COSC2674 & COSC2755 (PIoT) Semester 2, 2023
Assignment 2 Specification

Marks allocated:	100 (*worth 45% of the total score*)
Deadline:	Sunday 08.10.2023 11:59 pm
Submit via:	Canvas
Work mode:	In a group of 4 (*individual submissions required*)
Submission format:	Provided Form "Assignment2_GroupEffort.pdf"

---

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 ***>5 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**.

1 Scenario

Your team has been contacted by a scooter share company to develop an automatic Scooter Share Platform. This system will be used to find, book, use, return, and maintain the backend information. You have been tasked create an application for *three* types of users: customer and admin and engineer.

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 Platform (https://cloud.google.com/).

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

- Python documentation tools such as Sphinx
- 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. Only 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.

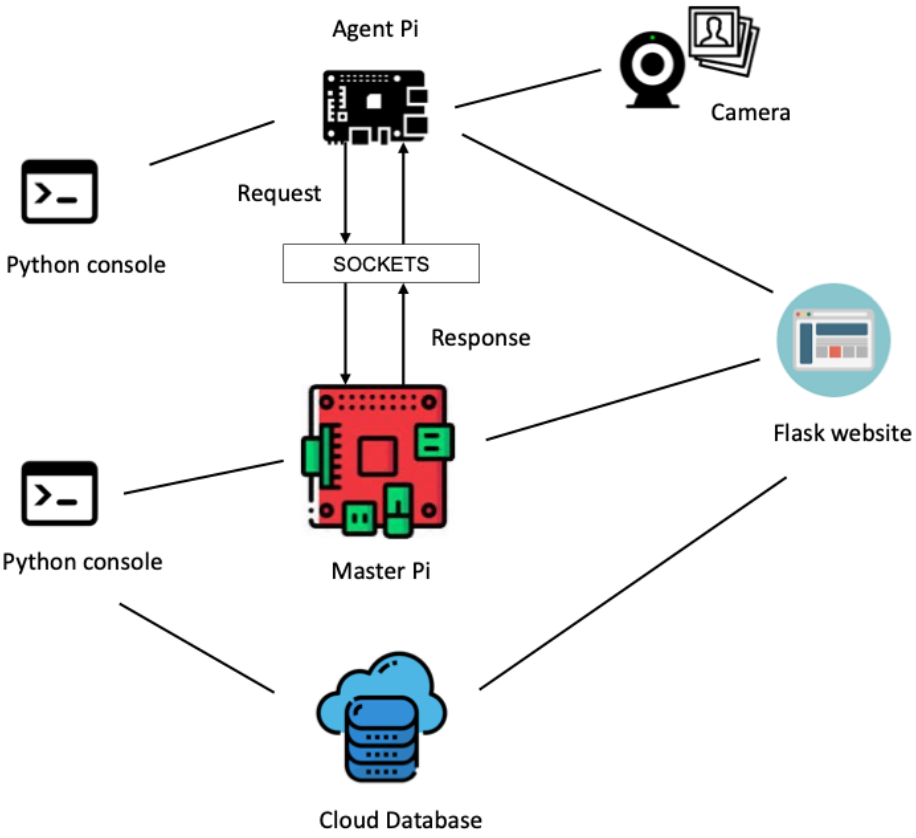
c. You must use a version control system i.e., *GitHub*. A private repository is to be used ONLY.

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 the **demo session** to get the assignment 2 marked during week 12 (maybe and week 13). 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 LMS

3.1 High level architecture diagram



3.1 LMS features

NOTE: You need to design the database by dividing the information into tables and turning information items into columns. You also need to populate some data into the tables to demonstrate you have completed the requirements for fair judgement from reviewers.

For customers: The customer can register, log in, search and book a scooter, and top up the account on the web-based system on Agent Pi (AP). The AP will execute all the operations of the customer by sending messages from AP to Master Pi (MP) and receiving responses from MP via sockets.

The customer registration is required for the first-time customer.

In the home page of the web-based application on AP 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 customer is now presented with another page including following functions:

- Show a list of scooters available, you need to show the detailed information of scooters in the list such as Make, Colour, Location, Remaining Power, Cost Per Time

- Show available money
- View scooter usage history
- Book a scooter
- Cancel a book
- Report scooter issues
- Top up
- Logout

The business rules for these are described along with the requirements. You can identify the cost by yourself.

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

- 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.

When the customer unlocks or locks the scooter by scanning QR code, the Agent PI (AP) will send message to MP in order to change the availability of the scooter. Once the customer locks a scooter, the corresponding bill will be generated, and the customer will be charged. The booked scooter cannot be unlocked by other customers. The engineer can unlock any scooter and not be charged.

For system admin: Admin can only login to the website on Master Pi (MP) to maintain the data of customers and scooters.

The system admin is able to

- View scooter usage history
- View scooter book history
- View the status of all the scooters
- View customer top-up history
- Add, remove and modify information of customers and scooters
- Report scooter issues
- Generate a data visualization report for scooter usage (day and week wise)

The database is stored on a cloud environment namely the Google's GCP platform (*Google Cloud Platform*).

For engineer: The engineers take responsibility to repair the reported scooters. They will receive a notification via email once admin reported a scooter that is needed to be repaired. After that, the engineer can login to the system on their own AP to check out and find the scooter's location.

The engineer is able to:

- Login

- Check the reported scooter's location in the web page
- Report the repair result

The AP attached in the scooter will have the ability to detect QR code carried by engineers in order to retrieve their personal information, which will help company to know who has done what.

