CPEN 291

Project 2 Report

A. Cover page

Smart Security Locker.

Contribution summary:

Zhihang Zhang 16.67%

Website development, server development, HTTP request API development

Sammy Brache 16.66%

Raspberry Pi live-streaming to website, HTTP request API development

Amir Tootooni 16.66%

App development back end, wiring

Robin Reyes 16.66%

App development UI

Richard Tian 16.66%

Lock and NFC interfacing and programming, Door design and building and circuit wiring

Johnny Ma 16.66%

Lock and NFC interfacing and programming, Door design and building and circuit wiring

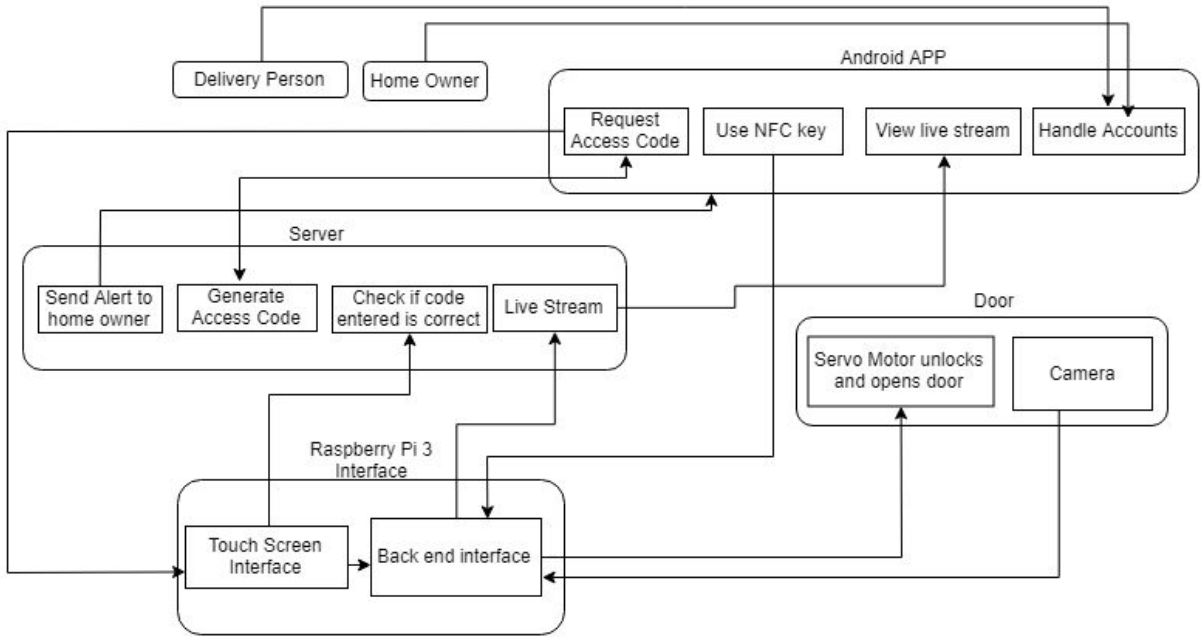
B. Introduction and motivations

This project report corresponds to the "Smart Security Locker" project that we created. Inspiration for this project came from the ongoing "smart lock" revolution, such as "Ring" and the "Amazon Lock". How we planned to largely differentiate from these companies' ideas was to implement the lock using an app and NFC. This new idea sprouted from the fact the smartphones are ubiquitous today. Also, unlike a key, if you lose your smartphone the lock is not vulnerable as the person who has your smartphone must bypass your phone's password as well as the app login. One idea that we had was to use bluetooth rather than NFC. The advantage of bluetooth is that it is more ubiquitous the NFC for smartphones and it is also more reliable. However, the trade-off is the difficulty in writing code for the app and raspberry pi to establish a connection.

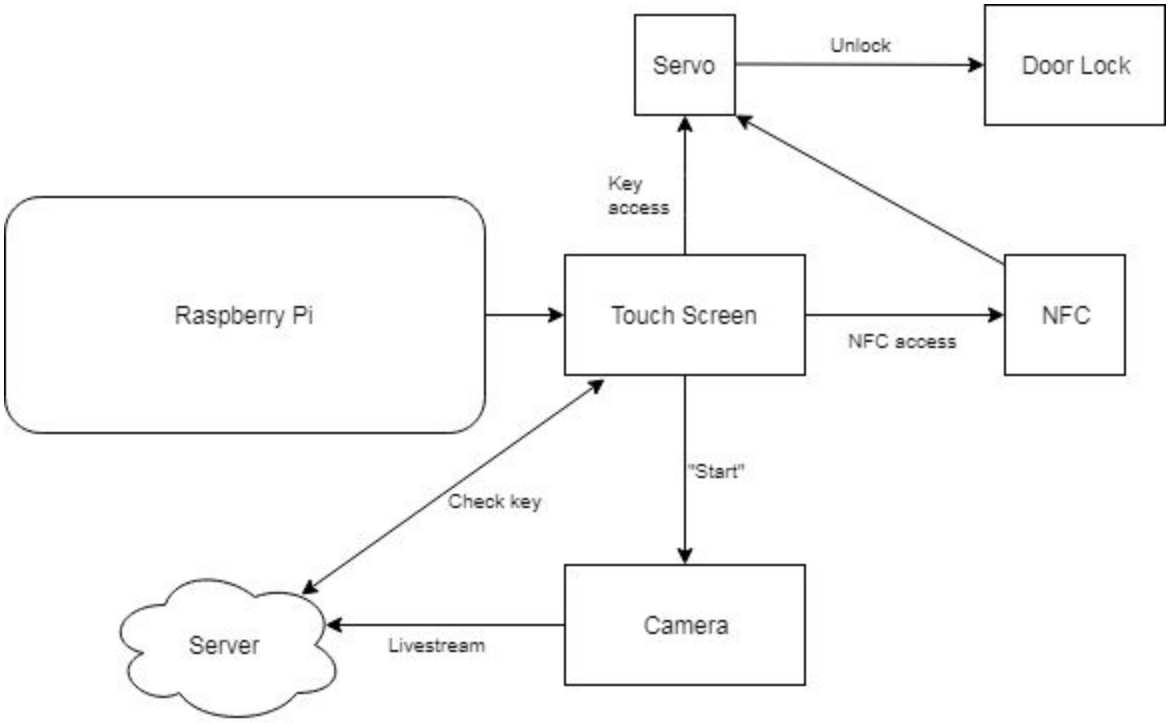
The lock currently has a LCD touchscreen displaying a user friendly interface, a camera showing guests at the door, NFC for easy and secure access, and a manual code input as a failsafe option as well as unlock sharing. The app allows for easy monitoring of multiple locks, as well as interactions with the camera and NFC mentioned above.

C. System diagrams

Software:



Hardware:



D. Project Description (the body of the report)

Product overview

Our product is the Smart Security Locker. It is designed to be suitable for a variety of applications, from serving as a temporary airport baggage locker to a home package reception locker. The Smart Locker's main goal is to have automated yet secure storage system.

The Smart Locker's operations are controlled by a Raspberry Pi 3. The Pi controls the Locker's servo motor (used to lock/unlock the door), NFC reader (reads smartphone NFC tags which allows users to unlock the door by tapping their phone to the lock), touchscreen (allows users to choose to unlock the door with a PIN or NFC, and is used as keypad for PIN entry), and camera (which livestreams a view of the locker door to our app and website attempted door entry).

The management of which users can open which doors, which lockers each user has access to, and the history of all locking and unlocking of each locker is done by an NGINX server. The server stores all this data in mySQL databases. The server is also responsible for keeping track of the URL each livestream is hosted on.

The website (https://hizhh.me) is where users are able to sign up for an account, chat with other proud Smart Locker enthusiasts, and view their lock's livestream. Users can also view their account statistics on the website. The statistics include number of locks held and authentication log.

On our "Raspberry P-eye" Android App, users can view their most recent lock activities, sign up for an account, view their locks' respective live-streams, add a lock to their account, and most importantly unlock their lockers with their NFC key.

The main objective of this project is to create a functional, secure and reliable Smart Locker interface so that the Smart Locker can easily be used in any environment.

Hardware

Lock

The lock mechanism uses a servo to turn a slide latch lock open and shut. By setting the servo to 180 degrees, the servo turns the knob, normally used to manually lock the lock, and setting it to 0 degrees will unlock the door in return. Further, to detect when the door should be relocked after being unlocked by one of the methods below, a micro switch is used. By closing the door, the switch is hit, signaling that the door should be relocked using a pull up resistor.

NFC Reader

When the NFC button is pressed, the pi will attempt to read a NDEF message sent from a phone. The nfcpy library we used to read the a NDEF message could only be run on python 2, so we created a function that calls the python 2 program as a subprocess. This way, we could also kill off the python 2 process after a while if no connection was established. The user must then log into the app and tap NFC access, which will send a request to the server. The server will then generate and store a key and send it to the app. Notice that the key generated is the authentication token generated by the http library in the server when a user logs in. This makes the key extremely secure as the token is not stored explicitly in the database and is only valid for a given amount of time, in our case 100 seconds. This key is then transferred to the pi through NFC when the phone is tapped on the NFC reader. To read the key, we used the NFC pi module which acts as an active NFC tag, meaning that there is no limit on the amount of data that could be transferred between the phone and the pi. The pi then uses an http GET request to authenticate the token. (for more info refer to the "server API" in the wiki tab in out github repository).

Camera

The camera we used was the Raspberry Pi Camera Module V2. We chose it because it was affordable, and interfacing it with the Pi consisted simply of plugging the camera into the Pi, and enabling the camera in the Pi's settings. To set up the livestream, the first thing we did was install a few libraries on the Pi to support streaming video to a website hosted by the Pi at http://<raspberry_pi_ip_address>:80. Once these libraries were installed and the stream was viewable at this URL, we found that this stream was only viewable on devices connected to the same wifi network as the Pi. To make the stream viewable to devices connected to any wifi network, we used a tool called "dataplicity". The dataplicity tool allows us to view the livestream on any wifi by activating what it calls a "wormhole". To activate the wormhole, the dataplicity tool must be downloaded on to the Pi, and the Pi connected to the internet. Then, a user must log in to their dataplicity account on the dataplicity website, link their account with their Pi, and simply click "activate wormhole". Once the stream was viewable to devices connected to

any network, we needed a way to turn the stream on and off easily. To do this, we wrote a shell script for both turning off and turning on the stream, and saved them to the Pi. If someone attempts to open the locker, the Pi runs the start stream script, and if the user opens the door successfully or fails to open the door, the server tells the Pi runs the stop stream script.

Touchscreen

The main form of entry control in the pi is through a LCD touchscreen. The LCD is essentially another monitor that can take taps as mouse clicks. The LCD-show library made it easy to switch between hdmi and touch screen control. To implement the touch screen, we created a GUI in python to display the various options one can use to open the door. Using the guizero library, we first created a "Start" push button that creates a new box and starts an online stream whenever pressed. The next menu contains two more push buttons for a manual code input option or NFC option. The manual code input menu displays a number pad, made of push buttons, that will turn the servo when the correct passcode is received. In order to deal with unresponsiveness, we created a timeout delay that will clear the current display and redisplay the main menu. To further enhance security, we coded the GUI to be full screen when run and disabled any forms of exiting the program by using the touchscreen on the outside.

Software

App

Our Android App's software was written entirely in Java using the Android Studio IDE. The first task we completed for the app was to design and implement the UI. The UI home page consists of a set of banners the user can tap to navigate to different pages. The pages available are the login & sign up page, home page, authentication log, locks menu, lock live stream page, add lock page, sidebar menu and NFC page. The UI was designed with the goal of looking clean and professional, and being easy to navigate. The implementation of the UI was done with reference to good UI design as described in our references (mainly https://developer.android.com/design/index.html). Our main goal when designing the UI was functionality and ease of use as described in our references.

In approaching the design of the app, initial code was written around the user interface first, modularizing the code around opportunities to integrate functionality for making requests to the server later. Executing on this strategy was only possible by having the members who worked on the server and the app agree early on on what each sub team was able to do and what they expected of the other sub team. This proved to be a useful strategy, as towards the end of project completion, very little of the code concerning the user interface had to be changed, which gave us more time to work on communication with the server.

Upon launching the app, the user is first taken to a basic login activity which reads text fields referring to the username and password of the user and uses those string to verify login. The app communicates with the server to verify that the username - password combination is associated with an account. The user could also be greeted with a variety of messages depending on poor connectivity with the server and invalid login information.

For the main functionality of the app, we decided that it was best to use one main activity that held a fragment for each of the main utilities of the app. This allows the user to freely navigate the app's UI, and minimizes the slow down of the app that comes from switching activities. Access to each fragment is given by the menu items in the navigation drawer. A NavigationItemSelectedListener was used to identify which menu item was selected, and the appropriate fragment was swapped for.

The authentication log fragment kept track of the user's logins and attempts at using NFC to unlock a lock. The fragment required that many distinct items were on screen at once. In particular, the fragment was implemented on a nested scroll view to allow for more screen real estate, and the view was littered with cardviews containing textviews pertaining to different kinds of information to be displayed. Because of the large amount of repetition in screen elements, it was decided to make a Java class for the card and its elements, and to treat each card as its own object. The method initializeCards() was implemented to gather all the screen elements and organize them into their own objects, and to add them to an ArrayList. This allowed each Visit Card to be easily accessible in the main code simply by calling the function at the beginning. To obtain the information to fill these visit cards, another function, getVisits(), was implemented that communicated with the server to request a JSON array of Visit objects. Each visit object obtained is then used to give information to a visit card.

The locks fragment manages the locks the user has authority to unlock as well as allows the user to add more locks to their account. Much like the authentication log fragment, the locks fragment has its own set of lock cards, and communicates with the server to request a JSON array of Lock objects for the cards to fill out according to. Each lock card is given the lock ID of the lock it pertains to. Each card has a button with an OnClickListener that requested the lock URL of the lock ID the card refers to, and launches a live stream with that URL. The fragment also comes with a Floating Action Button with an OnClickListener that brought up a dialog, and added a lock to the user's account pertaining to the lock URL the user put in the dialog box.

Once the apps' front-end implementation was done, we moved on to developing the app's back-end. In the development of the app's back-end, we first implemented server communication using sockets to send requests and receive replies. We found after completing the implementation that socket connections between the server and app were unreliable at times, and decided to make the switch to using HTTP requests for app-server communication. As a result, much of our code had to be re-written. To make this change on java, for which the support of http is considerably less compared to python, we needed to find a suitable library that will work with the server API's mentioned in the wiki tab of our github repository. Thankfully we found a condensed java file with an "Httprequest" object which had the necessary methods, mentioned in our references. Using said java file we were able to connect the app to the server and as a result the raspberry pi using http request.

With the inclusion of http request we were able to establish a number of useful client-server functionalities, which could not have been possible (or as efficient) with the socket implementation. Mainly the ability to have a tested, production-code-quality library generate and authenticate users made the system quite secure as the messages, username, passwords and token are all encrypted. This implementation also gave us the added ability to not store the tokens (or in our case, keys) in the

database or anywhere locally and also have time dependant tokens, so that we can have time dependant keys. Therefore a key shared with another user can only be used for a limited amount of time, making the idea of having one-time-use keys feasible. Thus we were able to securely login and sign up users through the app.

The other functions enabled by the server connection in the app are a log of all authentications made for a given user which includes the time, date and state of the authentication. Our communication API also enabled the app to have a log of all the locks in a given account and their specific live stream, plus the ability to add a lock to the account from the app. For the stream the app requests a stream url from the server with an http GET request and uses the value returned to start a stream in the stream fragment.

In our design the app needed to be able to communicate the token "key" with the lock with NFC. Therefore we needed to make the app have a p2p connection with the pi using NFC, similar to how android beam works with NFC. We defined an intent in the manifest which listens for NFC tags and when a tag is detected the handleNfcIntent() function is run and with the use of android.nfc library objects and functions the token (key) is beamed to the NFC receiver on the pi. Because we wanted the added functionality of sharing keys with others, the NFC fragment enables the user to send the current (server generated) key to a given number using an sms text message, code for which is in the NFC fragment file and an example picture is added in corresponding appendix. The NFC fragment also allows the user to set the key sent to the pi through NFC to whatever string they want, this is meant to be used by a person who receives a key from another user through sms and wants to unlock the door. The person who received the sms can simply copy the key and paste it to the dedicated textbox in the NFC fragment, then press set tag and use their phone to unlock the door. The receiver of a key can use any NFC beam app to unlock the door as long as they can set the p2p message to be key they go through sms, for example we tried unlocking the door using this app in the demo and we got the same result as our own app, it worked like a charm. https://play.google.com/store/apps/details?id=net.xerael.beam&hl=en

Server

At the beginning of our project, we hosted our server using Apache. The problem we had with Apache was that it can not handle concurrent socket connections very well, so we decided to make the switch to NGINX, because it has a few libraries supporting that. The code used by our server is in Python, and uses the Flask Python module to make a "Flask app". Not only does the Flask app can deliver the content dynamically on the website, but also it has packages that handle authentications and manage the database.

To store the data on the server, we use mySQL databases. The three databases on the server are the Lock History database, the User Locks database, and the User database. The Lock History database stores every opening and attempted opening event for each locker. Specifically, each row corresponds to a "lock event". The fields of a lock event are status, username, and date and time. The Locks database simply stores the locks held by each user, and the URL of where that lock's live-stream can be found.

Each row consists simply of the username of a user and a lockID corresponding to a lock held by that user. Finally, the User database is used to verify the login of a user into their account on the app or the website. Each row of this database corresponds to one user. The data fields for each user are the user's username and email, as well as an encrypted version of their password for login verification. We also designed RESTful APIs to make it easy for clients to communicate with the server. Apart from Flask's built-in support for http requests, we used flask-httpauth and itsdangerous to achieve time-based authentication with tokens. That means clients do not need to pass their username and password in the request every time for authentication. Instead, they can pass the token, which remains valid for a specific timeout.

Website

We developed our website using the Bootstrap library. We used Bootstrap because it allows us to customize our website thoroughly and has a pleasant and professional look. It also makes our website responsive to different devices. Our website allows users to sign-up, log-in, chat with other users, view their account statistics.

The sign-up and log-in functionalities are implemented by a few packages. First, user inputs are passed to backend via WTForms and Flask. Then the password will be encrypted by Passlib and saved in the database. When users log in, MySQL-Alchemy will retrieve the password and username from the database, and verify them using Passlib methods.

In addition, we bought a domain from https://ca.godaddy.com/ to make our website more accessible, and since our website provides log-in and sign-up functionalities, we used https protocol to eliminate man-in-the-middle attacks and to secure API calls.

E. Test, Evaluations and Challenges

Hardware:

The extra hardware we are using are the raspberry pi camera, a pn532 NFC module, a touch screen and a lock with a servo. The lock and servo were easy to test; we bought a lock and estimated how much torque the servo we would have to buy. The raspberry pi camera hardware was also easy to test. We used the built in console commands of the pi to take a picture and verify the camera was not broken. The touch screen was a little tricky to test as it used GPIO pins and thus we had to ensure that it would not interfere with the servo and pn532 NFC module. Fortunately, it did not and we just installed it using the instruction manual.

NFC: The pn532 NFC module was the most difficult to test. First, we had to solder the right connections on the NFC itself. There are two ways to communicate NFC data; UART or I2C. We initially chose I2C because the libnfc C library uses I2C. The library itself can test to see if the NFC module is broken as it will auto detect connected NFC devices. The main challenge however was to test if the NFC module could receive data. We decided to download some NFC data transfer apps and use some simple libnfc examples to check the data transfers. Unfortunately, the code in the libnfc library is very low level and difficult to understand. Thus, we chose to switch to nfcpy, a python library, which used serial instead of I2C. While serial is inferior to I2C because we could not have multiple serial devices sending

data at once, for the purposes of our project it is sufficient. Nfcpy is a higher level library and the code is easier to comprehend . After writing some test code, we were able to successfully transfer data using NFC.

Wiring: The biggest mistake and challenge involving hardware was the decision to not use a t-cobbler for pins. Since the LCD fit exactly on the pi and few other pins were used, it seemed wise to not crowd our door panel with an additional t-cobbler and breadboard. However, this made our lives very difficult, as the we had to connect nine wires to the GPIO pins themselves by creating wire loops. This made it extremely difficult to check hardware and wiring problems as we would have to remove all wire each time we needed to check a hardware module. Eventually, we soldered the wires to the GPIO pins once everything was confirmed to work.

Camera: The main challenge with using the camera was that it was delicate and finicky; making it hard to mount onto the board. A lot of the problems were fixed by just ensuring there was not too much pressure on the camera, resetting the pi and ensuring the camera interface was enabled.

Software:

Raspberry PI: For the raspberry pi UI, we main used GUIZERO, a higher level library based of tkinter. The main issue we ran into with this code was because GUIZERO is higher level and more abstracted, there would be unexpected behavior for when we instructed the UI to do an action. This was problematic as it made the UI less user friendly. While we were able to use the override function to use tkinter functions, it made programming the UI mode challenging. For the NFC code, the main trouble, as stated before, was selecting the right library to implement the NFC module. After we selected the right library the main issue we ran into integrating the library code with the UI; the NFC code ran on python 2 while the UI code ran on python 3. The remedy the issue, we used a library in python named subprocess. This allowed us the create fork system calls; similar to running programs in command line. The final challenge we ran into was that when we ran code using different files and subprocess, the main loop in GUIZERO ran into problems as we were exiting the loop and going into other processes. This messed with our timing for the UI. To fix this, we ran the processes in parallel with our UI code and we ensured that all processes have a timeout schedule. This ensured that our UI would never hang in one section of code for too long; it would always reset to the start screen if something went wrong.

