

Mobile and cyber-physical systems project specification

Dalla Noce Niko, Ristori Alessandro.

Master Degree in Computer science.

n.dallanoce@studenti.unipi.it, a.ristori5@studenti.unipi.it.

Mobile and cyber-physical systems, Academic Year: 2020/2021

Date: 31/03/2021

Team 8



Abstract

The document presents the problem we wish to address, a possible use case with the specifications necessary for the implementation of the project; there will also be a short paragraph in which we will talk about how we intend to carry out our idea.

1 Problem description

Le regolamentazioni vigenti riguardo la conservazione dei generi alimentari impone ad ogni commerciante l'obbligo di mantenere aggiornato un registro giornaliero contenente le temperature delle celle frigorifere usate; in caso di mancato rispetto, il commerciante è tenuto a pagare una penale molto consistente.

Inoltre, è conveniente per il commerciante sapere in automatico varie informazioni sullo stato dei frigoriferi (temperatura, umidità etc...) senza il bisogno di andare a controllare personalmente il loro stato, inoltre è conveniente avere tali informazioni sempre alla propria portata tramite l'uso di smartphone.

Per questo scopo desideriamo introdurre nomeprogetto, come valida soluzione sia al singolo commerciante e ad imprese che hanno il bisogno di tenersi sempre aggiornate sullo stato delle proprie celle frigorifere e, di conseguenza, sui loro prodotti che potrebbero finire in pessimo stato nel caso qualcosa potesse malfunzionare.

2 Use case(s)

Come abbiamo menzionato in 1, il nostro progetto ha lo scopo di fornire una soluzione efficiente e smart per la conservazione di prodotti in celle frigorifere, applicando elementi smart ed iot: per far ciò disporremo di sensori rilevanti le temperature ed umidità delle singole celle, tali dati saranno salvati al termine di ogni finestra temporale decisa dall'utente (la quale può variare anche da stagione a stagione) su un database apposito.

I clienti si potranno interfacciare tramite l'uso di un bot telegram che provvederà ad avvertire gli utenti nel caso ci fossero problemi urgenti alle proprie celle o per fornire i dati giornalieri al termine dell'attività lavorativa (anche questo parametro è impostabile dall'utente). Ogni sensore avrà un identificatore univoco che li contraddistinguerà che sarà salvato su un altro database in cui saranno contenute anche le specifiche desiderate dai clienti.

Con quanto abbiamo descritto brevemente in questo paragrafo sarà possibile mantenere aggiornati gli utenti nel caso di guasti improvvisi e/o altre problematiche legate all'utilizzo delle celle frigorifere, oltre alla possibilità di ottenere in modo repentino dei dati fondamentali da riportare sul registro giornaliero.

3 Requirements

We'll use the following software/hardware platforms:

- **Hardware** side:

- **IP camera** with Wi-Fi module for video streaming;
- **computer** to run the software.
- **Software side:**
 - **telegram bot**, built on Python by using python-telegram-bot;
 - **software for people detection** with the use of OpenCV and for sending data towards the ThingSpeak platform;
 - **Heroku** for bot hosting;
 - **Thingspeak** for the online data storing and visualization;
 - **MongoDB** for the storing of data regarding the affluence of people inside a building.

4 Solution description

Lo scopo del nostro progetto è fornire una soluzione iot ai commercianti e/o imprese che hanno il bisogno di essere sempre aggiornate sullo stato dei loro frigoriferi; per far fronte a ciò abbiamo intenzione di produrre un bot telegram che (a seguito di alcuni parametri definiti dall'utente) aggiorni il cliente sulle condizioni delle celle senza il bisogno di recarsi in loco.

Il nostro progetto sarà in scala minore rispetto a quanto ci attendiamo, di fatto utilizzeremo un singolo raspberry pi4 con tre sensori di temperatura annessi in modo da monitorare lo stato delle celle. I sensori saranno distinti da un codice identificativo univoco preceduto dal nome/codice dell'impresa che lo possiede (in questo modo possiamo sapere a chi fa riferimento quel sensore ed in quale delle celle si trova); i dati (temperatura ed umidità) saranno salvati, periodicamente su scelta dell'utente, su un server necessario per il loro immagazzinamento e visualizzazione. A questo punto i dati sono disponibili per il bot, ospitato sulla piattaforma Heroku, il quale potrà farne uso per i propri scopi.

5 Planned demo and future work

Presenteremo il nostro progetto tenendo in considerazione il suo fattore di scalabilità per un suo possibile impiego maggiore in futuro, ma per rimanere in termini strettamente accademici e per via del tempo a noi concessoci per la realizzazione, mostreremo una piccola implementazione del nostro lavoro.

La presentazione del nostro lavoro avverrà attraverso un video e/o una piccola dimostrazione su come fare il setup del bot per un singolo utente; nel caso decidessimo di

mostrare il video, faremo il possibile per elencare ogni singolo passaggio per il corretto funzionamento del bot e del ruolo dei sensori nel nostro progetto.

Siamo estremamente interessati ad espandere il nostro lavoro con la possibilità di inviare sms in parallelo col bot, in modo da rendere tutto più accessibile anche alle fasce più anziane dei clienti oltre alla creazione di un'app che possa dare maggiore capacità di gestione sui sensori da parte degli utenti.

Take a training example (a set of six movie preferences). Set the states of the visible units to these preferences. Next, update the states of the hidden units using the logistic activation rule described above: for the j th hidden unit, compute its activation energy $a_j = \sum_i w_{ij}x_i$, and set x_j to 1 with probability $\sigma(a_j)$ and to 0 with probability $1 - \sigma(a_j)$. Then for each edge e_{ij} , compute $Positive(e_{ij}) = x_i * x_j$ (i.e., for each pair of units, measure whether they're both on). Now reconstruct the visible units in a similar manner: for each visible unit, compute its activation energy a_i , and update its state. (Note that this reconstruction may not match the original preferences.) Then update the hidden units again, and compute $Negative(e_{ij}) = x_i * x_j$ for each edge. Update the weight of each edge e_{ij} by setting $w_{ij} = w_{ij} + L * (Positive(e_{ij}) - Negative(e_{ij}))$, where L is a learning rate. Repeat over all training examples.