CAPSTONE PROJECT CENG355

Humber College Institute of Technology & Advanced Learning

MAIDS HOME/BUSINESS INTRUSION DETECTION SYSTEM REPORT

Submitted by: Claudio, Meis – Mobile Application

Discipline: Computer Engineering Technology

Date Submitted: January 31, 2020

4.0 Development Platform

4.1 Mobile Application

MAIDS uses an Android-based application written in the Java language and built on the Android Studio IDE version 3.5.3 to control and test components, as well as, display intrusion related information. Powered by Gradle, Android Studio's build system allows for a customized MAIDS build and generates multiple build variants for different devices from a single project. In the case of the MAIDS project, the two variants devices created are:

1. A 10” display generic tablet running a quad-core CPU (Central Processing Unit) with 2 GB of RAM, using Android version 6.0 at a resolution of 1536 x 2048 pixels.
2. A 6” display Google Nexus 6 running a quad-core CPU, with 3 GB RAM, using Android version 6.0 at a resolution of 1440 x 2560 pixels.

In addition, the emulator permits virtual testing of the builds, simulates different MAIDS configurations and features, and provides feedback on feature response and configuration performance used to quickly modify the application.

Building the Android-based MAIDS application requires the use of three distinct file types:

1. manifest
2. activity
3. Drawable resource.

The Android manifest file is named AndroidManifest.xml and must be included with every application in the root directory. The file contains essential application metadata, a set of

data that describes and gives information about other data, in Extensible Markup Language (XML) format (a textual data format with strong support via Unicode for different languages). The manifest file presents essential information about the application to the Android system, information the system must have in order to run any of the application's code. Specifically, the MAIDS manifest file contains the basic building blocks of application (i.e. activities, services, permissions, etc.), details about resource permissions (i.e. access the Internet, remote storage devices, etc.) and the set of classes needed before launch.

Generally, an Android activity file refers to one screen of the Android application's user interface (more commonly referred to as the Application Program Interface (API)) and may contain one or more activities (screens); the main activity is shown first when the application starts. Subsequent screens, require their own activity. Specific to MAIDS, the two pertinent activities will be discussed below in a more detailed fashion under their own section of this report. Each section will outline a general description of its operation mode, as well as, detailed information about the inner workings of the program's code.

Finally, MAIDS uses a drawable resource file which is a graphic file (i.e. .png, .jpg, .gif) that can be drawn to the screen. Our MAIDS application uses exactly two .jpg format files on the second activity; one is used as a placeholder for the intrusion photo while the remote photo of the intrusion is retrieved and the other is the intrusion photo itself, once retrieved. The intrusion photo will have a caption which contains the following information:

1. MAIDS Alarm System Header
2. Address of Intrusion
3. Room where intrusion took place

4. Date and time of intrusion.

The Android application uses a landscape layout (relative layout to be more specific) on the tablet and a portrait layout for the phone, for better visualization of component views and is divided into two main activities:

1. A login activity (first screen), which allows for user authentication and access to the network
2. An data visualization display and component control activity (second screen) which relays database intrusion data (i.e. intrusion date and time, place of entry, owner contact information, etc.), displays a photograph of the intrusion with captioned information (place of entry, date time, etc.), as well as, a remote control mechanisms to activate and deactivate MAIDS and test some of its components (LED module, in particular).

4.2 Login Activity

The login activity is the first screen the user encounters to access MAIDS remotely. The main class of the program is: MainActivity. The class is responsible for presenting the user with a welcoming screen and a MAIDS promotional advertisement photo. In addition, the MainActivity class displays two rectangular, labelled (username and password), user input textboxes used for authentication purposes. Moreover, the class displays two labelled buttons (login and cancel) used to either login into the system and access the remote control features of the application or cancel access to it. It is worth noting that the application has been coded to allow the user only three tries at authentication; otherwise, the application closes and the login process has to be restarted. Once the user inputs the

correct username and password, control is transferred to the second screen, MainActivity3.

