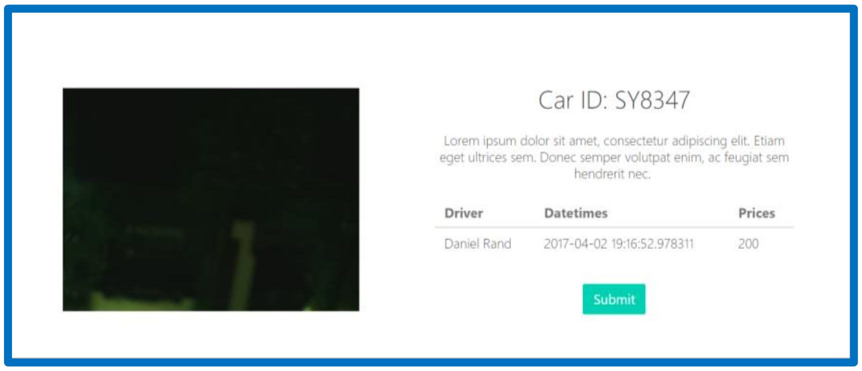
The Smart Toll Booth Project Report

Embedded System Subject, 2016



Car ID: SY8347

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eget ultrices sem. Donec semper volutpat enim, ac feugiat sem hendrerit nec.

Driver	Datetimes	Prices
Daniel Rand	2017-04-02 19:16:52.978311	200

Submitted by

Thanatcha Sangpetch (57010574)

Patcharapon Jantana (57010869)

Isara Naranirattisai (57011546)

Submitted to

Asst.Prof.Apinetr Unakul

Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang

Contents

Abstract 3
Requirements 4
    Functional Requirements 4
    Non-Functional Requirements 4
System Design 5
  Overview 5
  Class Diagram 6
  Use Case Diagram 7
  Sequence Diagram 8
  ER Diagram 9
  Architecture Diagram 9
  Database Structure 10
    Database Table 10
    Table “Car” (with sample data) 10
    Table “Transaction” (with sample data) 10
System Integration and Test 11
User Interface 12
Project Repository 14
References 15

Abstract

The purpose of this project, The Smart Tollbooth, is to make the system for the toll-booth for the car parking in a very cheap price (using raspberry pi3) that can detect rfid, connect to the database and more! (please look at the overview section), To make tutorial for the Interested people And to make it a learning source for the people who're interested in embedded system using raspberry pi, And also make it open source (under MIT License) which is mean everybody can help improve it to make the system better.

Source code is available on Github's repository (https://github.com/DreamN/Smart-Tollbooth)

*For using or editing in another project "Giving credit must be required"

Requirements

Functional Requirements

- User can scan the RFID Card
- Calculate fee from parking time
- Insert transaction into database
- See car’s photo
- View Transaction
- View Car and its description
- Detect the license plate
- Authorize car to pass
- Open and close the barrier

Non-Functional Requirements

- 24/7 working and less than 1% down time
- Access from the Internet
- Accept the rfid that registered

