**Project Technical Report**

*Name: Corey Lynch*

*Student ID: R00154863*

*Class: SD2-A*

*Lecturer: Byron Treacy*



The **objective** of this assignment is to develop a robust, standalone application using procedural language which uses an external DBMS to store data. I must document the research and the strategy/planning that was required to achieve this objective.

**This document contains a full technical report of my activities throughout the duration of the project. It includes a brief overview of the software required for the development of the application as well technical details and any new findings. It also includes technical issues I encountered and the troubleshooting of those issues.**

**Table Of Contents**

[**Introduction**](#_d5yf9sj1ke92) **3**

[**Python3.9**](#_6kzxw0aknqjs) **3**

[What Is Python?](#_7xdlb955n4ez) 3

[How to install Python:](#_t4u0inaulwzo) 3

[**PyCharm IDE**](#_ciocc9900p1) **4**

[What is PyCharm?](#_xz8c1sixl8ly) 4

[How to install PyCharm:](#_vai8oy3vkxx) 4

[Using PyCharm:](#_93s9rd9zsret) 4

[Issues with PyCharm:](#_dstp1prbx0di) 5

[**MySQL**](#_ig0kgopmldyh) **5**

[What is MySQL?](#_km2rwzwdtgs4) 5

[How to install MySQL:](#_htfhubq6273m) 6

[Using MySQL:](#_id8m5km59m53) 6

[Issues:](#_xpd8cqry5ie5) 6

[**SQLite3**](#_xv4hgvl5ey4a) **7**

[What is SQLite?](#_xahwaznzrkzn) 7

[Installing SQLite3:](#_hkzovmrhat4n) 7

[Using SQLite3:](#_i85k7bttm3ts) 7

[**Code Implementation**](#_6e11k1ogwahs) **7**

[Pseudocode:](#_htng7mdb3l3m) 7

[Issues:](#_5ru9tjw2ywfq) 9

[**Appendix**](#_yxqbwvxlztpj) **9**

[**References**](#_a6ib8oww4gdm) **11**

[**Declaration**](#_gc6lr0g7mjx6) **11**

# **Introduction**

My objective within this project is to develop a standalone, robust application which allows a car remapping and performance company to keep track of customers via a menu driven program. The user of the application can add or remove data, view data and quit out of the program.

The data is written to a database where it is saved and readily available.

I used a variety of software/applications to complete this project. Some of which worked as expected, others caused technical issues and forced me to change my plan of action which I will document clearly in this technical report.

# **Python3.9**

## What Is Python?

Python is a free and open-source, high level, general purpose language used in computer programming. It is defined as an interpreted language. This means that the python code can be executed directly without first having to be compiled into machine code in order to be executed.

Python was first developed in the late 1980’s however didn’t first appear until 1991. The language was founded by a dutch programmer named Guido van Rossum. Python quickly became a predominant language used within industry and rapidly grew a supporting community around the language. Today, Python boasts as the 2nd most popular programming language in the world, alongside Java.

## How to install Python:

Python is a free and open-source language. The python interpreter is available at [www.python.org/downloads](http://www.python.org/downloads).

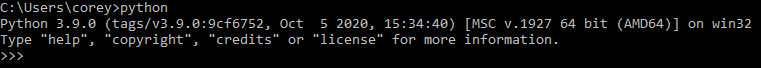
When downloading Python it is important to ensure the user downloads the latest version (Python 3.9.0) as often there can be backward compatibility issues with Python when operating with different versions.

Whilst on the website, I simply selected my operating system and the most recent version of Python and click download. Once the download was complete, I executed the .exe file which installed the Python interpreter on my machine.

It is important to take note as to where the base interpreter is being installed as you may need to point your IDE to this location at a later date.

Once the installation is complete, a check to see if Python is working can be done by opening the Command Prompt and typing ***python***. If the installation is successful, the command line will change to the following: >>>

I will include a snipping below.



This means the Python interpreter is successfully installed and the user can begin writing code in the command line.

To exit Python simply type ***exit()***.

# **PyCharm IDE**

## What is PyCharm?

PyCharm is an IDE or Integrated Development Environment used specifically for Python programming. PyCharm is available as both a paid, commercial environment as well as a free, community work environment.