Programmatically, the main activity consists of a public class named MainActivity. Inside the class, there is the protected onCreate() method containing two button views (login (b1) and cancel login (b2)), two EditText views (user input boxes ed1 and ed2) and one text view (tx1) which displays a red horizontal bar and are arranged in a relative layout. When the user is presented with the initial screen s/he has the option of login into the application or cancel the login. If the user chooses to launch the application s/he must first enter two pieces of information, username and password, into the textboxes available on the screen. When clicking the login button, the activity activates the setOnClickKistener(v) method which retrieves the EditText box inputs and checks the username and password imputed against the programmed login settings. If they match, the application will close the login activity screen and display the second activity screen which contains the data display activity and control activity on the same screen. At the same time the login button is clicked, a Toast message is display (“Redirecting…”) at the bottom of the screen through which the user is informed of the subsequent activity to be displayed (MainActivity3.javaj). The user has three opportunities to launch the application with each try displayed in the form of a horizontal red line increasing in length referencing the number of tries. If, upon the third try, the user does not input the correct username and password authentication fails and the application textbox inputs are cleared and the user is asked to re-enter the information.

4.3 Data Visualization Activity

The purpose of the data visualization activity is to present MAIDS-created intrusion data in a visual, easy-to-read-and-see format to the user. The data visualization activity is incorporated within the second activity (or screen). The second screen displays six, sequentially arranged, gray, rectangular buttons (views) situated on the upper portion of the screen. The button are labelled as follows:

1. retrieve intrusion information
2. retrieve photo from server
3. display photo locally
4. activate MAIDS
5. deactivate MAIDS
6. Test LED module.

Each button is a subclass onto itself performing different local and remote activities such as: data visualization, and action control. The main data visualization elements (views=buttons) for the MAIDS application are labelled: retrieve intrusion information and display photo locally. The main action control elements (views) are labelled: retrieve photo from server, activate MAIDS, deactivate MAIDS, and test LED module.

Programmatically, the data visualization activity (Main3Activity class) is contained inside the Main3Activiy.java file of the application. The data visualization portion of the activity consists of one WebView element (htmlWebView) which displays the database information gathered form MAIDS internal server (and reached via the Internet through https://singular-gar-5555.dataplicity.io/maidsintrusion.php link), an ImageView element (htmlImageView) which displays the intrusion photo of the incident and two buttons

(mButton1 and mButton3) which activate their respective setOnClickListener (v) method to retrieve the database and photo information remotely. All these view elements are located inside the onCreate () method of the Main3Activity. It is important to emphasize that during the database and photo information retrieval process, Toast messages (messages that provides simple feedback about an operation in a small popup) such as “Retrieving DB information…” and “Displaying Intrusion Image…” are displayed to the user informing them of the action being conducted.

4.4 Action control activity

There are four action control elements to the MAIDS project:

1. Test LED module
2. Retrieve photo information (from internal MAIDS server)
3. Activate MAIDS remotely
4. Deactivate Maids remotely.

All of these actions are contained inside the onCreate () method of the Main3Activity class and ran through their respective setOnClickListener (v) method.

When the button named mButton3 is clicked, the attached listener method displays a Toast message (“Testing LED module…”) and call upon the testLedsCommand () method. The method takes as input the username, password and host strings, as well as, a port integer value. Using these parameters, the method connects via SSH (inside the jsch library) to the MAIDS device and runs an internal python v3.0 program (python3 testleds.py) to test the LED module remotely. The test is programmed to assess the function of the green and red LED lights of the MAIDS device through two output GPIO

pins and to intermittently turn them ON and OFF five times for a period of 2 seconds, each.

When the button named mButton2 is clicked, the attached listener method displays a Toast message (“Retrieving Intrusion Photo…”) and calls upon the getRemoteFile() method to actually retrieve the photo form the MAIDS internal server. The getRemoteFile () method defines three strings: REMOTEDIR (/home/pi/webcam), REMOTEFILE (image.jpg) and LOCALDIR (/home/maids1/). These strings serve to define the remote directory where the photo is located, the generic name of the photo file to retrieve and the directory where the file is to be placed on the tablet/phone device once downloaded from the MAIDS device. Using the jsch library it initiates an SFTP session to the remote device which download the file form the remote system to the tablet or phone. Each step of the session creation, session connection, channel connection and the downloading of the remote file is presented to the user in the form of Toast messages at the bottom of the screen.