Android APP:

Because most of the app's functionality relies on communication with the server, work on the app had to be thought out well until the server was able to handle client requests. As previously stated, this forced us to modularize the program to build around the need for server requests, and give us the ability to run reasonable tests without the need for a server. More specifically, because most of the server requests our app makes expects JSON strings, we initially replaced the bit of code that would be used to request the appropriate JSON string from the server with just using a dummy JSON string that would serve as a good representation of what the real JSON string would look like. From this, when it came to working on the backend part of the app's program, almost none of the previous code had to be replaced.

Part of the difficulty of designing the app was the difficulty of passing around data between sections of the app. While initially the use of Intents and Bundles proved to be useful, for some transitions around the app, they would not work and we could not find an appropriate solution. What we did instead, which might not necessarily be in best practice, was to create a class called Constants that would be called in every fragment and activity in the app. The class would only have public static variables, which in nature, would have the same value in heap memory no matter when or where it was called in the app. This allowed us to manage our information more easily, and proved to simplify implementation of the app's utilities.

One of the primary issues we ran into when developing the mobile app was that we were sending requests to the server with the use of sockets, but the server was developed to handle requests in the form of HTTP requests. While HTTP requests rely on sockets at their core, it is quite far removed from simply using sockets as a method of client-server communication. To remedy this compatibility issue, all code written to send requests to the server had to be overhauled, and rewritten to use HTTP requests instead. Writing software to send HTTP requests in Java is quite difficult and requires a considerably high level of knowledge of server programming. Because of this, we used a library called HttpRequest (the code for the JavaDocs of mentioned library is this url and a link to the coressponding repository is added in refrences http://kevinsawicki.github.io/http-request/apidocs/index.html) to help with the transition from sockets to HTTP requests. This library made it easier for us to use HTTP requests by specifically designed for Android Studio, it also made our app much less laggy as the requests ran on a different thread than the main activity thread using the Android Studio provided AsyncTask class. After our software overhaul, the communication between the app and server was seamless and we had stable and secure connection.

Web Server:

For testing the server, we used different approaches for backend and frontend. For the frontend, we used the developer tool in chrome to select the components on our website and examine the corresponding html code, so that we can tweak the website more precisely. For the backend, we encountered a few problems when testing it. Initially, the server is built on Apache2 and mod_wsgi, which is fairly easy to set up. However, it turns out that Apache2 + mod_wsgi does not support concurrent connections very well with Flask-SocketIO. Therefore, we decided to migrate to Nginx, which has a better performance than Apache in terms of dealing concurrent connections and asynchronous tasks. Before the migration, we made a few backups of backend code and Apache configuration files in case that we could not set up our server on Nginx. After following some tutorials and reading pages of stackoverflow questions, we eventually finished migrating the server. In terms of testing the backend, we tried different test cases on the main functionalities. For example, for the registering functionality, we filled in an email address in an invalid format to check if the backend code validated it correctly. In addition to that, we checked the user table in our database to make sure that the new user appears, and more importantly, the hashed version of the user's password is saved in the table. We also spent a fair amount of time testing the RESTful APIs we wrote for clients. Specifically, we used curl to make different types of http requests in the command line and checked if the response contains correct information as a json object or an array of json objects.

F. References and bibliography

List of files submitted for lab

Raspberry Pi Code/beam.py - code for android nfc beam

Raspberry Pi Code/cli.py - command line interface for beam.py

Raspberry Pi Code/cli.py - command line interface for beam.py

Raspberry Pi Code/main.py - UI code. Also used as main

Raspberry Pi Code/nfc_subprocess.py - running beam.py in subprocess. used to allow main.py to use nfc

Raspberry Pi Code/servo.py - python file to hold servo functions and lever switch functions

Raspberry Pi Code/start_stream.sh - shell script to start stream

Raspberry Pi Code/stop_stream.sh - shell script to stop stream

Peye app code/build.gradle - includes the dependencies and libraries used for the App project

Peye app code/main/AndroidManifest.xml - file describing the minimum system requirements, the necessary permissions and intents

Peye app code/main/res/ - contains all resources used in the app project (ie.layouts, menus and constants)

Peye app code/main/java/com/g19p2/g19p2app - contains all .java files of the app

FlaskAppNginx/my_app.py - contains the backend of the server

FlaskAppNginx/templates/ - contains all the html code

References

raspberry pi UI documentation: https://lawsie.github.io/guizero/

nfcpy documentation: https://nfcpy.readthedocs.io/en/latest/

libnfc library: https://github.com/nfc-tools/libnfc

libnfc wiki: http://nfc-tools.org/index.php/Main_Page

LCD touchscreen driver: https://github.com/goodtft/LCD-show

File data encryption: http://docs.python-guide.org/en/latest/scenarios/crypto/

RESTful APIs: https://blog.miguelgrinberg.com/post/restful-authentication-with-flask

APIs: https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask

web server gateway interface: http://uwsgi-docs.readthedocs.io/en/latest/InternalRouting.html

Server:
https://iotbytes.wordpress.com/python-flask-web-application-on-raspberry-pi-with-nginx-and-uwsgi/

Flask + python backend: https://pythonprogramming.net/web-development-tutorials/

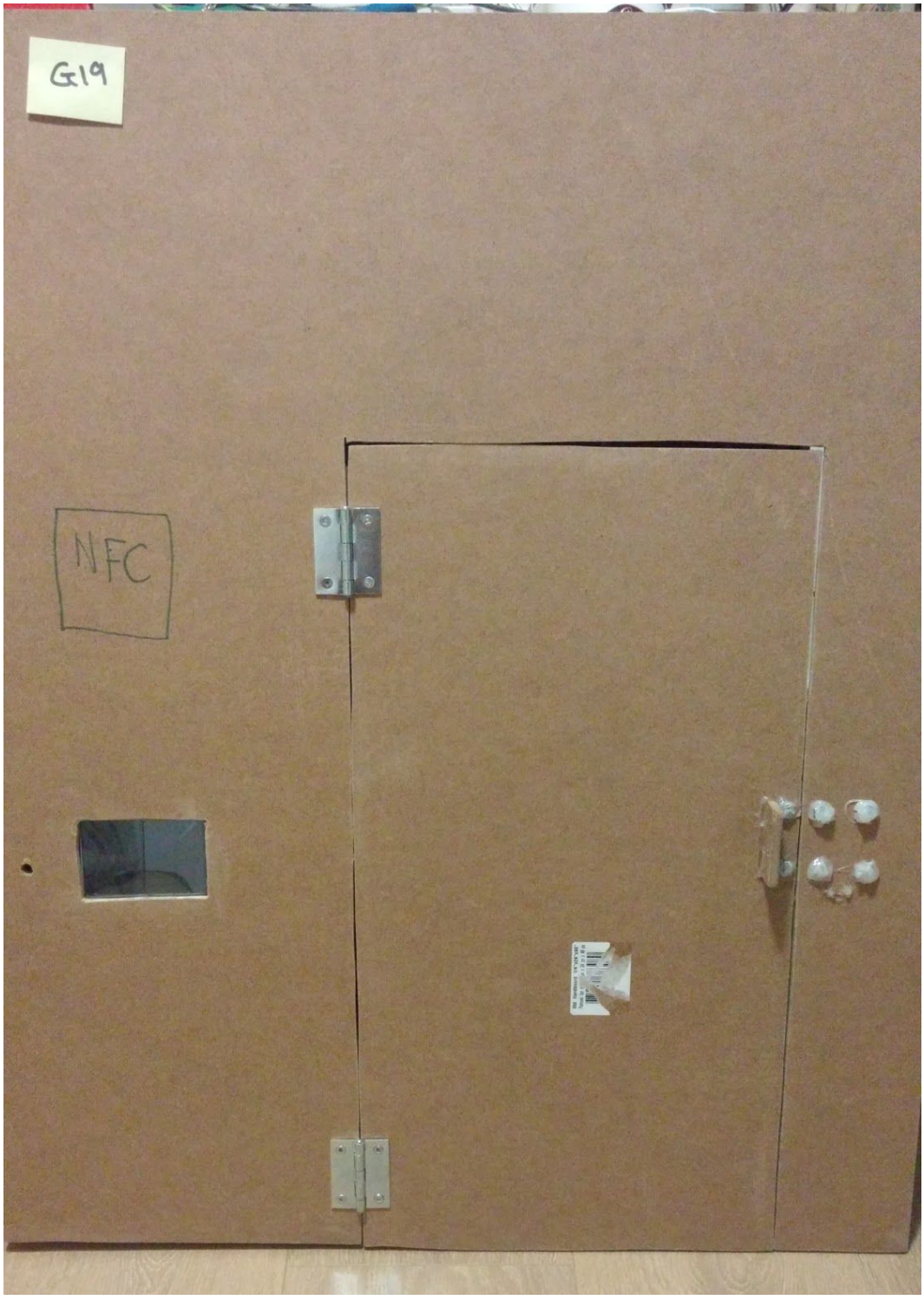
Packages:

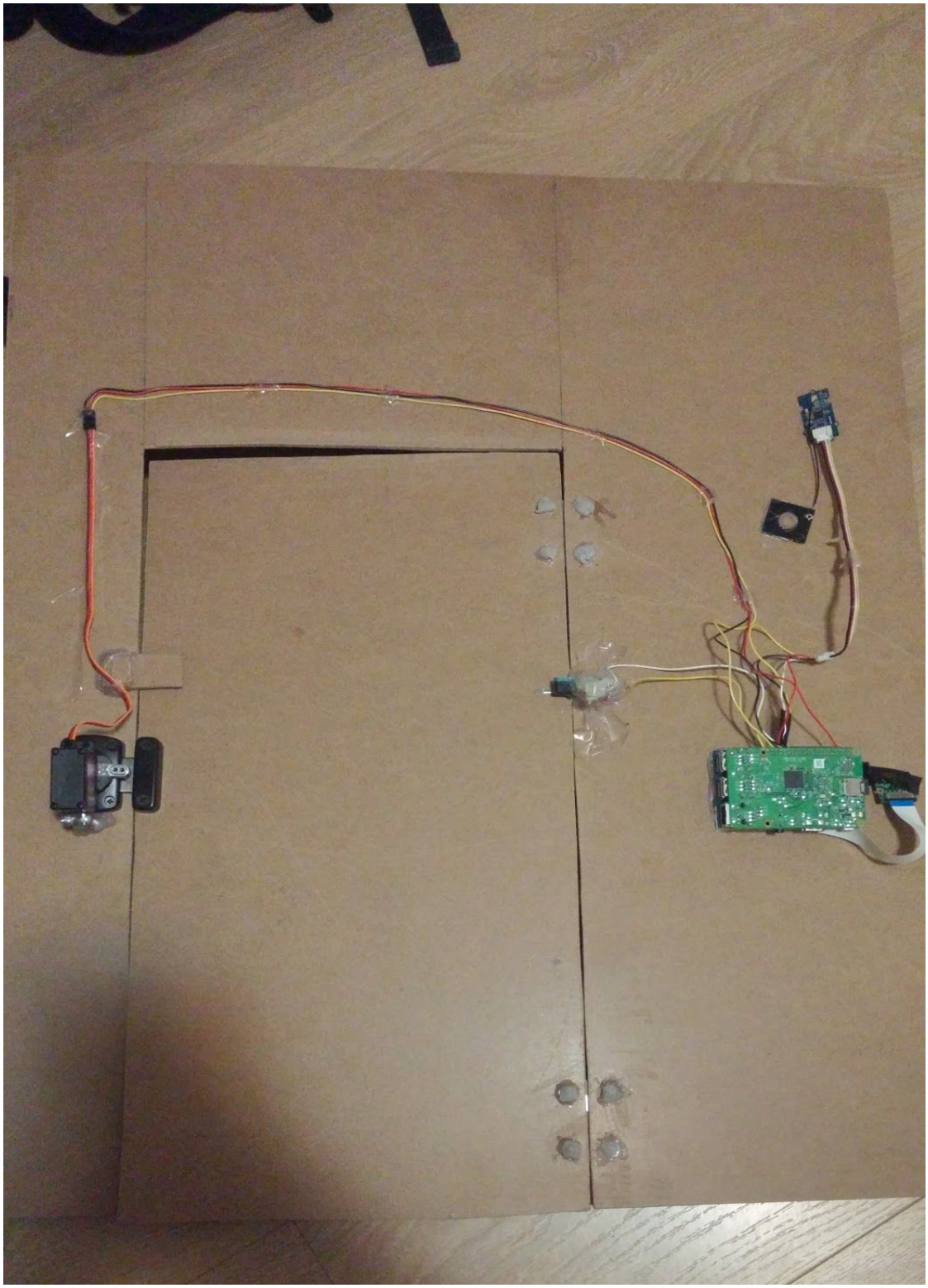
http://pythonhosted.org/itsdangerous/
http://flask-httpauth.readthedocs.io/en/latest/
https://passlib.readthedocs.io/en/stable/
http://flask.pocoo.org/
https://flask-socketio.readthedocs.io/en/latest/
https://certbot.eff.org/
https://flask-wtf.readthedocs.io/en/stable/
http://flask-sqlalchemy.pocoo.org/
http://docs.python-requests.org/en/master/

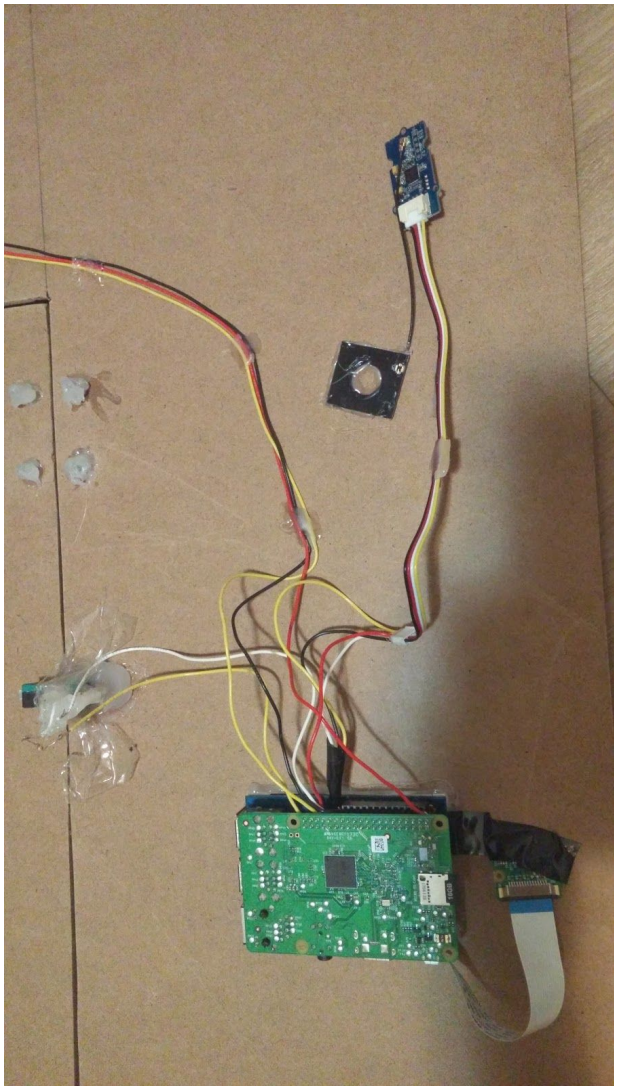
App references:

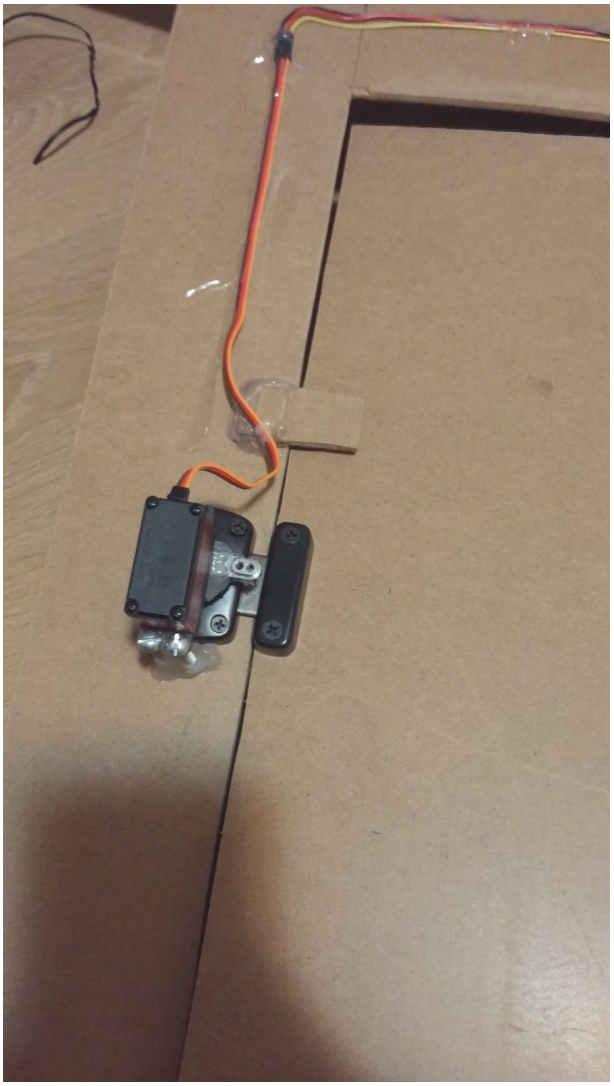
https://github.com/kevinsawicki/http-request
https://developer.android.com/guide/topics/connectivity/nfc/index.html
https://developer.android.com/training/basics/fragments/fragment-ui.html
https://developer.android.com/training/permissions/requesting.html
https://developer.android.com/guide/topics/connectivity/nfc/nfc.html
https://developer.android.com/guide/topics/connectivity/bluetooth.html
https://material.io/guidelines/components/bottom-navigation.html
https://material.io/guidelines/layout/principles.html
https://material.io/guidelines/usability/accessibility.html
https://material.io/guidelines/layout/principles.html
https://www.tutorialspoint.com/android/android_sending_sms.htm
http://sapandiwakar.in/replacing-fragments/
https://code.tutsplus.com/tutorials/sharing-files-with-nfc-on-android--cms-22501
http://androidsrc.net/android-client-server-using-sockets-client-implementation/
http://loopj.com/android-async-http/
https://code.tutsplus.com/tutorials/streaming-video-in-android-apps--cms-19888

Appendix A1 – Project pictures (hardware)


G19
NFC

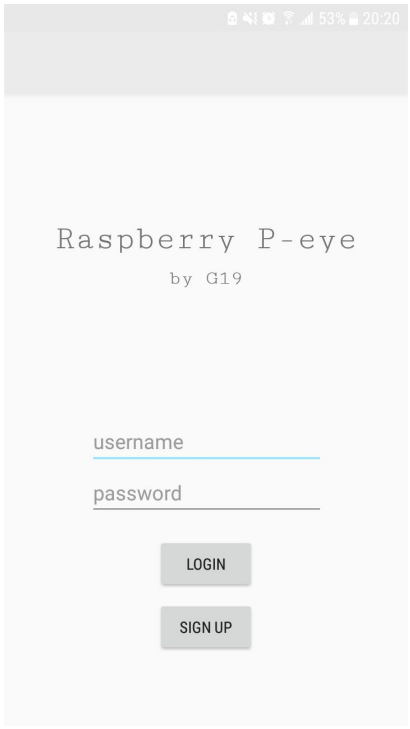


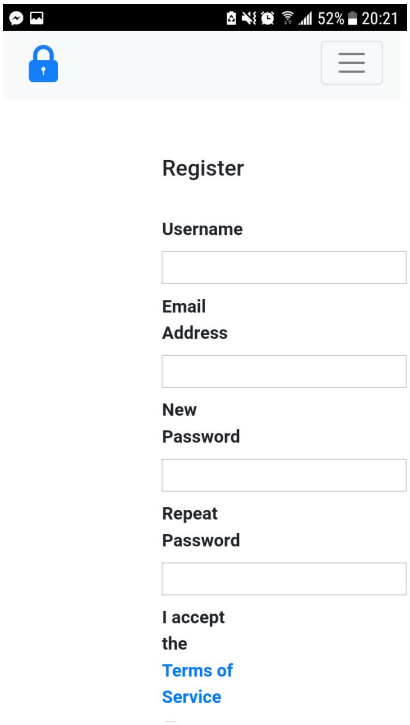


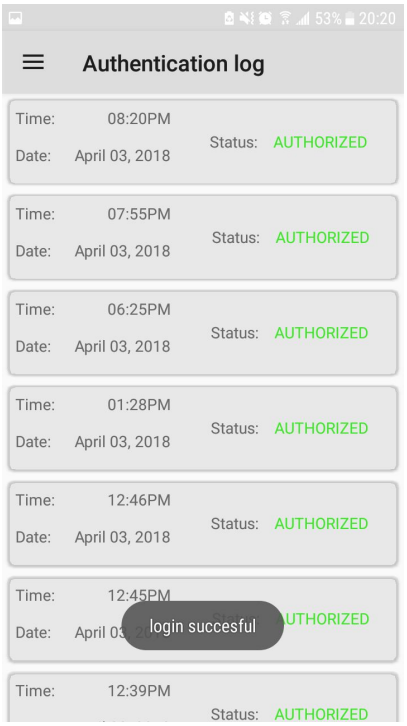


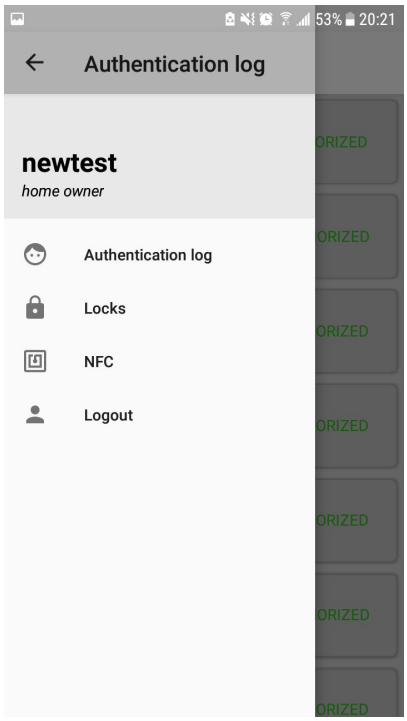
Appendix A2 – Project pictures (software)

