Attendance Tracking Via QR Codes

Benjamin Dean
Weisberg Division of Computer Science
Marshall University
Huntington, WV
deanb@marshall.edu

1. INTRODUCTION

The project started out with a concept to track beverages in a restaurant setting. The plan was to leverage QR codes and web cameras to simulate a possibly more elaborate setup such as RFID or NFC. The cost barrier was another deciding factor leading us down the path of QR codes and web camera design. The project morphed during the initial planning phases to attendance tracking of students within a classroom environment.

Although the idea of taking attendance has been around for a long time, making a better attendance tracking system has recently become a hot topic. For large institutions such as universities and some primary and secondary schools the sheer volume of classrooms can be a barrier with more expensive technologies. The camera we went with was $6 and is an USB device, but many devices such as laptops, tablets and even all in one computers come with web cameras built in. The QR codes would need to be added to student ID’s and could be phased in over a period of time.

2. BUSINESS PROBLEM

There are many benefits of a robust tracking system for student attendance that can help universities and students themselves see success. At the moment attendance is tracked mainly to award points for showing up to class. Sometimes it is used periodically throughout a time period to see the change in class size and who is showing up to class. This is not performed consistently or across the spectrum and by motivating a change in this process with simpler tools, data mining can be done to detect patterns and gather real insights into students, classes, and attendance in general.

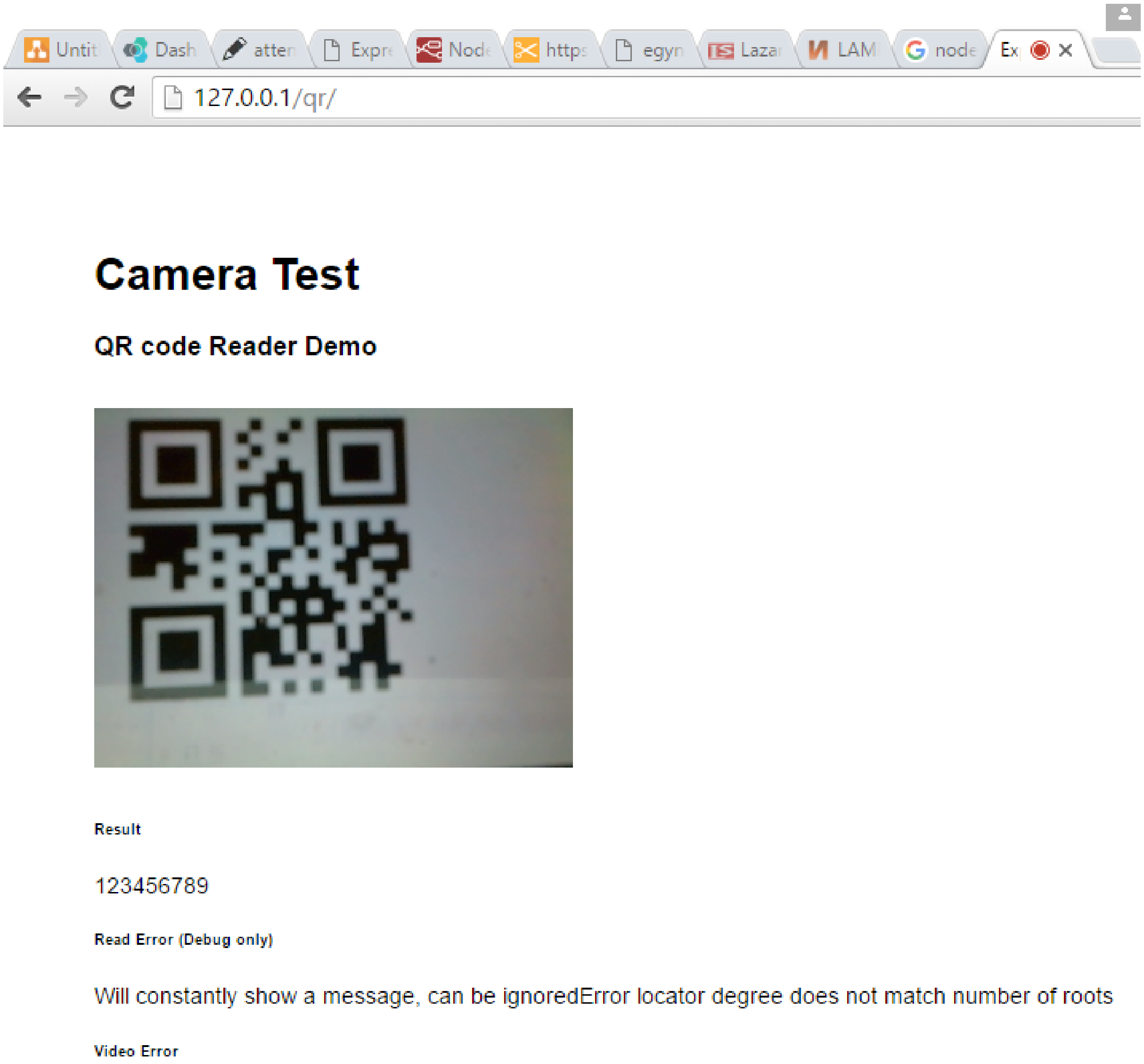
Aside from general tracking, i.e. present / not present, factors such as time arrived, time of day and many other types of information could now be leveraged. The reporting possibilities could really be enhanced, thus potentially leading to better scheduling, higher student success rates and better needs analysis to help struggling students. According to the National Center for Education Statistics, students who attend regularly achieve at higher levels than those who do not have regular attendance [1]. With better data and more meaningful insights the institutions will be able to better see and drive student success with these attendance tracking tools.