When the button named mButton4 is clicked, the attached listener method displays a Toast message (“Activating MAIDS remotely…”) and calls upon the maidsOnCommand () method. The method takes as input the username, password and host strings, as well as, the port integer value. Using these parameters, the method connects via SSH (using the JSCH library) to the MAIDS device and runs through the ‘exec’ command an internal python v3.0 program (python3 maids_final_python_code_22102019_bak1.py) that initializes GPIO pins, sounds audible warnings and activates the MAIDS alarm system. Each step of the session creation, session connection, channel connection and the

downloading of the remote file is presented to the user in the form of Toast messages at the bottom of the screen.

At this point in time, the MAIDS OFF command has not yet been implemented. However, the remote OFF command is envisioned to connect in the same manner as the ON command except that when it send the exec command ‘CTRL+C’ remotely to the MAIDS device, it runs the maidsOff.py program which deactivates the MAIDS alarm system. Deactivation of the MAIDS device results in the playing of three audible messages to the user informing them of the system shutdown.

4.5 Testing Screen Shots of MAIDS Android Application

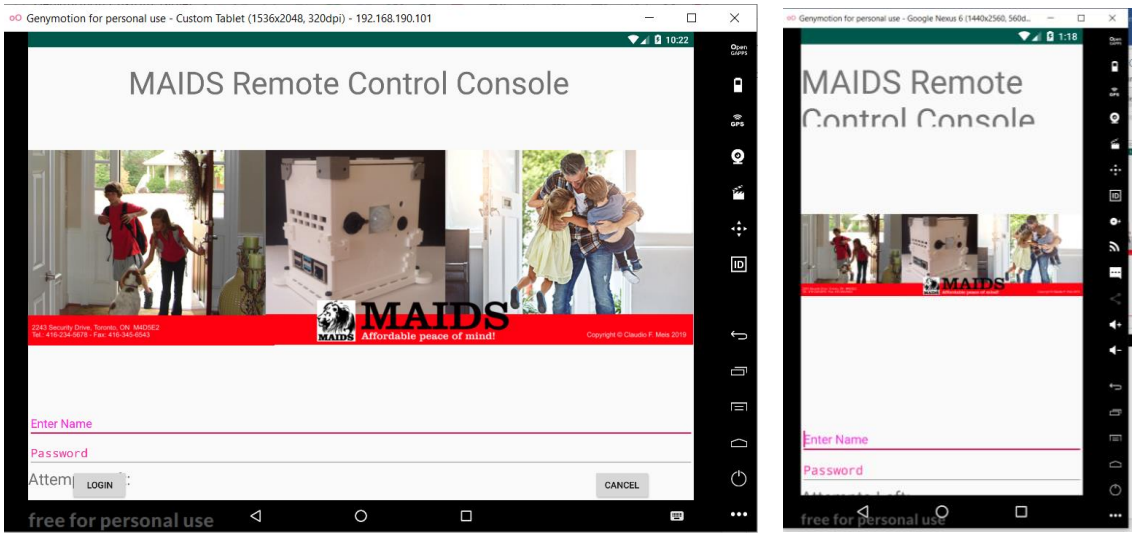


Figure 1 Generic tablet and Google Nexus 6 Login Activity screen shots.

