|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **vTask:** | | **3** | | |
| **Task Title:** | | **Project** | | |
| **Task Code:** | | **AT3 PRJ Project** | | |
|  | |  | | |
| Assessment type (): | | | | |
|  | Questioning (Oral/Written) | |  | Portfolio |
|  | Practical Demonstration | |  | Project |
|  | 3rd Party Report | |  | Other – Please Specify |

|  |
| --- |
| The base requirements this assessment task include:   * Web server, Python interpreter and database server * IDE or editor for developing Python programs (only PyCharm supported by the college) * Raspberry Pi with SenseHat * Access to Office 365 & Microsoft Word * GitHub Repository:  **https://github.com/NM-TAFE/civ-ipriot-proj-carpark**   Use of some of these items may not occur in this part of the assessment task. |
| Assessment Due This assessment is due:   * Week 18   Refer to Blackboard for most accurate date, which may alter due to unforeseen circumstances.  We also will endeavour to update these document(s) at the same time. |
| Instructions Follow the steps listed in this assessment item.  Submission of the documentation, code, and associated items is at the end of each part of the portfolio.  It is advantageous to you to attempt to meet the deadline provided. |
| Important If you are using a different configuration of tools and equipment for this assessment item, then assistance in this and subsequent parts of the portfolio to ensure the systems work correctly will be limited. |
| Scenario / background In this Project, you will be writing a combination of two or more Python programs that work together. The scenario is provided in the GitHub repository associated with this project.  You’ll need to follow the simple requirements outlined in the repository (docs/). You must follow OO (object-oriented) principles, and your solution must be implemented as a set of OO programs. You must also follow the guidelines in PEP 8.  More details can be found in the repository. |
| General Instructions If we provide a document template for your answers:  Save the file as:   * XXX-IoT-Prj.docx   Replacing the XXX with your initials.  For example, Adrian Gould would use AG-IoT-Prj.docx for his submitted filename.  Upload any code as a PyCharm project in a zip-file. Include your .git/ folder. Remove the virtual environment (**venv** or **.venv**) from the zip-file before uploading it to Blackboard. |
| Sources of Information In industry, it is good practice to keep track of where information was obtained. This is especially true if it is a written document, or even code.  If you answer any questions using information from web sites, please include the site name and URL (Web site address) after the answer. Likewise, include the title and author for books and magazine articles. For example:   * RS Electronics Ltd: <https://au.rs-online.com/> * Slack API Documentation, Users List Method: <https://api.slack.com/methods/users.list>   If you use ChatGPT or any other LLM/AI you must appropriately reference any conversations. The code you write must be original to you. **Copying code or text from an LLM without attribution is plagiarism.** Version Control You must use git and GitHub to complete this project. Refer to the project repository for details. You must show evidence of performing a logical commit on at least three separate instances (your .git file is sufficient in this regard) |

# WORKING WITH GIT/GITHUB

You must ensure that your git workflow (at a minimum consists of the following steps)

1. Forking of the original project repository
2. At least 3 local commits and 3 remote commits. Commits must include a reasonable commit message
3. Evidence of working in a feature branch (the project can be a feature branch) and merging the feature branch.
4. Both your origin and your local copy are synchronized at the time of submissions

Evidence for the above will be collected via access to your .git file and your remote repository. Ensure your lecturer is able to access the remote.

# EVIDENCE LOG

You must provide evidence of meeting user requirements, debugging, and clarifying requirements (each on at least one occasion).

Please complete this worksheet to provide evidence of your work on the Python IoT Carpark Application Project. Ensure that you include specific instances of meeting use cases, debugging, and clarifying requirements.

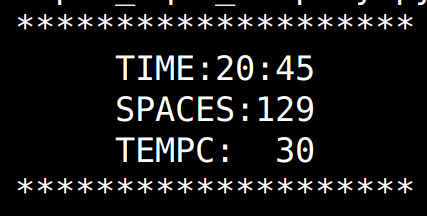
**Section 1: Meeting User Requirements**

| **Date** | **User Story** | **Description of Steps Followed** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