3. SOLUTION OVERVIEW

Our solution takes advantage of using a cloud environment, but is built on the technologies that can be used on-site as well. The benefit of the cloud is that the solution can be purchased as a hosted solution or an institution can take on the role of deploying, maintaining and extending the product as well. For the cloud we went with Bluemix [2], an IBM solution. We chose Bluemix because it seemed to be the easiest to bring up our platform and start developing right away. Within our cloud we have three main parts to our solution, an API server, Database Server, and Web Server. These three pieces along with our existing classroom hardware bring our solution to life.

For demonstration purposes we can use a laptop with a web camera and workflow through the entire process. Doing this we can stop along the way and identify the technologies in place to make each part of the process work. The main parts of the process are: 1. scanning and sending information to be tracked, and 2. processing and storing the information.

A student enters a classroom and presents their card in front of an attendance tracking device. This device can be, for example, a laptop with a web camera and the QR tracking webpage brought up.



Camera Test

QR code Reader Demo

Result

123456789

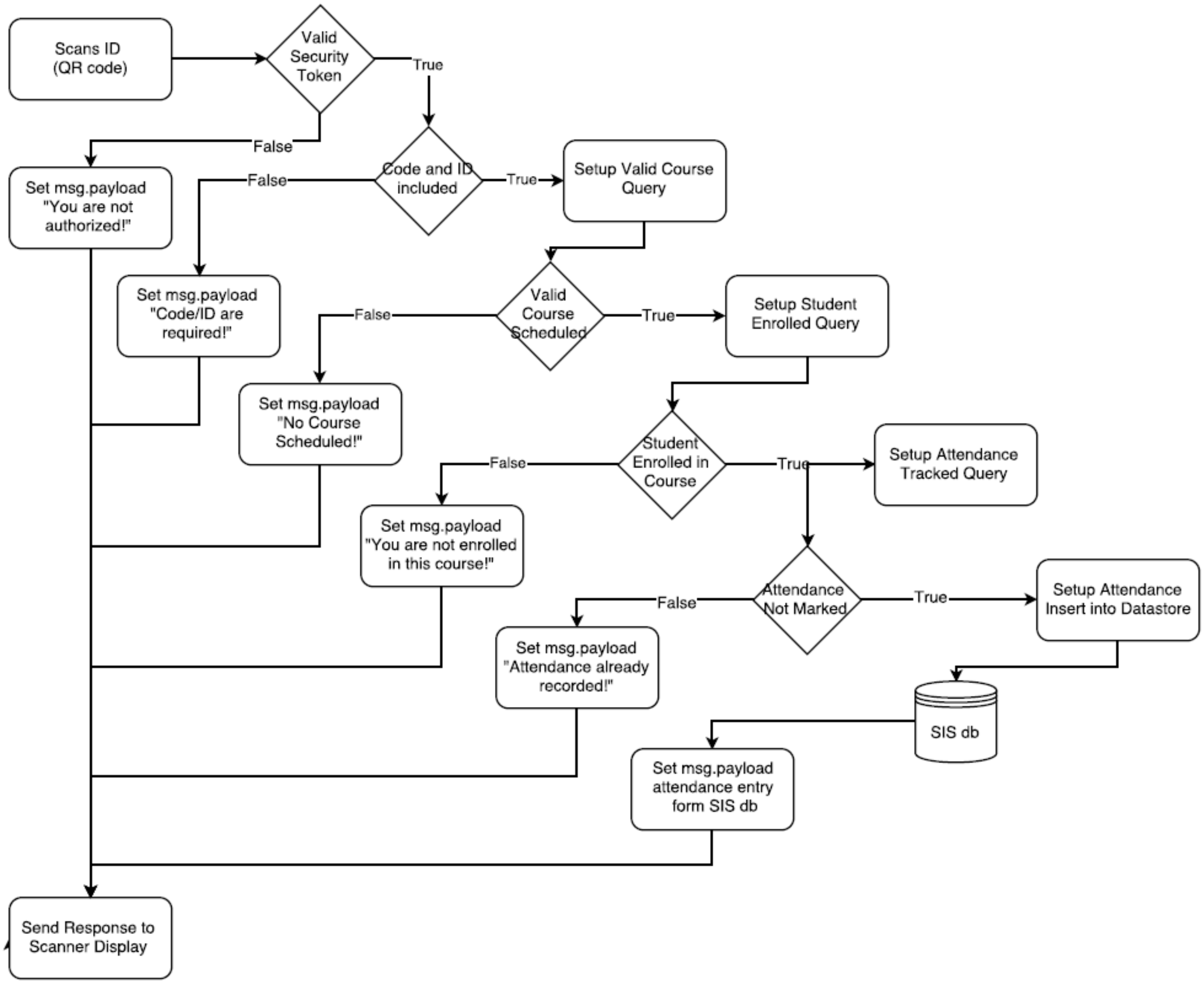
Read Error (Debug only)

Will constantly show a message, can be ignoredError locator degree does not match number of roots

Video Error

Figure 1: QR Reader Webpage

This webpage utilizes QR reading technology, from LazarSoft [3], built into a customized html page running JavaScript and jQuery. The page will scan the QR code which is a representation of the student's ID. The page upon successful scan will take its local IP address, student's ID, and a timestamp and send this information to the tracking API.


Scans ID (QR code)
Valid Security Token
True
False
Set msg.payload "You are not authorized!"
Code and ID included
False
True
Setup Valid Course Query
Set msg.payload "Code/ID are required!"
Valid Course Scheduled
False
True