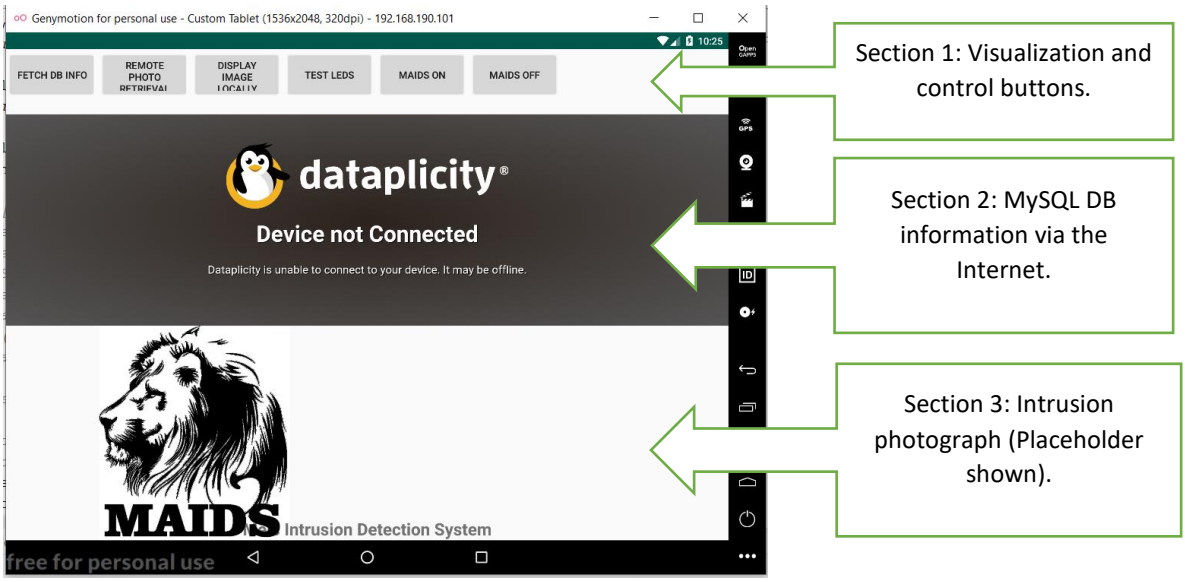


Figure 2 Data Visualization and Control Activity screen shot.

4.6 Android Application Status Report

Prepared by Claudio F. Meis, January 31, 2020.

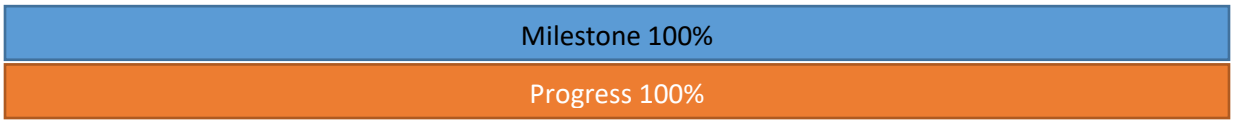
The presentation of the MAIDS project at the Capstone Project EXPO at 1:00 – 4:00 p.m. on Thursday, April 9, 2020, is still on track.

The following work has been completed on the MAIDS Android-based application:

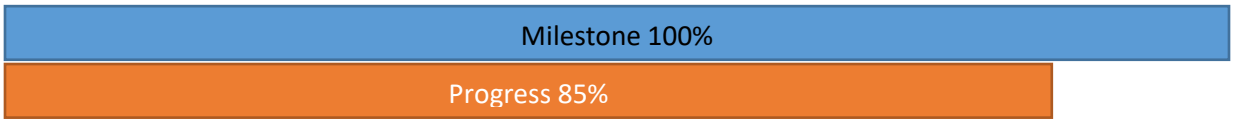
- Coding for the login/Visualization/Control activity.
- Coding for the database information retrieval.
- Coding for the remote testing of the LED module.
- Coding for the remote activation of MAIDS device.
- Coding to remotely turn OFF the MAIDS device.

Progress against Milestones

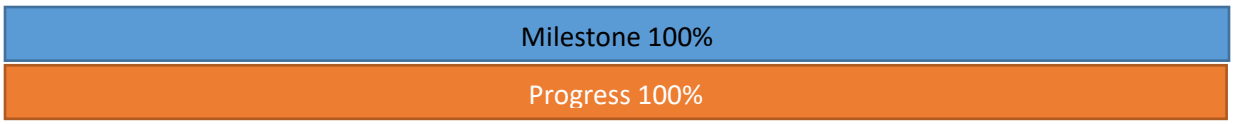
Login Activity



Data Visualization Activity



Control Activity



Key Issues

The two issues need to be resolved on or before February 25, 2020, to meet Capstone Project EXPO deadline:

- Finish coding for the data visualization (remote photo retrieval).
- Adjust tablet and phone resolution displays.

Action Steps

Task	Due Date	Responsibility
Visualization activity coding	February 12, 2020	Claudio Meis
Remote deactivation coding	February 18, 2020	Claudio Meis
Adjust resolution display	February 24, 2020	Claudio Meis