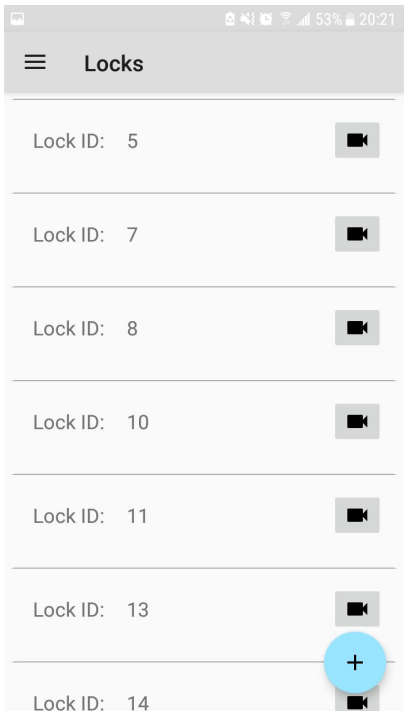
Android App pictures:

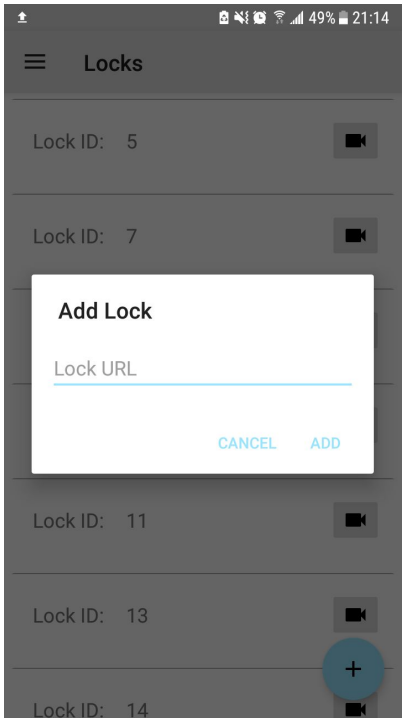

Raspberry P-eye
by G19
username
password
LOGIN
SIGN UP

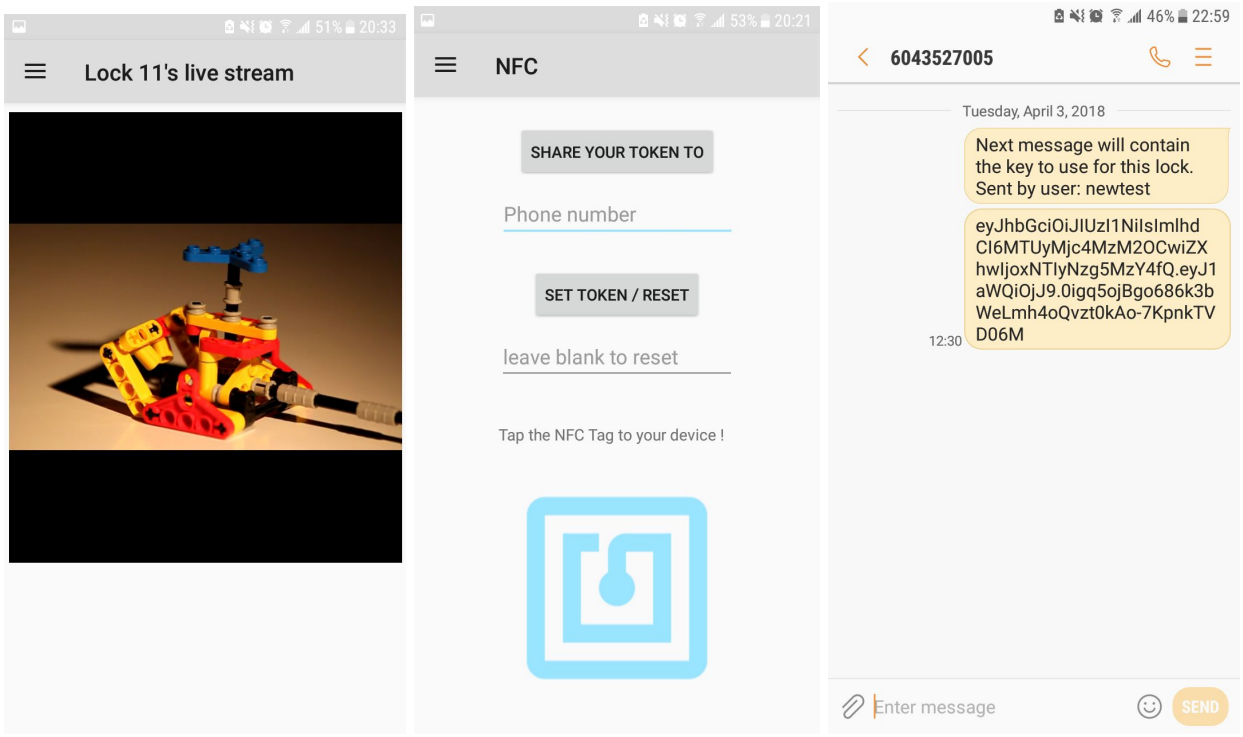

Register
Username
Email Address
New Password
Repeat Password
I accept the Terms of Service


Authentication log
Time: 08:20PM
Date: April 03, 2018
Status: AUTHORIZED
Time: 07:55PM
Date: April 03, 2018
Status: AUTHORIZED
Time: 06:25PM
Date: April 03, 2018
Status: AUTHORIZED
Time: 01:28PM
Date: April 03, 2018
Status: AUTHORIZED
Time: 12:46PM
Date: April 03, 2018
Status: AUTHORIZED
Time: 12:45PM
login succesful
Time: 12:39PM
Status: AUTHORIZED

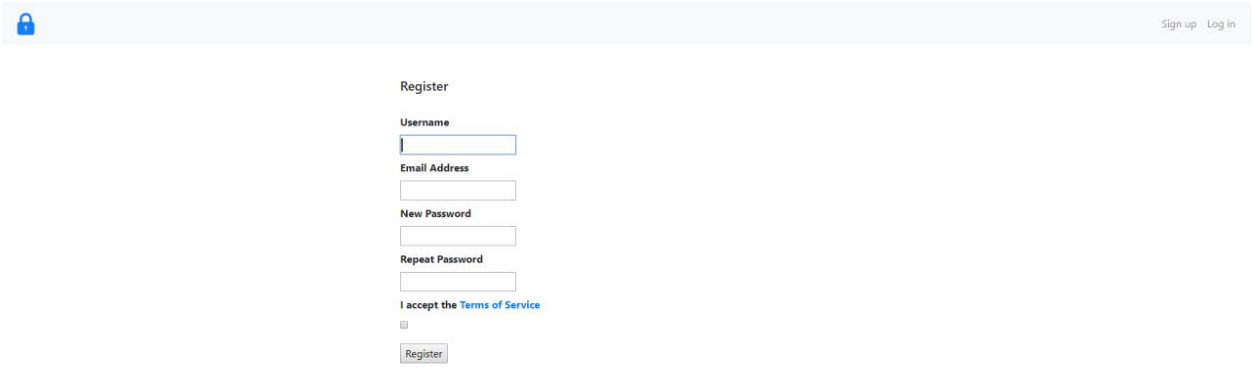

Authentication log
newtest
home owner
Authentication log
Locks
NFC
Logout

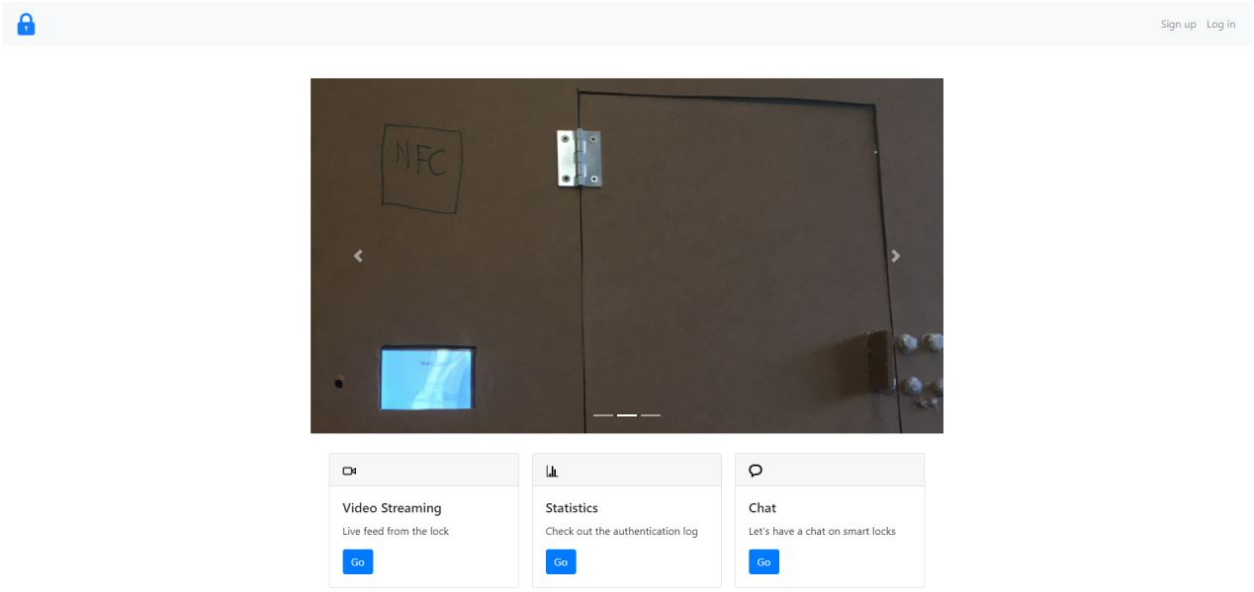

Locks
Lock ID: 5
Lock ID: 7
Lock ID: 8
Lock ID: 10
Lock ID: 11
Lock ID: 13
Lock ID: 14

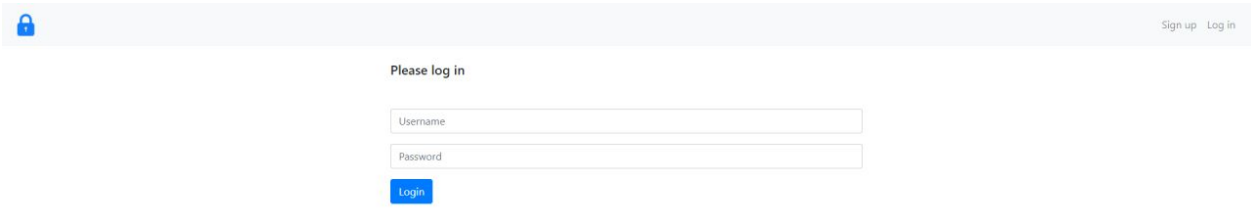

Locks
Lock ID: 5
Lock ID: 7
Add Lock
Lock URL
CANCEL
ADD
Lock ID: 11
Lock ID: 13
Lock ID: 14

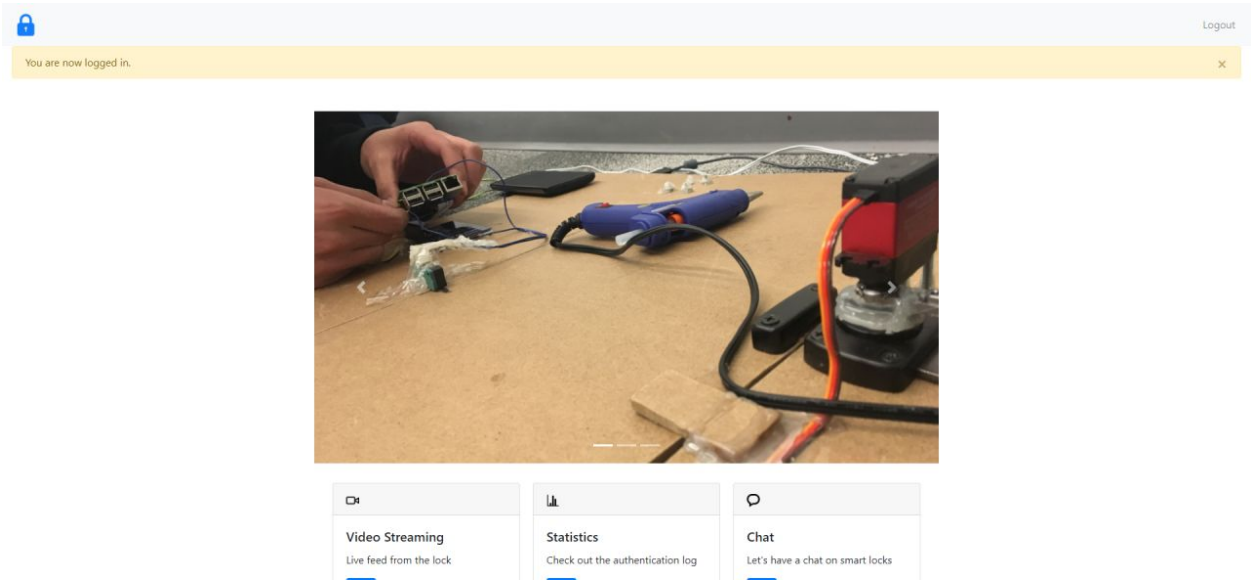

51% 20:33
Lock 11's live stream
53% 20:21
NFC
SHARE YOUR TOKEN TO
Phone number
SET TOKEN / RESET
leave blank to reset
Tap the NFC Tag to your device !
46% 22:59
6043527005
Tuesday, April 3, 2018
Next message will contain the key to use for this lock. Sent by user: newtest
eyJhbGciOiJIUzI1NiIsImlhdCI6MTUyMjc4MzM2OCwiZXhwIjoxNTIyNzg5MzY4fQ.eyJ1aWQiOjJ9.0igq5ojBgo686k3bWeLmh4oQvzt0kAo-7KpnkTVD06M
12:30
Enter message
SEND

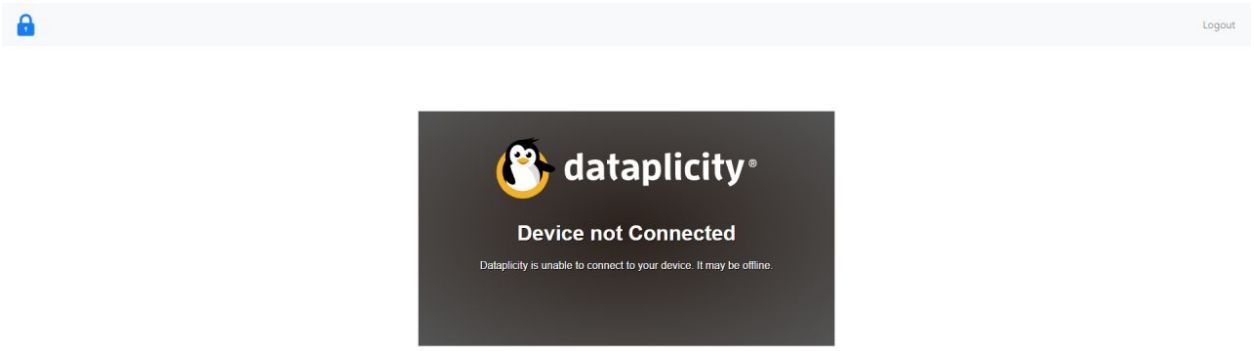
Website pictures:

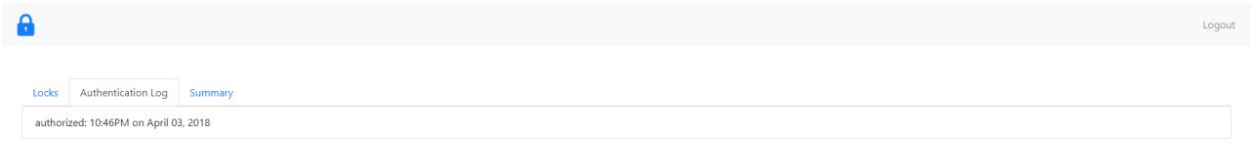


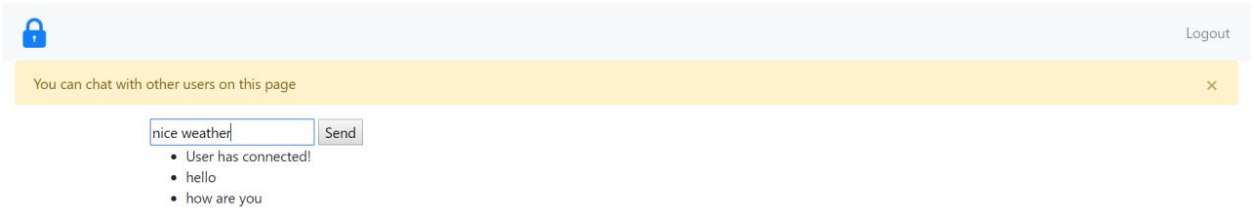



Sign up
Log in
Please log in
Username
Password
Login


Logout
You are now logged in.
Video Streaming
Live feed from the lock
Statistics
Check out the authentication log
Chat
Let's have a chat on smart locks


Logout
dataplicity®
Device not Connected
Dataplicity is unable to connect to your device. It may be offline.


Logout
Locks
Authentication Log
Summary
authorized: 10:46PM on April 03, 2018



Appendix B – Component list

Part #	Description	Quantity used
PN532	NFC Module	1
Camera Module V2	Raspberry PI Camera Module	1
Annimos B076CNKQX4	High torque servo used to unlock lock	1
Sayayo CMS220U-B	Lock used for the door	1
XPT2046 Touch Controller	Touchscreen for Raspberry PI	1
Raspberry PI 3	The Raspberry PI	1
Micro Switch	Micro switch used to detect the closing of the door	1

Item: PN532 NFC module
Vendor: https://leeselectronic.com/en/
Purchase link: https://leeselectronic.com/en/product/15408.html
Link to datasheet or wiki: http://wiki.seeedstudio.com/Grove_NFC/
Price: $28 CAD

Item: Camera Module V2
Vendor: https://www.amazon.ca
Purchase link: https://www.amazon.ca/Raspberry-Pi-Camera-Module-Megapixel/dp/B01ER2SKFS/
Link to datasheet or wiki: https://www.raspberrypi.org/documentation/hardware/camera/README.md
Price: $32.98 CAD

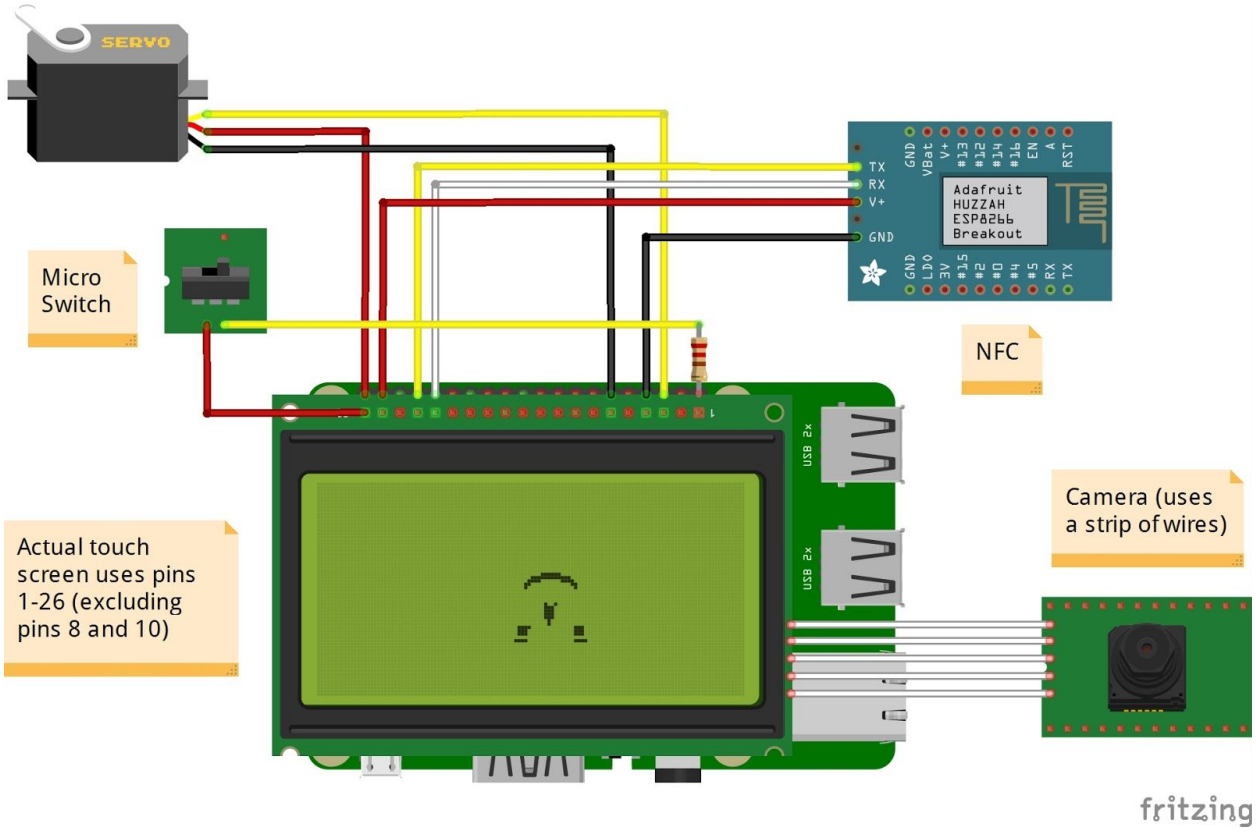
Item: Annimos digital high torque servo
Vendor: https://amazon.ca

Purchase link: https://www.amazon.ca/gp/product/B0769DFJVK
Link to datasheet or wiki: https://www.amazon.ca/dp/B0769DFJVK/
Price: $25.99 CAD