Setup Student Enrolled Query
Set msg.payload "No Course Scheduled!"
Student Enrolled in Course
False
True
Setup Attendance Tracked Query
Set msg.payload "You are not enrolled in this course!"
Attendance Not Marked
False
True
Setup Attendance Insert into Datastore
Set msg.payload "Attendance already recorded!"
SIS db
Set msg.payload attendance entry form SIS db
Send Response to Scanner Display

Fi

gure 3; Attendance tracking Student Process flow

Once the tracking API has received the information it begins the process with a SIS, or student information system, to run a few checks before storing the attendance. For our project we created a simplified SIS and served both the SIS and tracking database on the same database server, which was IBMs DB2. The APIs were built using Node Red and both the tracking and SIS API's were created here together since they were already coupled in the database. The first step in processing was to identify what class the scanner belongs to via a lookup table, attaching the Building and room information. Next, we utilized the location information and the time stamp to find a valid class in this location. Then, we verify the student is registered for this class. Finally if we have successfully passed all our checks, we insert the student along with the course ID, date and time into the attendance tracking table.

4. SYSTEM ARCHITECTURE

One of the nice aspects of the cloud was the variety of tools and services available to use allowing for quick changes in our architecture throughout our development process. Developing locally was how we began with the QR web page, my former teammate took on the initial development and design of this aspect of our solution. When he left I took on the code and it took me a bit to figure out how to implement a local solution to the cloud. After trying a Node.js server and having no luck, by chance I set it up on a local LAMP (Linux, Apache, MySQL, PHP) server which I started using for our user interface design. It so happened to work in this environment and I was back in business with the QR reading portion.

