Bowen Brooks and Samuel Wu
Contact: bojbrook@ucsc.edu, sazwu@ucsc.edu

Lock Management System

Research: Computer Engineering Senior Design Project

Abstract

This project is for the Computer Engineering Senior Design Capstone, at University of California, Santa Cruz, CMPE123. The project is meant to create an Internet of Things (Iot) capable smartlock to be used in a building. Each room will have a cloud-connected smartlock that is unlocked with Near-Field Communication (NFC) technology. The project emphasizes cloud connectivity, power management, and cost.

CMPE123 Senior Design Project: Lock Management System

Bowen Brooks, Samuel Wu *

Contents

*Thanks to Professor Anujan Varma and teaching assistants Sargis Yonan and Sam Mansfield for their guidance during this project

1 Summary of Project

This goal of this project is to create a lock management system that could potentially replace the BSOE current system. Our system will have real time access to the rooms through the Google cloud. Each lock will have a microcontroller connected to an NFC sensor. A user will unlock the room by tapping their android phone to the NFC sensor. The locks will have access to the Internet through a gateway that runs all of the queries for locks. Their will also be an administrative website that can view real time logs for each room and grant/revoke access at any given time.

2 Major Components

2.1 TI 3220S Microcontroller

The microcontroller is hooked up to an NFC sensor which reads in a CruzID. This be sent to Google Cloud which returns the room_numbers that the user has access to. If the given room is in the list it will turn a motor to unlock the door. The microcontroller will be cycling through low power mode in order to preserve the battery life from the AA batteries.

2.2 Raspberry Pi gateway

The gateway will run on a raspberry pi 3 which will serve as a wifi access point for all the locks connecting to it. It will connect to the micros using TCP and SSL. Once it has a roomid and cruzId it contact the google datastore and run a query to find if the given user has access to the room. Once it gets a respond to the correct micro with the information.

2.3 Google Cloud

Students, Faculty and rooms are all stored in Google Cloud. The cloud will respond from results from the microcontroller which will determine if a user has access to the rooms. The cloud storage can only be updated and from the administrative website. Students and rooms can be updated at anytime giving real-time access to the rooms. The cloud also logs all the login information for each room and which can be view on the administration website.