Item: Sayayo CMS220U-B
Vendor: https://amazon.ca
Purchase link:
https://www.amazon.ca/gp/product/B079JDZGNH/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1
Link to datasheet or wiki: None, its a bathroom door lock
Price: $14.99 CAD

Item: XPT2046 Touch Controller
Vendor: https://www.amazon.ca
Purchase link: https://www.amazon.ca/Raspberry-Display-Elegoo-320x480-Interface/dp/B071DF1QZZ/
Link to datasheet or wiki: https://www.buydisplay.com/download/ic/XPT2046.pdf
Price: $20.00 CAD

Appendix C - Fritzing



Appendix D – Raspberry Pi 3 Code

main.py (main UI code that controls everything)

```
from guizero import App, Text, TextBox, PushButton, Box, info
import time
from time import sleep
import servo
import tkinter as tk
import nfc_subprocess
import json


#requests
import requests
from requests.auth import HTTPBasicAuth

#PASSWORD (temporary)
global password
password = 1234

#nfc password
global nfc_password
nfc_password = 'hello1234'

#timer, timeout in seconds
global timeout_val
timeout_val = 20.0

#door open sleep time in seconds
global door_sleep
door_sleep = 5

def onclick_code():
    del_main()
    global input_text
    global numpad

    txt_sz = 25 #button text size
    num_padx = 55 # x axis padding for numbers
    input_text = TextBox(main_box, width = 40)
    numpad = Box(main_box, layout = "grid")
    #cols should have similar padding for numbers
    #col 1
    box_7 = PushButton(numpad, grid = [0,0], text = "7",padx = num_padx, pady = 15,
command = onclick_7)
    box_4 = PushButton(numpad, grid = [0,1], text = "4",padx = num_padx, pady = 15,
command = onclick_4)
    box_1 = PushButton(numpad, grid = [0,2], text = "1",padx = num_padx, pady = 15,
command = onclick_1)
    box_clear = PushButton(numpad, grid = [0,3], text = "CLEAR",padx = 12, pady =
15, command = onclick_del)
```

```
    box_7.text_size = txt_sz
    box_4.text_size = txt_sz
    box_1.text_size = txt_sz
    box_clear.text_size = txt_sz

    box_8 = PushButton(numpad, grid = [1,0], text = "8",padx = num_padx, pady = 15,
command = onclick_8)
    box_5 = PushButton(numpad, grid = [1,1], text = "5",padx = num_padx, pady = 15,
command = onclick_5)
    box_2 = PushButton(numpad, grid = [1,2], text = "2",padx = num_padx, pady = 15,
command = onclick_2)
    box_0 = PushButton(numpad, grid = [1,3], text = "0",padx = num_padx, pady = 15,
command = onclick_0)
    box_8.text_size = txt_sz
    box_5.text_size = txt_sz
    box_2.text_size = txt_sz
    box_0.text_size = txt_sz

    box_9 = PushButton(numpad, grid = [2,0], text = "9",padx = num_padx, pady = 15,
command = onclick_9)
    box_6 = PushButton(numpad, grid = [2,1], text = "6",padx = num_padx, pady = 15,
command = onclick_6)
    box_3 = PushButton(numpad, grid = [2,2], text = "3",padx = num_padx, pady = 15,
command = onclick_3)
    box_enter = PushButton(numpad, grid = [2,3], text = "ENTER", pady = 15, command
= onclick_enter)
    box_9.text_size = txt_sz
    box_6.text_size = txt_sz
    box_3.text_size = txt_sz
    box_enter.text_size = txt_sz

def onclick_1():
    input_text.append("1")
def onclick_2():
    input_text.append("2")
def onclick_3():
    input_text.append("3")
def onclick_4():
    input_text.append("4")
def onclick_5():
    input_text.append("5")
def onclick_6():
    input_text.append("6")
def onclick_7():
    input_text.append("7")
def onclick_8():
    input_text.append("8")
def onclick_9():
    input_text.append("9")
def onclick_0():
```

```
    input_text.append("0")
def onclick_del():
    input_text.clear()

def onclick_enter():
    global input_text
    global numpad
    global main_box
    enter_text = int (input_text.get())
    #do something with the password
    if(enter_text == password):
        open_door()
        time.sleep(5)
        #wait for lever sensor
        servo.wait_for_lever()
        #sleep(20)
        close_door()

    #example, you could probably add some encryption to this too by calling another
file
    f = open("input.txt","w+")
    f.write(str (enter_text))
    f.close()
    #example end
    main_box.destroy()
    show_start()
    nfc_subprocess.stop_stream()

def open_door():
    servo.SetAngle(0)

def close_door():
    servo.SetAngle(180)

def onclick_nfc():
    global main_box
    info("NFC", "Please tap OK when ready, then tap your phone to the NFC box
above.")
    del_main()
    nfc_subprocess.nfc_beam_recv()
    file = open("nfccode.txt", "r")
    nfc_code_get = file.read()
    print(nfc_code_get)
    r =
requests.get('https://hizhh.me/api/unlock/1',auth=HTTPBasicAuth(nfc_code_get,'sdf'),
verify=False)
    if(r.text != "Unauthorized Access"):
        open_door()
        print("OPEN SESAME")
        time.sleep(5)
```

```
            servo.wait_for_lever()
            #time.sleep(20)
            close_door()
        file.close()
        file = open("nfccode.txt", "w+")
        file.close()
        main_box.destroy()
        nfc_subprocess.stop_stream()
        show_start()

def del_main():
    global initial_box
    global welcome_message
    welcome_message.destroy()
    initial_box.destroy()

def make_main():
    global main_box
    global welcome_message
    global initial_box
    main_box = Box(app,layout = "auto")
    welcome_message = Text(main_box, text="Choose Method of Entry", size = 20)
    #box to hold 2 buttons
    initial_box = Box(main_box, layout = "grid")
    enter_code = PushButton(initial_box, grid = [0,0], text = "Enter Code", command
= onclick_code, padx = 30, pady = 30)
    enter_code.text_size = 20
    nfc = PushButton(initial_box, grid = [0,1], text = "NFC", command = onclick_nfc,
padx = 77, pady = 30)
    nfc.text_size = 20

def onclick_start():
    nfc_subprocess.start_stream()
    make_main()
    start_button.hide()
    timeout_en.value = 1
    start_time.value = time.time()

def show_start():
    start_button.show()


# Action you would like to perform
def counter():
    if(int(timeout_en.value) == 1 and time.time() - float(start_time.value) >
timeout_val):
        timeout_en.value = 0
        main_box.destroy()
        show_start()
        nfc_subprocess.stop_stream()
```

```
#main code loop


app = App(title="GUI")
app.tk.overrideredirect(True) #fullscreen mode
app.tk.overrideredirect(False)
app.tk.attributes('-fullscreen',True)
app.tk.config(cursor="none")

start_button = PushButton(app, text = "Start", command = onclick_start, padx = 50,
pady = 50)
start_button.text_size = 20

timeout_en = Text(app, text="0")
timeout_en.hide()

start_time = Text(app, text="0")
start_time.hide()

global my_counter
timeout_en.repeat(100, counter)  # Schedule call to counter() every 1000ms

#put all app things before this app.display()
app.display()
```

servo.py (used for servo and lever sensor control)

```
import RPi.GPIO as GPIO
from time import sleep

def SetAngle(angle):
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(36, GPIO.OUT)
        pwm=GPIO.PWM(36, 50)
        pwm.start(0)
        duty = angle / 18 + 2
        GPIO.output(36, True)
        pwm.ChangeDutyCycle(duty)
        sleep(1)
        GPIO.output(36, False)
        pwm.ChangeDutyCycle(0)
        pwm.stop()
        GPIO.cleanup()

def wait_for_lever():
```

```
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(32, GPIO.OUT)
        GPIO.output(32,1)
        GPIO.setup(40, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
        while(not(GPIO.input(40))):
                var = 1
        sleep(2)
        GPIO.cleanup()
```

nfc_subprocess.py (used to call the nfc library and start the stream)

```
import subprocess
import time
def nfc_beam_recv():
    #subprocess.call(["python", "beam.py", "--device", "tty:S0", "recv", "print"])
    python3_command ="python beam.py --device tty:S0 recv print"
    p = subprocess.Popen(python3_command.split(), stdout=subprocess.PIPE)
    start_time = time.time()
    while(p.poll() == None and time.time() - start_time < 20):
        var = 1
    if(p.poll == None):
        p.kill()
    print("Done")

def start_stream():
    python3_command ="sudo ./start_stream.sh"
    subprocess.Popen(python3_command.split(), stdout=subprocess.PIPE)

def stop_stream():
    python3_command ="sudo ./stop_stream.sh"
    subprocess.Popen(python3_command.split(), stdout=subprocess.PIPE)
```

start_stream.sh (script to start the stream)

```
if pgrep mjpg_streamer > /dev/null
then
  echo "mjpg_streamer already running"
else
  LD_LIBRARY_PATH=/opt/mjpg-streamer/ /opt/mjpg-streamer/mjpg_streamer -i
"input_raspicam.so -fps 10 -q 30 -x 640 -y 480" -o "output_http.so -p 80 -w
/opt/mjpg-streamer/www" > /dev/null 2>&1&
  echo "mjpg_streamer started"
fi
```

stop_stream.sh (script to end the stream)

```
if pgrep mjpg_streamer
then
  kill $(pgrep mjpg_streamer) > /dev/null 2>&1
  echo "mjpg_streamer stopped"
else
  echo "mjpg_streamer not running"
fi
```

Appendix E – Web app code

Server Backend

```
#!/usr/bin/python3.6
#
https://stackoverflow.com/questions/15279793/how-to-invoke-a-specific-python-version-within-
a-script-py-windows
# https://askubuntu.com/questions/528009/how-to-tell-ubuntu-to-use-python-3-4-instead-of-2-7
from flask import (Flask, render_template, flash, request, url_for,
                   redirect, session, jsonify, g, )
import sys
from wtforms import Form, BooleanField, StringField, PasswordField, validators
from flask_socketio import SocketIO, emit, send

from pyutil.mysql_connection import connection
from passlib.apps import custom_app_context as pwd_context
from passlib.hash import sha256_crypt
from pymysql import escape_string as thwart  # for SQL Injection
from itsdangerous import (TimedJSONWebSignatureSerializer
                          as Serializer, BadSignature, SignatureExpired)
import gc
import os
from functools import wraps
import requests
import json
import datetime

from flask_httpauth import HTTPBasicAuth
from flask_sqlalchemy import SQLAlchemy

import random
from bokeh.models import (HoverTool, FactorRange, Plot, LinearAxis, Grid,
                          Range1d)
from bokeh.models.glyphs import VBar
from bokeh.plotting import figure
from bokeh.charts import Bar
from bokeh.embed import components
from bokeh.models.sources import ColumnDataSource

first_app = Flask(__name__)
sio = SocketIO(first_app, engineio_logger=True)  # wrap first_app with flask socket io

#
https://stackoverflow.com/questions/35657821/the-session-is-unavailable-because-no-secret-ke
```

```
y-was-set-set-the-secret-key-on/35657961
# it is just A key, and this is not a good practice. We will change it later
first_app.secret_key =
b'\xc0\xa2V\xe1\xda\xfd@2B\xcc\xc0\x16U0\x88\x1c\xa6W\xe2^\xa6\x8e\xe4l'
first_app.config.update(
   TEMPLATES_AUTO_RELOAD=True
)

# use a flaks mysqlalchemy to map sql columns to python objects
first_app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:nZVLJudP4Qaz@localhost/291G19P2'
first_app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
first_app.config['SQLALCHEMY_COMMIT_ON_TEARDOWN'] = True

db = SQLAlchemy(first_app)
auth = HTTPBasicAuth()


# user model
class User(db.Model):
   __tablename__ = 'users'
   uid = db.Column('uid', db.Integer, primary_key=True)  # auto increment is set
automatically
   username = db.Column('username', db.Unicode)
   password = db.Column('password', db.Unicode)
   email = db.Column('email', db.Unicode)
   settings = db.Column('settings', db.Unicode)
   tracking = db.Column('tracking', db.Unicode)
   rank = db.Column('rank', db.Integer)

   def hash_password(self, password):
       self.password = pwd_context.encrypt(password)

   def verify_password(self, password):
       return pwd_context.verify(password, self.password)

   def generate_auth_token(self, expiration=600):
       s = Serializer(first_app.secret_key, expires_in=expiration)
       return s.dumps({'uid': self.uid})

   @staticmethod
   def verify_auth_token(token):
       s = Serializer(first_app.secret_key)
       try:
           data = s.loads(token)
       except SignatureExpired:
           return None  # valid token, but expired
       except BadSignature:
           return None  # invalid token
       user = User.query.get(data['uid'])
       return user


@auth.verify_password
def verify_password(username_or_token, password):
   # first try to authenticate by token
   # history is only saved someone wants to use your username and password to authenticate
```

```
    user = User.verify_auth_token(username_or_token)
    if not user:

        # try to authenticate with username/password
        user = User.query.filter_by(username=username_or_token).first()
        if not user:
            return False  # username is wrong, we are not gonna save it to the history
        if not user.verify_password(password):
            # denied
            history = LockHistory(username=username_or_token,
                                  time=datetime.datetime.now().strftime("%I:%M%p on %B %d,
%Y"), status='denied')
            db.session.add(history)
            db.session.commit()
            return False
        else:
            # authorized
            history = LockHistory(username=username_or_token,
                                  time=datetime.datetime.now().strftime("%I:%M%p on %B %d,
%Y"), status='authorized')
            db.session.add(history)
            db.session.commit()

    g.user = user
    return True


# lock model
class Locks(db.Model):
    __tablename__ = 'locks'
    lid = db.Column('lid', db.Integer, primary_key=True)  # auto-increment automatically
    username = db.Column('username', db.Unicode)
    streamURL = db.Column('streamURL', db.Unicode)
    # lock id is unique


# lock history model
class LockHistory(db.Model):
    __tablename__ = 'lock_history'
    hid = db.Column('hid', db.Integer, primary_key=True)  # auto-increment automatically
    username = db.Column('username', db.Unicode)
    time = db.Column('time', db.Unicode)
    status = db.Column('status', db.Unicode)  # status is either "authorized" or "denied"


api_username = "iwant2use8pi"
api_password = "291piapiapiapia"

# api links ------------------------------------------------------------------
cameraControl = {
    'left': 'https://nubblier-octopus-5424.dataplicity.io/cameraControl/api/v1.0/tasks/2',
    'right': 'https://nubblier-octopus-5424.dataplicity.io/cameraControl/api/v1.0/tasks/1',
    'start streaming':
'https://nubblier-octopus-5424.dataplicity.io/cameraControl/api/v1.0/tasks/3',
    'stop streaming':
'https://nubblier-octopus-5424.dataplicity.io/cameraControl/api/v1.0/tasks/4'
```

```
}

lockControl = {
   'lock': 'https://nubblier-octopus-5424.dataplicity.io/lockControl/api/v1.0/tasks/1',
   'unlock': 'https://nubblier-octopus-5424.dataplicity.io/lockControl/api/v1.0/tasks/2',
}


# user login, registering, authentication system --------------------------------
def logout_required(f):
   @wraps(f)
   def wrap(*args, **kwargs):
       if 'logged_in' in session:
           flash("You need to logout first!")
           return redirect(url_for('dashboard'))
       else:
           return f(*args, *kwargs)

   return wrap


@first_app.route('/login/', methods=["GET", "POST"])
@logout_required
def login_page():
   # return render_template("login.html")

   error = ''
   try:
       c, conn = connection()
       if request.method == "POST":
           data = c.execute("SELECT * FROM users WHERE username = (%s)",
                            thwart(request.form['username']))

           data = c.fetchone()[2]

           # if sha256_crypt.verify(request.form['password'], data):
           if pwd_context.verify(request.form['password'], data):
               session['logged_in'] = True
               session['username'] = request.form['username']
               history = LockHistory(username=request.form['username'],
time=datetime.datetime.now().strftime("%I:%M%p on %B %d, %Y"), status='authorized')
               db.session.add(history)
               db.session.commit()

               flash("You are now logged in.")
               return redirect(url_for('dashboard'))
           else:
               history = LockHistory(username=request.form['username'],
time=datetime.datetime.now().strftime("%I:%M%p on %B %d, %Y"), status='denied')
               db.session.add(history)
               db.session.commit()
               error = "Invalid credentials, try again."

       gc.collect()
       return render_template("login.html", error=error)
```

```
    except Exception as e:
        # flash(e)
        error = "Invalid credentials, try again."
        return render_template("login.html", error=error)


def login_required(f):
    @wraps(f)
    def wrap(*args, **kwargs):
        if 'logged_in' in session:
            return f(*args, *kwargs)
        else:
            flash("You need to login first!")
            return redirect(url_for('login_page'))

    return wrap


@first_app.route("/logout/")
@login_required  # you have to login first before log out
def logout():
    session.clear()
    flash("You have been logged out.")
    gc.collect()
    return redirect(url_for('dashboard'))


class SignupForm(Form):
    username = StringField('Username', [validators.Length(min=4, max=20)])
    email = StringField('Email Address', [validators.Length(min=6, max=50),
validators.Email("Invalid email")])
    password = PasswordField('New Password', [
        validators.DataRequired(),
        validators.EqualTo('confirm', message='Passwords must match')
    ])
    confirm = PasswordField('Repeat Password')
    accept_tos = BooleanField('I accept the <a href = "/tos/">Terms of Service</a>',
                              [validators.DataRequired()])


@first_app.route('/register/', methods=["GET", "POST"])
def register_page():
    try:
        form = SignupForm(request.form)  # create a form from html

        # user submit a form for registering
        if request.method == "POST" and form.validate():  # check if the method is post and
validate the form
            username = form.username.data  # get data from the wrtform we created
            email = form.email.data
            # password = sha256_crypt.encrypt((str(form.password.data)))  # encrypt password
            password = pwd_context.encrypt((str(form.password.data)))  # encrypt password
            c, conn = connection()  # connection and cursor

            # return rows of data from sql db
            x = c.execute("SELECT * FROM users WHERE username = (%s)", (thwart(username),))
```

```
            # if length of rows is greater than 1, that means that username is already taken
            if int(x) > 0:
                flash("That username is already taken, please choose another")
                return render_template('register.html', form=form)  # we will make the html
later
            else:
                c.execute("INSERT INTO users (username, password, email, tracking) VALUES
(%s, %s, %s, %s)",
                          (thwart(username)
                           , thwart(password), thwart(email), thwart("/dashboard/")))  # not
sure what tracking is doing
                conn.commit()  # save the changes in db

                flash("Thanks for registering!")
                c.close()
                conn.close()
                gc.collect()  # keep memory waste down.

                # session is used to track users
                session["logged_in"] = True
                session["username"] = username

                return redirect(url_for('dashboard'))

        # nothing has happened
        return render_template("register.html", form=form)
    except Exception as e:
        return render_template("500.html", error=str(e))


# Flask routing the user to different pages --------------------------------
@first_app.route('/')
def homepage():
    try:
        return render_template("dashboard.html")
    except Exception as e:
        return render_template("500.html", error=str(e))


@first_app.route('/dashboard/')
def dashboard():
    try:
        return render_template("dashboard.html")
    except Exception as e:
        return render_template("500.html", error=str(e))


@first_app.route('/statistics/')
@login_required
def statistics():
    locklistSet = []
    username = session['username']
    userLocks = Locks.query.filter_by(username=username).all()
    if userLocks is not None:
        for lock in userLocks:
```

```
            locklistSet.append({"lid": str(lock.lid), "streamURL": lock.streamURL,
"username": lock.username})

    denied = 0
    authorized = 0
    historySet = []
    histories = LockHistory.query.filter_by(username=username).all()
    if histories is not None:
        for history in histories:
            historySet.append(
                {"hid": str(history.hid), "username": history.username, "time": history.time,
"status": history.status})
            if history.status == "denied":
                denied += 1
            else:
                authorized += 1

    historySet = reversed(historySet)

    try:
        return render_template("statistics.html", locklistSet=locklistSet,
                               historySet=historySet, denied=denied, authorized=authorized)
    except Exception as e:
        return render_template("500.html", error=str(e))


@first_app.route('/chat/')
@login_required  # you have to login first before log out
def load_chat():
    try:
        flash("You can chat with other users on this page")
        return render_template("chat.html")
    except Exception as e:
        return render_template("500.html", error=str(e))


@first_app.route('/video/')
@login_required  # you have to login first before log out
def load_video():
    # return ":( This page is now under development..."
    return render_template("video.html")


@first_app.route('/tos/')
def load_terms():
    # return ":( This page is now under development..."
    return render_template("terms.html")


#  RESTful APIs
@first_app.route('/api/users', methods=['POST'])
def new_user():
    username = request.json.get('username')
    password = request.json.get('password')
    email = request.json.get('email')
    if username is None or password is None or email is None:
```

```
        abort(400)  # missing arguments
    if User.query.filter_by(username=username).first() is not None:
        abort(400)  # existing user
    user = User(username=username, email=email, tracking='/dashboard/')
    user.hash_password(password)
    db.session.add(user)
    db.session.commit()
    return (jsonify({'username': user.username, 'email': user.email}), 201,
            {'Location': url_for('get_user', id=user.uid, _external=True)})


@first_app.route('/api/users/<int:id>')
def get_user(id):
    user = User.query.get(id)
    if not user:
        abort(400)
    return jsonify({'username': user.username})


@first_app.route('/api/token/<int:duration>')
@auth.login_required
def get_auth_token(duration):
    token = g.user.generate_auth_token(expiration=duration)
    return jsonify({'token': token.decode('ascii'), 'duration': duration})


@first_app.route('/api/resource')
@auth.login_required
def get_resource():
    return jsonify({'data': 'Hello, %s!' % g.user.username})


@first_app.route('/api/unlock/<int:lock_id>', methods=['GET'])
@auth.login_required
def unlock(lock_id):
    return jsonify({('Lock ' + str(lock_id) + ' Status'): 'Authorized.'})


# additional APIs
@first_app.route('/api/resource/streamURL', methods=['GET'])
@auth.login_required
def get_stream_url():
    # username = request.json.get('username')
    # lid = request.json.get('lid')
    username = request.args.get('username')
    lid = request.args.get('lid')
    lid = int(lid)

    if username is None or lid is None:
        abort(400)

    userLocks = Locks.query.filter_by(username=username).all()
    if userLocks is None:
        abort(400)

    for lock in userLocks:
```

```
        if lock.lid == lid:
            streamURL = {'streamURL': lock.streamURL}
            return jsonify(streamURL)

    abort(400)


@first_app.route('/api/resource/lockList/<username>', methods=['GET'])
@auth.login_required
def get_locklist(username):
    locklistSet = []
    if username is None:
        abort(400)  # missing arguments
    lockList = Locks.query.filter_by(username=username).all()
    if lockList is None:
        abort(400)

    for lock in lockList:
        locklistSet.append({"lid": str(lock.lid), "streamURL": lock.streamURL})

    return json.dumps(locklistSet)


@first_app.route('/api/resource/entry_history/<username>', methods=['GET'])
@auth.login_required
def get_history(username):
    historySet = []
    histories = LockHistory.query.filter_by(username=username).all()
    if histories is not None:
        for history in histories:
            historySet.append(
                {"hid": str(history.hid), "username": history.username, "time": history.time,
"status": history.status})

    return json.dumps(historySet)


@first_app.route('/api/addLock', methods=['POST'])
@auth.login_required
def add_lock():
    username = request.json.get('username')
    streamURL = request.json.get('streamURL')
    if username is None or streamURL is None:
        abort(400)  # missing arguments
    if Locks.query.filter_by(streamURL=streamURL).first() is not None:
        abort(400)  # existing streaming address
    lock = Locks(username=username, streamURL=streamURL)
    db.session.add(lock)
    db.session.commit()
    return jsonify({'lid': lock.lid, 'username': lock.username, 'streamURL':
lock.streamURL}), 201


# error handler for 404 pages ------------------------------------------------------------
@first_app.errorhandler(404)
def page_not_found(e):
```

```
    return render_template("404.html", error=str(e))


# socket io handler
@sio.on('my_event', namespace='/chat')
def handleMessage(msg):
    print('Message: ' + msg)
    emit('my_response', msg, broadcast=True)


# communication with pi ----------------------------------------------------------------
@sio.on('pi_event', namespace='/pi')
def handlePiEvent(msg):
    if msg in cameraControl:
        r = requests.get(cameraControl[msg], auth=(api_username, api_password))
        str_json = json.dumps(r.json())
        emit('pi_response', str_json)
    else:
        emit('pi_response', msg)


# launch this app
if __name__ == "__main__":
    socketio.run(first_app, debug=False)
```