For the API development I took this section head on in conjunction with the database design and coupled the two together for this initial project. The database is on a light weight DB2 database and as such has a limited connection pool: 10 connections. After designing the initial ERD and uploading the initial data we found that there was not a direct development environment for the database. To circumvent this problem and to prepare for the needed API for the attendance tracking, we started developing the API's to interact with our simple SIS. The entire API is built in Node Red which is built on top of a Node.js server and uses Cloudant NoSQL DB. The first piece to be developed was the connection to the database from the QR reader. At first this took 4 database connections which are called nodes in Node Red. Next I started to make the Student API and soon ran out of database connections. The limitation of 10 connections was directly related to the DB nodes and this became a major problem. To circumvent this problem a middleware piece was created to handle API routing and a single database connection. One of the nice aspects of Node Red was how easy it is to add nodes and wire them up. This, however, with my new design, started to look monstrous and a bit daunting to maintain.

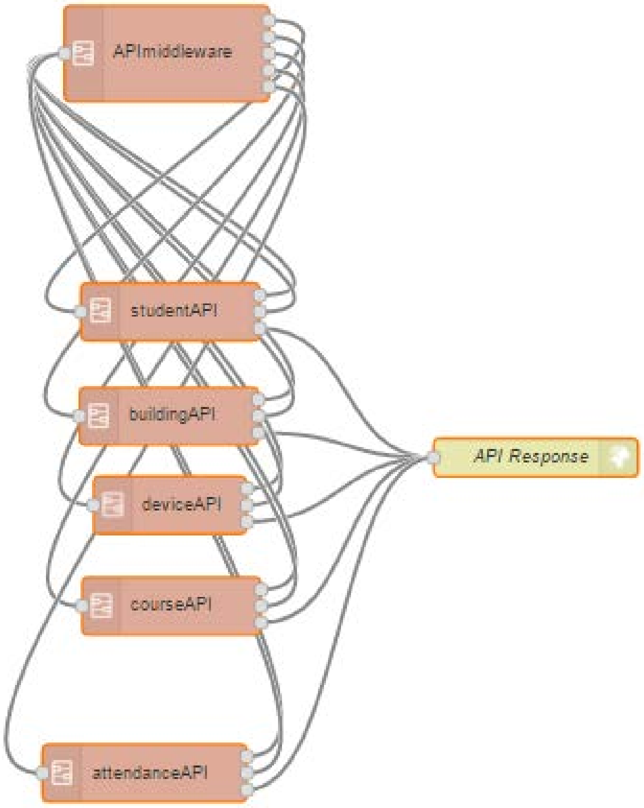


Figure 4: Reorganized API main flow

Another reorganization was done with the API to help maintain the code and make it easier to follow. After learning how to use sub-flows, each API such as student, course, or attendance was relocated to a sub-flow with a connection coming in and 3 connections going out.

