COSC2674/2755 Semester 1, 2020
Assignment 2 Specification

Marks allocated:	This assignment will be marked out of 100 and is worth 35% of your overall mark
Deadline:	Sunday 17.05.2020 (11:59 pm AEST)
Submit via:	Canvas
Work mode:	In a group of 4 (*individual submissions discouraged*)
Submission format:	**.zip** (*No other formats will be accepted*)
Online demo:	Week 11 *Monday - Friday* (No Demo→ No marks)

0 READ THIS FIRST

The real-life projects that you will face in Industry never come with crystal clear, direct list of instructions in a linear manner. In fact, the reality is far from that, the project requirements often come in bits and pieces from often a confused client who thinks that they know everything.
It is the job of requirement engineers to elicit the requirements. Business Analysts then spend good amount of time clarifying these requirements and creating more sensible, doable and negotiable list of deliverables.

When you read the specifications for this assignment, you will realise that some of the ones may have multiple ways of implementing them (*just like in real life software development*). So instead of blaming it on the specifications, clarify the requirement(s) by posting in discussion board for assignment 2.

Do not start this assignment late, you have ***four weeks*** to complete it which is more than enough time to do well and make sure that you use this time judiciously. Starting work at the last minute will only lead to poor outcome(s).

There are certain specifications which will push you out of the comfort zone. This has been done on purpose. There are certain parts of the assignment where you will need to do **self-research as you will not find answers in lectures, tute/labs**.

If you do a good job of this assignment, you can choose to **add it as a part of portfolio for future employers**. You are being prepared for potential employability prospects.

1 Scenario

Your team has been contacted by a car share company to develop an automatic Car Share System. This system is used to book, find and unlock and lock a car. In addition, the customer can report some issues with the car to help the company to maintain the cars. You will create an application for *four* types of users: customer, company manager, engineers and system administrator.

In this assignment,
For this assignment, you will be making extensive use of the Google Calendar API (https://developers.google.com/calendar/v3/reference/) to work with your Raspberry Pi.
You will also be using Google Cloud IoT Platform (https://cloud.google.com/solutions/iot/).
Besides, you will be using Pushbullet API (https://docs.pushbullet.com/) to send notifications to mobile devices.

In summary, the implementation of this assignment involves the following components:

- Python documentation tools such as Sphinx
- Practice third party API
- Unit testing in Python
- Socket Programming
- Writing your own API using Python's microframework *Flask*
- AI features such as facial recognition, object detection and Voice detection
- Programming with Cloud databases and,
- Selected Software Engineering Project Management/Tools

