UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

EPI-USE® LABS
value through innovation

COS 301 - SOFTWARE ENGINEERING

HOME SECURITY SYSTEM (ARGUS)

CODING STANDARDS DOCUMENT

Initials and Surname	Student Number
B.A. Zietsman	u15228194
B.R. Fourie	u16024002
J.K. Lutz	u15323413
J.S. Stadler	u16008554
S. Vermeulen	u16078625
B. Chu (Deregistered)	u18176951

August 21, 2020

Contents

Introduction

The purpose of this document is to make sure that the code written by team Sigma in the 'Home Security System' (also known as 'Argus') project is legible and consistent. It will lay out the structure followed when coding as well as the naming conventions used by the team.

Programs and languages used

The 'Web Dashboard' part of the project is being coded in Angular, which used HTML, CSS, TypeScript and JavaScript to work. The database component of the project is done in PostgreSQL on a Heroku server and is linked with the Angular application through the Spingboot backend. The neural network component is done in Python and is run on a Raspberry Pi. A RabbitMQ event handler is used to manage the events passed between the Angular frontend, Springboot backend and Python neural network.

Coding conventions

The 'Web application' done in Angular uses components to function as required. Each component is a functional page in the application and some components are made to be reused between different pages in the application. The components are ordered and grouped into the respective pages they are used in, or into the main category they fit in in terms of their main purpose. Examples include the Login and LiveFeed components being in the Dashboard subfolder, as they are both mainly used in the Dashboard page of the Web application, and then the ClearedList subfolder containing the ClearedListPersons.

The Springboot backend is also split up in to different folders and subfolders, grouping the models, controllers and configuration files together to make it easier to find them. The "src/main/java" subfolder also contains the files necessary for the RabbitMQ event handler, the files needed for sanitizing security and the service files necessary to run the backend application.

Naming conventions

The name of components, functions and variables are meant to be self-explanatory to the purpose they serve or for what they use will be used in the program. This will allow a team member or external person to be able to read the code and easily derive the use and purpose of a function, variable or why a certain component was generated.

The names of functions and variables are in 'camelCase', while the names of Angular components are 'hyphen-split'

Layout

There is a central CSS stylesheet that is used throughout the entire application located at 'src/styles.css' in the Angular-Frontend. There is more than one JavaScript file, as different team members took on different aspects of the application, but these files are however imported into the needed files. All of the JavaScript files are located in a central folder 'assets/js'. Any images or icons used throughout the application also reside in the 'assets' file and they are split up into respective sub-folders.

Each component used in the Angular app is also placed into a folder. These folders group components together that are used in the same page in the application, or that share a common class/purpose.

The components and services used in the Springboot backend were also grouped into subfolders. The service files, event handler, configuration, controller and model files were placed in different folders to enable faster access and a more ordered structure.

Indentation

Python relies on indentation to determine the scoping information. Without any indentation there will be errors and non-working code.

Commenting

Comments are added where a team member feels that a future change to his/her own code must take place or where a functions purpose is not easy to grasp from its name or its contents. Some parts of the code are commented out as these parts are used for testing purposes, and are therefore not critical to the function of the application.

Error checking and Error handling

Jasmin and Karma was used for unit testing for Angular, PyUnit for Python unit testing and JUnit for Spring unit testing. Travis was used to do the integration testing between the different components.

Review

Once a team member has completed a aspect of the application or has made noticeable changes to an already existing aspect or component, the team member pushes his/her code to Github. Other team members can then 'pull' these changes, compile and test them out on their own machine before delivering feedback in a follow up meeting.