Passenger Analytics and Card Validation System

Ciobanu Mihai, Feher Robert, Politehnica University of Timisoara May, 2018

# Repository

The project history, schematics, diagrams and codebase are contained under the following git repository:

**https://github.com/Coll199/ms-project**

# User requirements

1. The system must provide information about passenger analytics (no of passengers, duration of trip).
2. The system should be able to validate cards.
3. The system should run in an environment that provides a 24/24 access.
4. Passenger analytics should be accessed via a Web interface or a mobile application.
5. The system might provide access to history data for the last 2 weeks.
6. The system may provide a module for data interpretation.

# System overview

The overview of the system is depicted in Figure 1.

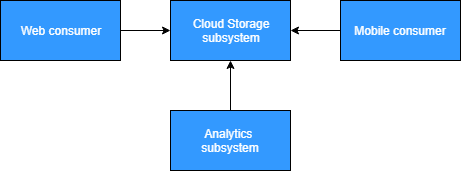


Figure 1: System overview diagram

Base (Analytics) Subsystem encompasses the measurement functionality. It has the sole purpose to acquire information from its sensors.

Cloud Storage Subsystem stores the data pushed by the Base Subsystem. Additionally, it offers a possibility to interpret stored information.

Web consumer provides a UI interpretation for the data stored in the Cloud Storage Subsystem. This view is accessed within a Web browser.

Mobile consumer also provides a method for creating an account and linking it with the Database.

# Circuit design

The hardware view of the system is depicted in Figure 2.

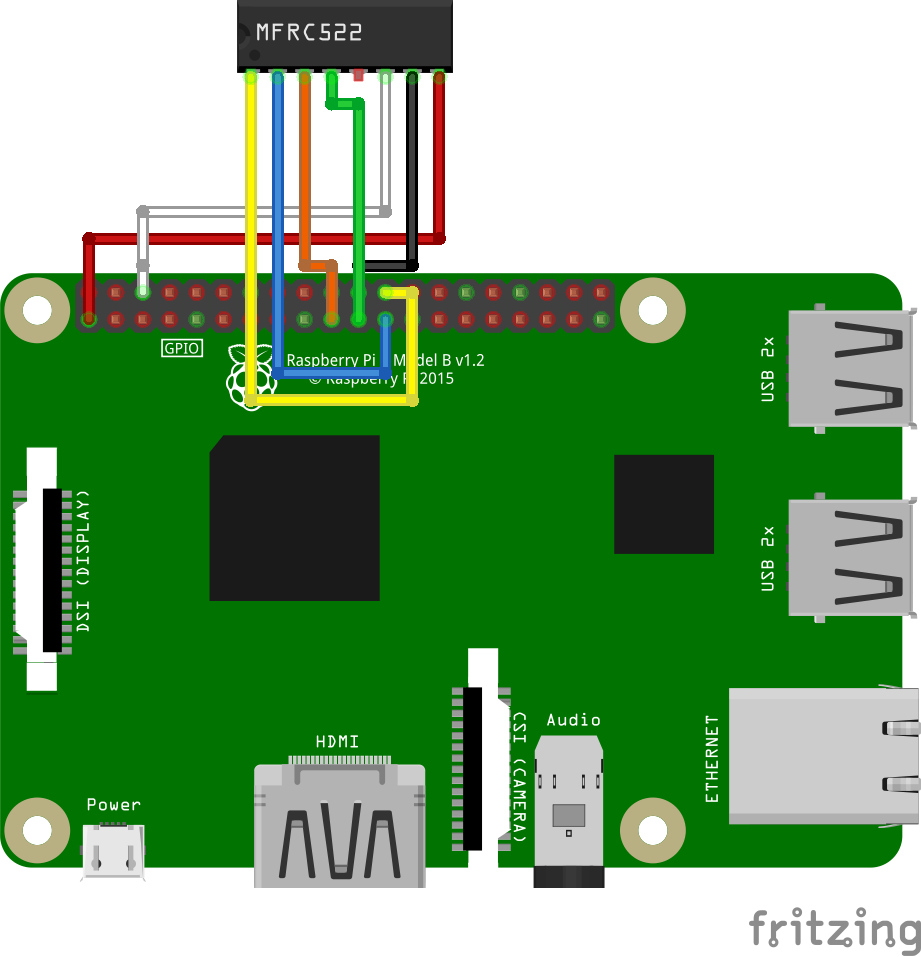


Figure 2: Circuit schematic

Raspberry Pi 3 provides support for quick prototyping. That makes it a perfect choice for quick prototyping but not adequate for real-time applications. We will use the Serial Peripheral Interface Bus it has, but also the implicit possibility of communicating with other devices over the Internet.

The MFRC522 is a highly integrated reader/writer IC for contactless communication at 13.56 MHz

The wiring of the components can be observed in Figure 2. An important aspect we need to mention is that the communication between the board and sensor is done via the Serial Peripheral Interface Bus (SPI).

# Software design

The software components and data flow directions are depicted in Figure 3. Each of these will be presented in the following subsections.

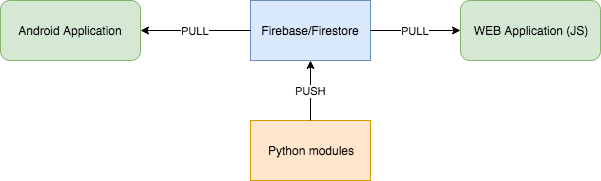


Figure 3: Software entities involved

## Python modules

Read.py: module which continuously waits for the card reader to read the card, after which it adds data into the Database

Write.py: module which writes the data needed for authentication to the card

Station.py: module which simulates going from a tram stop to another

MFRC522 library: it provides a quick implementation of reading and writing data to the card.

## Firebase

Firebase is a PaaS (Platform as a Service) which means it offers developers to a quick list of functionalities supported by a traditional backend. Realtime Database simplifies storing and synchronizing data between different devices in realtime using a noSQL database.

The reason we are using Firebase is for easier authentication trough Google services, for ease of use and because the realtime data we get through it is reliable.

**5.3 Android application**

The Android application is used for creating the account which will be linked together with the card and to be able to “pay” for services.

**5.4 Web Application**

The Web application retrieves data from the database to plot graphs.

# Results and further work

*Described what you accomplished so far, then what you plan to improve or extend.*

The current version of the project supports the following functionalities:

* storing data to Firebase Database
* card validation
* client implementations for retrieving data stored in Firebase Database (WEB)

The following list of extensions and improvements was identified to be supported in the future:

* use Wireless and Bluetooth to verify If client has gone out of range
* extend Web application to present the data more thoroughly
* GPS module to get better precision

# References

1. Fritzing [last seen: May 2018], <http://fritzing.org/>
2. Firebase Database [last seen: May 2018], [https://firebase.google.com/docs/ database/](https://firebase.google.com/docs/database/)
3. Draw.io [last seen: May 2018], <https://www.draw.io/>
4. MFRC522 Python Library [last seen: May 2018] , https://github.com/mxgxw/MFRC522-python