Subject : SECD2523 Database

Task : Database Conceptual Design (ERD)

Lecturer : Haslina Binti Hashim

Section : 03

Name	**Matric Id**
Adlyn Natasya Binti Aznul Rizal	A24CS0032
Afeefa Nazneen Binti Basheer @ Basheer Ahmed	A24CS0033
Ezralyn A/P Dayalan	A24CS0069
Syazwina Marsya Binti Shahrizan	A24CS0196

Table of contents

1.0 Introduction

The pervasive digital transformation has profoundly revolutionized daily life, placing transportation at the forefront of this significant shift. In this evolving landscape, commercial ride-hailing and carpooling platforms like Grab, Kumpool, and Maxim have undeniably become commonplace, offering unparalleled convenience and efficiency in urban mobility. They have successfully democratized access to transportation and redefined personal logistics for millions.

However, despite their widespread integration, these services often lack the bespoke features and targeted community focus particularly desired within a dynamic university environment. Their general-purpose design means they are not inherently optimized for the specific logistical demands and unique user groups present in such a setting.

Recognizing the escalating demand for e-hailing services among UTM students and observing certain limitations in existing general-purpose platforms, UTM Fleet has identified a critical need to implement a dedicated system. This new system is envisioned to provide an unparalleled level of satisfaction for both users and administrators, designed exclusively for UTM students to conveniently book or access rides from a demonstrably trusted and verified source, thereby enhancing daily convenience and operational efficiency within the campus.

With this forward-looking vision, we embark on the creation of UTM Drive, UTM Fleet's very own e-hailing service. Its initial implementation will be tailored specifically for UTM students, meticulously designed to achieve unparalleled convenience and operational excellence, thereby fostering its sustained success and future scalability. This proposal details our comprehensive roadmap for the development of the UTM Drive system, as a foundational component within UTM Fleet's evolving transportation strategy.