Server Frontend

```
{% extends "header.html" %} {% block body %}
<style>
    .top-buffer {
        margin-top: 20px;
    }

    .classWithPad {
        margin: 10px;
    }

</style>


<div class="container-fluid" style="margin-top:50px;">
    <div class="d-flex justify-content-center">
        <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel"
style="width: 60rem; ">
            <ol class="carousel-indicators">
                <li data-target="#carouselExampleIndicators" data-slide-to="0"
class="active"></li>
                <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
                <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
            </ol>
            <div class="carousel-inner">
                <div class="carousel-item active">
                    <img class="d-block w-100" src="{{ url_for('static', filename =
'images/2.jpg')}}"
                         alt="First slide">
```

```
                </div>
                <div class="carousel-item">
                    <img class="d-block w-100" src="{{ url_for('static', filename =
'images/4.jpg')}}"
                         alt="Second slide">
                </div>
                <div class="carousel-item">
                    <img class="d-block w-100" src="{{ url_for('static', filename =
'images/5.jpg')}}"
                         alt="Third slide">
                </div>
            </div>
            <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button"
data-slide="prev">
                <span class="carousel-control-prev-icon" aria-hidden="true"></span>
                <span class="sr-only">Previous</span>
            </a>
            <a class="carousel-control-next" href="#carouselExampleIndicators" role="button"
data-slide="next">
                <span class="carousel-control-next-icon" aria-hidden="true"></span>
                <span class="sr-only">Next</span>
            </a>

        </div>
    </div>


    <div class="row justify-content-center top-buffer">

        <div class="card classWithPad " style="width: 18rem; ">
            <div class="card-header">
                <img height="20" width="20" src="{{ url_for('static', filename =
'images/video.png')}}">
            </div>
            <div class="card-body">
                <h5 class="card-title">Video Streaming</h5>
                <p class="card-text">Live feed from the lock</p>
                <a href="/video/" class="btn btn-primary">Go</a>
            </div>
        </div>


        <div class="card classWithPad " style="width: 18rem; ">
            <div class="card-header">
                <img height="20" width="20" src="{{ url_for('static', filename =
'images/statistics.png')}}">
            </div>
            <div class="card-body">
                <h5 class="card-title">Statistics</h5>
                <p class="card-text">Check out the authentication log</p>
                <a href="/statistics/" class="btn btn-primary">Go</a>
            </div>
        </div>

        <div class="card classWithPad " style="width: 18rem; ">
            <div class="card-header">
```

```
                <img height="20" width="20" src="{{ url_for('static', filename =
'images/chat.png')}}">
            </div>
            <div class="card-body">
                <h5 class="card-title">Chat</h5>
                <p class="card-text">Let's have a chat on smart locks</p>
                <a href="/chat/" class="btn btn-primary">Go</a>
            </div>
        </div>


    </div>


</div>

{% endblock %}



{% extends "header.html" %} {% block body %}
<style>
    .classWithPad {
        margin: 10px;
    }

    .top-buffer {
        margin-top: 50px;
    }

</style>

<div class="container-fluid top-buffer" style="max-width: 40%">
    <div class="container-fluid justify-content-center">


        <h5> Register </h5>
        <br>

        {% from "_formhelpers.html" import render_field %}
        <form method="post" action="/register/">
            <dl>

                {{render_field(form.username)}}
                {{render_field(form.email)}}
                {{render_field(form.password)}}
                {{render_field(form.confirm)}}
                {{render_field(form.accept_tos)}}

            </dl>
            <p><input type=submit value=Register></p>
        </form>

        {% if error %}
        <p class="error"><strong>Error:</strong>{{error}}</p>
        {% endif %}
```

```
    </div>
</div>

{% endblock %}


{% extends "header.html" %} {% block body %}


<div class="container-fluid" style="margin-top:50px;">

    <ul class="nav nav-tabs" id="myTab" role="tablist">
        <li class="nav-item">
            <a class="nav-link active" id="home-tab" data-toggle="tab" href="#home" role="tab"
aria-controls="home"
               aria-selected="true">Locks</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" id="profile-tab" data-toggle="tab" href="#profile" role="tab"
aria-controls="profile"
               aria-selected="false">Authentication Log</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" id="contact-tab" data-toggle="tab" href="#contact" role="tab"
aria-controls="contact"
               aria-selected="false">Summary</a>
        </li>
    </ul>
    <div class="tab-content" id="myTabContent">
        <div class="tab-pane fade show active" id="home" role="tabpanel"
aria-labelledby="home-tab">
            <ul class="list-group">
                {% for lock in locklistSet %}
                <li class="list-group-item">Lock#{{lock.lid}} @{{lock.streamURL}}</li>
                {% endfor %}
            </ul>
        </div>
        <div class="tab-pane fade" id="profile" role="tabpanel" aria-labelledby="profile-tab">
            <ul class="list-group">
                {% for history in historySet %}
                <li class="list-group-item">{{history.status}}: {{history.time}}</li>
                {% endfor %}
            </ul>
        </div>
        <div class="tab-pane fade" id="contact" role="tabpanel" aria-labelledby="contact-tab">
            <ul class="list-group">
                <li class="list-group-item d-flex justify-content-between align-items-center">
                    Denied
                    <span class="badge badge-primary badge-pill">{{denied}}</span>
                </li>
                <li class="list-group-item d-flex justify-content-between align-items-center">
                    Authorized
                    <span class="badge badge-primary badge-pill">{{authorized}}</span>
                </li>
            </ul>
```

```
        </div>
    </div>
</div>

{% endblock %}

{% extends "header.html" %} {% block body %}
<style>
    .classWithPad {
        margin: 10px;
    }

    .top-buffer {
        margin-top: 50px;
    }

</style>


<!--<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.8/socket.io.min.js"></script>-->
<script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.0.4/socket.io.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
<script type="text/javascript" charset="utf-8">
    $(document).ready(function () {


        var namespace = '/pi';
        // The connection URL has the following format:
        //     http[s]://<domain>:<port>[/<namespace>]
        var socket = io.connect('https://' + document.domain + ':' + location.port + namespace);
        // var socket = io.connect('https://hizhh.me:8000/chat');
        // var socket = io.connect("https://nubblier-octopus-5424.dataplicity.io/" + namespace);


        // when connection is established successfully, send a message to server
        socket.on('connect', function () {
            // socket.emit('pi_event', 'User has connected!');
            console.log('Ready to send command to rpi');

        });

        //https://stackoverflow.com/questions/10437584/socket-io-reconnect

        socket.on('disconnect', function () {
            // socket.emit('pi_event', 'User has disconnected!');
            console.log('Disconnect..');

        });


        socket.on('pi_response', function (msg) {
            $("#messages").append('<li>' + msg + '</li>');
            console.log('Received message');
        });
```

```
        $('#camera_right').on('click', function () {
            socket.emit('pi_event', 'right');
        });

        $('#camera_left').on('click', function () {
            socket.emit('pi_event', 'left');
        });

        $('#start_streaming').on('click', function () {
            socket.emit('pi_event', 'start streaming');
            $("#stream_feed").attr('src',
"https://acerate-scorpion-5728.dataplicity.io/?action=stream");
        });

        $('#stop_streaming').on('click', function () {
            socket.emit('pi_event', 'stop streaming');
            $("#stream_feed").attr('src', "");
        });


    });


</script>

<div class="container-fluid" style="margin-top:50px;">
    <!--<form id="emit" method="POST" action='#'>-->
    <!--<input type="text" name="emit_data" id="emit_data" placeholder="Message">-->
    <!--<input type="submit" value="Echo">-->
    <!--</form>-->
    <!--<form id="broadcast" method="POST" action='#'>-->
    <!--<input type="text" name="broadcast_data" id="broadcast_data" placeholder="Message">-->
    <!--<input type="submit" value="Broadcast">-->
    <!--</form>-->

    <!--<form id="disconnect" method="POST" action="#">-->
    <!--<input type="submit" value="Disconnect">-->
    <!--</form>-->
    <!--<div id="log"></div>-->
    <!--<input type="text" id="myMessage">-->
    <div class="d-flex justify-content-center">

        <div class="embed-responsive embed-responsive-16by9 classWithPad top-buffer"
style="max-width: 640px">
            <iframe class="embed-responsive-item" style="background: #000000;" id="stream_feed"
                    src="https://acerate-scorpion-5728.dataplicity.io/?action=stream"></iframe>
        </div>
    </div>

    <!--<button id="run_script" class="btn btn-primary classWithPad">Run Script</button>-->

    <ul id="messages"></ul>


</div>
```

```
{% endblock %}


<!doctype html>
<html lang="en">

<head>
   <meta charset="utf-8">
   <title>Project2 G19</title>
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <script defer src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
   <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
           integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
           crossorigin="anonymous"></script>
   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
           integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
           crossorigin="anonymous"></script>
   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
           integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
           crossorigin="anonymous"></script>

   <link href="{{ url_for('static', filename='css/bootstrap.min.css') }}" rel="stylesheet">
   <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}">

</head>


<header>
   <nav class="navbar navbar-expand-lg  navbar-light bg-light">
       <a class="navbar-brand" href="/">
           <img width="40" height="40" src="{{ url_for('static', filename =
'images/lock.png')}}">
       </a>
       <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent"
               aria-controls="navbarSupportedContent"
               aria-expanded="false" aria-label="Toggle navigation">
           <span class="navbar-toggler-icon"></span>
       </button>


       <div class="collapse navbar-collapse" id="navbarSupportedContent">
           <ul class="navbar-nav ml-auto">
               {% if session.logged_in %}
               <li class="nav-item"><a class="nav-link" href="/logout/">Logout</a></li>

               {% else %}
               <li class="nav-item"><a class="nav-link" href="/register/">Sign up</a></li>
               <li class="nav-item"><a class="nav-link" href="/login/">Log in</a></li>

               {% endif %}
           </ul>
       </div>
   </nav>
```

```
</header>


<body>
<div class="container-fluid" style="min-height: 100%">

   <!-- flash messages  -->
   <!-- https://stackoverflow.com/questions/1369526/what-is-the-python-keyword-with-used-for
-->
   {% with messages = get_flashed_messages() %}
   {% if messages %}
   {% for message in messages %}
   <div class="alert alert-warning  alert-dismissible fade show" role="alert">
       <button type="button" class="close" data-dismiss="alert" aria-label="Close">
           <span aria-hidden="true">&times;</span>
       </button>
       {{message}}
   </div>
   {% endfor %}
   {% endif %}
   {% endwith %}


   <!-- Content here -->
   <!-- body is gonna be extended in other html files -->
   {% block body %}{% endblock %}


</div>
</body>


</html>


{% extends "header.html" %} {% block body %}


<div class="container-fluid" style="max-width: 40%">
   <br>
   <h5> Please log in </h5>
   <br>
   <p>{{ error }}</p>

   <form action="" class="form" method="post">
       <div class="form-group">
           <input type="text" class="form-control" id="usernameInput" placeholder="Username"
name="username" value="{{ request.form.username }}">
       </div>
       <div class="form-group">
           <input type="password" class="form-control" id="passwordInput"
placeholder="Password" name="password" value="{{ request.form.password }}">
       </div>


       <!-- value attributes will be passed back to python for validation -->
```

```
        <!-- <div class="form-check">
            <input type="checkbox" class="form-check-input" id="exampleCheck1">
            <label class="form-check-label" for="exampleCheck1">Remember Password</label>
        </div> -->

        <!-- implement remember password and usename later  -->


        <input class="btn btn-primary" type="submit" value="Login">
    </form>

</div>

{% endblock %}
```

Appendix F – Mobile app code

App

AddLockDialog.java (used to get lock URL from user)

```
package com.g19p2.g19p2app;

import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatDialogFragment;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.EditText;

/**
* Created by robin on 2018-03-30.
* Dialog used to get information for a new lock.
*/

public class AddLockDialog extends AppCompatDialogFragment {
    private EditText edit_lock_url;
    private AddLockDialogListener listener;

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

        LayoutInflater inflater = getActivity().getLayoutInflater();
        View view = inflater.inflate(R.layout.dialog_addlock, null);

        builder.setView(view)
                .setTitle("Add Lock")
                .setNegativeButton("cancel", new DialogInterface.OnClickListener() {
```

```
                    @Override
                    public void onClick(DialogInterface dialog, int which) {

                    }
                })
                .setPositiveButton("add", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        listener.applyTexts(edit_lock_url.getText().toString());
                    }
                });

        edit_lock_url = view.findViewById(R.id.edit_lock_url);


        return builder.create();
    }

    @Override
    public void onAttach(Context context) {
        try {
            listener = (AddLockDialogListener) getTargetFragment();
        } catch(ClassCastException e) {
            throw new ClassCastException (context.toString() + "must implement AddLockDialogListener");
        }

        super.onAttach(context);
    }
}
```

AddLockDialogListener.java (interface for lock fragment)

```
package com.g19p2.g19p2app;

/**
* Created by robin on 2018-03-30.
* Listener for transfering lock id and password to fragment.
*/

public interface AddLockDialogListener {
    void applyTexts(String lock_url);
}
```

Constants.java (holds app variables)

```
package com.g19p2.g19p2app;

/**
* Created by Amir on 3/31/2018.
*/

public class Constants {
    public static String userName = null;
    public static String password = null;
    public static String token = null;
    public static G19P2Client client = null;

    public static String lock_id = null;
```

```
    public static String url = null;
    public static boolean urlValid = false;

    public static String nfc_tag = null;
}
```

G19P2Client.java (sends HTTP requests to server)

```
package com.g19p2.g19p2app;

import android.os.AsyncTask;

import org.json.JSONObject;

/**
* Created by robin on 2018-03-22.
* Client that gets replies by sending requests to a server.
*
* Edited by Amir 2018-03-31
* Changed to a TCP client which operates in threads
*/

public class G19P2Client extends AsyncTask<String, Void, String> {

    Constants constants = new Constants();

    public static final String
            REQUEST_LOGIN = "0",
            REQUEST_GET_LOCKS = "1",
            REQUEST_GET_STREAM_URL = "2",
            REQUEST_GET_VISITS = "4",
            REQUEST_ADD_LOCK = "5";

    protected String doInBackground(String... args) {
        switch(args[0])
        {
            case REQUEST_LOGIN:
                return login();
            case REQUEST_GET_LOCKS:
                return locks();
            case REQUEST_GET_STREAM_URL:
                return stream(args[1]);
            case REQUEST_GET_VISITS:
                return visits();
            case REQUEST_ADD_LOCK:
                return addlock(args[1]);
        }

        return "Error";
    }

    private String login(){
        String s;
        try{
            // curl -u username:password -i https://hizhh.me/api/token/6000
            HttpRequest con =
HttpRequest.get("https://hizhh.me/api/token/6000").basic(constants.userName, constants.password);
```

```
            int result = con.code();
            if(result == 200)
            {
                JSONObject jo = new JSONObject(con.body());
                constants.token = jo.getString("token");
                constants.nfc_tag = constants.token;
                s = "2";
            } else if(result == 500){
                s = "1";
            } else
                s = "0";
            con.disconnect();
        } catch(Exception e){
            s = "Exception";
        }
        return s;
    }

    private String stream(String lockID){
        String s;
        try{
            // curl -u token:password
"https://hizhh.me/api/resource/streamURL?username=username&lid=lockID"
            HttpRequest con = HttpRequest.get("https://hizhh.me/api/resource/streamURL", true,
"username", constants.userName, "lid", lockID)
                    .basic(constants.token, "something");
            int result = con.code();
            if(result == 200)
            {
                JSONObject url = new JSONObject(con.body());
                s = url.getString("streamURL");
            } else
                s =  "Connection Issue: " + String.valueOf(result);
            con.disconnect();
        } catch(Exception e){
            s = "Exception";
        }
        return s;
    }

    private String locks(){
        String s;
        try{
            // curl -u newnewtest:password -i -X GET https://hizhh.me/api/resource/lockList/<username>
            HttpRequest con = HttpRequest.get("https://hizhh.me/api/resource/lockList/" +
constants.userName).basic(constants.token, "something");
            int result = con.code();
            if(result == 200)
                s = con.body();
            else
                s = "Connection Issue: " + String.valueOf(result);
            con.disconnect();
        } catch(Exception e){
            s = "Exception";
        }
        return s;
```

```
    }

    private String visits(){
        String s;
        try{
            // curl -u <token>:<whatever> -i -X GET
https://hizhh.me/api/resource/entry_history/<username>
            HttpRequest con = HttpRequest.get("https://hizhh.me/api/resource/entry_history/" +
constants.userName).basic(constants.token, "something");
            int result = con.code();
            if(result == 200)
                s = con.body();
            else
                s = "Connection Issue: " + String.valueOf(result);
            con.disconnect();
        } catch(Exception e){
            s = "Exception";
        }
        return s;
    }

    private String addlock(String URL){
        String s;
        try{
            // curl -u <token>:<whatever> -i -X POST -H "Content-Type: application/json" -d
'{"username":<username>,"streamURL":<streamURL>}' https://hizhh.me/api/addLock
            JSONObject jo = new JSONObject().put("username",constants.userName).put("streamURL", URL);
            HttpRequest con = HttpRequest.post("https://hizhh.me/api/addLock").basic(constants.token,
"something")
                    .contentType(HttpRequest.CONTENT_TYPE_JSON)
                    .send(jo.toString());
            int result = con.code();
            if(result == 201)
                s = con.body();
            else
                s = "Connection Issue " + String.valueOf(result);
        } catch(Exception e){
            s = "Exception";
        }
        return s;
    }
}
```

Listener.java

```
package com.g19p2.g19p2app;

public interface Listener {

    void onDialogDisplayed();

    void onDialogDismissed();
}
```

Lock.java (represents lock object)

```
package com.g19p2.g19p2app;
```

```
/**
* Created by robin on 2018-04-02.
*/

public class Lock {
   private String lock_id, lock_url;

   public Lock(String lock_id, String lock_url) {
       this.lock_id = lock_id; this.lock_url = lock_url;
   }

   public String getLock_id() {
       return lock_id;
   }

   public String getLock_url() {
       return lock_url;
   }
}
```