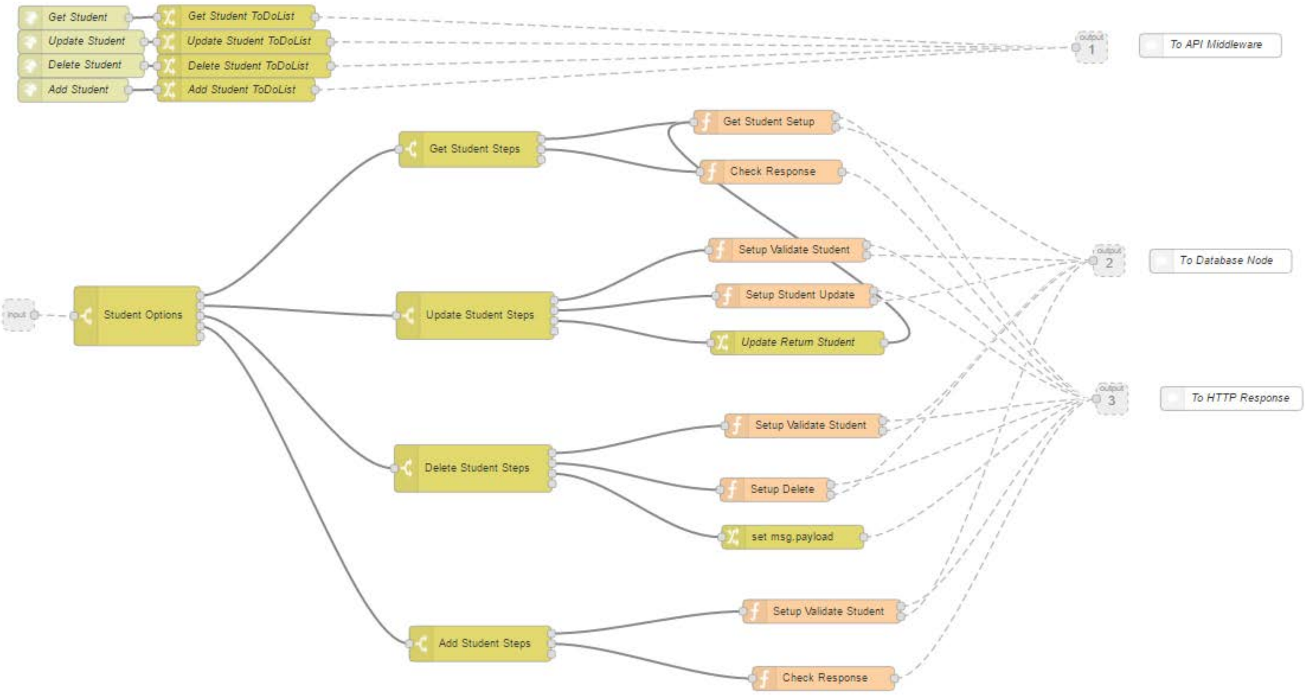


Figure 5: Student API sub-flow

The entire middleware process was also moved to a sub-flow, also with 1 connection in, but now a connection for each API section such as student or course coming out.

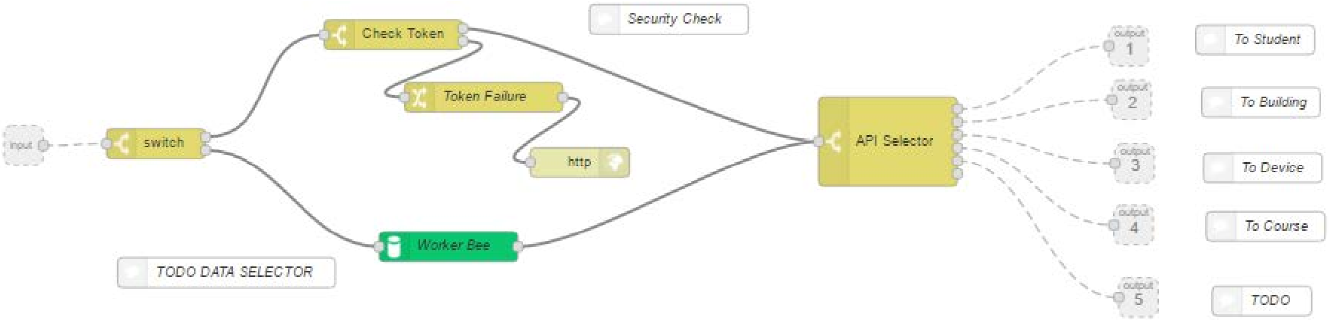


Figure 6: API Middleware sub-flow

This made building additional API's simpler as each basic function of the API, get (select), put (update), post (insert), delete (delete), operations.

The last piece of the cloud puzzle was the user interface to be able to get information out and display it in a stylized fashion. Unfortunately for me, this has been the more difficult part of the process. I tried to use Node.js but was not successful in the initial development and switched to PHP after working recently on a couple of work projects.

