CEF440: INTERNET PROGRAMMING (J2EE) AND MOBILE PROGRAMMING

Presented by:

Name	Matricule
Acha Rha'ah Achubang	FE20A001
Amehmbo Ngewung Sonia	FE20A008
Atem Randy Asong	FE20A014
Tambe Salome Ntoh	FE20A109
Tiokeng Samuel Edward	FE20A110

Table Of Content

I. Introduction

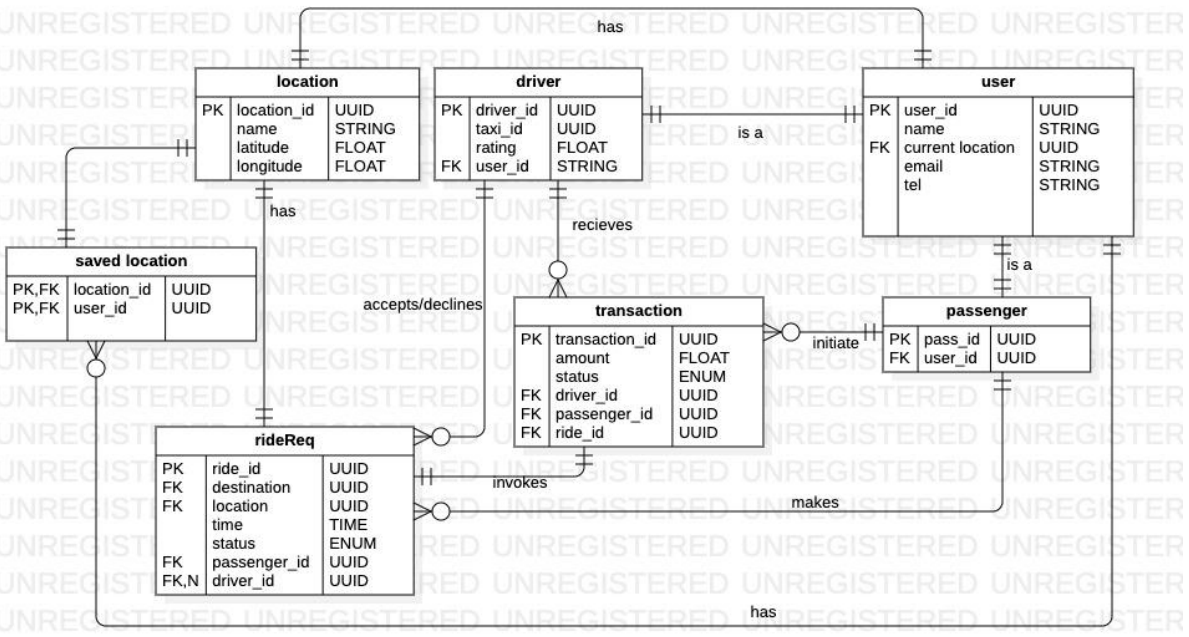
The purpose of this document for the passenger positioning app is to provide a guide for building and implementing the app's database. It will serve as a blueprint for developers involved in designing the app's database. This document will ensure that everyone involved in the project is on the same page regarding the requirements, features, and functions needed for the app's database to work effectively. Overall, the database design document aims to create a well-designed, efficient, and secure database that meets the needs of the passenger positioning app.

For this passenger positioning system, a relational database would be implemented. By using a relational database, we will enforce data integrity with the use of constraints which are rules that define how data is stored in the database. Relational databases are also highly performant in accessing and querying data from the database.

Below, we would elaborate on the database design by defining the entities in the system, the relationships that exist between them, their attributes and the constraints on those attributes.

II. Database Implementation

Designing a database for a passenger positioning system involves considering the various entities involved in the process, their attributes and relationships between various entities.



Entities

Here, we are going to discuss the various entities in our system and their attributes

User

This entity describes a user of the system. Attributes here will hold base user information and will be used by other entities.

Name	Type	Description
user_id (PK)	UUID	The Primary key for the identifying this entity
name	string	The user's full name
current location (FK)	string	User's current location
email	string	The user's email address
tel	string	User's phone number

Passenger

This entity describes a passenger.

Name	Type	Description
pass_id (PK)	UUID	Primary Key
user_id (FK)	UUID	Foreign key to user entity which holds the passenger's

Driver

This entity describes a driver who has a taxi number and a rating. Passenger's request and pay for drivers' services

Name	Type	Description
driver_id (PK)	UUID	Primary Key
user_id (FK)	UUID	Foreign key to user entity which holds the user's details
taxi_number	string	A unique identification number for a driver's taxi
rating	float	A number that describes the driver's rating.

Ride Request

This entity describes a request for a ride made by a passenger which can be accepted by any driver.

Name	Type	Description
ride_id (PK)	UUID	Primary Key
passenger_id (FK)	UUID	FK to passenger who made the request
driver_id (FK)	UUID	FK to the driver who accepts and executes the request.
destination(FK)	UUID	FK of the a location to of where a passenger is going to
location(FK)	UUID	FK to the passenger's current location
status	string	The current status of the request. It can either be Pending, Accepted, Completed, Cancelled.
time	time	The amount of time the passenger would need the driver to be present.

Location

This entity describes a user address.

Name	Type	Description
location_id (PK)	UUID	Primary Key
long	number	Longitude of the point
lat	number	Latitude of the point
name	string	The actual location name

Transaction

This entity describes the payments made in the system.

Name	Type	Description
transaction_id (PK)	UUID	Primary Key
amount	number	Amount paid in transaction
status	enum	The current status of the transaction. It can either be Pending, Accepted, Completed, Cancelled.
driver_id (FK)	UUID	FK to the driver who receives payment
passenger_id (FK)	UUID	FK to the passenger who makes the payment

Relationships

Entities	Relationship	Description
User - Passenger	Is A	A passenger is a type of user
User - Driver	Is A	A driver is a type of user
User - Location	One-to-one	A User has one current location (Location)
Location - SavedLocation	One-to-one	A saved location is a location
SavedLocation - User	many-to-one	A user can have many saved locations and a saved location can belong to one user.
RideRequest - Location	Many-to-one	A ride request has one location.
Passenger - RideRequest	One-to-many	A passenger can have many ride requests and a ride request can belong to one passenger.
Driver - RideRequest	One-to-many	A driver can have many ride requests and a ride request can belong to one driver.
Passenger - Transaction	One-to-Many	A passenger can have many transactions and a transaction can belong to one passenger
Driver - Transaction	One-to-many	A Driver can have many transactions and a transaction can belong to one driver
RideRequest - Transaction	One-to-One	A transaction is made for one ride request.