Version: 1.0.0

Created: 10/24/2019

LAST UPDATED: 10/26/2019

Car Maintenance Calendar

**CMC**

Presented by: Michael Mobilio

LUWAI WHEAGAR

MOHAMED SANGARE

KRUSHNA PATEL

Contents

[**README FILE** 2](#_Toc23185677)

[***DESCRIPTION*** 2](#_Toc23185678)

[***PREREQUISITES*** 2](#_Toc23185679)

[**Team Management tools** 2](#_Toc23185680)

[**Require Softwares** 2](#_Toc23185681)

[**Require Operating System** 2](#_Toc23185682)

[**Require Physical Network** 2](#_Toc23185683)

[**Require APIs’** 2](#_Toc23185684)

[***INSTALLATION SETUP*** 2](#_Toc23185685)

[***USER GUIDE*** 3](#_Toc23185686)

[***CONTRIBUTING*** 3](#_Toc23185687)

[***CITATION*** 3](#_Toc23185688)

[***CONTACT*** 3](#_Toc23185689)

[**API DOCUMENTATION** 4](#_Toc23185690)

[**CODE DOCUMENTATION** 5](#_Toc23185691)

[**TEAM ROLES AND RESPONSIBILITIES** 9](#_Toc23185692)

# **README FILE**

## ***DESCRIPTION***

Our project is on "Car Maintenance Calendar." CMC is created for the user to get their next schedule reminder of maintenance services for their cars. For instance, CMC will tell them when their next oil change, oil filter, air filter, fuel filter replacement is going to be, and so forth. Also, there will be a YouTube playlist provided according to their maintenance services for them to see how to fix of their own if they desire.

## ***PREREQUISITES***

### **Team Management tools**

1. Trello ([**click here**](https://trello.com/?&aceid=&adposition=1t3&adgroup=77960762125&campaign=1407850971&creative=389415002626&device=c&keyword=trello&matchtype=e&network=g&placement=&ds_kids=p48959053108&ds_e=GOOGLE&ds_eid=700000001557344&ds_e1=GOOGLE&gclid=CjwKCAjw3c_tBRA4EiwAICs8CgOHhOKXAC5wYaJQ8W0o4hHm89ySLckfomWFGqrt-Eg_eJ6UT5eVZRoCtt0QAvD_BwE&gclsrc=aw.ds))
2. Discord ([**click here**](https://discordapp.com/))
3. GitHub ([**click here**](https://github.com))

### **Require Softwares**

1. Virtual Box ([**click here**](https://www.virtualbox.org/wiki/Downloads))
2. RabbitMQ

### **Require Operating System**

Ubuntu 18.04.3 LTS ([**click here**](https://ubuntu.com/download/desktop))

### **Require Physical Network**

1. One Physical Router with four Ethernet ports
2. Four Ethernet cables

### **Require APIs’**

1. Google Calendar API <https://developers.google.com/calendar>
2. YouTube API <https://developers.google.com/youtube>
3. CarMD API <https://api.carmd.com/member/docs>

## ***INSTALLATION SETUP***

1. Install Virtual Box
2. Download Ubuntu image (while 1 & 2 downloading do 3 to 5)
3. Install & setup Trello
4. Install & setup Discord
5. Install & setup GitHub
6. Setup Ubuntu image on Virtual box ([**click here**](https://www.youtube.com/watch?v=QbmRXJJKsvs) for how)
7. Install RabbitMQ on Ubuntu Desktop

* Only one of the team members require to do this step whomever later going to handle role of RabbitMQ; however, it is good practice if all the team do this.
* Open terminal on Ubuntu and do the below commands:
  + $sudo apt-get update (enter)
  + $sudo apt-get upgrade (enter)
  + $sudo apt-get install erlang (enter)
  + $sudo apt-get install rabbitmq-server (enter)
  + $sudo systemctl enable rabbitmq-server (enter)
  + $sudo systemctl start rabbitmq-server (enter)
  + $sudo systemctl status rabbitmq-server (enter)
  + $sudo rabbitmq-plugins enable rabbitmq\_management (enter)
  + $sudo rabbitmq add\_user admin admin (enter)
  + $sudo rabbitmq set\_user\_tags administrator (enter)
  + $sudo rabbitmq set\_permissions -p / admin “ .\*” “ .\*” “ .\*” (enter)
  + Open web browser on Ubuntu and search (put your ubuntu ipv4 add) 192.168.7.153: 15672 (enter)
  + Login with username admin and pass admin or guest (enter)
* Now time to test if everything is working so do “Hello World”
* Open terminal on Ubuntu and do the below commands:
  + $sudo python3 -m pip install pika –upgrade (enter)
  + Now you create two .py files send.py and reciev.py ([**click here**](https://www.rabbitmq.com/tutorials/tutorial-one-python.html) for the code)
  + Open another terminal and run both files in separate terminal side by side
  + $python3 send.py (enter) (same goes for another file on other terminal)