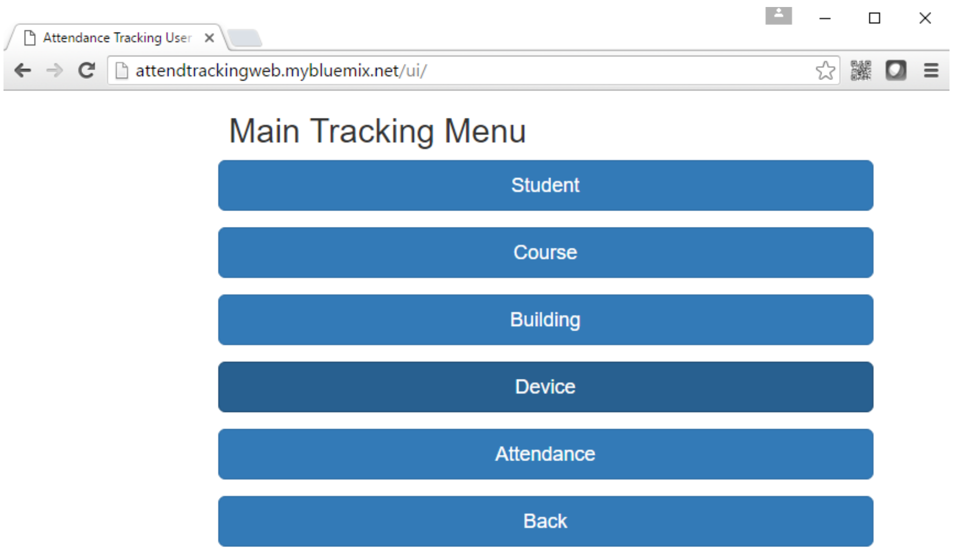


Figure 7: UI Main Menu

The UI could be developed in about any environment that can take advantage of restful calls to the API. The basic layout and design I went with was to facilitate use on either a computer, tablet or phone. Seeing as this is a prototype and time became more of a factor less of that time was spent on fully developing this aspect. Given a true SIS most of these pieces would be built-in and standard reporting tools for data analysis could be used to show metrics from the attendance tracking database.

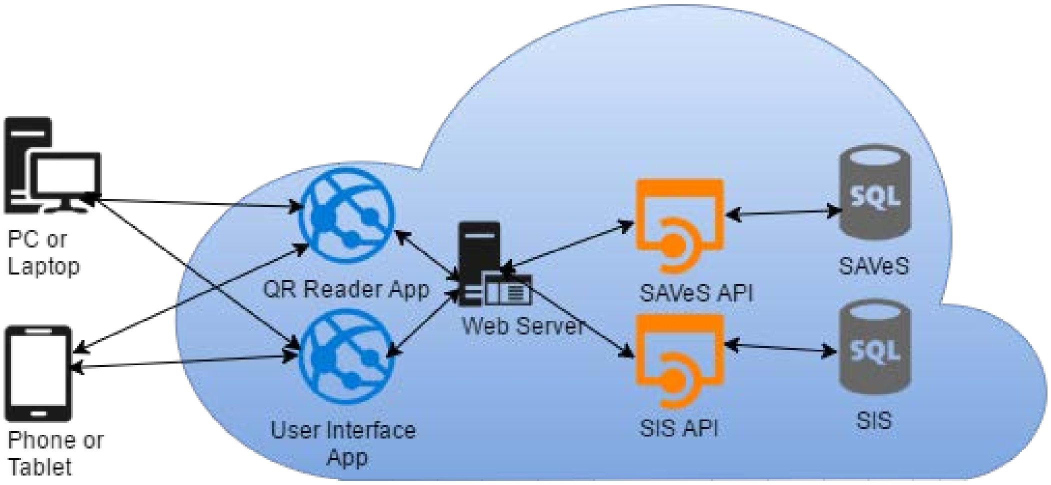
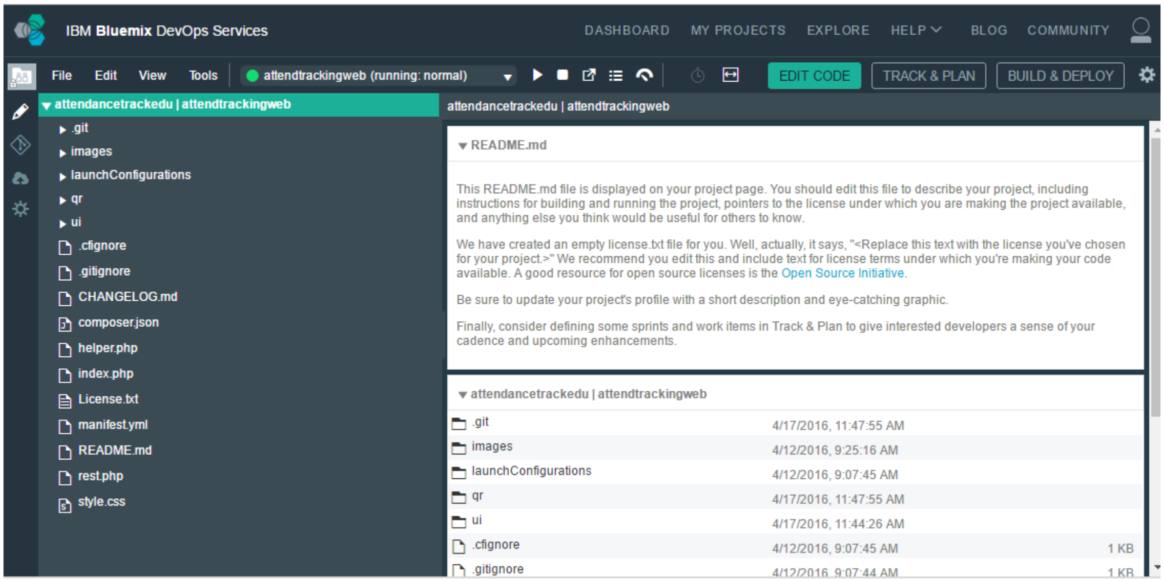