2.0 Data Flow Diagram (DFD) To-Be

Context Diagram

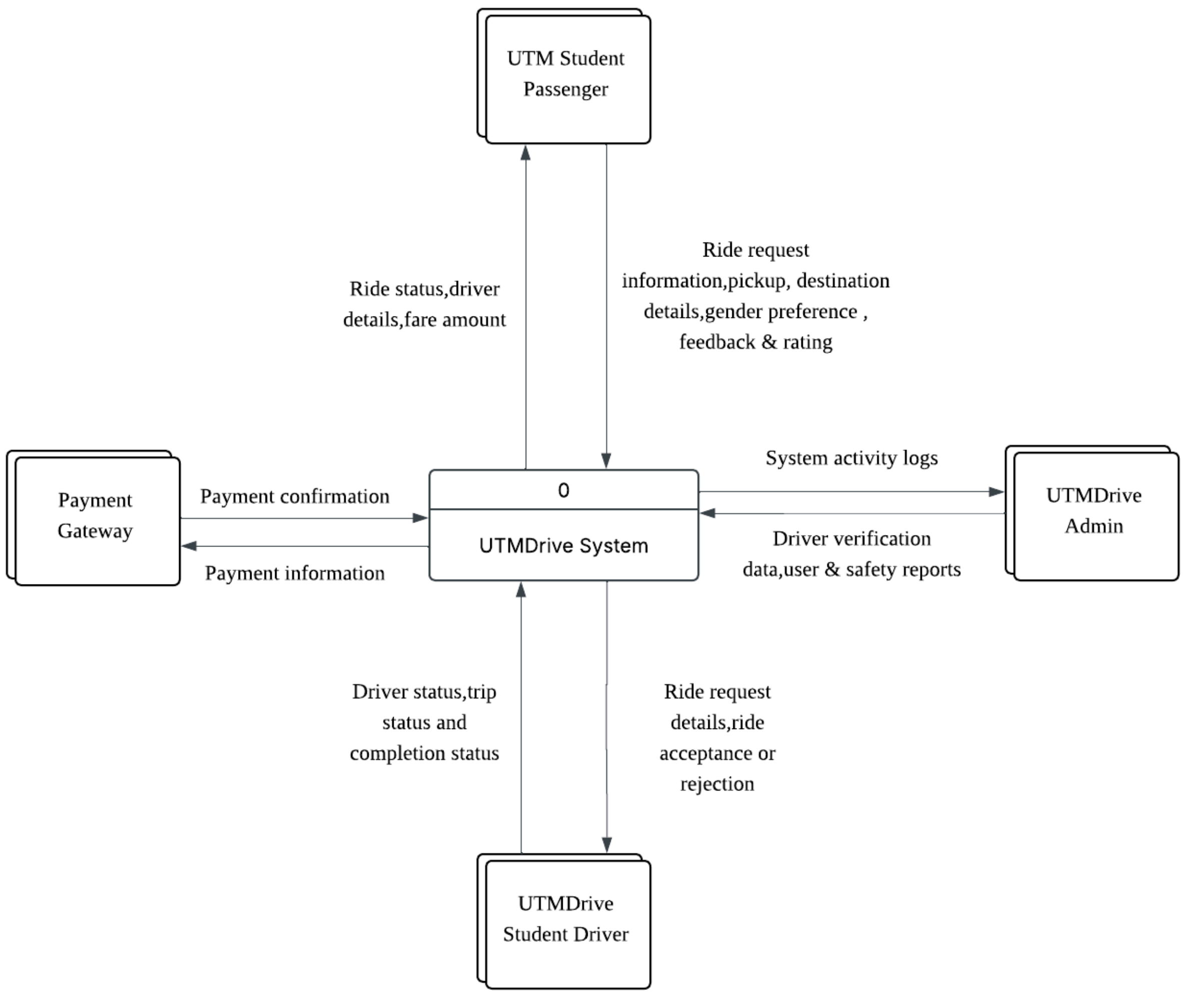
UTM Student Passenger
Ride status,driver details,fare amount
Ride request information,pickup, destination details,gender preference , feedback & rating
Payment Gateway
Payment confirmation
Payment information
0
UTMDrive System
System activity logs
Driver verification data,user & safety reports
UTMDrive Admin
Driver status,trip status and completion status
Ride request details,ride acceptance or rejection
UTMDrive Student Driver

