PEACE-WORK-FATHERLAND

MINISTRY OF HIHGER EDUCATION

UNIVERSITY OF BUEA

PAIX-TRAVAIL-PATRIE

MINISTERE DE L’ENSEIGNEMENT SUPERIEUR

UNIVERSITY OF BUEA



**UNIVERSITY OF BUEA**

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

CEF 440: INTERNET PROGRAMMING AND MOBILE PROGRAMMING

DATABASE implementation of a passenger positioning system

Presented by:

EMADE ROSINE NSAH........................................................,..FE20A032

ETUGE GIDEON.......................................................................FE20A036

NOUGHUE LEMOUPA FRANK..............................................FE20A092

TAJOUEGO DJUIDJA ESTRELA............................................FE20A107

TAMAH JUSTENE....................................................................FE20A108

**Course Instructor**: Dr Nkemeni Valery

Table of Contents

[II. INTRODUCTION 3](#_Toc136635502)

[Purpose 3](#_Toc136635503)

[III. WHAT IS A DBMS 4](#_Toc136635504)

[Types of database management system 4](#_Toc136635505)

[IV. WHY DID WE CHOOSE A RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS) AND WHY MYSQL 4](#_Toc136635506)

[V. ARCHITECTURE OF A DBMS 5](#_Toc136635507)

[VI. DESIGN OF OUR DBMS 6](#_Toc136635508)

[CONCEPTUAL DESIGNS 6](#_Toc136635509)

[Relational schema 7](#_Toc136635510)

[ER diagram 8](#_Toc136635511)

[VII. ARCHITECTURE OF THE JETME DATABASE 9](#_Toc136635512)

[VIII. IMPLEMENTATION OF Relational DBMS OF THE PASSENGER POSITIONING SYSTEM 10](#_Toc136635513)

# INTRODUCTION

Database design is the process of creating a data model for an application or system, which includes defining entities, their attributes, and relationships between them. The aim of a database management system is to provide efficient and secure storage of this data, as well as to provide tools for data retrieval and manipulation.

A taxi passenger positioning system is a type of location-based service that involves tracking the position of taxis in real-time and matching them with passenger requests. It therefore uses a database management system (DBMS) to store and manage information about passengers, rides and drivers. The data captured by the system can be used for analysis and reporting to improve the efficiency of the taxi service.

## Purpose

The purpose of the database of a taxi passenger positioning system is to store and manage all the information related to passengers, rides, drivers, and in real time. The system captures data about the present location, destination, date, time. This information is useful for operational and analytical purposes. From an operational perspective, the system can use the data to optimize routes and dispatch vehicles more efficiently. From an analytical perspective, the data can be used to generate reports on ridership trends, driver performance, and vehicle utilization.

# WHAT IS A DBMS

A Database Management System (DBMS) is a software system used to create, manage, and store databases. DBMS provides a way for users to interact with databases, allowing them to add, edit, and retrieve data in a secure and efficient manner. It also provides a way for administrators to control the access and security of the data stored in the database

## Types of database management system

There are mainly two types of DBMS which include the relational and the non relational database as seen below:

1. **Relational DBMS:** This is the most popular type of DBMS, where data is stored in tables of rows and columns, and the tables are related to each other through common fields.
2. **Non- relational database**: They include
   1. **Object-Oriented DBMS:** This type of DBMS stores data as objects, which can be defined and manipulated using object-oriented programming principles.
   2. **Document-Oriented DBMS**: This type of DBMS stores data in semi-structured documents, which can be retrieved and managed using document-oriented query languages.
   3. **Graph DBMS:** This type of DBMS stores data in the form of nodes and edges, and is used to manage complex relationships between data.
   4. **Key-Value DBMS:** This type of DBMS stores data as key-value pairs, and is optimized for high-speed retrieval and storage of data.

For the database implementation within the scope of this project, we will make use of the relational database management system (RDBMS)

# WHY DID WE CHOOSE A RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS) AND WHY MYSQL

We decided to make use of the relational database system because it is:

1. **Data Consistent:** RDBMS follows the ACID (Atomicity, Consistency, Isolation, and Durability) properties that ensure data consistency in a database and reduces the chances of any inconsistencies in data.
2. **Data Integrity:** RDBMS ensures data integrity in a database by enforcing relational integrity constraints using foreign keys and unique indexes.
3. **Scalability:** RDBMS is highly scalable, and it can handle an enormous amount of data. It supports distributed databases, which allows you to add more storage as and when required.
4. **Security:** RDBMS provides robust security features like authentication, authorization, and encryption that can help you safeguard your valuable data.
5. **Backup and Recovery:** RDBMS provides efficient backup and recovery options that can help you recover your data in case of any data loss.