Figure 8: System Architecture

Some very useful tools found along the way that became a part of our development process was GIT [4] and VirtualBox [5]. I ended up creating for the QR reader webpage and the UI webpage a local server on VirtualBox. Although you can do all the development on Bluemix, the biggest challenge was testing small changes or bug fixes. Every time you make a change on a DevOps instance of your web directory, it would require a complete restart of the webserver.



This README.md file is displayed on your project page. You should edit this file to describe your project, including instructions for building and running the project, pointers to the license under which you are making the project available, and anything else you think would be useful for others to know.

We have created an empty license.txt file for you. Well, actually, it says, "<Replace this text with the license you've chosen for your project.>" We recommend you edit this and include text for license terms under which you're making your code available. A good resource for open source licenses is the Open Source Initiative.

Be sure to update your project's profile with a short description and eye-catching graphic.

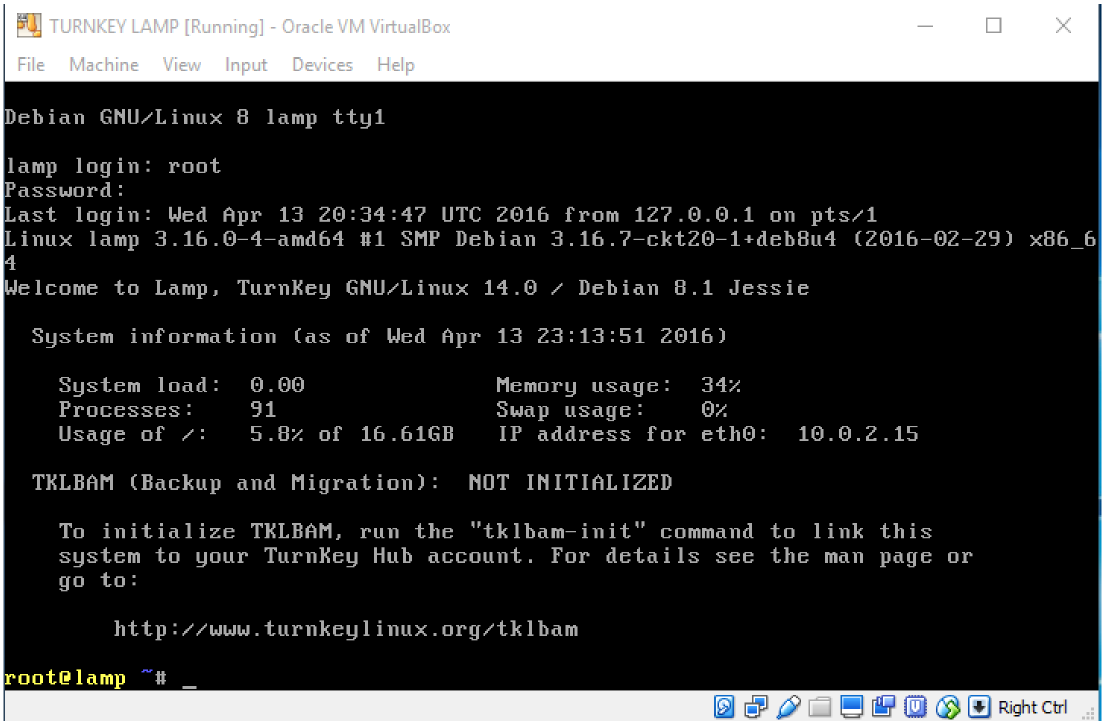
Finally, consider defining some sprints and work items in Track & Plan to give interested developers a sense of your cadence and upcoming enhancements.

