

**CIS020-1 – Introduction to Software Development**

**CIS093-1 – Mathematics and Concepts for Computational Thinking**

**Assignment 2 – Group / Individual Project –CaseStudy**

**(Taxi Booking System)**

**Student Name: Baibhav Paudel - 2146504X - University ID: 012345**

**Student Name: XXXX XXXXX - University ID: 012345**

**OR**

**Student Name: XXXX XXXXX - University ID: 012345**

**Table of Contents**

[Introduction/Overview 3](#__RefHeading___Toc42041_3358107187)

[Task Description 3](#__RefHeading___Toc42043_3358107187)

[Assumptions about the system : 4](#__RefHeading___Toc42045_3358107187)

[Project Plan/Schedule 4](#__RefHeading___Toc42047_3358107187)

[Requirements Analysis 5](#__RefHeading___Toc42153_3358107187)

[Functional Requirements 5](#__RefHeading___Toc42155_3358107187)

[Non-functional Requirements 5](#__RefHeading___Toc42261_3358107187)

[Usability Requirements 6](#__RefHeading___Toc42263_3358107187)

[Design 6](#__RefHeading___Toc42265_3358107187)

[UML Diagrams 6](#__RefHeading___Toc42267_3358107187)

[Uses Case Diagrams(s) 7](#__RefHeading___Toc42269_3358107187)

[Activity Diagram(s) 8](#__RefHeading___Toc42461_3358107187)

[9](#__RefHeading___Toc3950_2529199306)

[Class Diagram(s) 11](#__RefHeading___Toc42463_3358107187)

[Database Design 13](#__RefHeading___Toc42475_3358107187)

[13](#__RefHeading___Toc3952_2529199306)

[Logical Database Design 14](#__RefHeading___Toc42477_3358107187)

[Physical Database Design 14](#__RefHeading___Toc42485_3358107187)

[Skeleton Tables (with Primary Keys and Foreign Keys) 14](#__RefHeading___Toc42487_3358107187)

[Data Dictionary 14](#__RefHeading___Toc3954_2529199306)

[User Interface Design 16](#__RefHeading___Toc42491_3358107187)

[Algorithm Design (could be omitted) 16](#__RefHeading___Toc42493_3358107187)

[Test No 17](#__RefHeading___Toc42495_3358107187)

[Test Date 17](#__RefHeading___Toc42497_3358107187)

[Purpose of test 17](#__RefHeading___Toc42499_3358107187)

[Input data or action 17](#__RefHeading___Toc42501_3358107187)

[Expected result 17](#__RefHeading___Toc42503_3358107187)

[Actual result 17](#__RefHeading___Toc42505_3358107187)

[Action if the above are not the same 17](#__RefHeading___Toc42507_3358107187)

[Screenshot of output 17](#__RefHeading___Toc42509_3358107187)

CIS020-1 – Introduction to Software Development - 2021-2022

CIS093-1 – Mathematics and Concepts for Computational Thinking – 2021-2022  
Assignment 2 – Individual Project – Case Study (Taxi Booking System)

# Introduction/Overview

The Taxi Booking System project, together with the project brief and requirements to execute this assignment, was the final task handed to us. Our primary objective in order to finish Assignment 2 was to create a system for booking taxis. In order to book a cab, a user must first register and sign in using login information (username and password), hence a user registration system and a user login system were created. There must be a way for the system user to reserve a taxi for their journey, thus a booking system was developed. A mechanism for admin control was created to allocate cabs to user booking requests. A driver dashboard was established where he or she could monitor assigned bookings, mark them as completed when the journey was finished, and view forthcoming bookings. A database was made and the system was connected to it using a database connector in order to keep all of the user information and details in the system.

# Task Description

The main aim was to create software for a taxi company so that customers could schedule cabs and view their upcoming journeys online using a desktop application's graphical user interface. All the important data regarding user, booking, taxi, and the driver had to be kept in an external database system.

The user must first register as a user with accurate information in order to make a cab reservation. After registering, the user must be able to input the location, date, and time of their pickup as well as the destination of their journey throughout the booking process.

The system must have a capability that allows users to update or cancel their reservations and view forthcoming trips through a graphical user interface. Admin from the business side must

be allowed to approve the user's booking once they submit it and choose a taxi driver to complete the booking. Each driver must have a single forthcoming trip.