2.4 Battery

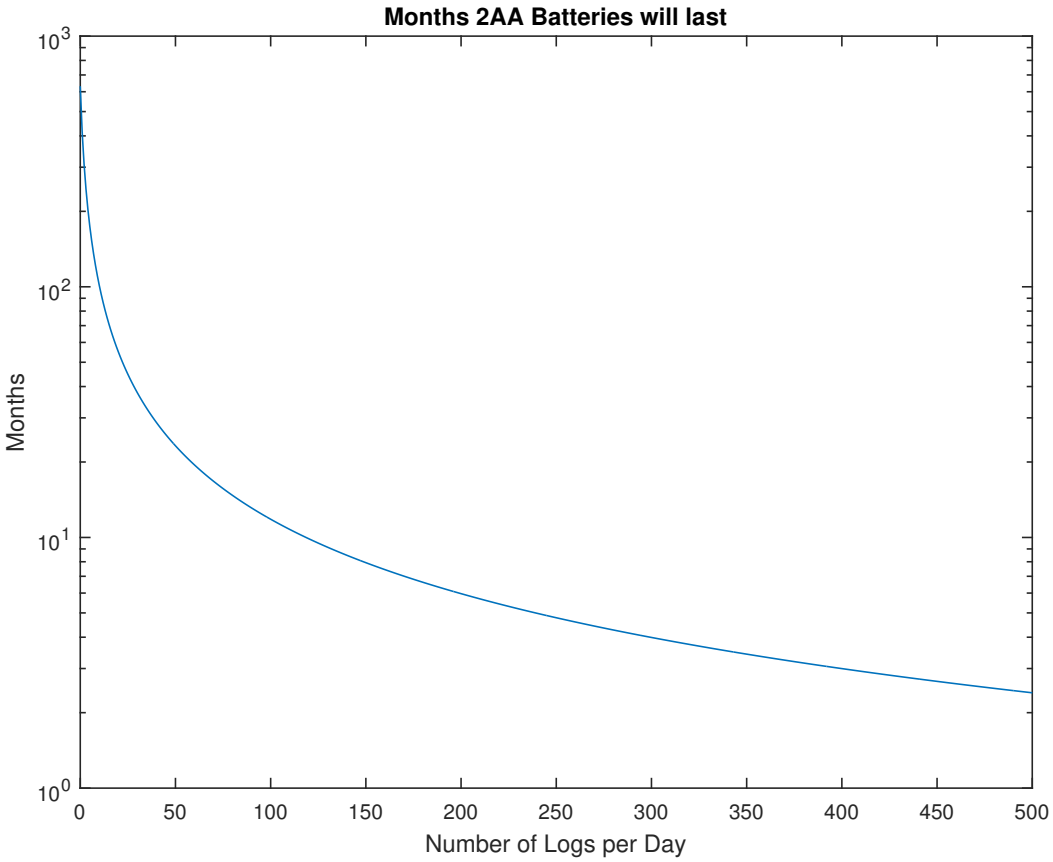

Figure 1: Theoretical Battery life using 2 AA batteris and the NXP MFRC522 Sensor

2.4.1 Entities

Room Entity		
ID	Room #	Classes with Access
name=index0	E2-399	[”AMS147”,”CMPS101”]
name=index1	BE340A	[”CMPE123A”,”CMPE123B”, ”CMPE129B”]

Student Entity			
ID	Name	cruzID	Classes
name=index0	Samuel Wu	sazwu	[”AMS147”,”CMPS12B”]
name=index1	Bowen Brooks	bojbrook	[”CMPE123A”]

BE340 Log Entity				
ID	cruzID	Enter Time	Exit Time	Cumulative
name=sazwu0	sazwu	02-07 17:26:55	02-07 17:27:02	0:00:07
name=bojbrook0	bojbrook	02-07 17:26:55	02-08 11:30:22	18:03:26
name=sazwu1	sazwu	02-08 11:27:19	02-08 11:27:57	0:00:44
name=hello0	hello	02-08 11:31:02	null	null

2.5 Administration Website

The administration website is where the faculty can add and revoke student access to the rooms. They can also view analytics from Google Cloud such as peak usage time and current room capacity.

2.6 Android App

The android application allows for a user to sign in using their CruzID and password. When the phone is tapped against an NFC sensor the application will transmit the CruzID to the microcontroller. The application will have access to the Google cloud in order to view which rooms they have access to.