System Design

Overview

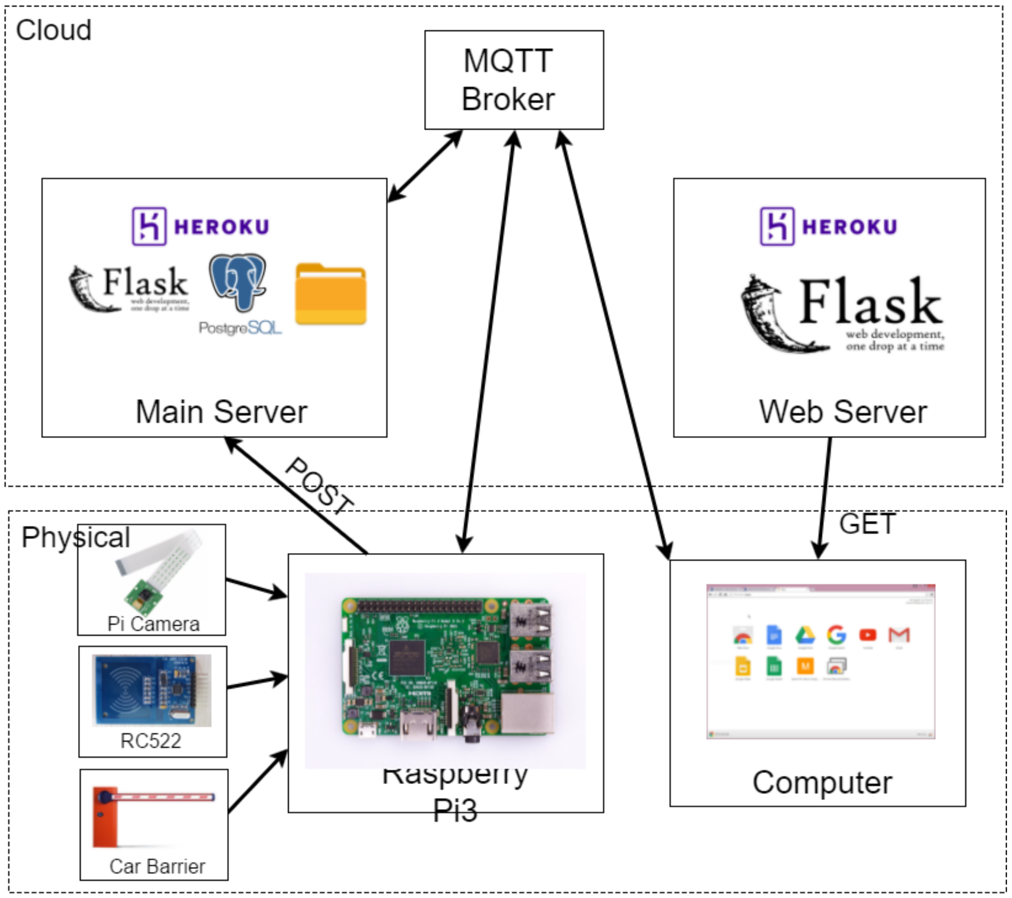


Figure 3.1.1 Overview of the system

Class Diagram

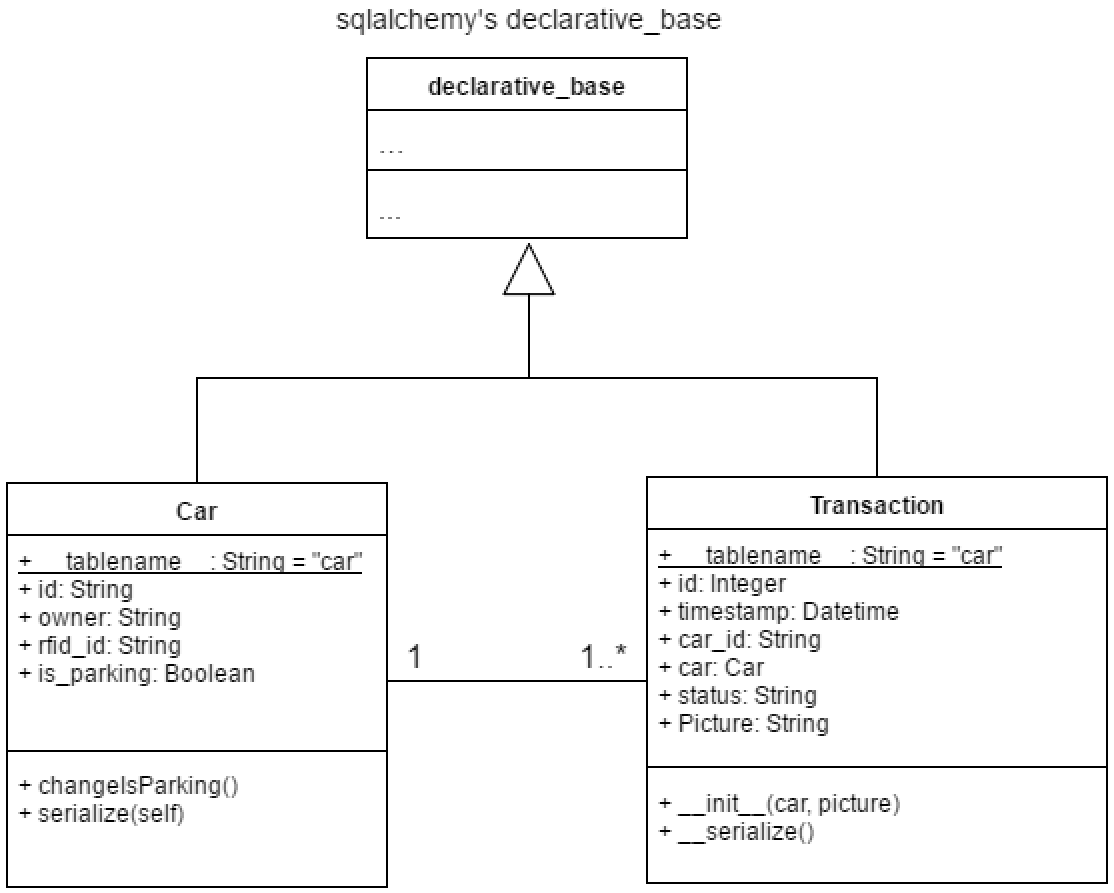


Figure 3.2.1 Class Diagram

Use Case Diagram

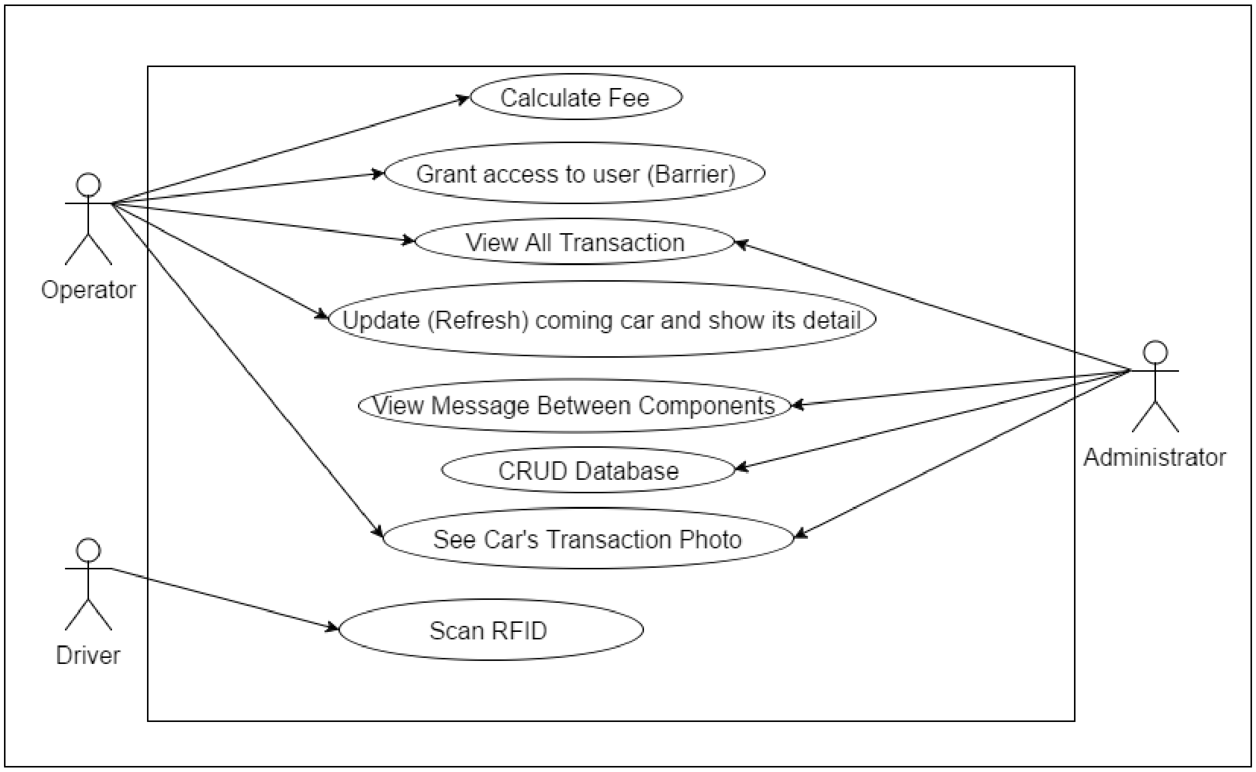

Calculate Fee
Grant access to user (Barrier)
View All Transaction
Update (Refresh) coming car and show its detail
View Message Between Components
CRUD Database
See Car's Transaction Photo
Scan RFID
Operator
Driver
Administrator