LockCard.java (holds UI elements concerning a lock object)

```
package com.g19p2.g19p2app;

import android.content.Context;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.widget.CardView;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

/**
* Created by robin on 2018-03-29.
* Collection of all elements of a card that displays
* the information of a lock, and the ability to
* acquire its key.
*/

public class LockCard {
   CardView card;
   TextView lock_id_label, lock_id;
   ImageButton unlock_button;
   View divider;
   String card_lock_id, card_lock_url;
   Context thisContext;
   FragmentManager manager;
   Constants c;

   public LockCard(CardView card, TextView lock_id_label, TextView lock_id,
                   ImageButton unlock_button, View divider, Context thisContext, FragmentManager
manager) {
       this.card = card;
       this.lock_id_label = lock_id_label;
       this.lock_id = lock_id;
```

```
        this.unlock_button = unlock_button;
        this.divider = divider;
        this.thisContext = thisContext;
        this.manager = manager;

        c = new Constants();
    }

    /**
     * Formats the text of the card elements to match that of the given lock.
     * precondition: the card is initially invisible
     * @param lock - Lock object to format the card with
     */
    public void formatCard(Lock lock) {
        // takes the lock ID and URL the card is associated to
        card_lock_id = lock.getLock_id();
        card_lock_url = lock.getLock_url();
        // sets the card's contents to visible
        card.setVisibility(View.VISIBLE);
        lock_id_label.setVisibility(View.VISIBLE);
        lock_id.setVisibility(View.VISIBLE);
        unlock_button.setVisibility(View.VISIBLE);
        lock_id.setText(lock.getLock_id());
        divider.setVisibility(View.VISIBLE);

        // on click, each card's button will produce the live stream
        // of its corresponding lock
        unlock_button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //c.url = "http://techslides.com/demos/sample-videos/small.mp4";
                c.url = card_lock_url;
                c.urlValid = true;
                c.lock_id = card_lock_id;

                Toast.makeText(thisContext, "loading lock " + c.lock_id + "'s stream",
Toast.LENGTH_SHORT).show();
                // create a stream fragment and set its arguments
                Fragment streamFragment = new StreamFragment();
                // switch the fragments
                FragmentTransaction transaction = manager.beginTransaction();
                transaction.replace(R.id.fragment_container, streamFragment);
                transaction.addToBackStack(null);
                transaction.commit();
            }
        });
    }

    /**
     * Makes all the contents of this card gone
     */
    public void setGone() {
        card.setVisibility(View.GONE);
        lock_id_label.setVisibility(View.GONE);
        lock_id.setVisibility(View.GONE);
        unlock_button.setVisibility(View.GONE);
```

```
        divider.setVisibility(View.GONE);
    }
}
```

LockFragment.java (app fragment that manages user's locks)

```
package com.g19p2.g19p2app;

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.Fragment;
import android.support.v7.widget.CardView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;


/**
* A simple {@link Fragment} subclass.
* Manages the locks the user has access to.
*/
public class LocksFragment extends Fragment implements AddLockDialogListener {

    private ArrayList<LockCard> LockCards;
    private ArrayList<Lock> Locks;
    private Constants c;

    public LocksFragment() {
        LockCards = new ArrayList<LockCard>();
        Locks = new ArrayList<Lock>();
    }

    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        // generate new instance of constants
        c = new Constants();

        // format each LockCard and set them each to empty
        initializeCards();
        for(LockCard lc : LockCards)
            lc.setGone();

        // get the Lock IDs of the user
        getLocks();

        // format each invisible card with the corresponding lock ID
        for(int i = 0; i < Math.min(LockCards.size(), Locks.size()); i++) {
```

```
            LockCards.get(i).formatCard(Locks.get(i));
        }

        FloatingActionButton add_lock_btn = (FloatingActionButton)
getView().findViewById(R.id.add_lock_btn);
        add_lock_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openDialog();
            }
        });
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_locks, container, false);
    }

    /**
     * Opens the dialog for adding a lock
     */
    public void openDialog() {
        AddLockDialog addLockDialog = new AddLockDialog();
        addLockDialog.show(getActivity().getSupportFragmentManager(), "add lock dialog");
        addLockDialog.setTargetFragment(LocksFragment.this, 1);
    }

    /**
     * Method where the dialog string end up.
     * Add the lock from here.
     * @param lock_url   stream url of the lock to add
     */
    @Override
    public void applyTexts(String lock_url) {
        addLock(lock_url);
    }

    /**
     * Attempts to add the lock associated to the lock ID
     * to the user's account.
     * @param lock_url
     */
    public void addLock(String lock_url) {
        c.client = new G19P2Client();
        try {
            c.client.execute(G19P2Client.REQUEST_ADD_LOCK, lock_url).get();
            LockCards.clear();  Locks.clear();
            onViewCreated(getView(), null);
        } catch (Exception e) {
            Toast.makeText(getActivity().getApplicationContext(), "failed to get reply",
Toast.LENGTH_SHORT).show();
            return;
        }
    }
```

```
    /**
     * Gets the Lock IDs of all the locks the user is authorized to have.
     */
    private void getLocks() {
        c.client = new G19P2Client();
        String reply;

        // attempt to get JSON string from server
        try {
            reply = c.client.execute(G19P2Client.REQUEST_GET_LOCKS).get();
        } catch(Exception e) {
            Toast.makeText(getActivity().getApplicationContext(), "failed to get reply",
Toast.LENGTH_SHORT).show();
            return;
        }
        //String reply = "[{\"lid\": \"1\", \"streamURL\": \"hi.com\"}, {\"lid\": \"2\", \"streamURL\":
\"hello.com\"}, {\"lid\": \"3\", \"streamURL\": \"helloagain.com\"}, {\"lid\": \"4\", \"streamURL\":
\"hihihi.com\"}]";


        // attempt to extract lock ID strings from JSON string
        try {
            JSONArray jsonArray = new JSONArray(reply);

            // take all the lock IDs and add them to LockIDs
            for(int i = 0; i < jsonArray.length(); i++) {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                String lock_id = jsonObject.getString("lid");
                String lock_url = jsonObject.getString("streamURL");
                Locks.add(new Lock(lock_id, lock_url));
            }
        } catch(JSONException e) {
            Toast.makeText(getActivity().getApplicationContext(), "reply was not JSON string",
Toast.LENGTH_SHORT).show();
            return;
        }
    }

    /**
     * Creates an object out of each CardView and its contents, and adds them to
     * an easily accessible ArrayList.
     */
    private void initializeCards() {
        CardView card0 = (CardView) getView().findViewById(R.id.lock0);
        TextView lock_id_label_0 = (TextView) getView().findViewById(R.id.lock_id_label_0);
        TextView lock_id_0 = (TextView) getView().findViewById(R.id.lock_id_0);
        ImageButton unlock_btn_0 = (ImageButton) getView().findViewById(R.id.unlock_btn_0);
        View divider_0 = (View) getView().findViewById(R.id.divider_0);

        LockCard lockCard0 = new LockCard(card0, lock_id_label_0, lock_id_0, unlock_btn_0, divider_0,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard0);

        CardView card1 = (CardView) getView().findViewById(R.id.lock1);
        TextView lock_id_label_1 = (TextView) getView().findViewById(R.id.lock_id_label_1);
```

```
        TextView lock_id_1 = (TextView) getView().findViewById(R.id.lock_id_1);
        ImageButton unlock_btn_1 = (ImageButton) getView().findViewById(R.id.unlock_btn_1);
        View divider_1 = (View) getView().findViewById(R.id.divider_1);

        LockCard lockCard1 = new LockCard(card1, lock_id_label_1, lock_id_1, unlock_btn_1, divider_1,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard1);

        CardView card2 = (CardView) getView().findViewById(R.id.lock2);
        TextView lock_id_label_2 = (TextView) getView().findViewById(R.id.lock_id_label_2);
        TextView lock_id_2 = (TextView) getView().findViewById(R.id.lock_id_2);
        ImageButton unlock_btn_2 = (ImageButton) getView().findViewById(R.id.unlock_btn_2);
        View divider_2 = (View) getView().findViewById(R.id.divider_2);

        LockCard lockCard2 = new LockCard(card2, lock_id_label_2, lock_id_2, unlock_btn_2, divider_2,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard2);

        CardView card3 = (CardView) getView().findViewById(R.id.lock3);
        TextView lock_id_label_3 = (TextView) getView().findViewById(R.id.lock_id_label_3);
        TextView lock_id_3 = (TextView) getView().findViewById(R.id.lock_id_3);
        ImageButton unlock_btn_3 = (ImageButton) getView().findViewById(R.id.unlock_btn_3);
        View divider_3 = (View) getView().findViewById(R.id.divider_3);

        LockCard lockCard3 = new LockCard(card3, lock_id_label_3, lock_id_3, unlock_btn_3, divider_3,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard3);

        CardView card4 = (CardView) getView().findViewById(R.id.lock4);
        TextView lock_id_label_4 = (TextView) getView().findViewById(R.id.lock_id_label_4);
        TextView lock_id_4 = (TextView) getView().findViewById(R.id.lock_id_4);
        ImageButton unlock_btn_4 = (ImageButton) getView().findViewById(R.id.unlock_btn_4);
        View divider_4 = (View) getView().findViewById(R.id.divider_4);

        LockCard lockCard4 = new LockCard(card4, lock_id_label_4, lock_id_4, unlock_btn_4, divider_4,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard4);

        CardView card5 = (CardView) getView().findViewById(R.id.lock5);
        TextView lock_id_label_5 = (TextView) getView().findViewById(R.id.lock_id_label_5);
        TextView lock_id_5 = (TextView) getView().findViewById(R.id.lock_id_5);
        ImageButton unlock_btn_5 = (ImageButton) getView().findViewById(R.id.unlock_btn_5);
        View divider_5 = (View) getView().findViewById(R.id.divider_5);

        LockCard lockCard5 = new LockCard(card5, lock_id_label_5, lock_id_5, unlock_btn_5, divider_5,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard5);

        CardView card6 = (CardView) getView().findViewById(R.id.lock6);
        TextView lock_id_label_6 = (TextView) getView().findViewById(R.id.lock_id_label_6);
        TextView lock_id_6 = (TextView) getView().findViewById(R.id.lock_id_6);
        ImageButton unlock_btn_6 = (ImageButton) getView().findViewById(R.id.unlock_btn_6);
        View divider_6 = (View) getView().findViewById(R.id.divider_6);

        LockCard lockCard6 = new LockCard(card6, lock_id_label_6, lock_id_6, unlock_btn_6, divider_6,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
```

```
        LockCards.add(lockCard6);

        CardView card7 = (CardView) getView().findViewById(R.id.lock7);
        TextView lock_id_label_7 = (TextView) getView().findViewById(R.id.lock_id_label_7);
        TextView lock_id_7 = (TextView) getView().findViewById(R.id.lock_id_7);
        ImageButton unlock_btn_7 = (ImageButton) getView().findViewById(R.id.unlock_btn_7);
        View divider_7 = (View) getView().findViewById(R.id.divider_7);

        LockCard lockCard7 = new LockCard(card7, lock_id_label_7, lock_id_7, unlock_btn_7, divider_7,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard7);

        CardView card8 = (CardView) getView().findViewById(R.id.lock8);
        TextView lock_id_label_8 = (TextView) getView().findViewById(R.id.lock_id_label_8);
        TextView lock_id_8 = (TextView) getView().findViewById(R.id.lock_id_8);
        ImageButton unlock_btn_8 = (ImageButton) getView().findViewById(R.id.unlock_btn_8);
        View divider_8 = (View) getView().findViewById(R.id.divider_8);

        LockCard lockCard8 = new LockCard(card8, lock_id_label_8, lock_id_8, unlock_btn_8, divider_8,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard8);

        CardView card9 = (CardView) getView().findViewById(R.id.lock9);
        TextView lock_id_label_9 = (TextView) getView().findViewById(R.id.lock_id_label_9);
        TextView lock_id_9 = (TextView) getView().findViewById(R.id.lock_id_9);
        ImageButton unlock_btn_9 = (ImageButton) getView().findViewById(R.id.unlock_btn_9);
        View divider_9 = (View) getView().findViewById(R.id.divider_9);

        LockCard lockCard9 = new LockCard(card9, lock_id_label_9, lock_id_9, unlock_btn_9, divider_9,
getActivity().getApplicationContext(), getActivity().getSupportFragmentManager());
        LockCards.add(lockCard9);
    }
}
```

LoginActivity.java

```
package com.g19p2.g19p2app;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {

    private static final String sign_up_link = "https://hizhh.me/register/";

    private Constants c = new Constants();


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_login);

        // set the title to an empty string, because it looks nice
        setTitle("");

        Button login_btn = (Button) findViewById(R.id.login_btn);
        login_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText username = (EditText) findViewById(R.id.email);
                EditText password = (EditText) findViewById(R.id.password);

                String s_username = username.getText().toString();
                String s_password = password.getText().toString();

                // if either field is empty, display the appropriate message
                if (s_username.matches("")) {
                    Toast.makeText(getApplicationContext(), "username required",
Toast.LENGTH_SHORT).show();
                    return;
                } else if (s_password.matches("")) {
                    Toast.makeText(getApplicationContext(), "password required",
Toast.LENGTH_SHORT).show();
                    return;
                } else {
                    // communicate with server to see if email and password match
                    int verified = verifyLogin(s_username, s_password);

                    switch (verified) {
                        // if communication with the server failed
                        case 0:
                            Toast.makeText(getApplicationContext(), "server is down",
Toast.LENGTH_SHORT).show();
                            break;
                        // if login information is not valid
                        case 1:
                            Toast.makeText(getApplicationContext(), "login failed",
Toast.LENGTH_SHORT).show();
                            break;
                        // if login information is valid
                        case 2:
                            Toast.makeText(getApplicationContext(), "login succesful",
Toast.LENGTH_SHORT).show();
                            Intent go_to_main = new Intent(getApplicationContext(), MainActivity.class);
                            startActivity(go_to_main);
                            break;
                        default:
                            Toast.makeText(getApplicationContext(), "LOLOLOLOL",
Toast.LENGTH_SHORT).show();
                    }

                }
            }
        });

        Button sign_up_btn = (Button) findViewById(R.id.sign_up_btn);
```

```
        sign_up_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            // on click, send the user to the sign up page on their mobile browser
            public void onClick(View v) {
                startActivity(new Intent("android.intent.action.VIEW", Uri.parse(sign_up_link)));
            }
        });
    }

    /**
     * Attempts to communicate with the server to verify the user's login information,
     * and also initializes the static values in constants for use around the app.
     *
     * @param s_username username used to verify the user's account
     * @param s_password password used to verify the user's account
     * @return 0 if communication with the server failed
     *         1 if the login information could not be verified
     *         2 if the login information was verified
     */
    private int verifyLogin(String s_username, String s_password) {
        c.client = new G19P2Client();
        c.userName = s_username;
        c.password = s_password;
        String reply;
        try {
            reply = c.client.execute(G19P2Client.REQUEST_LOGIN).get();
        }catch(Exception e){
            reply = "THREAD ERROR";
        }
        return Integer.valueOf(reply);
    }
}
```

MainActivity.java

```
package com.g19p2.g19p2app;

import android.content.Intent;
import android.content.pm.PackageManager;
import android.nfc.NdefMessage;
import android.nfc.NdefRecord;
import android.nfc.NfcAdapter;
import android.nfc.NfcEvent;
import android.os.Build;
import android.os.Bundle;
import android.os.Parcelable;
import android.support.annotation.NonNull;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
```

```
import java.nio.charset.Charset;

public class MainActivity extends AppCompatActivity implements NfcAdapter.OnNdefPushCompleteCallback,
       NfcAdapter.CreateNdefMessageCallback{

   private DrawerLayout drawer_layout;
   private ActionBarDrawerToggle toggle;
   private NfcAdapter mNfcAdapter;
   private Constants c = new Constants();
   private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 42069;

   // disable the back button
   @Override
   public void onBackPressed() {}


   @Override
   protected void onCreate(final Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       // visit log is the first page upon launch
       setTitle("Authentication log");
       FragmentTransaction init = getSupportFragmentManager().beginTransaction();
       init.add(R.id.fragment_container, new VisitLogFragment());
       init.commit();

       // toggle open and close the action bar
       drawer_layout = findViewById(R.id.drawer_layout);
       toggle = new ActionBarDrawerToggle(this, drawer_layout, R.string.open, R.string.close);

       drawer_layout.addDrawerListener(toggle);
       toggle.syncState();

       getSupportActionBar().setDisplayHomeAsUpEnabled(true);

       // display the username on the navigaion drawer
       NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
       View headerView = navigationView.getHeaderView(0);
       TextView username_field = (TextView) headerView.findViewById(R.id.name);
       username_field.setText(c.userName);

       TextView account_type = (TextView) headerView.findViewById(R.id.account_label);
       account_type.setText("home owner");

       // format the navigation drawer
       navigationView.setNavigationItemSelectedListener(
               new NavigationView.OnNavigationItemSelectedListener() {
                   @Override
                   public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                       // set item as selected to persist highlight
                       item.setChecked(true);
                       // close drawer when item is tapped
                       drawer_layout.closeDrawers();
```

```
                Fragment newFragment;
                FragmentManager manager = getSupportFragmentManager();
                FragmentTransaction transaction = manager.beginTransaction();

                // new fragment to be loaded depends on the id of item
                switch (item.getItemId()) {
                    case R.id.nav_visitlog:
                        setTitle("Authentication log");
                        newFragment = new VisitLogFragment();
                        break;
                    case R.id.nav_locks:
                        setTitle("Locks");
                        newFragment = new LocksFragment();
                        break;
                    case R.id.nav_nfc:
                        setTitle("NFC");
                        newFragment = new NFCfragment();
                        break;
                    case R.id.nav_logout:
                        Intent logout = new Intent(getApplicationContext(), LoginActivity.class);
                        startActivity(logout);
                    default:
                        setTitle("Visit log");
                        newFragment = new VisitLogFragment();
                        break;

                }

                transaction.replace(R.id.fragment_container, newFragment);
                transaction.addToBackStack(null);
                transaction.commit();

                item.setChecked(false);

                return true;
            }
        }
);

String channel_name = "main channel";

//Check if NFC is available on device
mNfcAdapter = NfcAdapter.getDefaultAdapter(this);
if(mNfcAdapter != null) {
    Toast.makeText(this, "NFC connected",
            Toast.LENGTH_SHORT).show();

    //This will refer back to createNdefMessage for what it will send
    mNfcAdapter.setNdefPushMessageCallback(this, this);

    //This will be called if the message is sent successfully
    mNfcAdapter.setOnNdefPushCompleteCallback(this, this);
}
else {
    Toast.makeText(this, "NFC not available on this device",
            Toast.LENGTH_SHORT).show();
```

```
        }
    }

    @Override
    public void onNdefPushComplete(NfcEvent event) {
        //This is called when the system detects that our NdefMessage was
        //Successfully sent.
        c.nfc_tag = c.token;
        Toast.makeText(this, "Unlocked",
                Toast.LENGTH_SHORT).show();
    }

    @Override
    public NdefMessage createNdefMessage(NfcEvent event) {
        //This will be called when another NFC capable device is detected.
        //We'll write the createRecords() method in just a moment
        NdefRecord[] recordsToAttach = createRecords(c.nfc_tag);

        //When creating an NdefMessage we need to provide an NdefRecord[]
        return new NdefMessage(recordsToAttach);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (toggle.onOptionsItemSelected(item)) return true;

        return super.onOptionsItemSelected(item);
    }

    public NdefRecord[] createRecords(String message) {

        NdefRecord[] records = new NdefRecord[2  ];
        //To Create Messages Manually if API is less than
        if (Build.VERSION.SDK_INT < Build.VERSION_CODES.JELLY_BEAN) {

            byte[] payload = message.getBytes(Charset.forName("UTF-8"));
            NdefRecord record = new NdefRecord(
                    NdefRecord.TNF_WELL_KNOWN,      //Our 3-bit Type name format
                    NdefRecord.RTD_TEXT,            //Description of our payload
                    new byte[0],                    //The optional id for our Record
                    payload);                       //Our payload for the Record
            records[0] = record;
        }
        //Api is high enough that we can use createMime, which is preferred.
        else {
            byte[] payload = message.getBytes(Charset.forName("UTF-8"));
            NdefRecord record = NdefRecord.createMime("text/plain",payload);
            records[0] = record;
        }
        records[1] = NdefRecord.createApplicationRecord(getPackageName());
        return records;
    }

    private void handleNfcIntent(Intent NfcIntent) {
        if (NfcAdapter.ACTION_NDEF_DISCOVERED.equals(NfcIntent.getAction())) {
            Parcelable[] receivedArray =
```

```
                    NfcIntent.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
            if(receivedArray != null) {
                Toast.makeText(this, "Received Messages", Toast.LENGTH_LONG).show();
            }
            else {
                Toast.makeText(this, "Received Blank Parcel", Toast.LENGTH_LONG).show();
            }
        }
    }

    @Override
    public void onNewIntent(Intent intent) {
        handleNfcIntent(intent);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode,String permissions[], int[] grantResults) {
        switch (requestCode) {
            case MY_PERMISSIONS_REQUEST_SEND_SMS: {
                if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                } else {
                    Toast.makeText(getApplicationContext(),
                            "SMS faild, please try again.", Toast.LENGTH_LONG).show();
                    return;
                }
            }
        }

    }
}
```

NFCfragment.java

```
package com.g19p2.g19p2app;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.DialogFragment;
import android.support.v4.content.ContextCompat;
import android.telephony.SmsManager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class NFCfragment extends DialogFragment {

    Constants c = new Constants();
    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 42069;

    public NFCfragment() {
```

```
        // Required empty public constructor
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        askPermission();

        Button share_btn = (Button) getView().findViewById(R.id.share_btn);
        share_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText phone_number = (EditText) getView().findViewById(R.id.phone_number);
                shareToken(phone_number.getText().toString());
            }
        });

        Button set_token = (Button) getView().findViewById(R.id.set_token);
        set_token.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText token_field = (EditText) getView().findViewById(R.id.token_field);
                setToken(token_field.getText().toString());
            }
        });
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_write,container,false);
        return view;
    }

    public void shareToken(String phone_number) {
        if(phone_number.equals(""))
            Toast.makeText(getActivity().getApplicationContext(), "please enter a number",
Toast.LENGTH_SHORT).show();
        else {
            Toast.makeText(getActivity().getApplicationContext(), "sending to: " + phone_number,
Toast.LENGTH_SHORT).show();
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phone_number, null, "Next message will contain the key to use for
this lock. Sent by user: " + c.userName, null,null);
            smsManager.sendTextMessage(phone_number, null, c.token, null,null);
            Toast.makeText(getActivity().getApplicationContext(), "sent!", Toast.LENGTH_SHORT).show();
        }
    }

    public void setToken(String token) {
        if(!token.equals("")) {
            Toast.makeText(getActivity().getApplicationContext(), token, Toast.LENGTH_SHORT).show();
            c.nfc_tag = token;
        }
        else {
            Toast.makeText(getActivity().getApplicationContext(), "reset", Toast.LENGTH_SHORT).show();
```

```
            c.nfc_tag = c.token;
        }
    }

    private void askPermission() {
        if (ContextCompat.checkSelfPermission(this.getActivity(), Manifest.permission.SEND_SMS) !=
PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(this.getActivity(),
Manifest.permission.SEND_SMS)) {
            } else {
                ActivityCompat.requestPermissions(this.getActivity(),
                        new String[]{Manifest.permission.SEND_SMS},
                        MY_PERMISSIONS_REQUEST_SEND_SMS);
            }
        }
    }
}
```