Figure 2.1

Level 0 Diagram

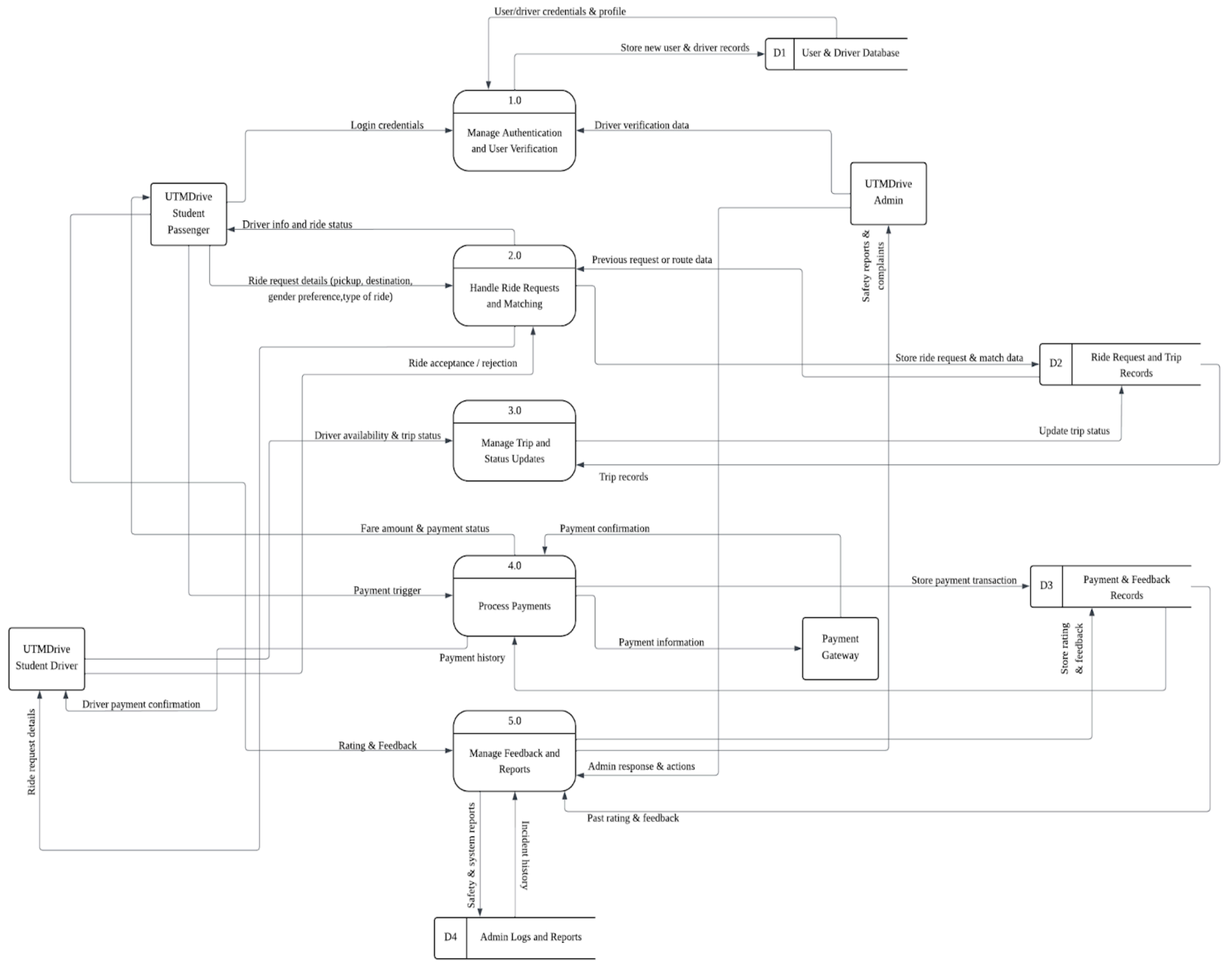
User/driver credentials & profile
Store new user & driver records
D1
User & Driver Database
1.0
Manage Authentication and User Verification
Login credentials
Driver verification data
UTMDrive Admin
UTMDrive Student Passenger
Driver info and ride status
2.0
Handle Ride Requests and Matching
Previous request or route data
Ride request details (pickup, destination, gender preference,type of ride)
Safety reports & complaints
Store ride request & match data
D2
Ride Request and Trip Records
Ride acceptance / rejection
3.0
Manage Trip and Status Updates
Driver availability & trip status
Update trip status
Trip records
Fare amount & payment status
Payment confirmation
4.0
Process Payments
Payment trigger
Store payment transaction
D3
Payment & Feedback Records
Payment information
Payment Gateway
UTMDrive Student Driver
Payment history
Store rating & feedback
Driver payment confirmation
5.0
Manage Feedback and Reports
Rating & Feedback
Admin response & actions
Ride request details
Past rating & feedback
Safety & system reports
Incident history
D4
Admin Logs and Reports