Figure 3.3.1 Use Case Diagram

Sequence Diagram

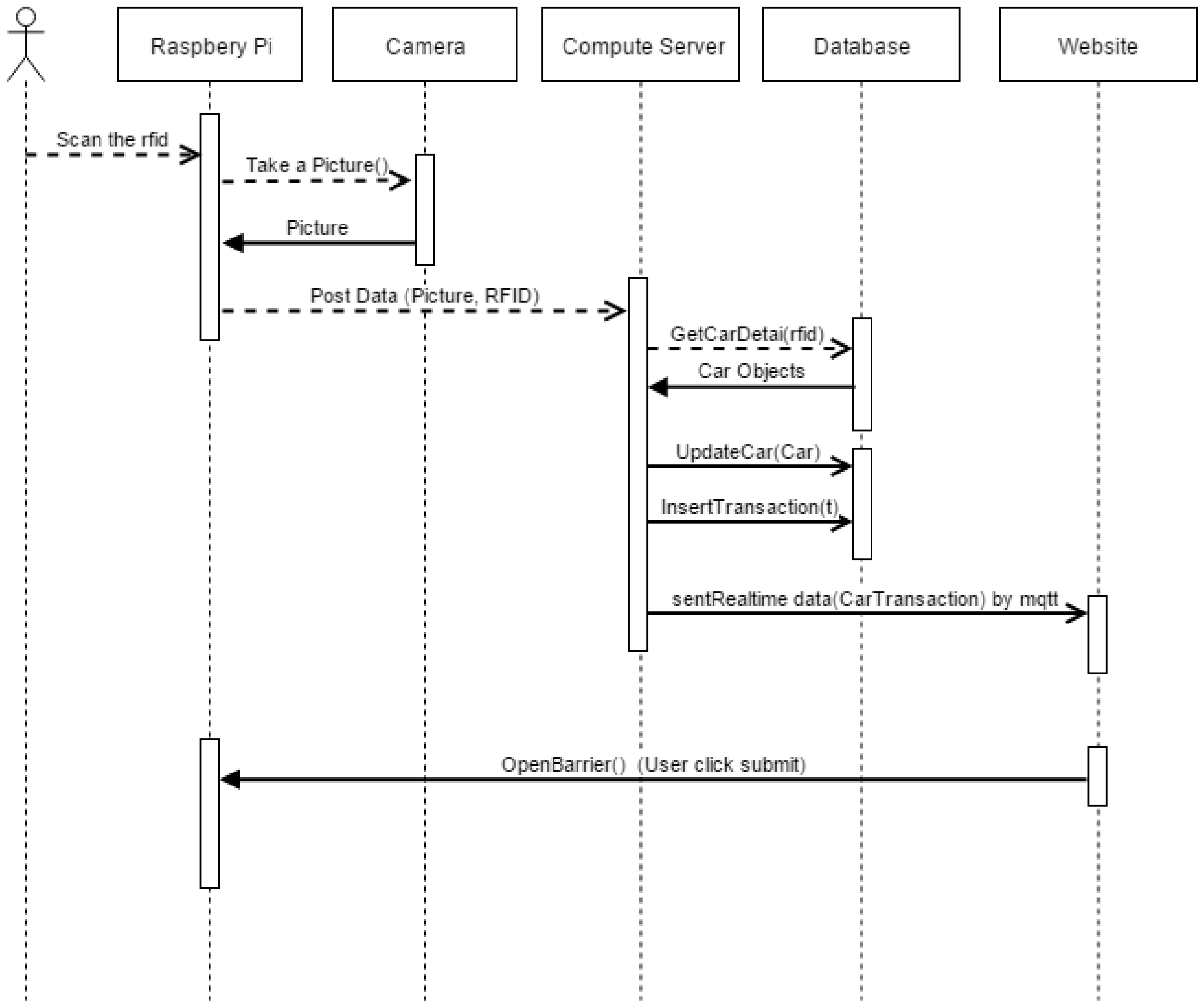


Figure 3.4.1 Sequence Diagram

ER Diagram

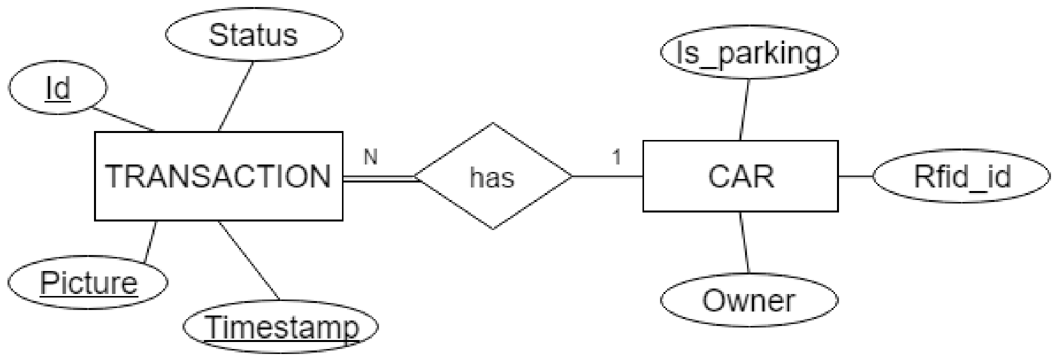


Figure 3.5.1 ER Diagram

Architecture Diagram

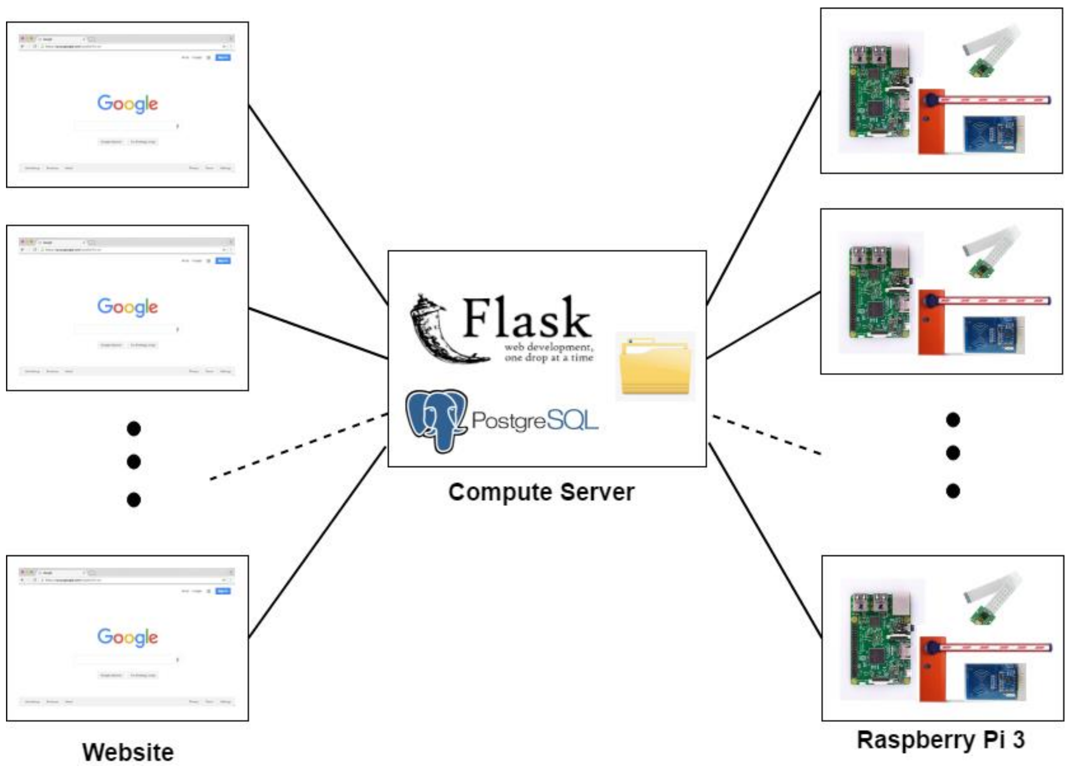
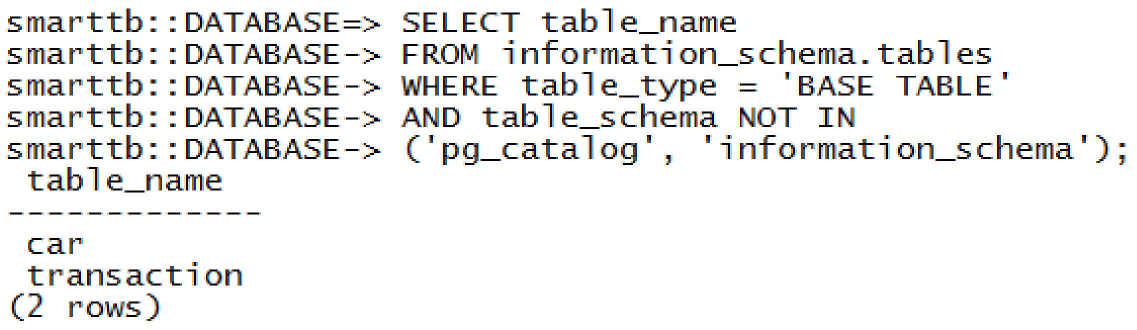