YOU ARE NOT ALLOWED TO USE ANY OTHER PLATFORM WITHOUT PRIOR PERMISSION FROM THE LECTURER/HEAD TUTOR.

4 Tasks

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

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

Note: If all of your team members do not have a USB camera, there is a solution for you, you can use the QR code picture saved in local file system to test the system performance.

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

(5 marks) An option to login to the system for three types of users including admin, engineer and customer. They will be directed to different web pages regarding to their user type. Note: Both username and password of admin and engineer must be stored in an encrypted (you may hash and salt) format in advance. In addition, the login of the admin can only happen on MP, while the login of the customer and engineer can only happen on AP.

(5 marks) Design database and populate some data based on the requirements.

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

(5 marks) Create your own RESTful API to communicate with the cloud database.
Note: All database operations related to data retrieve and update must utilise API to talk to the cloud database.

(15 marks) Once the customer successfully logs in on AP, AP should display a menu which will allow the customer to access the system. The main page should display the available money, the operation buttons, and a list of available scooters with necessary information such as Make, Colour, Location, Remaining Power, Cost Per Time. The operations include:

- View scooter usage history: show a list of scooters that current customer has used
- Book a scooter: the customer needs to select the scooter to book. A scooter that is booked and used cannot be booked again until returned. Note the book event should be added to Google Calendar, detailing the scooter information (the default duration is 10 min). *Google calendar will be tied to the Google login of the user.*

- Cancel a book: the event added to Google Calendar will be removed and the information in the database need to be modified (instead of deleting the record, the status should be changed to 'canceled').
- Top up: input the money to top up the account. This will be updated in the database.
- Unlock a scooter: scan the unlock QR code on the scooter, unlock it, and send the message to MP.
- Report scooter issues: report scooters with issues
- Return/Lock a scooter: scan the lock QR code on the scooter, only the customer who unlock this scooter can return this scooter. The account should be charged accordingly.
- Logout: after logout, the customer needs to log in again to access the system.

Please note: the customer can only unlock or book the scooter with enough money in the account.

(15 marks) Once the admin logs in successfully on MP, the system should display a menu which will allow the admin to manipulate the data regarding to customers and scooters. The operations include:

- View scooter usage history: show a list of scooter usage history
- View scooter booking history: show a list of scooter booking history
- View scooters: show a table of all the scooters with information including make, colour, location, remaining power, cost per time and status (such as Available, Occupying, To Be Repaired, Under repair)
- View customer information: first name, last name, phone, email and available money
- Add, remove and modify information of customers and scooters: be able to edit information of customers or scooters
- Report scooter issues: select and report scooters with issue in the scooter view
- Generate a data visualization report for scooter usage: generate the image based on the information in database. You can design it by yourself.

(5 marks) Once the engineer logs in on AP, the system should display a menu which allow the engineer to do the following operations:

- Check the reported scooter's location in the web page: show a list of reported scooters with enough details, including id, location, make, colour, power, etc.
- Unlock a scooter: scan the unlock QR code on the scooter, unlock the scooter and send message to MP.
- Return/Lock a scooter: scan the lock QR code on the scooter, lock the scooter and send message to MP.
- Report the repair result: once the engineer finishes the repair, the engineer can click the button and the message will be sent to MP.

(2 marks) AP on the scooter should display its correct status on Sense HAT, such as Available, Booked, Occupying, To Be Repaired, Under repair.

(2 marks) Send an email to engineer's email address when a scooter with issue is reported by admin.

(2 marks) Once the scooter is reported, the location of the scooter needs to show in Google Map of engineer's page, which will guide engineer to find the scooter.

(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 scooter in cloud database.

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

(10 marks) Complete unit test suite for the whole project. This is where you can decide what kind of unit tests are required.

(10 marks) Professional use of

- Documentation
- GitHub from day 1 of the development
- Trello board (or other similar apps) for the development cycle

(15 marks) Enhancements: design two your own features in this system. Those features should not be mentioned in the above requirements. Specifically, the novelty/uniqueness (5 marks), usefulness (5 marks) and complexity (5 marks) will be considered.

5 Adding a README file

In order to get fair marks, it is your responsibility to make sure that all your effort will be fairly judged. You must have a README.md file written in markdown format in your project root path, which will guide reviewer go through your project smoothly.

- Adding a README file to introduce your project.
- Write an instruction of your project in the README.md.
- Username and password of 3 types of user are included in the README.md.
- Report your repository usage in GitHub by attaching some screen shots in README.md.
- Report your Trello board usage by attaching some screen shots in README.md.
- Well organised file structure and also separate website and API code into two folders.
- Well explained enhancement feature(s)
- Make sure having one main method to run in main.py.

The last but most important thing is to make sure that your program is runnable. Besides, you need to have a requirement.txt file in your project to show which pre-request package are needed to install. You can browse GitHub to get some idea what requirement.txt is and how it can be used.

6 Who does what

Please include the evidence of who does what in this assignment, i.e., a file explaining what specific task each team member did. The evidence needs to be consistent to your GitHub and Trello board. If not consistent, your maker may deduce marks. Although each member did different tasks, all the members should know each other's tasks.

7 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 (xiaoyu.xia@rmit.edu.au)

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

8 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 (*ie 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.

ONCE AGAIN:

Do not start this assignment late. 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**.