2 Important

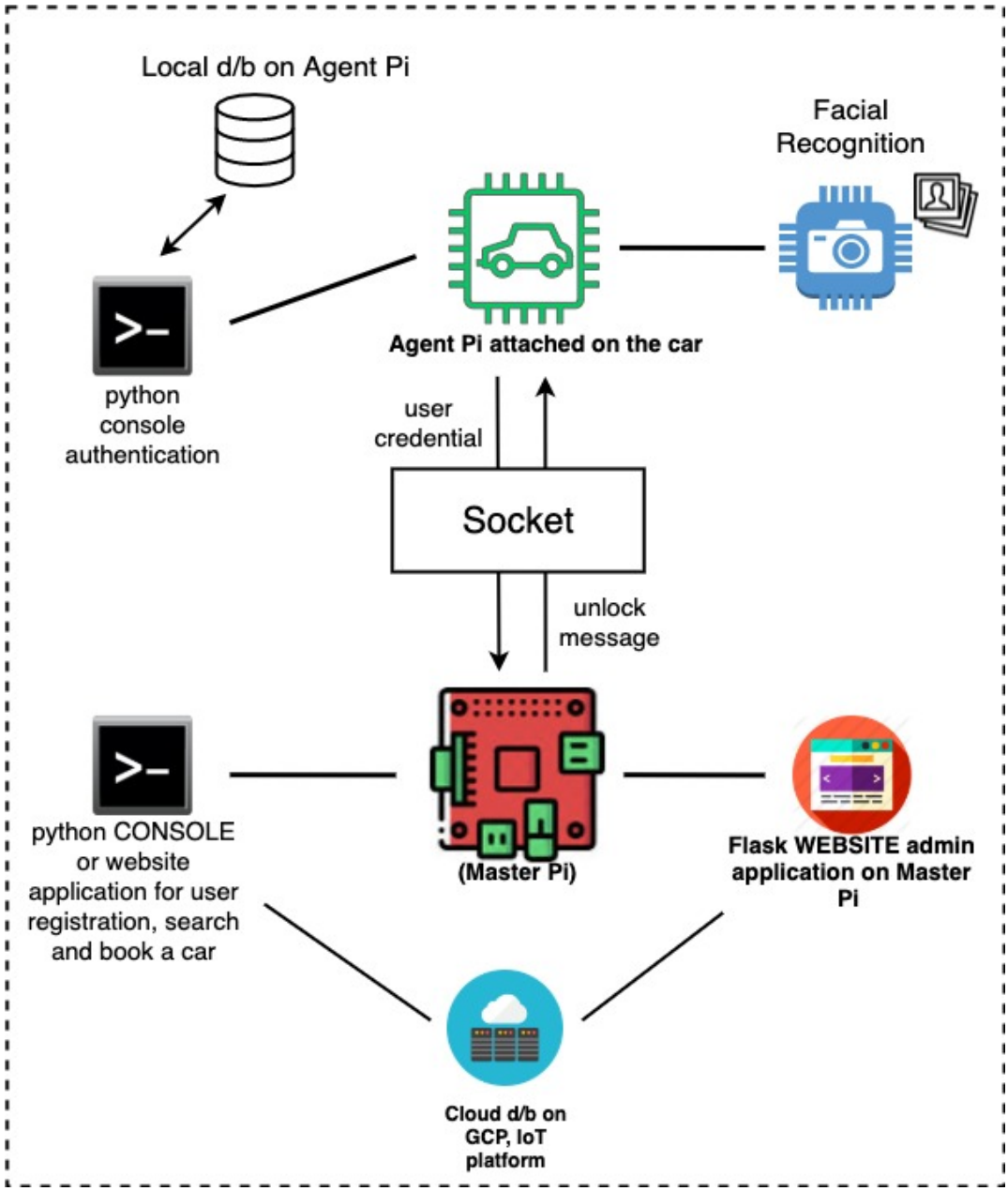
You must adhere to the following requirements:

a. Raspberry Pi model 4 or 3 should be used.
b. You must use Python 3.* to complete the tasks. Older versions must not be used.
c. You must use a version control system of some sorts such as *GitHub, Bitbucket,* etc. A private repository is to be used ONLY.
d. You must stick to the standard *style guide for your Python code:* (https://www.python.org/dev/peps/pep-0008/)
e. You must attend a **25 minutes** demo session to get the assignment 2 marked. A schedule and a booking document will be published later. You must submit the assignment prior to demo. No submission → No demo → No marks.

3 Detail(s) of CSS

3.1 High level architecture diagram

NOTE: You only need to implement user-side application in this assignment, the rest parts of the system will remain for the other assignment.



3.1 CSS features

NOTE: You need to design the database by dividing the information into tables and turning information items into columns. Future, you also need populate some data into the tables in order to illustrate the system during the demo session.

For customers: The customers can register, logging in, search and book a car on the web-based system in MASTER PI (MP).

The user registration on MP is required for the first-time user. In the home page of the web-based application provides only two options:

- registration
- log in

Upon registration the details are stored in cloud database. *You may use MySQL database*.

Upon logging in, the user is now presented with another page including following functions:

- show a list of cars available, you need to show the detailed information of cars in the list such as Make, Body Type, Colour, Seats, Location, Cost per hour.
- search for a car based on body type or other features.
- book a car based on car identity, the user will be asked to input booking details.
- cancel a booking
- logout

When the customer arrives at the car booked, the Agent PI (AP) provides two options available for unlocking the car:

- using console-based system which allows them to type in the user credentials or,
- using a facial recognition system

Upon logging in, the user's credential will be sent from AP to MP via sockets. At the same time, MP will check the credential and send the response message back to AP. The AP will execute the operation according to the message from MP. Once the operation (unlock the car) is successfully executed, the AP need to send a message to nearby the device via Bluetooth, simulating notification in your smart phone.

When the customer leaves the booked car, they can lock the car by choose one option:

- return the car

Once the "lock car" operation is executed, the AP need to send a message to nearby the device via Bluetooth.

When the customer enters and leaves the booked car, the Agent PI (AP) will send message to MP in order to change the availability of the car.

YOU ARE NOT ALLOWED TO USE ANY OTHER PLATFORM WITHOUT PRIOR PERMISSION.

4 Tasks

Part A (50 marks)

You will now implement either a web-based system or console menu-based system on MP and a console menu-based system on AP.