PyCharm, and other IDE’s, allow the user to develop their code with the added benefit of a number of added assists to make the programmer's life easier. Some of these assists and features include coding assists and analysis, an integrated code debugger, an integrated unit for testing code, a smart file structure to keep projects organised, amongst many more.

PyCharm also provides an API so that developers have the ability to write their own PlugIn’s and further extend the features of PyCharm.

PyCharm is developed by a Czech company called JetBrains. The PyCharm IDE is available for download at [www.jetbrains.com/pycharm](http://www.jetbrains.com/pycharm).

## How to install PyCharm:

To install PyCharm I simply visited [www.jetbrains.com/pycharm](http://www.jetbrains.com/pycharm) and clicked on the download button on the home page. This brought me to another page where I was asked to select if I wanted to install the Professional (paid) version or the Community (free) version. I clicked on the download button under the Community Version title and the download began.

Once the download was complete I executed the .exe file and followed the instructions in order to install the software on my machine.

## Using PyCharm:

I am using PyCharm as my IDE for this project. The application will allow me to write my code within the environment. It will also allow me to test my code by running it. As well as display and errors or bugs I may encounter, pointing me in the right direction to resolving them.

In order to get started with the project, I opened PyCharm and created a new project.

From here I could specify the location of my new project.

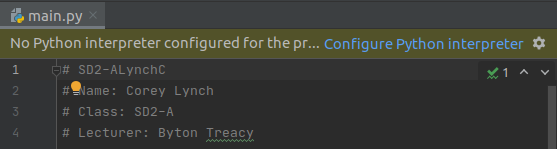
PyCharm will default to a PyCharmProject folder but that can be changed by selecting a different directory.

After this I could decide if I wanted to set up a new environment or use an existing environment. To set up a new environment you simply select where you would like to create it and point to the Python base interpreter which should be installed on your machine.

## Issues with PyCharm:

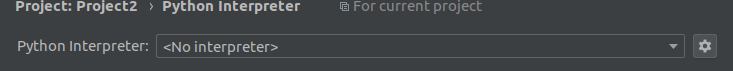
During the early stages of the project, when I was first setting up a virtual environment for my Python application, I encountered an error.

Anytime I ran my code I was told that *NO PYTHON INTERPRETER CONFIGURED FOR THE PROJECT* followed by a link which would take me to a setting menu.



After some research I discovered that I had not pointed to the Python base interpreter when I was setting up the project. Thus, the code could not execute as there was no interpreter there to interpret the code.

To resolve the issue, I clicked on the link in the error message which opened a setting menu. Within the interpreter settings menu I could see that I had no interpreter selected.



I simply clicked on the drop down box, selected the location of my base interpreter, pointed to the virtual environment I was using (venv file within the project file) and the issue was resolved right away. I could now execute code.

# **MySQL**

## What is MySQL?

MySQL is an open-source relational database management system. It is denoted as a structured query language or SQL as abbreviated. It can be used to store anything from a single record to an entire inventory or usernames and passwords for a web application. SQL can be used to access, update, delete and manipulate data within the database. It is used to query the database.

MySQL is widely used worldwide and is one of the most widely used in industry. Major companies such as Twitter, Netflix and Verizon use MySQL databases to power their businesses due to its free and open source nature.

## How to install MySQL:

To install MySQL I went to [www.mysql.com/](http://www.mysql.com/), navigated to the downloads page and selected the MySQL Community Downloads for my specified operating system.

Once the download was complete, I executed the file which began the MySQL Installer. I selected the Install MySQL option and followed the instructions displayed on the MySQL GUI.

Once everything was finished, I received a message on the GUI stating the Installation was complete.

## Using MySQL:

My intended use for MySQL was to use it in parallel with Python. MySQL was going to be used to save the data that was generated by the python application to a database.

However, I experienced issues with MySQL which I could not resolve. After hours upon hours of troubleshooting, I was able to narrow down the potential problems, however I could not resolve it.