Figure 3.6.1 Architecture Diagram

Database Structure

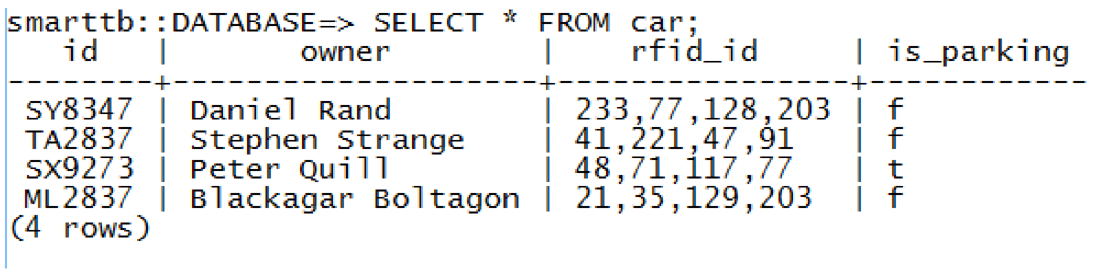
Database Table



```
smarttb::DATABASE=> SELECT table_name
smarttb::DATABASE-> FROM information_schema.tables
smarttb::DATABASE-> WHERE table_type = 'BASE TABLE'
smarttb::DATABASE-> AND table_schema NOT IN
smarttb::DATABASE-> ('pg_catalog', 'information_schema');
 table_name
-------------
 car
 transaction
(2 rows)
```

Figure 3.7.1 Database table

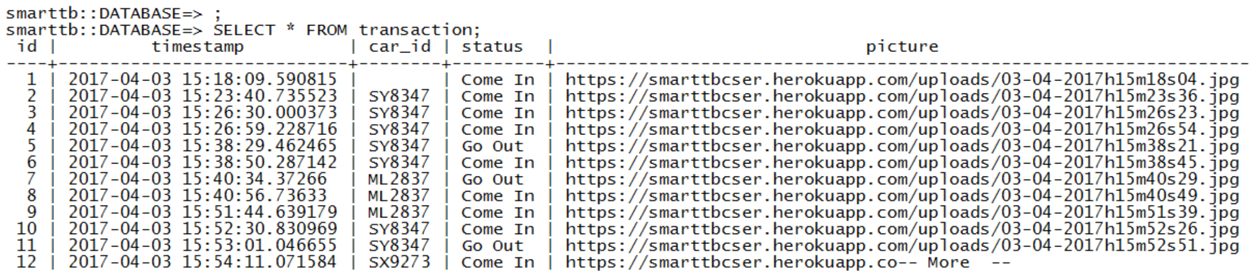
Table "Car" (with sample data)



```
smarttb::DATABASE=> SELECT * FROM car;
   id    |        owner         |     rfid_id      | is_parking
---------+----------------------+------------------+------------
 SY8347  | Daniel Rand          | 233,77,128,203   | f
 TA2837  | Stephen Strange      | 41,221,47,91     | f
 SX9273  | Peter Quill          | 48,71,117,77     | t
 ML2837  | Blackagar Boltagon   | 21,35,129,203    | f
(4 rows)
```

Figure 3.7.2 Car table

Table "Transaction" (with sample data)



```
smarttb::DATABASE=> ;
smarttb::DATABASE=> SELECT * FROM transaction;
 id |         timestamp          | car_id | status  |                              picture
----+----------------------------+--------+---------+--------------------------------------------------------------------
  1 | 2017-04-03 15:18:09.590815 |        | Come In | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m18s04.jpg
  2 | 2017-04-03 15:23:40.735523 | SY8347 | Come In | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m23s36.jpg
  3 | 2017-04-03 15:26:30.000373 | SY8347 | Come In | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m26s23.jpg
  4 | 2017-04-03 15:26:59.228716 | SY8347 | Come In | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m26s54.jpg
  5 | 2017-04-03 15:38:29.462465 | SY8347 | Go Out  | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m38s21.jpg
  6 | 2017-04-03 15:38:50.287142 | SY8347 | Come In | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m38s45.jpg
  7 | 2017-04-03 15:40:34.37266  | ML2837 | Go Out  | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m40s29.jpg
  8 | 2017-04-03 15:40:56.73633  | ML2837 | Come In | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m40s49.jpg
  9 | 2017-04-03 15:51:44.639179 | ML2837 | Come In | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m51s39.jpg
 10 | 2017-04-03 15:52:30.830969 | SY8347 | Come In | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m52s26.jpg
 11 | 2017-04-03 15:53:01.046655 | SY8347 | Go Out  | https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m52s51.jpg
 12 | 2017-04-03 15:54:11.071584 | SX9273 | Come In | https://smarttbcser.herokuapp.co-- More  --
```

Figure 3.7.3 Transaction table

System Integration and Test

For the system integration strategy and testing strategy we'll separate the system into a small module and testing each module (Unit test). And if every module is working correctly we'll Integrate it into big project system (which is our whole project)

The Module that connect with raspberry pi (RFID, Car barrier, Camera) we will develop it separately (create function for each work e.g., "OpenBarrier()", "takePicture()") and if it all working correctly we'll integrate it into the bigger module called 'RaspberryPi' (which is the directory in the same name in our Git repository).

Module website on everywhere either cloud or local (internet connection is required)