## ***USER GUIDE***

1. User Registration page
2. User Login page
3. User Database Entry
4. User Car Maintenance Calendar Data

## ***CONTRIBUTING***

Issue Tracker: github.com/project/issues

## ***CITATION***

Main register website link goes here

## ***CONTACT***

[Kp592@njit.edu](mailto:Kp592@njit.edu)

# **API DOCUMENTATION**

# **CODE DOCUMENTATION**

Project Tree:

-it490-car-calendar

-broker

-README.md

-provision-broker.sh

-db

-data

-empty.txt

-logs

-empty.txt

-services

-auth-consumer.service

-data-consumer.service

-log-consumer.service

-sql

-0-users.sql

-1-cars.sql

-src

-database

-\_\_init\_\_.py

-auth.py

-cars.py

-db.py

-users.py

-example\_senders

-auth\_sender\_rpc.py

-dmz\_sender\_rpc.py

-log\_sender.py

-.env

-\_\_init\_\_.py

-\_env

-consume\_auth.py

-consume\_data.py

-consume\_log.py

-requirements.txt

-README.md

-motd

-provision-db.sh

-dmz (The directory which contains all the files for the DMZ vm)

-logs (The directory for all the locally stored log files)

-empty.txt (Test file test log file)

-services (The directory for the service bash script to start receiving/sending data)

-dmz-consumer.service (The bash script which starts the the consume\_dmz.py process)

-src (The directory for the source code used to create functionality)

-.env (Bash script which sets the values for the environment variables)

-\_env (Bash script which sets the values for the environment variables)

-carmd.py (Python script which retrieves the car’s maintenance/recall information from the CarMD API)

-consume\_dmz.py (The main python script which listens for actions from Rabbitmq, then runs functions based upon the received action)

-email.py (Python module script to send emails)

-google\_calendar.py (Python module script to gain access to Google Calendar, and schedule events)

-requirements.txt (Text file containing the dependencies to run the consum\_dmz.py python script)

-youtube.py (Python module script to retrieve YouTube playlist)

-README.md (Description of what the DMZ does)

-provision-dmz.sh (Bash script to configure the DMZ vm upon creation)

-packages (The directory for the custom Python classes)

-amqp (The directory for the custom Python consumer classes)

-\_\_init\_\_.py (Python script to initialize amqp package modules)

-consumer.py (Python class to abstract the process of consuming Rabbitmq data)

-ez.py (Python module to consume/produce to Rabbitmq)

-producer.py (Python class to abstract the process of producing Rabbitmq data)

-setup.py (Python script to setup package modules)

-logger (The directory for the custom Python logger classes)

-\_\_init\_\_.py (Python script to initialize amqp package modules)

-logger.py (Python class to generate and send log files)

-logger\_tests.py (Tester Python class for generating/sending log files)

-setup.py (Python script to setup package modules)

-web

-services

-gunicorn.service

-src

-public

-empyt.txt

-static

-sunflower.jpg

-templates

-auth

-login.html

-register.html

-cars

-create.html

-display.html

-display\_add\_events.html

-display\_maintenance.html

-display\_recalls.html

-display\_video.html

-display\_videos.html

-list.html

-list\_item.html

-update.html

-base.html

-index.html

-utils

-dueshit.py

-views

-auth.py

-cars.py

-general.py

-.env

-\_env

-main.py

-producers.py

-requirements.txt

-run\_dev\_server

-run\_prod\_server

-README.md

-motd

-nginx.conf

-provision-web.sh

-.gitignore

-README.md

-Vagrantfile

# **TEAM ROLES AND RESPONSIBILITIES**