StreamFragment.java

```
package com.g19p2.g19p2app;

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;


/**
* A simple {@link Fragment} subclass.
* Displays a livestream of a view just outside the door, and
* allows for remote lock and unlock of the door.
*/
public class StreamFragment extends Fragment {
    private static final String DUMMY_URL = "http://techslides.com/demos/sample-videos/small.mp4";
    private static final String REAL_URL = "https://acerate-scorpion-5728.dataplicity.io/?action=stream";

    private String streamURL;
    private Constants c;

    public StreamFragment() {
        // Required empty public constructor
    }

    @Override
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        getActivity().setTitle("Lock " + c.lock_id + "'s live stream");
        WebView webview = (WebView) getView().findViewById(R.id.webview);
        // get URL from the server
        streamURL = getURL();

        // play the video linked in the URL
        webview.loadUrl(streamURL);
```

```
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        c = new Constants();
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_stream, container, false);
    }

    private String getURL() {
        c.client = new G19P2Client();
        String reply;
        try {
            reply = c.client.execute(G19P2Client.REQUEST_GET_STREAM_URL, c.lock_id).get();
        }catch(Exception e){
            reply = "THREAD ERROR";
        }
        return reply;
    }
}
```

Visit.java

```
package com.g19p2.g19p2app;

/**
* Created by robin on 2018-03-18.
* Class describing the characteristics of a visit.
*/

public class Visit {

    private String date;
    private String time;
    private boolean authorized;

    public Visit(String date, String time, boolean authorized) {
        this.date = date;
        this.time = time;
        this.authorized = authorized;
    }

    public String getDate(){
        return date;
    }

    public String getTime(){
        return time;
    }

    public boolean getAuthorized() {
        return authorized;
    }
}
```

VisitCard.java

```
package com.g19p2.g19p2app;

import android.content.Context;
import android.support.v7.widget.CardView;
import android.view.View;
import android.widget.TextView;

/**
* Created by robin on 2018-03-18.
* Collection of all elements of a card that displays
* the information of a visit.
*/

public class VisitCard {
   CardView card;
   TextView date_label, time_label, status_label,
           date, time, status;
   Context context;

   public VisitCard(CardView card, TextView date_label, TextView time_label,
                    TextView status_label, TextView date, TextView time, TextView status,
                   Context context) {
       this.card = card;
       this.date_label = date_label;
       this.time_label = time_label;
       this.status_label = status_label;
       this.date = date;
       this.time = time;
       this.status = status;
       this.context = context;
   }

   /**
    * Formats the text of the card elements to match that of the given visit.
    * precondition: the card is initially invisible
    * @param visit - Visit whose information is to be displayed on the card
    */
   public void formatCard(Visit visit) {
       card.setVisibility(View.VISIBLE);

       date_label.setVisibility(View.VISIBLE);
       date.setVisibility(View.VISIBLE);
       date.setText(visit.getDate());

       time_label.setVisibility(View.VISIBLE);
       time.setVisibility(View.VISIBLE);
       time.setText(visit.getTime());

       status_label.setVisibility(View.VISIBLE);
       status.setVisibility(View.VISIBLE);
       if(visit.getAuthorized()) {
           status.setText("AUTHORIZED");
           status.setTextColor(context.getResources().getColor(R.color.colorAuthorized));
       }
       else {
           status.setText("DENIED");
```

```
            status.setTextColor(context.getResources().getColor(R.color.colorDenied));
        }
    }

    /**
     * Makes all the contents of this card gone
     */
    public void setGone() {
        card.setVisibility(View.GONE);
        date_label.setVisibility(View.GONE);
        time_label.setVisibility(View.GONE);
        status_label.setVisibility(View.GONE);
        date.setVisibility(View.GONE);
        time.setVisibility(View.GONE);
        status.setVisibility(View.GONE);
    }
}
```

VisitLogFragment.java

```
package com.g19p2.g19p2app;


import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v7.widget.CardView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Collections;


/**
* A simple {@link Fragment} subclass.
* Displays the user's most recent visits on cardviews.
*/
public class VisitLogFragment extends Fragment {

    ArrayList<VisitCard> VisitCards;
    ArrayList<Visit> Visits;
    Constants c;

    public VisitLogFragment() {
        VisitCards = new ArrayList<VisitCard>();
        Visits = new ArrayList<Visit>();
    }

    @Override
```

```
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {
        // initialize instance of constants
        c = new Constants();

        // load the cards and visits into their specified ArrayLists
        initializeCards();  getVisits();

        // initially set all cards to be gone
        for(VisitCard vc : VisitCards)
            vc.setGone();

        // format a visit card for each visit, or until number of cards
        //is exhausted
        for(int i = 0; i < Math.min(Visits.size(), VisitCards.size()); i++)
            VisitCards.get(i).formatCard(Visits.get(i));
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_visit_log, container, false);
    }


    /**
     * Generates the visits whose information is to be displayed on the visit cards,
     * and adds them to an easily accessible ArrayList.
     * postcondition: most recent visits are put at lower indices of Visits
     *
     * Requests a JSON Array from the server containing recent visits, and adds each
     * JSON Object as a Visit object to the Visits ArrayList.
     * postcondition: most recent visits are put at lower indices of Visits
     */
    private void getVisits() {
         // Expecting JSON string formatted like:
         // {"visits":[ {"date":"2018-03-22" , "time":"23:44" , "authorized":true , "method":1} ,
{"date":... } ] }

        c.client = new G19P2Client();
//        String reply = "[{\"hid\": \"4\", \"username\": \"newtest\", \"time\": \"12:28AM on April 02,
2018\", \"status\": \"denied\"}, {\"hid\": \"5\", \"username\": \"newtest\", \"time\": \"12:28AM on
April 02, 2018\", \"status\": \"authorized\"}, {\"hid\": \"6\", \"username\": \"newtest\", \"time\":
\"12:31AM on April 02, 2018\", \"status\": \"authorized\"}]";

        String reply;
        // attempt to get JSON string from request
        try {
            reply = c.client.execute(G19P2Client.REQUEST_GET_VISITS).get();
        } catch(Exception e) {
            Toast.makeText(getActivity().getApplicationContext(), "failed to get reply",
Toast.LENGTH_SHORT).show();
            return;
        }

        try {
```

```
            JSONArray jsonArray = new JSONArray(reply);
            // takes the information out of each JSON object to construct
            // a Visit object to add to Visits
            for(int i = 0; i < jsonArray.length(); i++) {
                JSONObject visit = jsonArray.getJSONObject(i);

                String[] time = visit.getString("time").split(" on ");

                String status = visit.getString("status");
                boolean status_bool = status.equals("denied") ? false : true;

                Visits.add(new Visit(time[1], time[0], status_bool));
            }
        } catch(JSONException e) {
            Toast.makeText(getActivity().getApplicationContext(), "reply was not JSON string",
Toast.LENGTH_SHORT).show();
            return;
        }

        Collections.reverse(Visits);
    }

    /**
     * Creates objects out of each CardView and its contents, and adds them to
     * an easily accessible ArrayList.
     */
    private void initializeCards() {
        CardView card0 = (CardView) getView().findViewById(R.id.card0);
        TextView time_label_0 = (TextView) getView().findViewById(R.id.time_label_0);
        TextView time_0 = (TextView) getView().findViewById(R.id.time_0);
        TextView date_label_0 = (TextView) getView().findViewById(R.id.date_label_0);
        TextView date_0 = (TextView) getView().findViewById(R.id.date_0);
        TextView status_label_0 = (TextView) getView().findViewById(R.id.status_label_0);
        TextView status_0 = (TextView) getView().findViewById(R.id.status_0);

        VisitCard visitCard0 = new VisitCard(card0, date_label_0, time_label_0, status_label_0,
                date_0, time_0, status_0, getActivity().getApplicationContext());
        VisitCards.add(visitCard0);

        CardView card1 = (CardView) getView().findViewById(R.id.card1);
        TextView time_label_1 = (TextView) getView().findViewById(R.id.time_label_1);
        TextView time_1 = (TextView) getView().findViewById(R.id.time_1);
        TextView date_label_1 = (TextView) getView().findViewById(R.id.date_label_1);
        TextView date_1 = (TextView) getView().findViewById(R.id.date_1);
        TextView status_label_1 = (TextView) getView().findViewById(R.id.status_label_1);
        TextView status_1 = (TextView) getView().findViewById(R.id.status_1);

        VisitCard visitCard1 = new VisitCard(card1, date_label_1, time_label_1, status_label_1,
                date_1, time_1, status_1, getActivity().getApplicationContext());
        VisitCards.add(visitCard1);

        CardView card2 = (CardView) getView().findViewById(R.id.card2);
        TextView time_label_2 = (TextView) getView().findViewById(R.id.time_label_2);
        TextView time_2 = (TextView) getView().findViewById(R.id.time_2);
        TextView date_label_2 = (TextView) getView().findViewById(R.id.date_label_2);
        TextView date_2 = (TextView) getView().findViewById(R.id.date_2);
```

```
TextView status_label_2 = (TextView) getView().findViewById(R.id.status_label_2);
TextView status_2 = (TextView) getView().findViewById(R.id.status_2);

VisitCard visitCard2 = new VisitCard(card2, date_label_2, time_label_2, status_label_2,
        date_2, time_2, status_2, getActivity().getApplicationContext());
VisitCards.add(visitCard2);

CardView card3 = (CardView) getView().findViewById(R.id.card3);
TextView time_label_3 = (TextView) getView().findViewById(R.id.time_label_3);
TextView time_3 = (TextView) getView().findViewById(R.id.time_3);
TextView date_label_3 = (TextView) getView().findViewById(R.id.date_label_3);
TextView date_3 = (TextView) getView().findViewById(R.id.date_3);
TextView status_label_3 = (TextView) getView().findViewById(R.id.status_label_3);
TextView status_3 = (TextView) getView().findViewById(R.id.status_3);

VisitCard visitCard3 = new VisitCard(card3, date_label_3, time_label_3, status_label_3,
         date_3, time_3, status_3, getActivity().getApplicationContext());
VisitCards.add(visitCard3);

CardView card4 = (CardView) getView().findViewById(R.id.card4);
TextView time_label_4 = (TextView) getView().findViewById(R.id.time_label_4);
TextView time_4 = (TextView) getView().findViewById(R.id.time_4);
TextView date_label_4 = (TextView) getView().findViewById(R.id.date_label_4);
TextView date_4 = (TextView) getView().findViewById(R.id.date_4);
TextView status_label_4 = (TextView) getView().findViewById(R.id.status_label_4);
TextView status_4 = (TextView) getView().findViewById(R.id.status_4);

VisitCard visitCard4 = new VisitCard(card4, date_label_4, time_label_4, status_label_4,
        date_4, time_4, status_4, getActivity().getApplicationContext());
VisitCards.add(visitCard4);

CardView card5 = (CardView) getView().findViewById(R.id.card5);
TextView time_label_5 = (TextView) getView().findViewById(R.id.time_label_5);
TextView time_5 = (TextView) getView().findViewById(R.id.time_5);
TextView date_label_5 = (TextView) getView().findViewById(R.id.date_label_5);
TextView date_5 = (TextView) getView().findViewById(R.id.date_5);
TextView status_label_5 = (TextView) getView().findViewById(R.id.status_label_5);
TextView status_5 = (TextView) getView().findViewById(R.id.status_5);

VisitCard visitCard5 = new VisitCard(card5, date_label_5, time_label_5, status_label_5,
        date_5, time_5, status_5, getActivity().getApplicationContext());
VisitCards.add(visitCard5);

CardView card6 = (CardView) getView().findViewById(R.id.card6);
TextView time_label_6 = (TextView) getView().findViewById(R.id.time_label_6);
TextView time_6 = (TextView) getView().findViewById(R.id.time_6);
TextView date_label_6 = (TextView) getView().findViewById(R.id.date_label_6);
TextView date_6 = (TextView) getView().findViewById(R.id.date_6);
TextView status_label_6 = (TextView) getView().findViewById(R.id.status_label_6);
TextView status_6 = (TextView) getView().findViewById(R.id.status_6);

VisitCard visitCard6 = new VisitCard(card6, date_label_6, time_label_6, status_label_6,
        date_6, time_6, status_6, getActivity().getApplicationContext());
VisitCards.add(visitCard6);

CardView card7 = (CardView) getView().findViewById(R.id.card7);
```

```
        TextView time_label_7 = (TextView) getView().findViewById(R.id.time_label_7);
        TextView time_7 = (TextView) getView().findViewById(R.id.time_7);
        TextView date_label_7 = (TextView) getView().findViewById(R.id.date_label_7);
        TextView date_7 = (TextView) getView().findViewById(R.id.date_7);
        TextView status_label_7 = (TextView) getView().findViewById(R.id.status_label_7);
        TextView status_7 = (TextView) getView().findViewById(R.id.status_7);

        VisitCard visitCard7 = new VisitCard(card7, date_label_7, time_label_7, status_label_7,
                date_7, time_7, status_7, getActivity().getApplicationContext());
        VisitCards.add(visitCard7);

        CardView card8 = (CardView) getView().findViewById(R.id.card8);
        TextView time_label_8 = (TextView) getView().findViewById(R.id.time_label_8);
        TextView time_8 = (TextView) getView().findViewById(R.id.time_8);
        TextView date_label_8 = (TextView) getView().findViewById(R.id.date_label_8);
        TextView date_8 = (TextView) getView().findViewById(R.id.date_8);
        TextView status_label_8 = (TextView) getView().findViewById(R.id.status_label_8);
        TextView status_8 = (TextView) getView().findViewById(R.id.status_8);

        VisitCard visitCard8 = new VisitCard(card8, date_label_8, time_label_8, status_label_8,
                date_8, time_8, status_8, getActivity().getApplicationContext());
        VisitCards.add(visitCard8);

        CardView card9 = (CardView) getView().findViewById(R.id.card9);
        TextView time_label_9 = (TextView) getView().findViewById(R.id.time_label_9);
        TextView time_9 = (TextView) getView().findViewById(R.id.time_9);
        TextView date_label_9 = (TextView) getView().findViewById(R.id.date_label_9);
        TextView date_9 = (TextView) getView().findViewById(R.id.date_9);
        TextView status_label_9 = (TextView) getView().findViewById(R.id.status_label_9);
        TextView status_9 = (TextView) getView().findViewById(R.id.status_9);

        VisitCard visitCard9 = new VisitCard(card9, date_label_9, time_label_9, status_label_9,
                date_9, time_9, status_9, getActivity().getApplicationContext());
        VisitCards.add(visitCard9);
    }


}
```

Activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.g19p2.g19p2app.LoginActivity">

    <EditText
        android:id="@+id/password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="184dp"
        android:layout_marginEnd="8dp"
```

```
        android:layout_marginStart="8dp"
        android:ems="10"
        android:hint="password"
        android:inputType="textPassword"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.503"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:hint="username"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toTopOf="@+id/password"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.503"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0" />

    <Button
        android:id="@+id/login_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:fontFamily="sans-serif-condensed"
        android:text="Login"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/password" />

    <TextView
        android:id="@+id/app_title"
        android:layout_width="wrap_content"
        android:layout_height="36dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="48dp"
        android:fontFamily="serif-monospace"
        android:text="Raspberry P-eye"
        android:textAlignment="center"
        android:textSize="30sp"
        app:layout_constraintBottom_toTopOf="@+id/g19"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0" />
```

```
    <TextView
        android:id="@+id/g19"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="112dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:fontFamily="serif-monospace"
        android:text="by G19"
        android:textSize="18sp"
        app:layout_constraintBottom_toTopOf="@+id/email"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/sign_up_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:fontFamily="sans-serif-condensed"
        android:text="SIGN UP"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/login_btn"
        app:layout_constraintVertical_bias="0.0" />

</android.support.constraint.ConstraintLayout>

Activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<!-- Use DrawerLayout as root container for activity -->
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <!-- Layout to contain contents of main body of screen (drawer will slide over this) -->
    <LinearLayout
        android:id="@+id/fragment_container"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <!-- Container for contents of drawer - use NavigationView to make configuration easier -->
    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
```

```
        android:fitsSystemWindows="true"
        app:menu="@menu/navigation_menu"
        app:headerLayout="@layout/navigation_header"/>

</android.support.v4.widget.DrawerLayout>

Dialog_addlock.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent" android:padding="16dp">

   <EditText
       android:id="@+id/edit_lock_url"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:layout_alignParentLeft="true"
       android:layout_alignParentStart="true"
       android:layout_alignParentTop="true"
       android:ems="10"
       android:hint="Lock URL"
       android:inputType="textPersonName" />
</RelativeLayout>

Fragment_locks.xml
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   tools:context="com.g19p2.g19p2app.LocksFragment">

   <android.support.v4.widget.NestedScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
       xmlns:tools="http://schemas.android.com/tools"
       xmlns:card_view="http://schemas.android.com/apk/res-auto"
       android:id="@+id/scrollview"
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       tools:context="com.g19p2.g19p2app.VisitLogFragment">

       <LinearLayout
           android:orientation="vertical"
           android:layout_width="match_parent"
           android:layout_height="wrap_content">

           <android.support.v7.widget.CardView
               android:id="@+id/lock0"
               android:layout_width="match_parent"
               android:layout_height="75dp"
               android:layout_margin="5dp"
               card_view:cardElevation="0dp"
               card_view:cardBackgroundColor="@android:color/transparent"
               card_view:cardCornerRadius="5dp">

               <RelativeLayout
                   android:layout_width="match_parent"
```

```
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_0"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_0"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_0"
                android:layout_alignBottom="@+id/lock_id_label_0"
                android:layout_toEndOf="@+id/lock_id_label_0"
                android:layout_toRightOf="@+id/lock_id_label_0"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_0"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/lock_id_0"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_marginEnd="13dp"
                android:layout_marginRight="13dp"
                android:src="@mipmap/ic_videocam_black_24dp" />

            <View
                android:id="@+id/divider_0"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:background="@android:color/darker_gray" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/lock1"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardElevation="0dp"
        card_view:cardBackgroundColor="@android:color/transparent"
        card_view:cardCornerRadius="5dp">
```

```
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_1"
                android:layout_alignBottom="@+id/lock_id_label_1"
                android:layout_toEndOf="@+id/lock_id_label_1"
                android:layout_toRightOf="@+id/lock_id_label_1"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/lock_id_1"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_marginEnd="13dp"
                android:layout_marginRight="13dp"
                android:src="@mipmap/ic_videocam_black_24dp" />

            <View
                android:id="@+id/divider_1"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:background="@android:color/darker_gray" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/lock2"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
```

```
        card_view:cardElevation="0dp"
        card_view:cardBackgroundColor="@android:color/transparent"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_2"
                android:layout_alignBottom="@+id/lock_id_label_2"
                android:layout_toEndOf="@+id/lock_id_label_2"
                android:layout_toRightOf="@+id/lock_id_label_2"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/lock_id_2"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_marginEnd="13dp"
                android:layout_marginRight="13dp"
                android:src="@mipmap/ic_videocam_black_24dp" />

            <View
                android:id="@+id/divider_2"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:background="@android:color/darker_gray" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/lock3"
```

```
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardElevation="0dp"
        card_view:cardBackgroundColor="@android:color/transparent"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_3"
                android:layout_alignBottom="@+id/lock_id_label_3"
                android:layout_toEndOf="@+id/lock_id_label_3"
                android:layout_toRightOf="@+id/lock_id_label_3"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/lock_id_3"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_marginEnd="13dp"
                android:layout_marginRight="13dp"
                android:src="@mipmap/ic_videocam_black_24dp" />

            <View
                android:id="@+id/divider_3"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:background="@android:color/darker_gray" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>
```

```
<android.support.v7.widget.CardView
    android:id="@+id/lock4"
    android:layout_width="match_parent"
    android:layout_height="75dp"
    android:layout_margin="5dp"
    card_view:cardElevation="0dp"
    card_view:cardBackgroundColor="@android:color/transparent"
    card_view:cardCornerRadius="5dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/lock_id_label_4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_centerVertical="true"
            android:layout_marginLeft="11dp"
            android:layout_marginStart="11dp"
            android:padding="10dp"
            android:text="Lock ID:"
            android:textSize="18sp" />

        <TextView
            android:id="@+id/lock_id_4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/lock_id_label_4"
            android:layout_alignBottom="@+id/lock_id_label_4"
            android:layout_toEndOf="@+id/lock_id_label_4"
            android:layout_toRightOf="@+id/lock_id_label_4"
            android:padding="10dp"
            android:text="FFFFFF"
            android:textSize="18sp" />

        <ImageButton
            android:id="@+id/unlock_btn_4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBottom="@+id/lock_id_4"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_marginEnd="13dp"
            android:layout_marginRight="13dp"
            android:src="@mipmap/ic_videocam_black_24dp" />

        <View
            android:id="@+id/divider_4"
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@android:color/darker_gray" />
```

```
        </RelativeLayout>

    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/lock5"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardElevation="0dp"
        card_view:cardBackgroundColor="@android:color/transparent"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_5"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_5"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_5"
                android:layout_alignBottom="@+id/lock_id_label_5"
                android:layout_toEndOf="@+id/lock_id_label_5"
                android:layout_toRightOf="@+id/lock_id_label_5"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_5"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/lock_id_5"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_marginEnd="13dp"
                android:layout_marginRight="13dp"
                android:src="@mipmap/ic_videocam_black_24dp" />