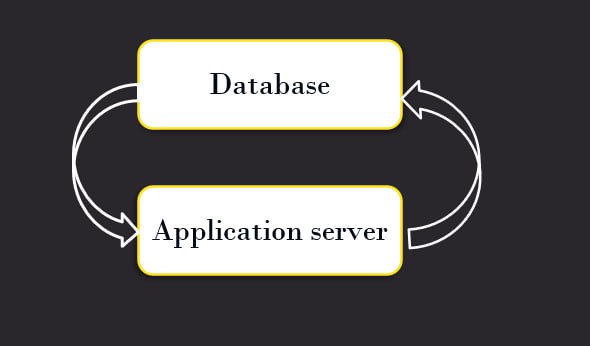
**Why MySQL ?**

MySQLwas chosen for the database implementation because of the following reasons

1. **Open-Source:** It is open-source software, which means it can be downloaded and used for free.
2. **High-Performance:** It is capable of handling high-load environments, and it offers high-speed access to data.
3. **Compatibility**: It is highly compatible with different platforms and programming languages.
4. **Ease of Use:** It is easy to install, manage and it provides a user-friendly interface for managing databases.
5. **Community Support:** It has a large community of developers who provide support through forums, tutorials, and documentation.

# ARCHITECTURE OF A DBMS

The architecture of the DBMS in the taxi passenger positioning system involves the integration of various software components to exchange data and provide taxi booking and tracking. The architecture must be scalable and secure, providing high availability and reliability to the users.



The architecture of the RDBMS for the passenger positioning system contain the following components:

1. **User Interface**: This component provides a graphical or command-line interface for users to interact with the database. It allows users to input, modify, and retrieve data from the database.
2. **Database Engine**: This component is responsible for managing the storage and retrieval of data in the database. It includes several sub-components such as the query processor, transaction manager, buffer manager, and storage manager.
3. **Data Storage:** This component stores the actual data in the database. It includes the physical storage devices such as hard drives or solid-state drives, as well as the data files and indexes used by the database engine to access and manipulate the data.

# DESIGN OF OUR DBMS

The design of this system is intended to provide real-time information on the location of taxis and enable passengers to book a taxi from their current location.

### CONCEPTUAL DESIGNS

**Entities:**

* Passenger
* Drivers
* Admin
* Free Lance Taxi

**Attributes**

* Passenger: Passenger\_ld, passenger\_Location, Telephone, Names, Destination, Password
* Admin: Names, AdminID, AdminPassword
* Free Lance Taxi: Names, Plate\_Number, Freelance\_Id, First Name, Last\_NamesFreelance\_password, Driver\_license
* Hired Taxi: First Name, Last Name, Plate Number, Driver license, Employer, password, hired\_id

**Relationships:**

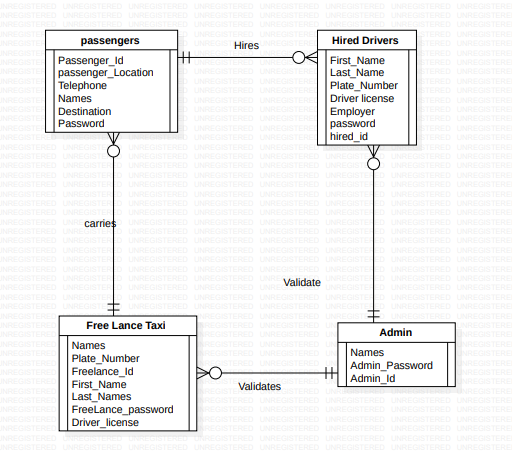
* One passenger can book only one taxi at a time
* One hired taxi can accept only one passenger at a time
* Many hire taxi can request for validation at a time
* An admin can validate only one hired taxi at a time
* An admin can validate only one free-lance taxi at a time
* Many free-lance taxi can be sent for validation at a time
* One free-lance taxi can carry many passenger at a time
* One passenger can take only one taxi at a time