Figure 2.2

Child Diagram

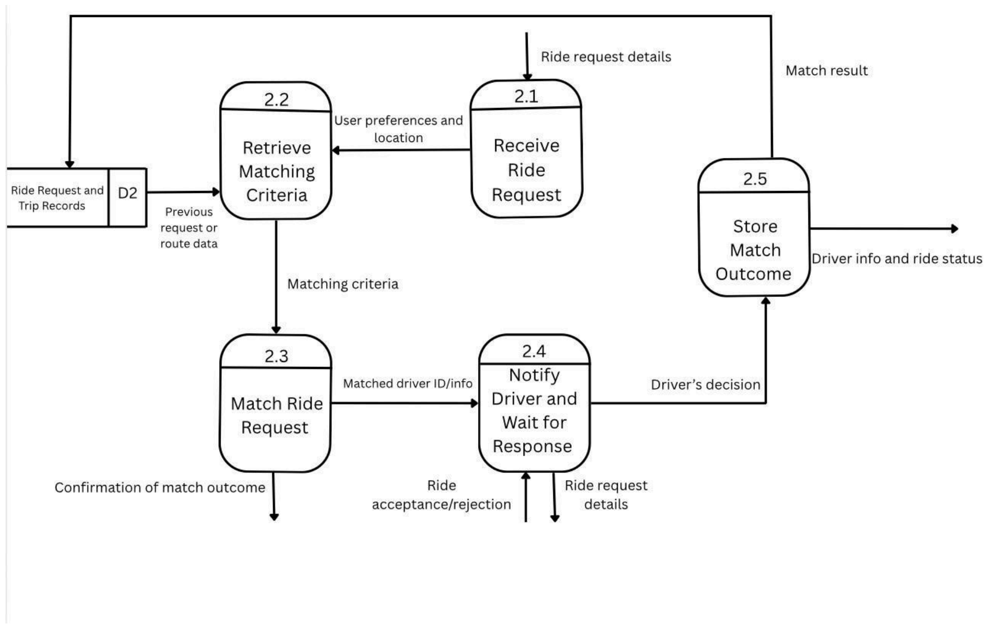
Ride request details
Match result
2.2
Retrieve Matching Criteria
User preferences and location
2.1
Receive Ride Request
Ride Request and Trip Records
D2
Previous request or route data
2.5
Store Match Outcome
Driver info and ride status
Matching criteria
2.3
Match Ride Request
Matched driver ID/info
2.4
Notify Driver and Wait for Response
Driver's decision
Confirmation of match outcome
Ride acceptance/rejection
Ride request details

Figure 2.3

3.0 Data and Transaction Requirements

3.1 Proposed Business rule

1. **UTM Student (passenger)**

Account and Access

- Login using UTM Single Sign-On (SSO) account to access UTMDrive
- Only verified UTM students can request rides
- Passenger profiles must include name, matrc_ID and verified gender

Ride Request

- Input pickup and destination locations before booking
- Choose gender preference for driver (female only or no preference)
- Choose between individual ride or carpool option

During Ride

- Track ride in real-time via in-app map for safety
- Passenger ready at the pickup point
- Payment is automatically processed via the app upon ride completion (fixed fare)

After Ride

- Provide rating (1–5 stars) and leave review for driver
- Submit complaints or incident reports

2. **UTM Student (driver):**

Account & Access

- Login using UTM Single Sign-On (SSO) account to access UTMDrive
- Verified UTM student with an active driver profile

Availability

- Go “Online” to receive ride requests

Ride Acceptance

- Receives notification of nearby booking request
- View passenger info, gender preference, route and fare
- Accept bookings that match gender preference requirements

During Ride

- Use in-app navigation to ensure route accuracy

After Ride

- Complete trip in the app once drop-off is done
- Fare and trip details are automatically recorded in the driver’s dashboard
- Driver receives passenger’s rating and review after completion

3. **Admin:**

User Management

- Verify student status (via SSO integration)
- Manages driver approvals, license validation and vehicle verification
- Authority to suspend or deactivate users for policy violations

System Management

- Manages fare structures, fixed fare rates and carpool pricing
- Oversees real-time ride monitoring and safety alerts
- Handles reported incidents, disputes and feedback moderation

Data & Analytics

- Tracks ride history, performance and usage statistics
- Generates reports on active drivers, completed trips and user ratings
- Ensures data privacy compliance and proper handling of user information

