

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

**Assignment 2 –Individual Project – Case Study**

**(Taxi Booking System)**

**Student Name: Santosh Tamang**

**University ID: 2147440**

Table of Contents

Introduction / Overview………………………………………………………………………… 5

Task Description………………………………………………………………………………... 5

Project Plan / Schedule….

Tasks:

Requirements Analysis

Overview of Functional, Technical (Non-Functional) and Usability Requirements

Design:

* UML Diagrams
* Use Case Diagrams
  + Use Case Specifications/Description
  + Use Case Scenario
* Activity Diagrams
* Class Diagram
* Database Design
* Logical Database Design
* Entity Relationship Model
  + ERM Diagram
  + List of Entities
* Physical Database Design (including Data Dictionary)
  + Skeleton Tables
  + Data Dictionary
* User Interface Design

Implementation

Testing

List of Group Members – Roles and Responsibilities (if Group Work)

Discussion / Reflection / Critical Analysis

Conclusion

References

Appendix

* Complete code with comments and class / filenames.

You will need to complete the Table of Contents and Page Nos manually

PLEASE NOTE THAT THIS IS A PROFESSIONAL REPORT

ALL SCREENSHOTS, DIAGRAMS, etc SHOULD BE GIVEN A FIGURE NUMBER, A CAPTION AND AN EXPLANATION

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

Assignment 2 – Group or Individual Project – Case Study (Taxi Booking System)

Introduction/Overview

The S&S Taxi Service has been providing taxi services for a long time. They now want to provide the booking facilities through an online medium and a desktop application was requested to be created.

Currently, all the details of their customer, driver and staff are registered in hard copy format along with the trips they have booked. The customer must come to the office, send an e-mail or call the office to book a trip, or to cancel the current trip. Searching for data takes a long time as it was hard to search through all the data. It was hard for the customer to just book or cancel a trip.

For the easiness of the customer to book, cancel and view trips and the staff to search and fill data by converting data into softcopy version. A desktop application is purposed to solve all the existing problem and creating user-friendly environment. The application allows different users (i.e. customer, staff and drive) to perform different activity. The application allows customer to login, book a trip, update the trip, view the trip and cancel the trip. The application allows staff to login, confirm a trip booked by the customer and assign a suitable driver. The application allows driver to login and view their upcoming trips.

To fulfil the requirement of the given assignment, a desktop application was created to book, view, update and cancel trips via online medium, with database to store all data. PostgreSQL database was used to create the required RDBMS with all the required queries and data dictionary.

Task Description

The task requires the creation of a desktop application with a functioning database for a taxi company that is attempting to offer its services online. There are only three users i.e., customer, driver and staff.

The customer should be able to book a trip, i.e., provide pickup location and time as well as the number of passengers, destination location and payment method. They should be able to view their bookings and booking can be cancelled if needed. To be able to do the above-mentioned activity customer must log in first. If not registered, the customer must register providing their name, address, email and telephone number.

The driver should be able to view all the upcoming trips after logging in.

The trip booking is confirmed and drivers are assigned by the taxi company administrative after logging in. The staff can see all trips booked.

Include the scenario, and Use Case Diagram from the Brief.

Include any assumptions you make about the system.

Project Plan/Schedule

|  |  |  |
| --- | --- | --- |
| Week No. | Tasks | Priority |
| 1 | Research in the related field | MUST |
| 2 | Requirement Analysis and GUI Prototype Design | MUST |
| 3 | Database Design | MUST |
| 4 | GUI Implementation | MUST |
| 5 | Database Implementation and Testing the Application | MUST |
| 6 | Finalizing Product | MUST |
| 7 | Optimizing Product | SHOULD |
| 8 | Submit Group Report, Project Code and Video Recording | MUST |
| 9 | Project Presentation | MUST |

Requirements Analysis

Overview of Functional, Technical (Non-Functional Requirements) and Usability Requirements

Functional Requirements

TBS = Taxi Booking System

Customer

1. A customer must register.
2. A customer must log in.
3. A customer should be able to select the pickup locations and the destination.
4. A customer should be able to view their trips.
5. A customer can cancel the booking.
6. A customer should be able to make multiple bookings using the same id.

Staff

1. A staff must login.
2. A staff should be able to view all the customers with the booking they made.
3. A staff should be able assign a driver to a booking.
4. A staff should be able view driver schedule.

Driver

1. A driver must login.
2. Drivers should be able to view their upcoming trip.

Non-functional Requirements

1. The system should respond as fast as possible.
2. The system must be platform independent.
3. The system must be online 24/7.
4. Future amendments must be allowed in the system.

Usability Requirements

1. The system should be user-friendly.
2. There should be validation to enter data and the forms should be short and easy to fill.
3. Enough information about the booking should be provided to meet users’ needs.
4. The navigation in the application should be easy the user should not need a guide to do the basics.

Design

* UML Diagrams
* Uses Case Diagrams(s)

Diagram

Description automatically generated

Fig 1: Taxi Booking System - Use Case Diagram (Sea Level)

* + - Use Case Specifications/Description
    - Use Case Scenario
    - Use Case Diagrams at more than one level (Sea Level and Fish Level)

EXAMPLE OF USE CASE SPECIFICATIONS (NOT FOR THIS SCENARIO)

Use Case Specifications / Description

Allocate Accommodation

A specific room is allocated to each guest.

