Executive Summary

The overall purpose of this project is to analyze the requirements, design, implement, document, and test a database application for Cuber RidesRUs that will allow their users to manage information for its drivers, vehicles, customers, and trips. All required portions of this project, as detailed by the project description, are attached in a zip file that includes this report. This document details the reports of each phase of the project. The final result of the project is able to fulfill are functional requirements of the project description. However, there are some improvements that could be made to this project. First for the E-R model, generalization could be applied to driver and customer with person being the parent entity. In my project, I modeled driver and customer as completely different entities. There is more information for this choice later in the report for Phase C. Second, query 7 in the functional requirements could be improved as well. The user could still be able to gain the information described in the requirement, but some work on the user's part must be done. More information on this improvement is available in the Database_interface Jupyter notebook.

Further, createALL contains the SQL statements needed to create and use both the database for the project and all the tables detailed in Phase C. dropALL contains the statements needed to drop all the tables in the database. queryALL contains all the SQL statements needed to fulfill the functional requirements of the project. loadALL contains the SQL statements needed to insert the data needed to demonstrate the database's function capabilities. Finally, transactionsALL is empty as I was able to fulfill the functional requirements of the project without the need for transaction control.

Introduction

This document is a detailed report for UMBC's CMSC 461 semester project as required by the project description. All users of this database application must have network computers capable of running Web browsers, which is assumed in the System Scope of the project description. The data requirements given in Appendix A of the project were translated into a relevant E-R diagram which is the basis of this database design, and is attached in the PhaseB file.

Phase A - Requirement Analysis

The tasks involved with this project include the design, implementation, documentation, and testing of a database application for the use of Cuber RidesRUs' employees. These tasks will be accomplished throughout various phases of the project. There are two different kinds of requirements associated with the implementation of this application. They include both data and functional requirements. The data requirements include storing information pertaining to drivers, vehicles, customers, trips, and payments within the context of the Cuber RidesRUs ride sharing business. The data requirements will be modeled and defined thoroughly in Phase B of the project through the construction of an Entity-Relationship diagram along with further conceptual design. As for the functional requirements of the database application, they include the entering, updating, deletion of the data associated with drivers, vehicles, customers, trips, and payments. Additionally, there is a set of queries that the database must support, as defined by the functional requirements. These functional requirements will be implemented in Phase D and Phase E of the project through the design of a MySQL database along with the use of SQL scripts and Python Jupyter notebook to interact with the database.

Phase B - Conceptual Design Constraints

For the driver entity, the license information as well as the contact information of the driver can be thought of as a candidate key. Within the drives relationship, drivers are able to drive 1 or more cars within the ride sharing service. For a vehicle, the make, model, year, and driver of the vehicle form a candidate key. However, since a driver can drive multiple cars, these cannot be a part of the primary key. For the make relationship, vehicles can make any number of trips. However, these trips must be associated with different timestamps as it is impossible for two trips to be taken at the same time by the same vehicle. For the trips entity, there is no candidate key outside of the primary key. Also, the stops attribute is multivalued as there are 2 or more associated stops with every trip. The trajectory of a trip is defined as the primary key for a trip as two trips can not be taken at the same place and at the same time. For the take relationship, customers may take any number of trips. Finally, there may be any number of payments for a trip, each payment being made at different times.

Phase C - Logical Design - Needs to be finished

The file containing my logical design of my project is attached in the PhaseC file. In this phase, I mapped the conceptual model of the previous phased onto the relational data model as well as designing all the tables for all my entities and relations. I decided to take email and phone out of both the customer and driver relations. I also decided to keep driver and customer as completely separate entities, and not apply generalization on them. While this design choice could perhaps be improved on, I was still able to fulfill all functional requirements of the project with my design choice.

Phase D - Physical Design

In this phase, I designed my MySQL database based on the design developed in phase B, implemented tables for the relations and constraints in the createALL file, created scripts for dropping these tables in dropALL, created scripts for loading data into your tables in loadALL, and designed the user interface for my application. My main decision in terms of my user interfaces was to implement each query that is described in the functional requirements as its own cell in the Jupyter Notebook that is attached. This choice will allow users to pick and choose exactly what queries they want to run, and not be forced into waiting for queries to finish that they potentially have no use for. The scripts created in this phase contain the basic data needed to perform the queries described in the functional requirements.

Phase E - Development and Testing

All of my development and testing was done within Jupyter Notebook. The code used to connect to the database as well as the code used to generate the queries reside here. In this phase, I developed a Jupyter notebook with code to access, populate, update, and administer the SQL tables made in the above phases, developed a user interface using Python Jupyter notebook which satisfies all the functional user requirements of the project, and SQL scripts for creating indices for the database application. Justify the reasons for creating any such indices.

Phase F - User Guide

The user guide for my application is integrated in both the Jupyter Notebook and README that are attached in the zip file. All default behavior/input for the queries in functional requirements reside in the Jupyter Notebook that is attached.

Conclusion

I was able to provide all functional requirements of the problem, aside from one improvement in query 7. In addition, the discussed change could be made to the conceptual design of the project. As for the demonstration and testing of the database application, all the information and steps needed are located in the README file attached. In conclusion, I feel that, through the items attached, I was able to fulfill the purpose of this project by meeting both the data and functional requirements given in the project description.