## Assumptions about the system :

* The cost of a journey is determined during the booking process based on the distance between the pickup place and the destination.
* Credit card payments will be added in the driver's account balance.
* The driver is paid immediately once the journey is finished.
* Even after cabs have been assigned, reservations can still be canceled, but only with the admin's permission if an issue occurs.
* Taxi assignments cannot be changed.
* The driver keeps his or her earnings throughout the month, but at the end of the month, he or she is required to pay the corporate service fee which has already been computed and is shown to him in his driver dashboard in order to continue using the system the following month.

# Project Plan/Schedule

|  |  |  |
| --- | --- | --- |
| Week No. | Tasks | Priority |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | Submit Group Report, Project Code and Video Recording (if required) |  |
|  | Project Presentation (if required) |  |

# Requirements Analysis

## Functional Requirements

TTBS = Turbo Taxi Booking System

|  |  |  |
| --- | --- | --- |
| Req. No | Requirement | Priority\* |
| 1 | A signup system allows customers to register. | MUST |
| 2 | To access the system, users must log in. | MUST |
| 3 | Request booking from booking form. | MUST |
| 4 | User and reservation information is saved to a separate file. | MUST |
| 5 | An administrator must be able to add a cab to a user's reservation. | MUST |
| 6 | All booking requests will be visible to the administrator. | MUST |
| 7 | Bookings must be able to be canceled by admin. | MUST |
| 8 | The user must be able to see information about forthcoming bookings. | MUST |
| 9 | A booking must be able to be cancelled by the user. | MUST |
| 10 | Booking specifics must to be editable by the user. | SHOULD |
| 11 | he ability to log into the system is required for cab drivers. | MUST |
| 12 | An forthcoming journey must be visible to taxi drivers. | MUST |
| 13 | It should be possible for taxi drivers to see their earnings. | SHOULD |

## Non-functional Requirements

|  |  |  |
| --- | --- | --- |
| Req. No | Requirement | Priority\* |
| 1 | The system's processing speed should be high. | MUST |
| 2 | Must be screen-adaptive for devices. | SHOULD |
| 3 | System upgrades must be possible. | MUST |
| 4 | must be free of crashes or run-time problems. | MUST |

## Usability Requirements

|  |  |  |
| --- | --- | --- |
| Req. No | Requirement | Priority\* |
| 1 | System GUI must be user friendly. | MUST |
| 2 | All forms for entering data should be brief, simple to complete, and include entry validation. | MUST |