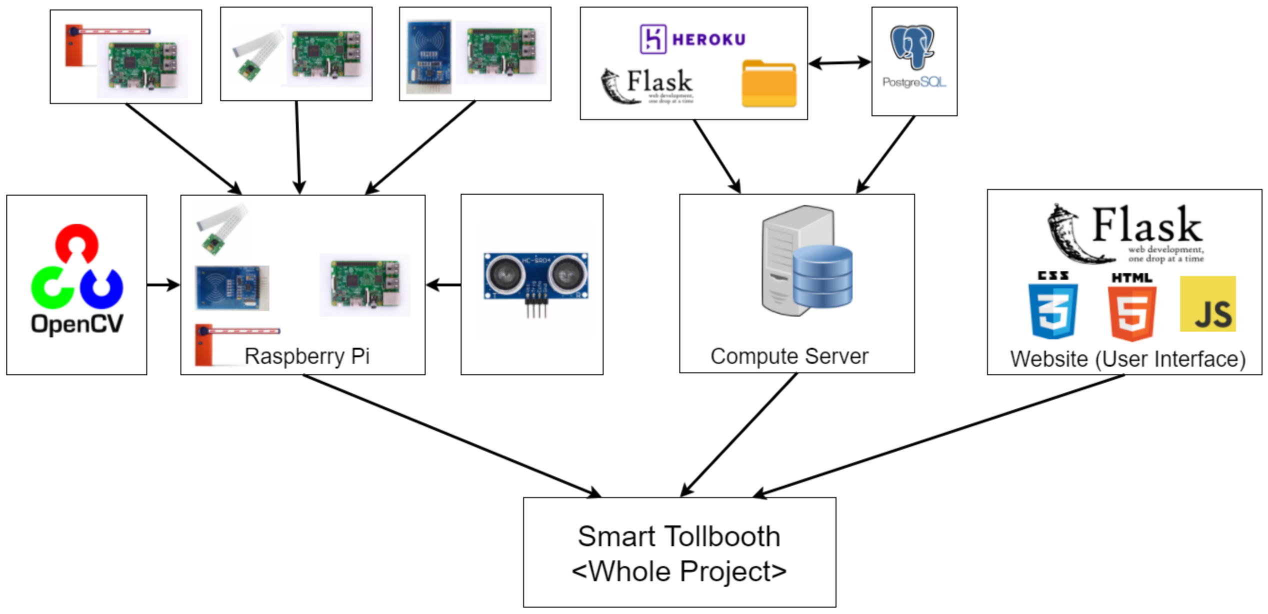
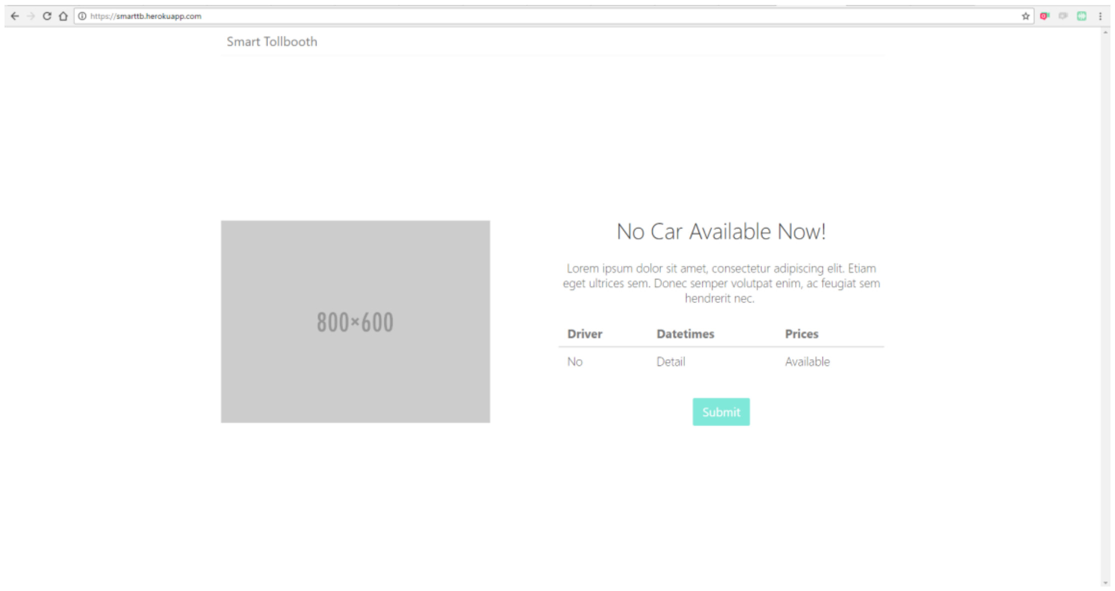


Figure 4.1 System Integration Strategy

User Interface

Main Page



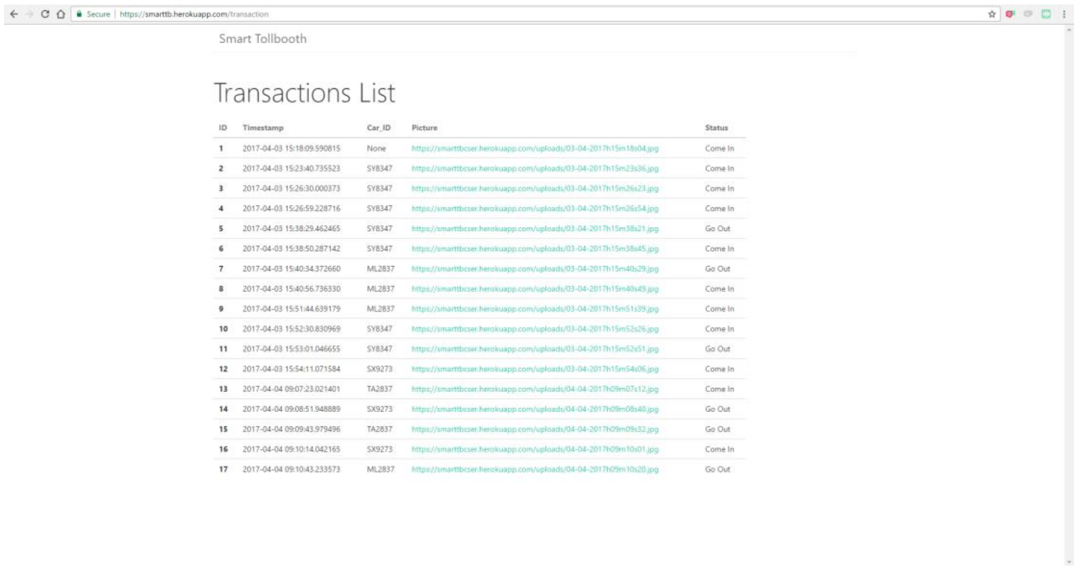
No Car Available Now!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eget ultrices sem. Donec semper volutpat enim, ac feugiat sem hendrerit nec.