            <View
                android:id="@+id/divider_5"
                android:layout_width="match_parent"
```

```
                android:layout_height="1dp"
                android:background="@android:color/darker_gray" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/lock6"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardElevation="0dp"
        card_view:cardBackgroundColor="@android:color/transparent"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_6"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_6"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_6"
                android:layout_alignBottom="@+id/lock_id_label_6"
                android:layout_toEndOf="@+id/lock_id_label_6"
                android:layout_toRightOf="@+id/lock_id_label_6"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_6"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/lock_id_6"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_marginEnd="13dp"
                android:layout_marginRight="13dp"
                android:src="@mipmap/ic_videocam_black_24dp" />
```

```
            <View
                android:id="@+id/divider_6"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:background="@android:color/darker_gray" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/lock7"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardElevation="0dp"
        card_view:cardBackgroundColor="@android:color/transparent"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_7"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_7"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_7"
                android:layout_alignBottom="@+id/lock_id_label_7"
                android:layout_toEndOf="@+id/lock_id_label_7"
                android:layout_toRightOf="@+id/lock_id_label_7"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_7"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/lock_id_7"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_marginEnd="13dp"
```

```
                android:layout_marginRight="13dp"
                android:src="@mipmap/ic_videocam_black_24dp" />

            <View
                android:id="@+id/divider_7"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:background="@android:color/darker_gray" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/lock8"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardElevation="0dp"
        card_view:cardBackgroundColor="@android:color/transparent"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_8"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_8"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_8"
                android:layout_alignBottom="@+id/lock_id_label_8"
                android:layout_toEndOf="@+id/lock_id_label_8"
                android:layout_toRightOf="@+id/lock_id_label_8"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_8"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBottom="@+id/lock_id_8"
```

```
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_marginEnd="13dp"
                android:layout_marginRight="13dp"
                android:src="@mipmap/ic_videocam_black_24dp" />

            <View
                android:id="@+id/divider_8"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:background="@android:color/darker_gray" />

        </RelativeLayout>

    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/lock9"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardElevation="0dp"
        card_view:cardBackgroundColor="@android:color/transparent"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/lock_id_label_9"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_centerVertical="true"
                android:layout_marginLeft="11dp"
                android:layout_marginStart="11dp"
                android:padding="10dp"
                android:text="Lock ID:"
                android:textSize="18sp" />

            <TextView
                android:id="@+id/lock_id_9"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/lock_id_label_9"
                android:layout_alignBottom="@+id/lock_id_label_9"
                android:layout_toEndOf="@+id/lock_id_label_9"
                android:layout_toRightOf="@+id/lock_id_label_9"
                android:padding="10dp"
                android:text="FFFFFF"
                android:textSize="18sp" />

            <ImageButton
                android:id="@+id/unlock_btn_9"
```

```
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_alignBottom="@+id/lock_id_9"
                    android:layout_alignParentEnd="true"
                    android:layout_alignParentRight="true"
                    android:layout_marginEnd="13dp"
                    android:layout_marginRight="13dp"
                    android:src="@mipmap/ic_videocam_black_24dp" />

                <View
                    android:id="@+id/divider_9"
                    android:layout_width="match_parent"
                    android:layout_height="1dp"
                    android:background="@android:color/darker_gray" />

            </RelativeLayout>

        </android.support.v7.widget.CardView>

    </LinearLayout>

</android.support.v4.widget.NestedScrollView>

<android.support.design.widget.FloatingActionButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/add_lock_btn"
    android:layout_gravity="end|bottom"
    android:src="@mipmap/ic_add_black_24dp"
    android:layout_margin="16dp"/>


</FrameLayout>

Fragment_stream.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent" >

   <WebView
       android:id="@+id/webview"
       android:layout_width="352dp"
       android:layout_height="400dp"
       android:layout_marginEnd="8dp"
       android:layout_marginStart="8dp"
       android:layout_marginTop="8dp"
       app:layout_constraintEnd_toEndOf="parent"
       app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>

Fragment_visit_log.xml
```

```
<android.support.v4.widget.NestedScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:id="@+id/scrollview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.g19p2.g19p2app.VisitLogFragment">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <android.support.v7.widget.CardView
            android:id="@+id/card0"
            android:layout_width="match_parent"
            android:layout_height="75dp"
            android:layout_margin="5dp"
            card_view:cardBackgroundColor="@color/colorPrimaryDark"
            card_view:cardCornerRadius="5dp">

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent">

                <TextView
                    android:id="@+id/time_label_0"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_alignParentLeft="true"
                    android:layout_alignParentStart="true"
                    android:layout_alignParentTop="true"
                    android:padding="7dp"
                    android:text="Time:" />

                <TextView
                    android:id="@+id/time_0"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_alignEnd="@+id/date_0"
                    android:layout_alignParentTop="true"
                    android:layout_alignRight="@+id/date_0"
                    android:padding="7dp"
                    android:text="23:59" />

                <TextView
                    android:id="@+id/date_label_0"
                    android:layout_width="53dp"
                    android:layout_height="match_parent"
                    android:layout_alignParentLeft="true"
                    android:layout_alignParentStart="true"
                    android:layout_below="@+id/time_label_0"
                    android:padding="7dp"
                    android:text="Date:" />

                <TextView
```

```
            android:id="@+id/date_0"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/date_label_0"
            android:layout_alignBottom="@+id/date_label_0"
            android:layout_toEndOf="@+id/date_label_0"
            android:layout_toRightOf="@+id/date_label_0"
            android:padding="7dp"
            android:text="2018-05-15" />

        <TextView
            android:id="@+id/status_label_0"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/status_0"
            android:layout_alignBottom="@+id/status_0"
            android:layout_toLeftOf="@+id/status_0"
            android:layout_toStartOf="@+id/status_0"
            android:padding="7dp"
            android:text="Status:" />

        <TextView
            android:id="@+id/status_0"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:layout_marginEnd="21dp"
            android:layout_marginRight="21dp"
            android:padding="7dp"
            android:text="AUTHORIZED"
            android:textAlignment="textEnd" />


    </RelativeLayout>


</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/card1"
    android:layout_width="match_parent"
    android:layout_height="75dp"
    android:layout_margin="5dp"
    card_view:cardBackgroundColor="@color/colorPrimaryDark"
    card_view:cardCornerRadius="5dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/time_label_1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:padding="7dp"
        android:text="Time:" />

    <TextView
        android:id="@+id/time_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/date_1"
        android:layout_alignParentTop="true"
        android:layout_alignRight="@+id/date_1"
        android:padding="7dp"
        android:text="23:59" />

    <TextView
        android:id="@+id/date_label_1"
        android:layout_width="53dp"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/time_label_1"
        android:padding="7dp"
        android:text="Date:" />

    <TextView
        android:id="@+id/date_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/date_label_1"
        android:layout_alignBottom="@+id/date_label_1"
        android:layout_toEndOf="@+id/date_label_1"
        android:layout_toRightOf="@+id/date_label_1"
        android:padding="7dp"
        android:text="2018-05-15" />

    <TextView
        android:id="@+id/status_label_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/status_1"
        android:layout_alignBottom="@+id/status_1"
        android:layout_toLeftOf="@+id/status_1"
        android:layout_toStartOf="@+id/status_1"
        android:padding="7dp"
        android:text="Status:" />

    <TextView
        android:id="@+id/status_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="19dp"
```

```
            android:layout_marginRight="19dp"
            android:padding="7dp"
            android:text="AUTHORIZED" />


    </RelativeLayout>


</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/card2"
    android:layout_width="match_parent"
    android:layout_height="75dp"
    android:layout_margin="5dp"
    card_view:cardBackgroundColor="@color/colorPrimaryDark"
    card_view:cardCornerRadius="5dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/time_label_2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:padding="7dp"
            android:text="Time:" />

        <TextView
            android:id="@+id/time_2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignEnd="@+id/date_2"
            android:layout_alignParentTop="true"
            android:layout_alignRight="@+id/date_2"
            android:padding="7dp"
            android:text="23:59" />

        <TextView
            android:id="@+id/date_label_2"
            android:layout_width="53dp"
            android:layout_height="match_parent"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_below="@+id/time_label_2"
            android:padding="7dp"
            android:text="Date:" />

        <TextView
            android:id="@+id/date_2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
            android:layout_alignBaseline="@+id/date_label_2"
            android:layout_alignBottom="@+id/date_label_2"
            android:layout_toEndOf="@+id/date_label_2"
            android:layout_toRightOf="@+id/date_label_2"
            android:padding="7dp"
            android:text="2018-05-15" />

        <TextView
            android:id="@+id/status_label_2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/status_2"
            android:layout_alignBottom="@+id/status_2"
            android:layout_toLeftOf="@+id/status_2"
            android:layout_toStartOf="@+id/status_2"
            android:padding="7dp"
            android:text="Status:" />

        <TextView
            android:id="@+id/status_2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:layout_marginEnd="20dp"
            android:layout_marginRight="20dp"
            android:padding="7dp"
            android:text="AUTHORIZED" />


    </RelativeLayout>


</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/card3"
    android:layout_width="match_parent"
    android:layout_height="75dp"
    android:layout_margin="5dp"
    card_view:cardBackgroundColor="@color/colorPrimaryDark"
    card_view:cardCornerRadius="5dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/time_label_3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:padding="7dp"
```

```
        android:text="Time:" />

    <TextView
        android:id="@+id/time_3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/date_3"
        android:layout_alignParentTop="true"
        android:layout_alignRight="@+id/date_3"
        android:padding="7dp"
        android:text="23:59" />

    <TextView
        android:id="@+id/date_label_3"
        android:layout_width="53dp"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/time_label_3"
        android:padding="7dp"
        android:text="Date:" />

    <TextView
        android:id="@+id/date_3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/date_label_3"
        android:layout_alignBottom="@+id/date_label_3"
        android:layout_toEndOf="@+id/date_label_3"
        android:layout_toRightOf="@+id/date_label_3"
        android:padding="7dp"
        android:text="2018-05-15" />

    <TextView
        android:id="@+id/status_label_3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/status_3"
        android:layout_alignBottom="@+id/status_3"
        android:layout_toLeftOf="@+id/status_3"
        android:layout_toStartOf="@+id/status_3"
        android:padding="7dp"
        android:text="Status:" />

    <TextView
        android:id="@+id/status_3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="20dp"
        android:layout_marginRight="20dp"
        android:padding="7dp"
        android:text="AUTHORIZED" />
```

```
        </RelativeLayout>


    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/card4"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardBackgroundColor="@color/colorPrimaryDark"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/time_label_4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_alignParentTop="true"
                android:padding="7dp"
                android:text="Time:" />

            <TextView
                android:id="@+id/time_4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignEnd="@+id/date_4"
                android:layout_alignParentTop="true"
                android:layout_alignRight="@+id/date_4"
                android:padding="7dp"
                android:text="23:59" />

            <TextView
                android:id="@+id/date_label_4"
                android:layout_width="53dp"
                android:layout_height="match_parent"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_below="@+id/time_label_4"
                android:padding="7dp"
                android:text="Date:" />

            <TextView
                android:id="@+id/date_4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/date_label_4"
                android:layout_alignBottom="@+id/date_label_4"
                android:layout_toEndOf="@+id/date_label_4"
                android:layout_toRightOf="@+id/date_label_4"
```

```
            android:padding="7dp"
            android:text="2018-05-15" />

        <TextView
            android:id="@+id/status_label_4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/status_4"
            android:layout_alignBottom="@+id/status_4"
            android:layout_toLeftOf="@+id/status_4"
            android:layout_toStartOf="@+id/status_4"
            android:padding="7dp"
            android:text="Status:" />

        <TextView
            android:id="@+id/status_4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:layout_marginEnd="20dp"
            android:layout_marginRight="20dp"
            android:padding="7dp"
            android:text="AUTHORIZED" />


    </RelativeLayout>


</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/card5"
    android:layout_width="match_parent"
    android:layout_height="75dp"
    android:layout_margin="5dp"
    card_view:cardBackgroundColor="@color/colorPrimaryDark"
    card_view:cardCornerRadius="5dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/time_label_5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:padding="7dp"
            android:text="Time:" />

        <TextView
            android:id="@+id/time_5"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignEnd="@+id/date_5"
            android:layout_alignParentTop="true"
            android:layout_alignRight="@+id/date_5"
            android:padding="7dp"
            android:text="23:59" />

        <TextView
            android:id="@+id/date_label_5"
            android:layout_width="53dp"
            android:layout_height="match_parent"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_below="@+id/time_label_5"
            android:padding="7dp"
            android:text="Date:" />

        <TextView
            android:id="@+id/date_5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/date_label_5"
            android:layout_alignBottom="@+id/date_label_5"
            android:layout_toEndOf="@+id/date_label_5"
            android:layout_toRightOf="@+id/date_label_5"
            android:padding="7dp"
            android:text="2018-05-15" />

        <TextView
            android:id="@+id/status_label_5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/status_5"
            android:layout_alignBottom="@+id/status_5"
            android:layout_toLeftOf="@+id/status_5"
            android:layout_toStartOf="@+id/status_5"
            android:padding="7dp"
            android:text="Status:" />

        <TextView
            android:id="@+id/status_5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:layout_marginEnd="19dp"
            android:layout_marginRight="19dp"
            android:padding="7dp"
            android:text="AUTHORIZED" />


    </RelativeLayout>
```

```
        </android.support.v7.widget.CardView>

        <android.support.v7.widget.CardView
            android:id="@+id/card6"
            android:layout_width="match_parent"
            android:layout_height="75dp"
            android:layout_margin="5dp"
            card_view:cardBackgroundColor="@color/colorPrimaryDark"
            card_view:cardCornerRadius="5dp">

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent">

                <TextView
                    android:id="@+id/time_label_6"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_alignParentLeft="true"
                    android:layout_alignParentStart="true"
                    android:layout_alignParentTop="true"
                    android:padding="7dp"
                    android:text="Time:" />

                <TextView
                    android:id="@+id/time_6"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_alignEnd="@+id/date_6"
                    android:layout_alignParentTop="true"
                    android:layout_alignRight="@+id/date_6"
                    android:padding="7dp"
                    android:text="23:59" />

                <TextView
                    android:id="@+id/date_label_6"
                    android:layout_width="53dp"
                    android:layout_height="match_parent"
                    android:layout_alignParentLeft="true"
                    android:layout_alignParentStart="true"
                    android:layout_below="@+id/time_label_6"
                    android:padding="7dp"
                    android:text="Date:" />

                <TextView
                    android:id="@+id/date_6"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_alignBaseline="@+id/date_label_6"
                    android:layout_alignBottom="@+id/date_label_6"
                    android:layout_toEndOf="@+id/date_label_6"
                    android:layout_toRightOf="@+id/date_label_6"
                    android:padding="7dp"
                    android:text="2018-05-15" />

                <TextView
```

```
                android:id="@+id/status_label_6"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/status_6"
                android:layout_alignBottom="@+id/status_6"
                android:layout_toLeftOf="@+id/status_6"
                android:layout_toStartOf="@+id/status_6"
                android:padding="7dp"
                android:text="Status:" />

            <TextView
                android:id="@+id/status_6"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_centerVertical="true"
                android:layout_marginEnd="20dp"
                android:layout_marginRight="20dp"
                android:padding="7dp"
                android:text="AUTHORIZED" />


        </RelativeLayout>


    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/card7"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardBackgroundColor="@color/colorPrimaryDark"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/time_label_7"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_alignParentTop="true"
                android:padding="7dp"
                android:text="Time:" />

            <TextView
                android:id="@+id/time_7"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignEnd="@+id/date_7"
                android:layout_alignParentTop="true"
```

```
            android:layout_alignRight="@+id/date_7"
            android:padding="7dp"
            android:text="23:59" />

        <TextView
            android:id="@+id/date_label_7"
            android:layout_width="53dp"
            android:layout_height="match_parent"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_below="@+id/time_label_7"
            android:padding="7dp"
            android:text="Date:" />

        <TextView
            android:id="@+id/date_7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/date_label_7"
            android:layout_alignBottom="@+id/date_label_7"
            android:layout_toEndOf="@+id/date_label_7"
            android:layout_toRightOf="@+id/date_label_7"
            android:padding="7dp"
            android:text="2018-05-15" />

        <TextView
            android:id="@+id/status_label_7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/status_7"
            android:layout_alignBottom="@+id/status_7"
            android:layout_toLeftOf="@+id/status_7"
            android:layout_toStartOf="@+id/status_7"
            android:padding="7dp"
            android:text="Status:" />

        <TextView
            android:id="@+id/status_7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:layout_centerVertical="true"
            android:layout_marginEnd="20dp"
            android:layout_marginRight="20dp"
            android:padding="7dp"
            android:text="AUTHORIZED" />


    </RelativeLayout>


</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    android:id="@+id/card8"
```

```
android:layout_width="match_parent"
android:layout_height="75dp"
android:layout_margin="5dp"
card_view:cardBackgroundColor="@color/colorPrimaryDark"
card_view:cardCornerRadius="5dp">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/time_label_8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:padding="7dp"
        android:text="Time:" />

    <TextView
        android:id="@+id/time_8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/date_8"
        android:layout_alignParentTop="true"
        android:layout_alignRight="@+id/date_8"
        android:padding="7dp"
        android:text="23:59" />

    <TextView
        android:id="@+id/date_label_8"
        android:layout_width="53dp"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/time_label_8"
        android:padding="7dp"
        android:text="Date:" />

    <TextView
        android:id="@+id/date_8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/date_label_8"
        android:layout_alignBottom="@+id/date_label_8"
        android:layout_toEndOf="@+id/date_label_8"
        android:layout_toRightOf="@+id/date_label_8"
        android:padding="7dp"
        android:text="2018-05-15" />

    <TextView
        android:id="@+id/status_label_8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/status_8"
```

```
                android:layout_alignBottom="@+id/status_8"
                android:layout_toLeftOf="@+id/status_8"
                android:layout_toStartOf="@+id/status_8"
                android:padding="7dp"
                android:text="Status:" />

            <TextView
                android:id="@+id/status_8"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_centerVertical="true"
                android:layout_marginEnd="18dp"
                android:layout_marginRight="18dp"
                android:padding="7dp"
                android:text="AUTHORIZED" />


        </RelativeLayout>


    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
        android:id="@+id/card9"
        android:layout_width="match_parent"
        android:layout_height="75dp"
        android:layout_margin="5dp"
        card_view:cardBackgroundColor="@color/colorPrimaryDark"
        card_view:cardCornerRadius="5dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TextView
                android:id="@+id/time_label_9"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_alignParentTop="true"
                android:padding="7dp"
                android:text="Time:" />

            <TextView
                android:id="@+id/time_9"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignEnd="@+id/date_9"
                android:layout_alignParentTop="true"
                android:layout_alignRight="@+id/date_9"
                android:padding="7dp"
                android:text="23:59" />
```

```
            <TextView
                android:id="@+id/date_label_9"
                android:layout_width="53dp"
                android:layout_height="match_parent"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_below="@+id/time_label_9"
                android:padding="7dp"
                android:text="Date:" />

            <TextView
                android:id="@+id/date_9"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/date_label_9"
                android:layout_alignBottom="@+id/date_label_9"
                android:layout_toEndOf="@+id/date_label_9"
                android:layout_toRightOf="@+id/date_label_9"
                android:padding="7dp"
                android:text="2018-05-15" />

            <TextView
                android:id="@+id/status_label_9"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignBaseline="@+id/status_9"
                android:layout_alignBottom="@+id/status_9"
                android:layout_toLeftOf="@+id/status_9"
                android:layout_toStartOf="@+id/status_9"
                android:padding="7dp"
                android:text="Status:" />

            <TextView
                android:id="@+id/status_9"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_centerVertical="true"
                android:layout_marginEnd="19dp"
                android:layout_marginRight="19dp"
                android:padding="7dp"
                android:text="AUTHORIZED" />


        </RelativeLayout>


    </android.support.v7.widget.CardView>

  </LinearLayout>
</android.support.v4.widget.NestedScrollView>

Fragment_write.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="@dimen/activity_horizontal_margin">

    <ImageView
        android:id="@+id/logo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="8dp"
        android:src="@drawable/ic_nfc"
        android:tint="@color/colorAccent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_message" />

    <ProgressBar
        android:id="@+id/progress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="@dimen/activity_horizontal_margin"
        android:visibility="gone" />

    <TextView
        android:id="@+id/tv_message"
        android:layout_width="211dp"
        android:layout_height="23dp"
        android:layout_gravity="center"
        android:layout_marginTop="36dp"
        android:text="@string/message_tap_tag"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.503"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/token_field" />

    <Button
        android:id="@+id/share_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="share your token to"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
```

```
        android:id="@+id/phone_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="phone"
        android:hint="Phone number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.503"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/share_btn" />

    <Button
        android:id="@+id/set_token"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="24dp"
        android:text="set token / reset"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/phone_number" />

    <EditText
        android:id="@+id/token_field"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="leave blank to reset"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/set_token" />

</android.support.constraint.ConstraintLayout>

Navigation_header.xml
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="192dp"
    android:background="?attr/colorPrimaryDark"
    android:padding="16dp"
    android:theme="@style/ThemeOverlay.AppCompat.Dark"
    android:orientation="vertical"
    android:gravity="bottom">

    <TextView
        android:id="@+id/name"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="20dp"
        android:text="Robin"
        android:textColor="@android:color/background_dark"
        android:textSize="24sp"
        android:textStyle="bold" />

    <TextView
    android:id="@+id/account_label"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:text="Please login"
    android:textColor="@android:color/background_dark"
    android:textStyle="italic" />

/>

</LinearLayout>
```

Appendix G – GitHub/Version Control

Every group member has reasonably and equally contributed to the Project 2 github repository.

Repository link:
https://github.com/CPEN-291/G19_B_P2

Wiki (Server API's):
https://github.com/CPEN-291/G19_B_P2/wiki/Server-APIs