## Issues:

The initial problem began when I installed MySQL on my Linux Ubuntu 20.04 machine. I followed all steps carefully to install the software through the Terminal. When I would try to run MySQL I was given an error message about no password set in a config file. Upon closer inspection I noticed that I had installed MySQL for Ubuntu 18.04 so I thought it may have been a compatibility issue.

I uninstalled MySQL for 18.04 and installed the correct software for 20.04.

However, the issue continued to persist after the fresh install.

I carried out some troubleshooting in my file manager and on Google. It appeared I was missing required contents from a configuration file that was installed with the MySQL software.

After reading multiple forums of people with similar issues, I discovered that the config file was a global file and could be written myself.

I googled the contents of this file and began writing the missing data to the file from the terminal.

Having done this, I attempted to save the file, only to be greeted by yet another error message stating I only had read permissions for the file. I was unable to change my permissions, even as the root user for unknown reasons.

Having evaluated the situation further, it appears that the missing data was a password for MySQL. When installing the software I set a password and anytime I wanted to carry out any functionality with MySQL within the terminal I had to input the password first for security purposes.

So, for some reason, when I set my password upon initial installation, it did not write the password to the config file as it should have. Therefore, MySQL was asking me for a password which it didn’t have and therefore the software was stuck as it needed the valid password in order to carry out any functionality. I was unable to even change the password.

I attempted to reinstall the software numerous times. The same issue occurred on both my Linux machine and my Windows machine. Time was of great importance at this stage of the project and I couldn’t waste anymore chasing this issue so I was forced to rethink my strategy and find a way around the issue.

# **SQLite3**

Finally, I resolved the issue I was facing and managed to set up an effective database which would work in conjunction with my python code and store data which was generated from the program on a database. My solution to this issue was to use SQLite.

## What is SQLite?

SQLite is an embedded relational database management system. As the name suggests it is a lightweight version of the SQL DBMS (database management system).

SQLite is commonly used on small and portable devices such as Android and iOS phones and tablets.

SQLite and Python are often used together, so commonly in fact, SQLite comes as a pre-installed library within Python since Python2.5 and therefore the module does not need to be installed separately.

SQLite provides a standardised Python DB API interface to the SQLite database.

## Installing SQLite3:

Since SQLite3 comes with any installation of Python from version 2.5 on, no installation is required. The module is simply installed alongside Python.

## Using SQLite3:

To use the SQLite3 module, you must first create a connection object within the Python code. This connection object represents the database from the Python program. You then create a cursor object which will allow you to input SQL code and execute it in order to manipulate the database.

*View the appendix below to see code.*

# **Code Implementation**

## Pseudocode:

***Initialisation:***

* Import sqlite module
* Make connection to database
* Create cursor in order to execute SQL statements within program

***User Menu:***

* Print menu to screen with menu options
* Ask user to input their menu option
  + 1. Edit Table
  + 2. View Table
  + 3. Quit Table
  + User Input >>>

***If Menu Choice = 1:***

* User selects edit table
* Print menu asking how user would like to edit table
  + 1. Insert
  + 2. Delete
    - If Menu Choice = 1
      * User wants to insert new data
      * Ask user for user input & assign input to variable
      * Call cursor to execute SQL
      * “INSERT INTO table(colName) VALUES(?)”, (data\_input)
      * DB-API parameter substitution to prevent SQL injection attacks
      * Commit changes
      * Print success message
      * Close connection
    - If Menu Choice = 2
      * User wants to delete data
      * Use primary key to delete row of data
      * Primary key => carReg
      * Ask user for car reg & assign to variable
      * Call cursor to execute SQL command
      * DELETE FROM table WHERE colName=(?),(user input,)
      * Assign user input to a tuple, DB-API parameter substitution for security purposes. Prevent SQL injections.
      * Commit changes
      * Print success message
      * Close connection
    - Error handling

***If Menu Choice = 2:***

* User selects to view table
* Print table to screen
* Call cursor to execute SQL command
* SELECT \*(all) FROM table
* Use fetchall function and assign return value to variable
* Use for loop to iterate through tables
* Print return variable to screen, in turn will print table

***If Menu Choice = 3:***

* Quit program
* Print goodbye message to user

***Error Handling***

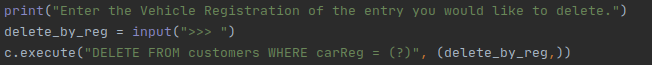
* Else statement - Error if user input int other than 1/2/3
* ValueError - Error if user input other than int 1/2/3

## Issues:

I encountered one issue whilst I was implementing the code. When I was passing in an input to the SQL code, I had issues with the datatype of that input.