### Relational schema

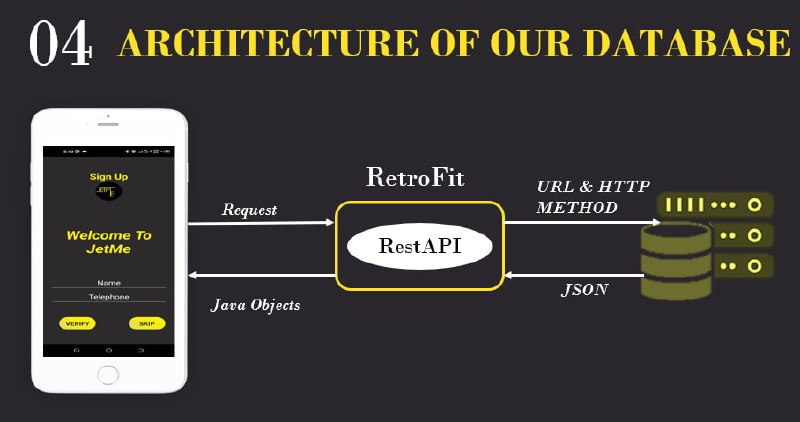
* Admin( admin-Id, name, password-password)
* Passenger(Passenger-ID, name, phone number, location, destination, password,\*freelance-id)
* Hired drivers (hired-ID, name, phone number, driver license, plate-number, employer , Hpassword, f-name, l-name, \*Admin-id)
* Freelance Taxis (freelance-id, license, firstname, lastname plate number, password, username, \*admin-id)

The above information can be illustrated on the following diagrams

### ER diagram



# ARCHITECTURE OF THE JETME DATABASE



The architecture of the database for a taxi passenger positioning system typically involves the following components:

* **Storage:** The database stores the data about passengers and taxis in a structured way, using tables and relationships between them.
* **Query processing:** The DBMS provides tools for querying and retrieving data from the database using SQL (Structured Query Language).
* **Access control:** The DBMS controls access to the data in the database, using security features such as user authentication and permission management.
* **Recovery:** The DBMS provides features for managing transaction logging and recovery, ensuring that the database can be restored to a consistent state in case of data corruption or other issues.
* **Performance tuning:** The DBMS provides tools for optimizing the performance of the database, including indexing and other optimization techniques.

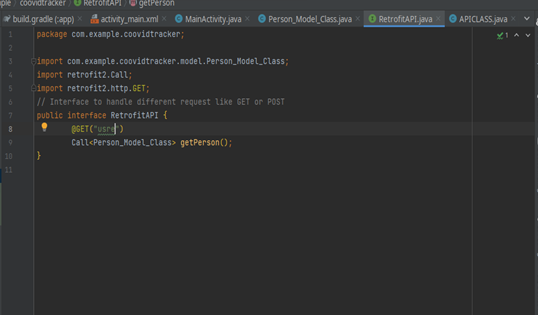
# IMPLEMENTATION OF Relational DBMS OF THE PASSENGER POSITIONING SYSTEM

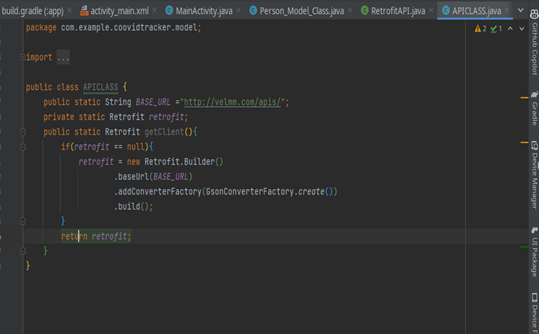
The implementation of this system was done using the XAMP as the software tool and retrofit as the library tool

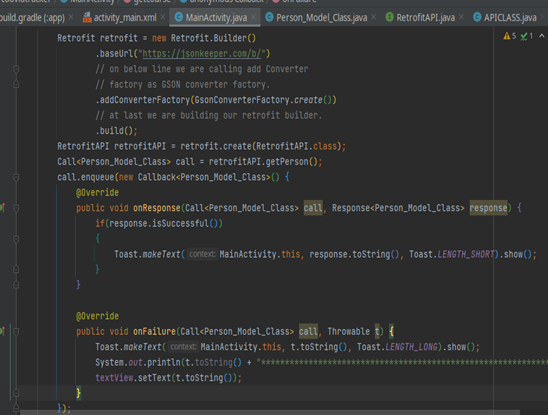
**What is retrofit?**

Retrofit is a type of library for Android that allows developers to easily make HTTP requests to a web service or API. It simplifies the process of sending and receiving data from a server by providing a high-level interface that abstracts away many of the low-level details

Below is the implementation of the retrofit to permit interaction with the database from android studio







Below is the database implementation files that enable communication with the application