Driver	Datetimes	Prices
No	Detail	Available

Figure 5.1 Main page

Transaction Page

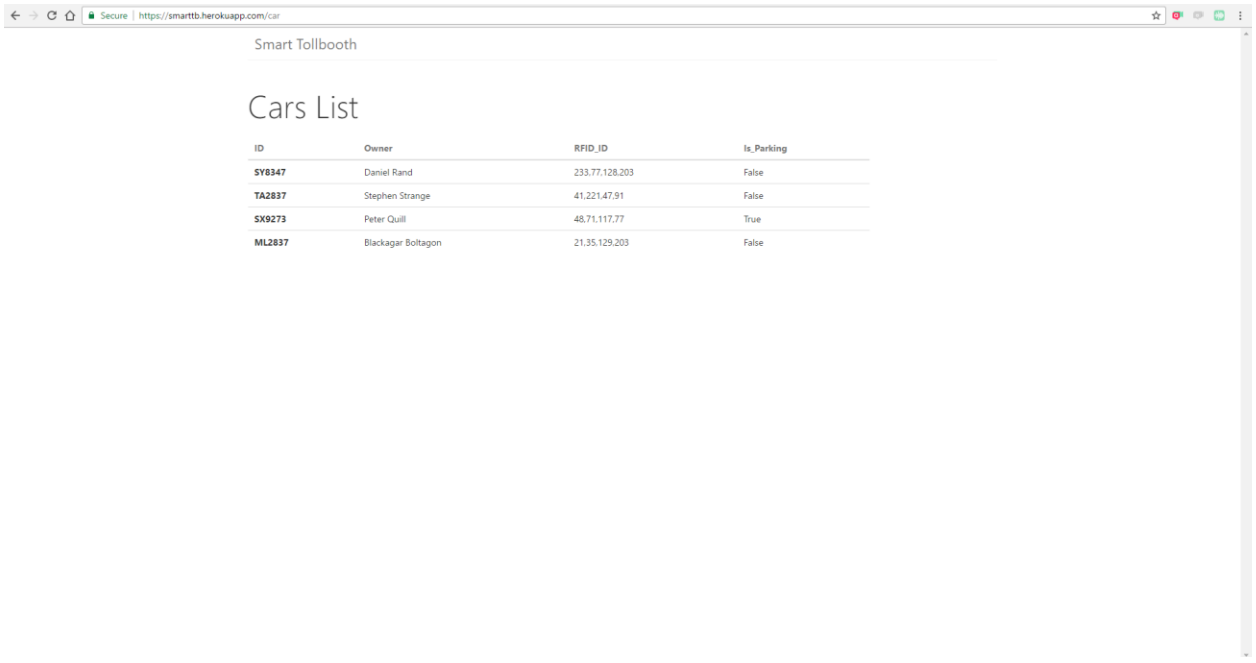


Transactions List

ID	Timestamp	Car_ID	Picture	Status
1	2017-04-03 15:18:09.590815	None	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m18s04.jpg	Come In
2	2017-04-03 15:23:40.735523	SY8347	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m23s36.jpg	Come In
3	2017-04-03 15:26:30.000373	SY8347	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m26s23.jpg	Come In
4	2017-04-03 15:26:59.228716	SY8347	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m26s54.jpg	Come In
5	2017-04-03 15:38:29.462465	SY8347	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m38s21.jpg	Go Out
6	2017-04-03 15:38:50.287142	SY8347	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m38s45.jpg	Come In
7	2017-04-03 15:40:34.372660	ML2837	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m40s29.jpg	Go Out
8	2017-04-03 15:40:56.736330	ML2837	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m40s49.jpg	Come In
9	2017-04-03 15:51:44.639179	ML2837	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m51s39.jpg	Come In
10	2017-04-03 15:52:30.830969	SY8347	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m52s26.jpg	Come In
11	2017-04-03 15:53:01.046655	SY8347	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m52s51.jpg	Go Out
12	2017-04-03 15:54:11.071584	SX9273	https://smarttbcser.herokuapp.com/uploads/03-04-2017h15m54s06.jpg	Come In
13	2017-04-04 09:07:23.021401	TA2837	https://smarttbcser.herokuapp.com/uploads/04-04-2017h09m07s12.jpg	Come In
14	2017-04-04 09:08:51.948889	SX9273	https://smarttbcser.herokuapp.com/uploads/04-04-2017h09m08s40.jpg	Go Out
15	2017-04-04 09:09:43.979496	TA2837	https://smarttbcser.herokuapp.com/uploads/04-04-2017h09m09s32.jpg	Go Out
16	2017-04-04 09:10:14.042165	SX9273	https://smarttbcser.herokuapp.com/uploads/04-04-2017h09m10s01.jpg	Come In
17	2017-04-04 09:10:43.233573	ML2837	https://smarttbcser.herokuapp.com/uploads/04-04-2017h09m10s20.jpg	Go Out

Figure 5.2 Transaction page

Car Page



Secure | https://smarttb.herokuapp.com/car

Smart Tollbooth

Cars List

ID	Owner	RFID_ID	Is_Parking
SY8347	Daniel Rand	233,77,128,203	False
TA2837	Stephen Strange	41,221,47,91	False
SX9273	Peter Quill	48,71,117,77	True
ML2837	Blackagar Boltagon	21,35,129,203	False