3.2 Proposed Data and Transactional Requirement

3.2.1 Data Requirement

ENTITY	DATA TO BE STORED	REQUIREMENTS OF DATA
Student	1. matric_id 2. username 3. name a. first_name b. last_name 4. email 5. password 6. phone_num 7. program_code 8. faculty 9. role	- matric_ID and username is unique - Students need to login using their username and password before using the app - Data 3-9 will be taken from UTM database
Driver	1. driver_id 2. matric_id 3. license_num 4. car_plate 5. car_model 6. car_color 7. verification_status	- driver_id is unique and must match a matric_id in Student - verification_status must be “Accepted” for driver to start driving
Admin	1. admin_id 2. password	- Admin_id is unique
Ride_request	1. request_id 2. driver_id 3. pickup_location 4. destination	- request_id is unique - driver_id in Ride_request must match the driver_id in Driver or the driver_id

	5. distance 6. matric_id 7. gender_pref 8. ride_type 9. seater 10. request_status	is null in case of cancel/ongoing request
Trip	1. trip_id 2. matric_id 3. driver_id 4. pickup_location 5. destination 6. pickup_time 7. dropoff_time 8. date	- trip_id is unique - matric_id is the student that rides with driver - driver_id in Trip must match the driver_id in Driver
Payment	1. transaction_num 2. total_price 3. trip_id	- transaction_num is unique - trip_id in Payment must match the trip_id in Trip
Feedback	1. feedback_id 2. rating 3. comment 4. matric_id 5. driver_id	- feedback_id is unique - driver_id in Feedback must match the driver_id in Driver - matric_id is the student that give feedback to a specific driver

3.2.2 Transactional Requirement

Transaction	Description	Actors Involved	Data Entry	Data Update / Deletion	Data Querries
Student	New users (students) register into the system using UTM Single Sign-On (SSO)	Student Driver Admin	Register new student details into the system.	Update student profile, contact info and other details. Delete student account by admin.	Search and view student information by admin.
Driver	New drivers (students) register and submit all the details using UTM Single Sign-On (SSO)	Driver	Register new driver details into the system.	Update driver profile, contact info and other details. Delete driver account by admin.	Search and view driver information by admin.
Admin	Verify UTMstudents as drivers by submitting vehicle and driving license details	Admin	Register new driver details into the system.	Delete invalid or canceled trip, invalid payment record, student and driver account by admin	Admin can list verified, pending, and rejected drivers for monitoring purposes.
Ride_reque st	A passenger submits ride details including pickup and dropoff location points,gender preference, ride type (solo or carpool) and seater (4-seater or 6-seater)	Student	Enter ride request details into the system.	Update or cancel ride request before confirmation.	Search ride requests or view ride details by passenger or admin.

Trip	Passengers create or join shared rides with others traveling in the same direction.	Passenger	Create or join trip record in system (carpool).	Update trip status (requested, accepted, ongoing, completed). Delete invalid or canceled trips by admin.	View current and past trips by passenger, driver, or admin.
Payment	The system records fare payment after ride completion and updates wallet balance.	Passenger / System	Record payment transaction after ride completion.	Update wallet balance or transaction history. Delete invalid payment record by admin.	View wallet balance, payment history, or transaction details.
Feedback	Users submit ratings and reviews after completing a ride.	Passenger	Submit new feedback or rating.	Edit or delete feedback within a limited time period.	View driver feedback history and ratings by passengers or admin.