# Design

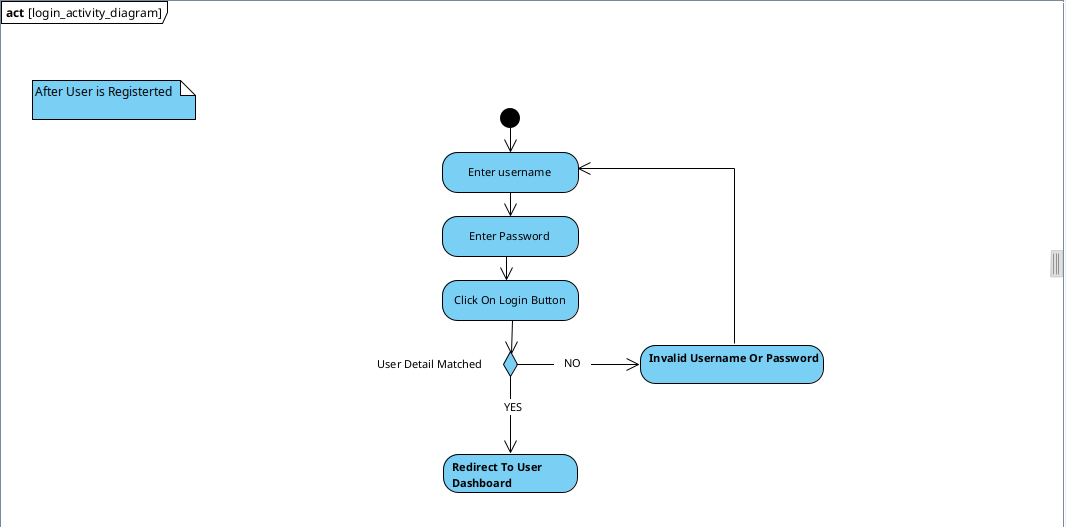
## UML Diagrams

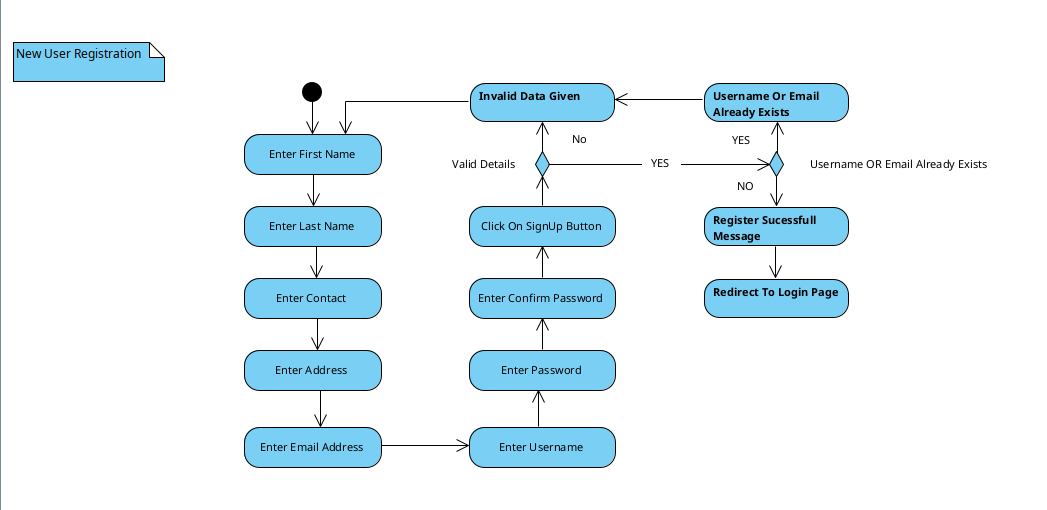
### Uses Case Diagrams(s)

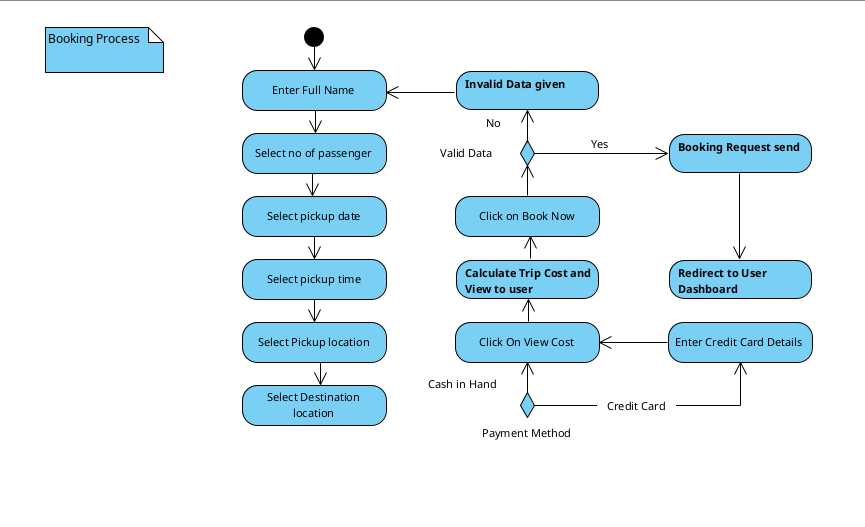
Diagram

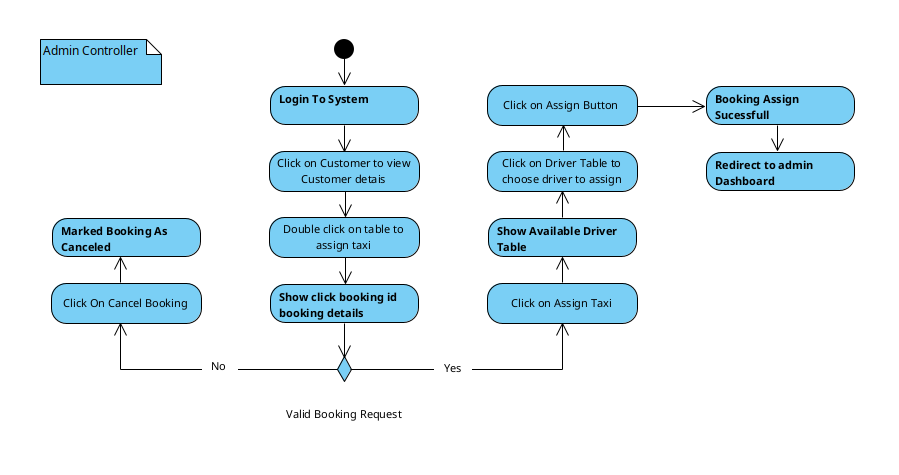
Description automatically generatedFigure 1: Taxi\_Booking\_System\_Usecase\_diagram