Change Booking

An unguaranteed booking may become guaranteed on provision of credit card details. Other changes (for example to the composition of the party or to the dates booked) may be made on the customer’s request.

Calculate Bill Total

The total bill for a room comprises the cost of the room plus any items charged to it (for example, from the minibar, hotel bar or restaurant and any additional services provided) plus the appropriate rate of VAT.

Cancel Booking

A customer may cancel a booking any time before checking in. The booked accommodation is de-allocated.

Charge Item to Room

A guest may use items from the minibar in the room, make external telephone calls, make purchases from the hotel bar or restaurant or order additional services such as laundry. These items are charged to the guest’s room. An external service ordered by the guest may also be charged to their room (although some services, such as taxis, will be paid for directly by the guest).

Check In Guest

When a party (one or more guests) arrives at the hotel each guest is checked in and the booking becomes registered. Details of all the guests are confirmed, or if they have not been given at the time of booking they are recorded. The method of payment (by which guest(s) or whether to be charged to a corporate account) is confirmed. Accommodation is allocated if not already done at the time of booking and special requests are recorded.

Check Out Guest

Before the guest leaves the hotel they need to check out and the booking becomes completed. The receptionist provides them with the total room bill, which an individual customer must pay. If the bill is to be charged to a corporate customer the guest is asked to check and sign the bill, it is then added to the appropriate corporate account.

Make a Booking

When a customer contacts the receptionist with a booking request the receptionist checks the room occupancy for the required period and if the customer's needs can be met, makes a booking.

A guaranteed booking is only made for corporate customers or for individual customers who supply credit card details.

Accommodation may be allocated at this stage; alternatively where it is a group booking (for example, for 6 twin-bedded rooms) the allocation of each guest in the party to a specific room may be done on checking in. Either way, occupancy records are updated to avoid double booking.

Order External Service

Services not offered by the hotel itself can be ordered, once their availability has been checked. The guest name, room number and details of the service required are taken when the service is ordered.

Provide Guest with Current Bill

On request a guest may be provided with the current total of their bill.

* Activity Diagram(s)
* Class Diagram(s)

Table

Description automatically generated with medium confidence

* Classes (omitting methods / operations)
  + Customer (customertid, title, firstname, lastname, email, telno, password, address1, town, county, postcode, paymentmethod)
  + Booking (bookingid, datebooked, startaddress, destinationaddress, date, time, status, paid)
  + Taxidriver (driverid, title, firstname, lastname, email, password, regno)
  + Administrator (adminid, firstname, lastname, email, password)
* Database Design
* Logical Database Design
  + Entity Relationship Model (ERM)
    - ERM Diagram

Diagram

Description automatically generated

Diagram

Description automatically generated

* + - List of Entities

Customer (**customertid**, title, firstname, lastname, email, telno, password, address1, town, county, postcode, paymentmethod)

Booking (**bookingid**, customerid\*, driverid\*, datebooked, startaddress, destinationaddress, date, time, status, paid)

Taxidriver (**driverid**, title, firstname, lastname, email, password, regno)

Administrator (**adminid**, firstname, lastname, email, password)

Primary Key = Bold and Underlined

Foreign Key = \*

* Physical Database Design
  + Skeleton Tables (with Primary Keys and Foreign Keys)

Customer (**customertid**, title, firstname, lastname, email, telno, password, address1, town, county, postcode, paymentmethod)

Booking (**bookingid**, customerid\*, driverid\*, datebooked, startaddress, destinationaddress, date, time, status, paid)

Taxidriver (**driverid**, title, firstname, lastname, email, password, regno)

Administrator (**adminid**, firstname, lastname, email, password)

Primary Key = Bold and Underlined

Foreign Key = \*

* + Data Dictionary

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer | | | | | | | |
| Description: Customer details | | | | | | | |
| Field Name | **Datatype** | **Length** | **Index** | **Null** | **Default** | **Validation rule** | **Description** |
| customerid *(Primary)* | int (10)  unsigned | 10 | PK | No |  |  | Autoincremented Uniquely identifies every customer |
| title | varchar (10) | 10 |  | No |  |  | Title of customer |
| firstname | varchar (20) | 20 |  | No |  |  | First name of customer |
| lastname | varchar (30) | 30 |  | No |  |  | Last name of customer |
| email | varchar (50) | 50 |  | No |  | Must be email format containing an @ and a ‘.’  Regex expression used | Email of customer |
| telno | varchar (20) | 20 |  | No |  |  | Telephone number of customer |
| password | varchar (20) | 20 |  | No |  |  | TBS password |
| address1 | varchar (30) | 30 |  | No |  |  | First line of address |
| town | varchar (30) | 30 |  | No |  |  | Customer town |
| county | varchar (30) | 30 |  | No |  |  | Customer county |
| postcode | varchar (15) | 15 |  | No |  |  | Customer postcode |
| paymentmethod | varchar (20) | 20 |  | No |  |  | Customer payment method |

Indexes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Column** | **Null** |
| PRIMARY | BTREE | Yes | customerid | No |

* 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.