4.0 Database Conceptual Design

4.1 Conceptual ERD

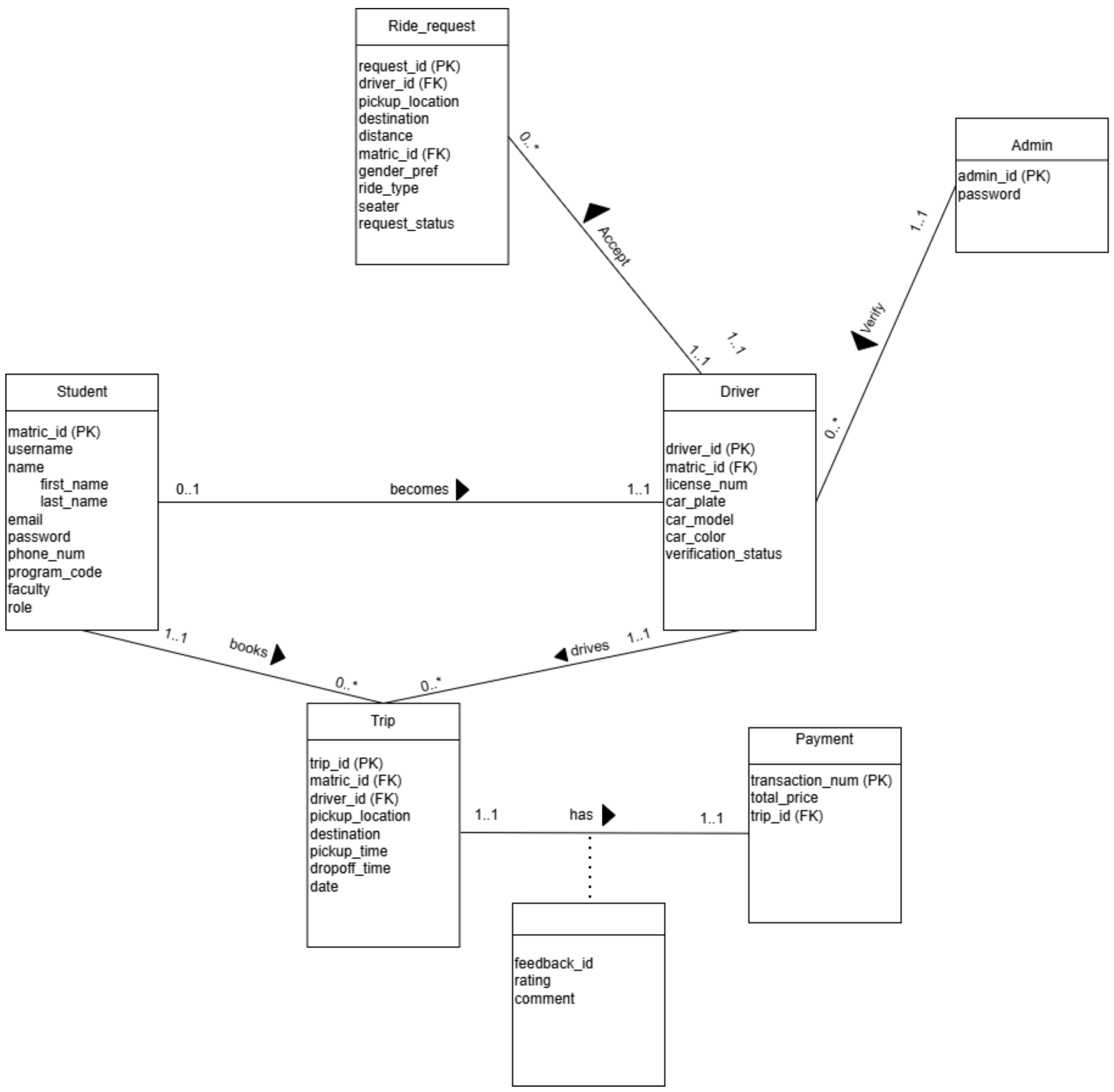

Figure 4.1.1

5.0 Data Dictionary

5.1 Description of Entity

Entity	Description	Occurence
Student	Stores details of UTM students who register and use the app as passengers or drivers.	Each student can make many ride requests, give feedback, and optionally register as a driver.
Driver	Store car and license details of UTM Driver	Each driver belongs to one student and may handle multiple trips and payments.
Admin	Contains information about system administrators responsible for managing users, rides, and reports.	Each admin can monitor multiple students, drivers, and payments.
Ride_Request	Keeps record of ride bookings, including pickup, destination, and gender preference.	Each student can make many ride requests; each request may be assigned to one driver.
Trip	Holds trip details for completed rides.	Each trip belongs to one student and one driver, and generates one payment.
Payment	Records payment information for each completed trip.	Each trip produces one payment; managed by admin.