### Activity Diagram(s)

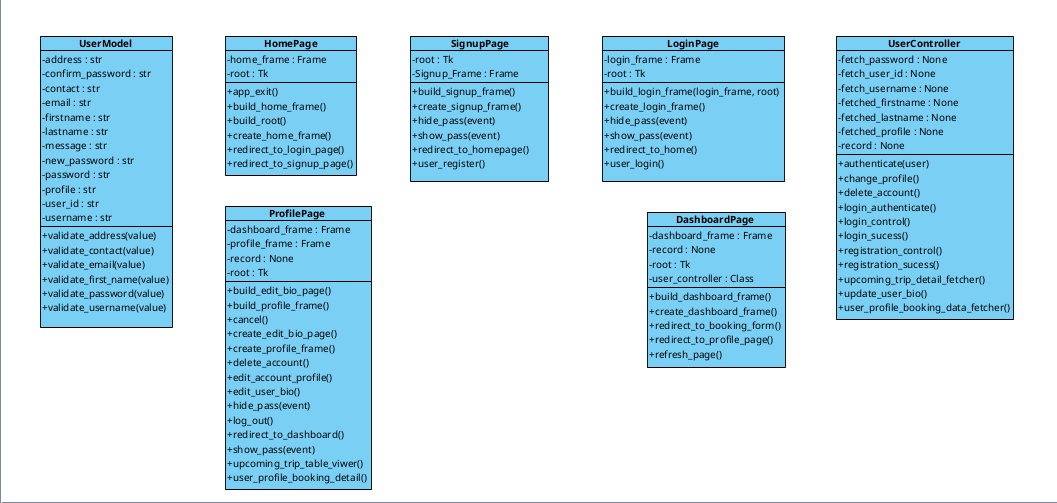
Figure 2: Login\_System\_Activity\_Diagram

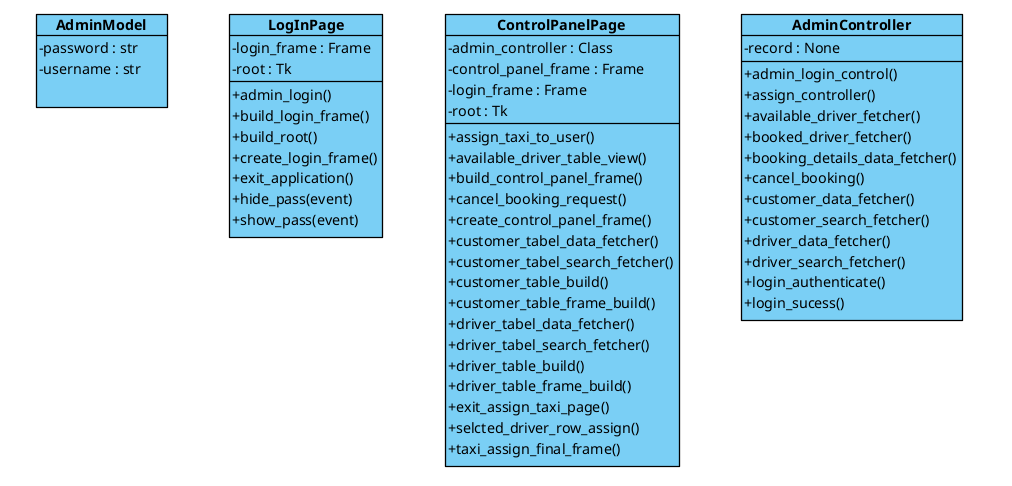
Figure 3: Register\_System\_Activity\_Diagram