3 Block Diagram

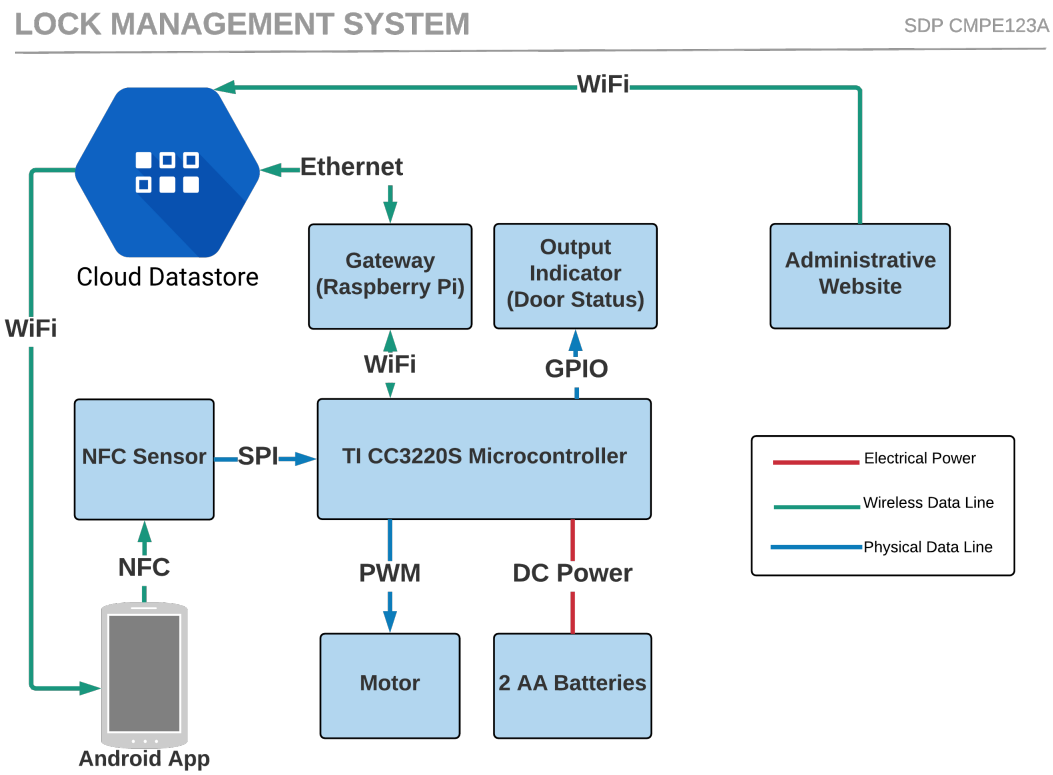

Figure 2: Block Diagram of System

4 Software Flow Chart

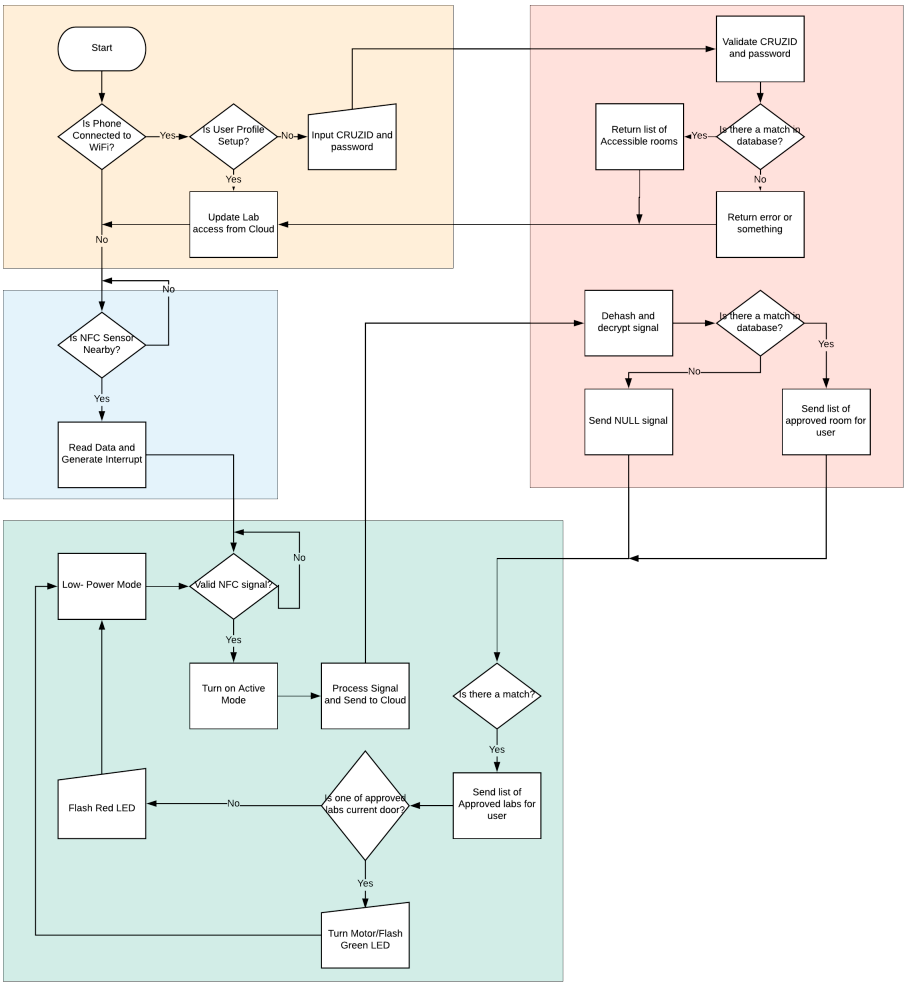

Figure 3: Yellow: Android app; Blue: NFC sensor; Green: microcontroller; Red: Google Cloud

5 Schedule

5.1 Winter Quarter

5.1.1 Quarter Goals

We want to be able to have the microcontroller send and receive information to and from the cloud. Additionally, the NFC sensor should be able to read unique *RFID* tags.

5.1.2 Tasks for the Quarter

1. RFID/MCU communication
2. Cloud/MCU communication
3. Database
4. LEDs
5. Power management

Bowen	
Week 4	Design database, Design cloud API, Populate database
Week 5	Design cloud API
Week 6	Cloud AUTH/access, Design cloud API
Week 7	MCU Push/pull database, MCU Log interaction
Week 8	Cloud push results to MCU
Week 9	Finish MCU, Cloud clean up
Week 10	Start API calls for website

Sam	
Week 4	Learn basic Google Cloud, Design database, Design cloud API
Week 5	Populate the database
Week 6	MCU internet access, SPI interface for sensor
Week 7	MCU/RFID communication (MRFC522)
Week 8	MCU/RFID communication (PN522)
Week 9	Differentiate unique RFID tags, power management
Week 10	RFID wakeup MCU, sleep/wake modes

5.2 Spring Quarter

5.2.1 Quarter Goals

Everything should be completely finished. This includes full functionality of the Android App for communicating with the cloud and the NFC sensor with a *NFC* signal. Additionally, there will be an administrative website that pulls data from the cloud. The website will have administrative functionalities such as: adding, removing, or modifying user privileges. The administrator will also be able to navigate a clean UI to view analytics and data.