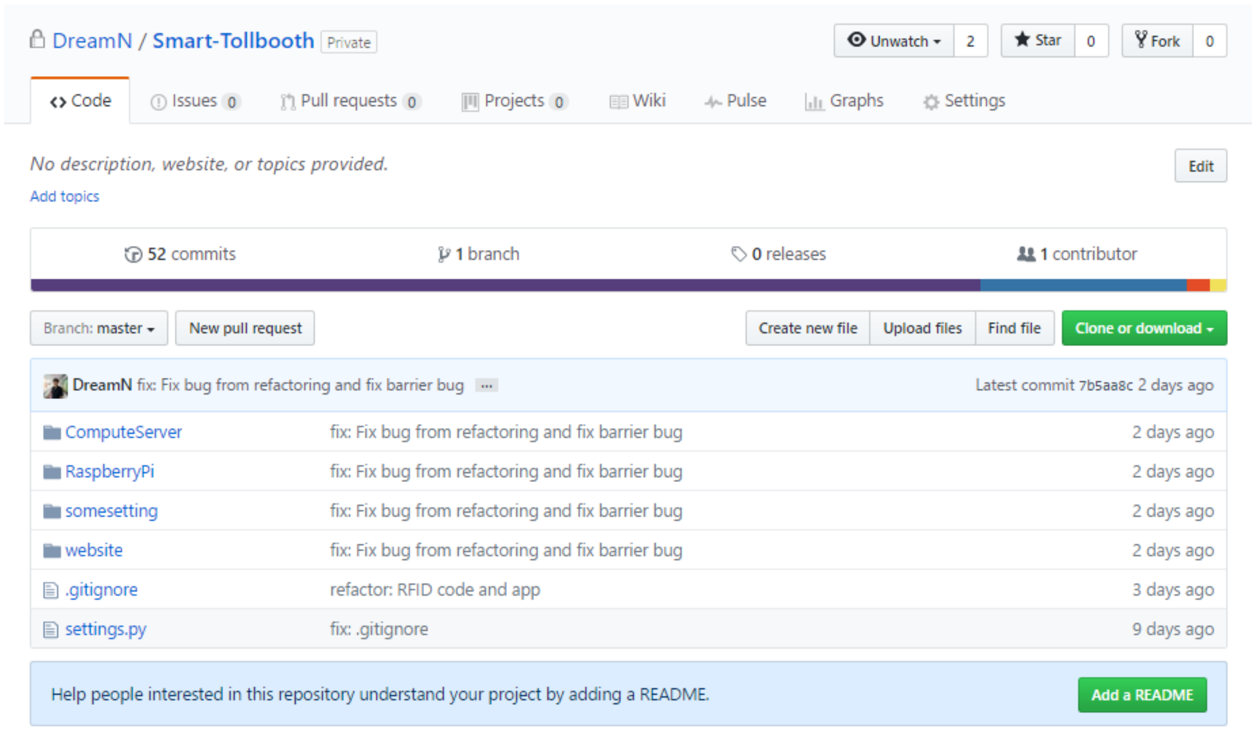
Figure 5.3 Car page

The User interface (Main, Transaction list, Car list) will be display in the web browser (HTML page) which is rendered from the web server ("website" directory in our repository (you can see our "Github Repository" on the next sections))

The main page will display real-time the coming cars's detail (included its photos) for the operator to validate the license plate. If the license plate is valid then click "Submit" to accept the car and the web will communicate with the Raspberry Pi 3 to open the barrier

For the "Transaction page" and "Car page" the web will display the list of transactions and cars in the table (the data is selected from database)

Project Repository


DreamN / Smart-Tollbooth Private

No description, website, or topics provided.

52 commits | 1 branch | 0 releases | 1 contributor

DreamN fix: Fix bug from refactoring and fix barrier bug ... Latest commit 7b5aa8c 2 days ago

ComputeServer	fix: Fix bug from refactoring and fix barrier bug	2 days ago
RaspberryPi	fix: Fix bug from refactoring and fix barrier bug	2 days ago
somesetting	fix: Fix bug from refactoring and fix barrier bug	2 days ago
website	fix: Fix bug from refactoring and fix barrier bug	2 days ago
.gitignore	refactor: RFID code and app	3 days ago
settings.py	fix: .gitignore	9 days ago

Help people interested in this repository understand your project by adding a README.

Figure 6.1 Project's Github Repository

Project Repository (Using GIT Version Control) is available at https://github.com/DreamN/Smart-Tollbooth under "the MIT License"

*For using or editing in another project "Giving credit must be required"

References

[1] Raspberry Pi Foundation, 'Camera Module', 2017. [Online]. Available: https://www.raspberrypi.org/documentation/usage/camera/ [Accessed: 28- Feb- 2017]

[2] Thaieasyelec, 'How to use RC522 ReadWrite Module', http://www.thaieasyelec.com/downloads/EFDV558/How_to_use_RC522_ReadWrite_Module_(SPI)_with_Raspberry_Pi.pdf [Accessed: 28- Feb- 2017]

[3] Gaven MacDonald, 'Ultrasonic Sensor with the Raspberry Pi', 2013. [Online]. Available: https://www.youtube.com/watch?v=xACy8l3LsXI [Accessed april. 8, 2017]

[4] Chris Dahms, 'License_Plate_Recognition_Python', 2016. [Online]. Available: https://github.com/MicrocontrollersAndMore/OpenCV_3_License_Plate_Recognition_Python [Accessed april. 6, 2017]