attendancetrackedu | attendtrackingweb

.git	4/17/2016, 11:47:55 AM	
images	4/12/2016, 9:25:16 AM	
launchConfigurations	4/12/2016, 9:07:45 AM	
qr	4/17/2016, 11:47:55 AM	
ui	4/17/2016, 11:44:26 AM	
.cfignore	4/12/2016, 9:07:45 AM	1 KB
.gitignore	4/12/2016, 9:07:44 AM	1 KB

Figure 9: Bluemix DevOps

The restarts took several minutes, and at one time there were connectivity issues with the cloud services in which an entire day was lost. I found a premade virtual machine called Turnkey LAMP [6] which is based off of Turnkey Linux, and put this into place to do local development.



```
Debian GNU/Linux 8 lamp tty1

lamp login: root
Password:
Last login: Wed Apr 13 20:34:47 UTC 2016 from 127.0.0.1 on pts/1
Linux lamp 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt20-1+deb8u4 (2016-02-29) x86_6
4
Welcome to Lamp, TurnKey GNU/Linux 14.0 / Debian 8.1 Jessie

  System information (as of Wed Apr 13 23:13:51 2016)

    System load:  0.00              Memory usage:  34%
    Processes:    91                Swap usage:    0%
    Usage of /:   5.8% of 16.61GB   IP address for eth0:  10.0.2.15

  TKLBAM (Backup and Migration):  NOT INITIALIZED

    To initialize TKLBAM, run the "tklbam-init" command to link this
    system to your TurnKey Hub account. For details see the man page or
    go to:

        http://www.turnkeylinux.org/tklbam

root@lamp ~# _
```

Figure10: Turnkey Linux

Once I had Turnkey up and running I was able to then clone the GIT repository on the DevOps instance on Bluemix. Now I could make quick edits and tests and see the results right away without restarts. Once I got to a major milestone or even quick little fixes along the way I could always push the updates back to the cloud and redeploy to publish the changes.

5. Code

The code for this project was done in many different formats and some is quite difficult to read as just plain text. This is especially true for the Node Red portion. A project repository for all of the main parts has been created on github and available at the following location: https://github.com/bbd999/CS652_Project .

6. Conclusion

This project was a lot of fun to work on and has the potential to serve a very real need in education at all levels. The proof of concept and design work with very simple technologies to show what can be done and what little is needed to get up and running with a solution. Although there are many other technologies already in place such as NFC and RFID the concept can be altered to work with these as well. Hybrid solutions to work with existing technologies and fallback technologies such as the QR code or bar codes could be implemented as well. The biggest driving force for this project would be the data analysis and insights that could be attained by a robust attendance tracking system. This driving force ties in directly with where education appears to be headed: student success.

7. References

[1] Every School Day Counts: The Forum Guide to Collecting and Using Attendance Data - Why Does Attendance Matter? (2009, February). Retrieved April 25, 2016, from https://nces.ed.gov/pubs2009/attendancedata/chapter1a.asp

[2] IBM Bluemix - Next-Generation Cloud App Development Platform. (n.d.). Retrieved April 25, 2016, from https://console.ng.bluemix.net/

[3] LazarSoft. (n.d.). Retrieved April 25, 2016, from http://www.lazarsoft.info/ .

[4] Build software better, together. (n.d.). Retrieved April 25, 2016, from https://github.com/

[5] Welcome to VirtualBox.org! (n.d.). Retrieved April 25, 2016, from https://www.virtualbox.org/

[6] LAMP Stack. (n.d.). Retrieved April 25, 2016, from https://www.turnkeylinux.org/lampstack