Figure 4: Booking\_System\_Activity\_Diagram

Figure 5: Admin\_Control\_System\_Activity\_Diagram

### Class Diagram(s)

Figure 6: user\_class\_diagram

Figure 7: admin\_class\_diagram

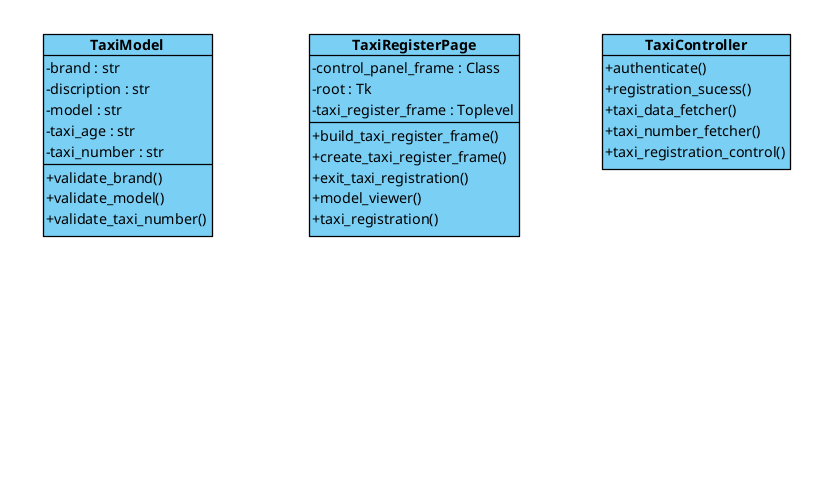
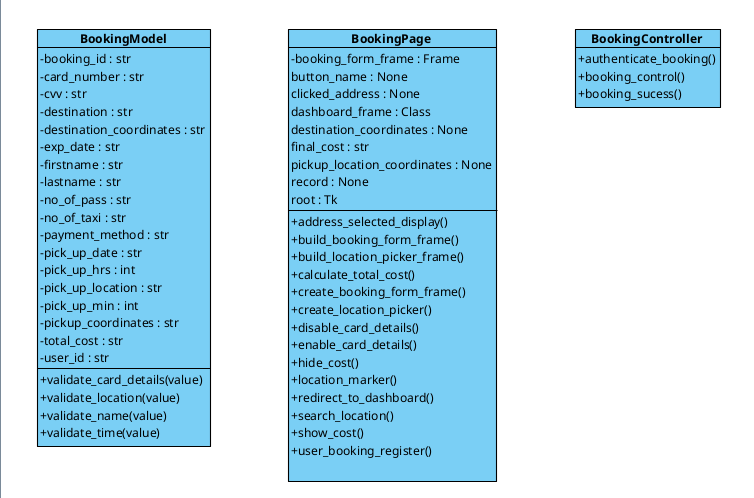
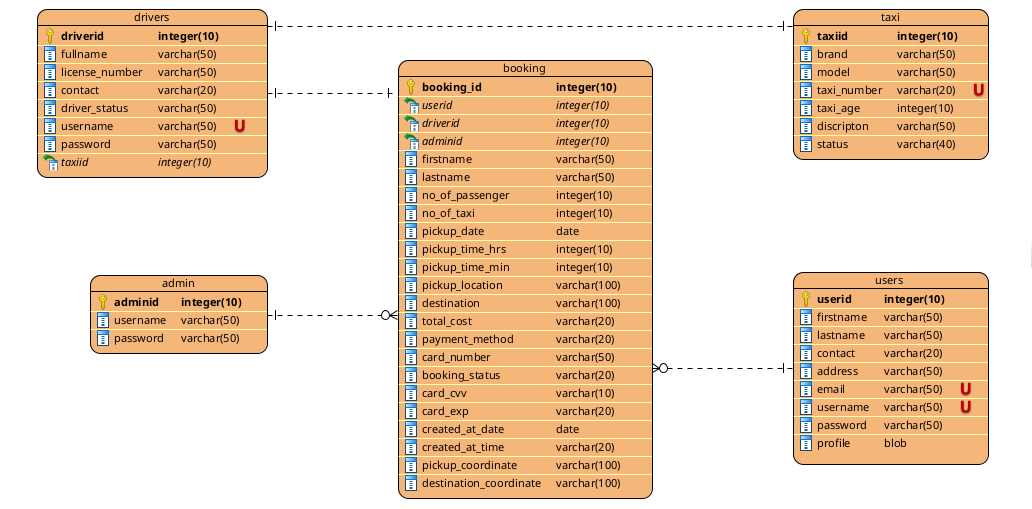
Figure 8: taxi\_class\_diagram

Figure 9: driver\_class\_diagram

Figure 10: booking\_class\_diagram

## Database Design

Figure 11: turbo\_tb\_database\_ERD

### Logical Database Design

Diagram

Description automatically generatedFigure 12: databae\_ERM\_figure

### Physical Database Design

#### Skeleton Tables (with Primary Keys and Foreign Keys)

Users(**userid**, firstname, lastname, contact, address, email, username,

password, profile)