NOTE: The console version can get up to 80% of the full mark for Part A. Please do not be ambitious as the web-based system can be difficult and time consuming. Developing a console menu-based system will be easier and less time consumed.

1) (5 marks) An option to register a new user on MP. The username, password and other necessary details (first name, last name and email) must be stored on cloud database. The password must be stored in an **encrypted (you may hash and salt)** format.

2) (2 marks) An option to login into the system and gain access to the application on MP.

3) (5 marks) Develop and implement a robust input validation scheme.

4) (3 marks) Design database based on the requirements.

   Make sure that the database is normalized - *this means if you only have one table, you will lose marks.* The car database is stored on a cloud environment namely the Google's GCP IoT platform (Google Cloud Platform).

5) (5 marks) Create your own RESTful API to talk to the cloud database.

6) (15 marks) Once MP receives the login information, the system should display a menu which will allow the user to access the booking system:

   - *view the history of my booked cars*
   - *show all the available cars*
   - *search a car*
   - *book a car*
   - *cancel a booking*
   - *logout*

   YOU WILL RECEIVE ZERO for storing data in a local database.

   All the car related information is stored in a Cloud-based database hosted on Google Cloud IoT Platform (GCP). It is your responsibilities to make sure that you do not exceed the free tier limit on the GCP.

   View history:

   Be able to show a list of cars that current user has booked.

   Search a car:

   Be able to search by any of the car's properties and display fields neatly (e.g., column's aligned) in the console.

   Book a car:

   The uses need to select which car they want to book and input the duration they want

to use. A car that is booked cannot be booked again until returned. Note when booked an event should be added to **Google Calendar**, detailing the car, who book the car and the booked duration. *Google calendar will be tied to the Google login of the user*.

Cancel a booking:

The event added to Google Calendar will be removed, and the related information in the database need to be modified

7) (5 marks) Complete documentation using Sphinx

8) (10 marks) Professional use of

   - GitHub from day 1 of the development
   - Trello board for the development cycle

Part B (20 marks)

1) (3 marks) Implement Console-based system on AP, providing the functionality for user to unlock and return the car.

2) (5 marks) Sending user's credential from AP to MP via sockets. The message needs to include some important information, such as current date and time, username/password, car ID.

3) (2 marks) MP needs to check the credential from AP and also send back the response message via sockets. If the credential is correct, MP need to modify related information about the car in cloud database.

4) (5 marks) Being able to show all the car's location by using Google Map API.

5) (5 marks) Complete documentation using Sphinx, Trello board and Git Hub for this part

Part C (15 marks)

1) (12 marks) Implement the facial recognition requirement of the system using OpenCV. This applies to the customers who want to use facial recognition authentication instead of console-based authentication to unlock the car.

   Note: If all of your team members do not have a USB camera, below a solution for you. You can use the camera on your smart phone to capture the face photo and send the photos to the AP. Since the photo is saved in the AP already, you need to add a menu option that uses local photo to complete authentication in you console menu-based system. However, you need to demonstrate that the facial recognition system does work during the demo session.

2) (3 marks) Complete documentation using Sphinx, Trello board and Git Hub for this part.

Part D (15 marks): Unit Test

1) (15 marks) Complete unit test suite for the whole project (i.e., A, B, C and D parts). This is where you can decide what kind of unit tests are required.

5 Demo

In prior to Demo,

a. create 10-20 Cars on your own (e.g., 2 Admins, 4 Engineers, 6 Users)

6 Late submission and Extension

a. A penalty of 10% per day of the total marks will apply for each day late, including both weekend and weekdays.

b. After five days, you will receive a zero for the whole assignment.

c. Extension requests should only be emailed to the lecturer.

d. Extension offered to a group member(s) does not qualify for a global extension for the whole of group.

7 Plagiarism

All assignments will be checked with plagiarism-detection software; any student found to have plagiarised would be subject to disciplinary action. Plagiarism includes

- submitting work that is not your own or submitting text that is not your own
- allowing others to copy your work via email, printouts, social media etc.
- posting assignment questions (in full or partial) on external technical forums
- copying work from/of previous/current semester students
- sending or passing your work to your friends
- posting assignment questions on technical forums to get them solved
- someone else writing your code (*i.e., contract cheating*)

A disciplinary action can lead to

- a meeting with the disciplinary committee
- a score of zero for the assignment
- a permanent record of copying in your personal university records and/or
- expulsion from the university, in some severe cases

All plagiarism will be penalised. There are no exceptions and no excuses. You have been warned.