I was specifying the user input as a string -> str(input(“Enter input”)).

This was generating an error within the SQLite module. As it turns out the input needs to be declared as a tuple in order to be passed into the SQL command.



This is a safety feature that is used to prevent SQL injections.

# **Appendix**

# SD2-ALynchC

# Name: Corey Lynch

# Class: SD2-A

# Lecturer: Byton Treacy

########################################################################################################################

# Import sqlite3 module

import sqlite3

conn = sqlite3.connect("project.db")

c = conn.cursor()

# Create 1 instance of Customers table

# NOTE: This line of code is only executed once, once the table is created & exists, cannot be executed again.

# c.execute("""CREATE TABLE customers(fName text, lName text, telNo text, carReg text, carMake text, carModel text)""")

# conn.commit()

########################################################################################################################

# Print menu to user and ask for menu choice

print("Please choose an option from the menu:\n")

print("1. Edit table")

print("2. Display table")

print("3. Quit")

try:

user\_input = int(input("\n>>> "))

########################################################################################################################

# If user wants to edit the table | 1 = add data, 2 = delete data

if user\_input == 1:

# Insert new data to table

print("\n...EDITING TABLE...")

print("[1] to insert new data | [2] to delete existing data")

edit\_input = int(input("\n>>> "))

# If user wants to add new data:

if edit\_input == 1:

fName = input("First Name: ")

lName = input("Last Name: ")

telNo = input("Contact Number: ")

carReg = input("Vehicle Registration: ")

carMake = input("Car Manufacturer: ")

carModel = input("Car Model: ")

c.execute("""INSERT INTO customers(fName, lName, telNo, carReg, carMake, carModel)

VALUES (?,?,?,?,?,?)""", (fName, lName, telNo, carReg, carMake, carModel))

conn.commit()

print('Data entered successfully.')

conn.close()

# Delete data from table

elif edit\_input == 2:

print("Enter the Vehicle Registration of the entry you would like to delete.")

delete\_by\_reg = input(">>> ")

c.execute("DELETE FROM customers WHERE carReg = (?)", (delete\_by\_reg,))

conn.commit()

print("Data deleted successfully.")

conn.close()

# Error Handling

else:

print("Error! Please choose option [1] or [2].")

########################################################################################################################

# If user wants to display table to screen

elif user\_input == 2:

# Display table

print("...DISPLAYING TABLE...")

c.execute("""SELECT \* FROM customers""")

rows = c.fetchall()

for row in rows:

print(row)

########################################################################################################################

# User wants to quit the program

elif user\_input == 3:

# Quit Program

print("Thank you. Goodbye.")

########################################################################################################################

# Error handling

else:

print("Error! Please choose an option [1-3].")

except ValueError:

print("Error! Please choose an option [1-3].")

########################################################################################################################

# **References**

* <https://dev.mysql.com/doc/>
* <https://www.digitalocean.com/community/tutorial_series/how-to-troubleshoot-issues-in-mysql>
* <https://www.w3schools.com/python/python_mysql_getstarted.asp>
* <https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html>
* <https://www.datacamp.com/community/tutorials/mysql-python>
* <https://www.jetbrains.com/help/pycharm/configuring-local-python-interpreters.html>
* <https://www.jetbrains.com/pycharm/guide/tips/create-sqlite-connection/>
* <https://www.jetbrains.com/help/pycharm/connecting-to-a-database.html>
* <https://www.jetbrains.com/help/pycharm/relational-databases.html>

# **Declaration**

I hereby certify that this material which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent, that such work has been cited and acknowledged within the text of my work. I confirm that I have read and abided by the CIT policy on plagiarism.

Signed: Corey Lynch.