
+ Emergency Vehicles Sound Detection Models
+ Gunshot Sound Detection Models
+ Urban Sound Detection Models

- Start application threads

+ Thread 1: Emergency Sound Detection
+ Thread 2: Model Update Requester

#2. THREAD 1: EMERGENCY SOUND DETECTION

- Detect sounds above a threshould
- Save a snapshot with 30 seconds of the detected sound
- Save the current GPS position if available
- Save the sound metadata localy (XML file)
- Pre-classify the sound using TinyML in the classification model table
- If the sound is an EmergencySound, then...
- Send SOUND and SOUND METADATA to server

#3. THREAD 2: MODEL UPDATE REQUESTER

- Load initial parameters
- Start Model Update Requester
- Pool the server for an updated model
- If has any updated model in the server, then...
- Load the updated model in background
- Block model table access for update
- Update the sound classification model table
- Release model table access for processing

II. Emergency Sound Detection Server Application
--

#1. INITIALIZATION

- Load initial parameters
- Load accurated sound classification model table
- Load TinyML sound classification model table

- Start application threads

+ Thread 1: Emergency Sound Detection Listener
+ Thread 2: Model Update Listener
+ Thread 3: Local Server Model Update

#2. THREAD 1: EMERGENCY SOUND DETECTION LISTENER
- Wait for detected EmergencySound
- If received SOUND + SOUND METADATA, then...
- Classify the sound using accurated classification model table
- If the sound is an EmergencySound, then...
- Save SOUND and SOUND METADATA to Database

#3. THREAD 2: MODEL UPDATE LISTENER
- Load initial parameters
- Start Model Update Listener
- If receive a updated model request, and...
- Has any updated model in the server, then...
- Send the updated model to the mobile application

#4. THREAD 3: LOCAL SERVER MODEL UPDATE
- Load initial parameters
- Set server model update Timeout (= 60 seconds)
- If more models then a threashold is marked for update, then...
- Update the local accurated models and TinyML models
- Clean the updated mark in the classification table
- Set the update request mark in the classification table

III. Emergency Sound Model Trainning
--
#1. INITIALIZATION
- Load initial parameters
- Load the sound dataset

- Start application threads
+ Local Server Model Trainning

#2. THREAD 1: LOCAL SERVER MODEL TRAINNING
- Set server model update Timeout (= 60 seconds)
- Select a random dataset (+/-10%) from the all saved sounds
- Trainning using random part of the selected dataset (+/-90%)
- Test using random part of the selected dataset (+/-10%)
- If predictor is better then previous loaded predictor, then...
- Update the model localy
- Generate the TinyML model localy
- Mark the model as updated in the classification table
- If more then a threashold is marked for update

IV. Emergency Sound Dashboard
--
#1. INITIALIZATION
- Load initial parameters
- Start map server
- Show in the map the last 10 minutes of emergency sound detected

#2. QUERY OPTIONS
- Show in the map the last 30 minutes of emergency sound detected
- Show in the map the last 1 hour of emergency sound detected
- Show in the map the last 24 hours of emergency sound detected
- Show in the map the last 1 week of emergency sound detected
- Show in the map the last 1 month of emergency sound detected

#3. LIST EMERGENCY SOUND METADATA
- GPS Position
- Pre-Classification score
- Classification score
- Audio link