5.2 Description of Relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
Student	1..1	provide	1..1	Feedback
	1..1	make	1..1	Ride_request
	0..1	become	1..1	Driver
	1..1	books	0..*	Trip
Driver	1..1	accept	0..*	Ride_request
	1..1	receive	0..*	Feedback
	1..1	have	0..*	Trip
Admin	1..1	verify	0..*	Driver
	1..1	review	0..*	Feedback
Trip	1..1	has	1..1	Payment

5.3 Description Attributes

Entity	Attribute	Description	Data Type	Null	Multi-valued
Student	matric_id	Uniquely identifies a student (Primary Key)	VARCHAR(9)	no	no
	username	Student’s login username	VARCHAR(20)	no	no
	first_name	Student’s first name	VARCHAR(30)	no	no
	last_name	Student’s last name	VARCHAR(30)	no	no
	email	Student’s utm email	VARCHAR(30)	no	no
	password	Student’s login password	VARCHAR(20)	no	no
	phone_num	Student’s contact number	VARCHAR(15)	no	no
	program_code	Student’s program code	VARCHAR(5)	no	no
	faculty	Student’s faculty name	VARCHAR(20)	no	no
	role	Specifies whether the student is a passenger or driver	VARCHAR(10)	no	no
Driver	driver_id	Uniquely identifies driver (Primary Key)	VARCHAR(10)	no	no
	matric_id	References the student who registered as a driver	VARCHAR(9)	no	no
	license_num	Driver’s license number	VARCHAR(15)	no	no
	car_plate	Vehicle registration number	VARCHAR(10)	no	no

	car_model	Model of the driver’s car	VARCHAR(10)	no	no
	car_color	Color of the driver’s car	VARCHAR(15)	no	no
	verification_status	Indicates whether the driver has been verified by the admin	VARCHAR(10)	no	no
Admin	admin_id	Unique identifier for each admin (Primary key)	VARCHAR(10)	no	no
	password	Admin’s login password	VARCHAR(20)	no	no
Trip	request_id	Uniquely identifies each ride request	VARCHAR(10)	no	no
	driver_id	References the driver who got accepted or completed the trip	VARCHAR(10)	no	no
	pickup_location	Location where the student will be picked up	VARCHAR(50)	no	yes
	destination	Location where the student will be dropped off	VARCHAR(50)	no	yes
	pickup_time	Time of the student got picked up	DATETIME	no	no
	dropoff_time	Time of the student got dropped off	DATETIME	no	no
	date	Date of the trip	DATE	no	no
Payment	transaction_num	Unique identifier for each payment transaction (Primary Key)	VARCHAR(10)	no	no

	total_price	Total fare amount for the trip	DECIMAL(8,2)	no	no
	trip_id	References the completed trip associated with the payment	VARCHAR(10)	no	no
	feedback_id	Unique identifier for each feedback entry (Primary key)	VARCHAR(10)	no	no
	rating	Rating score given by the student (1–5)	INT	no	no
	comment	Written feedback about the driver or trip	TEXT	yes	no

6.0 Summary

In this phase, our group has gained better understanding while creating the conceptual ERD diagram and enhanced ERD diagram for our proposed system, UTMDrive. We started by updating our data and transactional requirements, and proceeded to identify the entities and attributes of each entity.

We also created the Data Flow Diagram (DFD) to-be, which included context diagram, level 0 diagram and child diagram, clearly showing the process of managing authentication and user verification, handling ride requests and matching, managing trip and status updates, process payment and managing feedback and reports. As a result, we understand better about the interaction between entities and processes.

After that, we explored how to determine all possible relationships between entities, including the multiplicities for each relationship. There are entities Student, Driver, Admin, Ride_request, Trip, Payment and Feedback. We successfully designed the conceptual ERD diagram to show how the data related to each entity. We also included the enhanced ERD diagram to provide a more detailed relationship between entities.

Lastly, we provided the data dictionary, which describes the entity, relationship and attributes in the system. In conclusion, we managed to complete this phase together by giving ideas and getting ready for the next phase of the project.