Booking (**booking\_id**, userid\*, driverid\*, adminid\*, firstname,lastname, no\_of\_passenger, no\_of\_taxi, pickup\_date, pickup\_time\_hrs, pickup\_time\_min, pickup\_location, destination, total\_cost, payment\_method, card\_number, booking\_status, card\_cvv, card\_exp, created\_at\_date, created\_at\_time, pickup\_coordinates, destination\_coordinates)

Taxi (**taxiid**, brand, model, taxi\_number, taxi\_age, discription, status)

driver (**driverid**, fullname, license\_number, contact, driver\_status, username, password, taxiid\*)

Admin (**adminid**, username, password)

#### Data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Users** | | | | | |
| **Discription :** Users Details | | | | | |
| **Field Name** | **Data Type** | **Length** | **Index** | **Null** | **Discription** |
| **Userid**  **(Primary)** | **int** | **10** | **PK** | **No** | **Auto Increment** |
| **firstname** | **varchar** | **50** |  | **No** | **Firstname of user** |
| **lastname** | **varchar** | **50** |  | **N0** | **Lastname of user** |
| **contact** | **varchar** | **20** |  | **No** | **Contact detail of user** |
| **address** | **varchar** | **50** |  | **No** | **User’s location detail** |
| **email** | **varchar** | **50** |  | **No** | **Valid email address of user** |
| **username** | **varchar** | **50** |  | **No** | **Username for login** |
| **password** | **varchar** | **50** |  | **No** | **Password for personal security** |
| **profile** | **blob** |  |  | **No** | **User’s profile picture** |

* User Interface Design

Sketches of your GUI or text-based interfaces

You should use Balsamiq Wireframes (see under Announcements) or an online prototyping tool (for example, <https://marvelapp.com/>)

* Algorithm Design (could be omitted)

For CSI093-1 only

Examples later!

Implementation

You should discuss YOUR application. How was the program created?

You should describe the development stages.

Explain how you manged the implementation. How did you ensure all group members actively contributed to writing code? How did you distribute the workload? Was it successful?

What Integrated Development Environment (IDE) did you use? Justify your choice of IDE, Python Framework (if used), packages / libraries.

What problems did you encounter during the coding, and explain how you overcame these?

Include snippets of code here – that either caused problems or show original, novel approaches

Include a couple of screen shots of your application here

What rDBMS did you use? Possibly, SQLite, MySQL. Justify your choice of database system.

Did you use WAMP / XAMPP / MAMP or an equivalents stack on your own PC?

Detailed discussion of your experience coding is required here.

Did you incorporate any novel approaches to coding and meeting the Assignment requirements?

Testing

Include evidence of detailed and robust testing.

There are various ways to document this…

Test Plan / Test Log (tabular format) with date, input/expected output/actual output and screen shot of result.

OR

Documentation of each test – one after the other

You need to include:

* Test No
* Test Date
* Purpose of test
* Input data or action
* Expected result
* Actual result
* Action if the above are not the same
* Screenshot of output

Discussion / Reflection / Critical Analysis

* What went well and what went wrong?
* What would you have done differently?
* How would you improve your application in the future?

This part is very important. You must discuss the project, group work, and time management skills.

Basically, you are evaluating what you have done. Did you achieve all the required specification of the Assignment brief? If not, why not? Discuss the problems you encountered and how you overcame them.

What stopped you achieving what you had hoped to achieve and to the standard you know you are capable of achieving? Remote learning, lack of access to BREO, COVID, lack of suitable devices remotely, Internet connection issues etc.

Did you give yourselves enough time to accomplish this work? Was it harder than you expected? What have you learned from tackling this Assignment?

If working as a group, did your group work well together? What did you learn about the dynamics of group work?

Did this work enhance your understanding of the Python programming language? Did it motivate you to learn more Python, and other programming languages?

If you had to undertake this or something similar again would you tackle it differently?

What improvements or enhancements would you recommend for a future iteration of this project?

Conclusion

What were you asked to do? Did you achieve it?

References

Use the Harvard Referencing System. Any reference must be cited in text -otherwise it should go in a Bibliography section.

Appendix

Make sure that all content has a Fig. No., caption and explanation.

You MUST include:

Complete project code.

Include file/class names.

Make sure you code is fully commented and well presented with correct indentation and colour coding.

You could include:

Details of you database/table structure and content (from PHPMyAdmin / SQLite Studio)

Screen shots of your GUI or text-based interfaces

Anything else that you might want to include which does not fit well into the main report body.