5.2.2 Tasks for the Quarter

1. NFC/MCU communication
2. NFC/App communication
3. Website
4. DC Motor/H-Bridge
5. Web/Cloud communication
6. App/Cloud communication

Bowen	
Week 1	Familiarize with Android API, Design UI
Week 2	Clean up anything from Winter, NFC/App comm
Week 3	NFC/App comm, App/Cloud comm
Week 4	App/Cloud comm, start design for website
Week 5	Finish app, cleanup app UI, website UI
Week 6	Website/cloud comm, website UI
Week 7	Finish website, debug all comm
Week 8	Debug all comm, debug any small things
Week 9	Buffer week, start report/presentation
Week 10	Finish everything

Sam	
Week 1	MCU/NFC comm (M6E Nano), App/NFC comm
Week 2	MCU/NFC debug, App/NFC comm
Week 3	DC Motor, clean up MCU code
Week 4	App/Cloud comm, Website
Week 5	Website UI, Web/Cloud comm
Week 6	Fix any small bugs, DC motor/H-Bridge
Week 7	Cleanup App UI, website UI
Week 8	Debug everything, buffer week
Week 9	Buffer week, start report/presentation
Week 10	Finish everything

6 List of hardware components

Component	Cost	Quantity	Total
TI CC3200S	$39.99	2	$79.98
Battery Case	$1.50	2	$3.00
Op Amp	$0.95	4	$3.80
Resistor Kit	$7.95	1	$7.95
NFC Sensor MRFC522	$9.99	0	$0.00
AA battery 20 pack	$8.54	1	$8.54
NFC Sensor PN532	$12.99	2	$25.98
Motor	$1.95	2	$3.90
LEDs 5 Pack	$2.95	1	$2.95
H-Bridge	$2.35	2	$4.70
Total Cost			$140.80