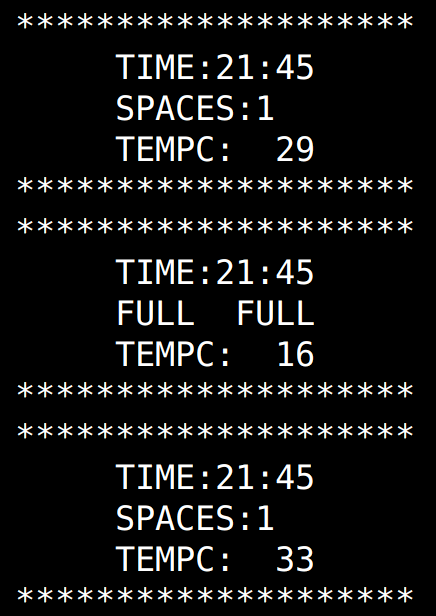
**Instructions**: On at least one occasion, describe how you met specific user stories for the Python IoT carpark application. Include the date and a detailed explanation of the steps you followed.

**### User Story 1**

As a driver, I want to know the number of available parking spots in a parking lot so that I can quickly find a spot to park my car.



**The driver can see from the display there is 129 spaces**



**The driver see there is one space,**

**then the car park is full, then a free spot**

**Section 2: Debugging Process**

| **Date** | **Debugging Tool Used** | **Description of Debugging Process (Issue Encountered, Steps to Resolve)** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

**Instructions**: On at least one occasion, describe your debugging process. Include the date, the debugging tool you used, the issue encountered, and the steps you took to resolve the issue.

**Section 3: Clarifying Requirements**

| **Date** | **Requirement in Question** | **Description of Steps Followed (Clarification Process, Actions Taken Based on Clarification)** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

**Instructions**: On at least one occasion, describe how you clarified user requirements. Include the date, the specific requirement in question, and a detailed explanation of the steps you followed to clarify the requirement and any actions taken based on the clarification.

# Appendix A: Code Style Guidelines

Your code will follow the PEP 8 standard.

Readability Counts  
- Zen of Python

Explicit is better than implicit.  
- Zen of Python

Other code standards available in the Presentation, “Python Coding Standards for North Metropolitan TAFE”.

# APPENDING B – MARKING

Please note the following is for reference purposes only and the actual documentation will be completed on blackboard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assessment # and title | | Assessment 3: Project | | | |
| **Lecturer name** | |  | | | |
| **Student name** | |  | | | |
| **Student ID number** | |  | | | |
| **Telephone contact number** | |  | **Email:** | | |
| By completing and submitting this signed form to my lecturer, I am stating that:   1. The attached submission is completely my own work. 2. I have correctly cited all sources of information used in this work (if required) 3. I have kept a copy of this assessment (where practicable) 4. I understand a copy of my assessment will be kept by the NMTAFE for their records. 5. I understand my assessment may be selected for use in the NMTAFE’s validation and audit process to ensure student assessment meets requirements | | | | | |
| **Student Signature** |  | | | **Date** |  |

**Assessor please note: Where verbal clarification has been sought from a student to gather additional assessment evidence from an assessment item, question/s and response/s must be recorded, signed, and dated by the assessor, against the relevant assessment item/s.**

**Submission 3**  **Assessment Result Satisfactory / Not Yet Satisfactory** *(please circle)* **Date: \_\_\_\_\_\_\_\_\_**

*To satisfy requirements for this assessment, you need to complete the following:*

|  |  |  |
| --- | --- | --- |
| **No.** | **Complete the task by following these steps** | **D or NYD** |
| 1 | Submitted working code consisting of at least two classes in two separate files |  |
| 2 | Submitted at least two documented\* test cases that run and pass  \* Documented: verbose unit test method names, docstring. Link to either the technical or user requirements is apparent. |  |
| 3 | Uses PEP8 in code, particularly adheres to all cases highlighted in the technical specification |  |
| 4 | Reads configuration from a text file |  |
| 5 | Writes to a file on at least one occasion |  |
| 6 | Documents application using docstrings and comments as appropriate |  |
| 6 | Includes .git folder with source |  |
| 7 | Git repo shows evidence of origin (their fork) and upstream (repository forked from) |  |
| 8 | Shows at least three commits with meaningful messages |  |
| 9 | Show evidence of branching and merging |  |
| 10 | Provides evidence of meeting requirements, consulting with stakeholders, and debugging |  |

**Student Feedback**

|  |
| --- |
|  |

Lecturer Signature: \_\_\_\_\_\_\_\_\_ Student Signature: