TRANSIT TRACKER

IS 6420-002 Fall 2023 Database Theory & Design

Group Members: -

Akesh Anumalasetty
Akilesh Thuniki
Fnu Adarsh
Raja Sirum

Table of Contents

EXECUTIVE SUMMARY

Optimizing public transportation in the digital transformation requires real-time processing and intelligent data management. Due to growing urban populations and rising mobility demands, public transportation services rely on the use of technology to improve operational efficiency and commuter experience. Initiated to address these needs, this project aims to improve route efficiency, increase user engagement, and streamline access to transit data within the Utah Transit Authority (UTA).

Our mission is to transform UTA's transportation by offering a reliable data management system that combines easily navigable accessibility with real-time transit information. We aimed to decrease wait times, increase commuter satisfaction, and maximize transit resources.

VISION

To provide an integrated transit system that redefines the concept of city travel, ensuring it is effortless, quick, and pleasant for every citizen. By harnessing advanced technology, we aim to significantly enhance efficiency, dependability, and overall quality of transportation services, making our urban landscape more accessible than ever.

OBJECTIVE

The objective of this project is to improve public transportation by implementing a comprehensive and user-friendly system that provides updates on buses and trains, ensuring passengers have accurate arrival information. This will be accomplished by utilizing cutting-edge technology to monitor and improve service efficiency while putting accessibility for people of all ages and abilities first. Furthermore, the project intends to promote the environmental benefits of taking public transportation, emphasizing its role in lowering air pollution. Regular community engagement will take place to ensure that transit services meet the diverse needs of our city's residents. Furthermore, the project will constantly investigate new ideas and technologies to improve the travel experience, focusing on adaptability to meet the changing needs of the city's resident’s population and infrastructure.

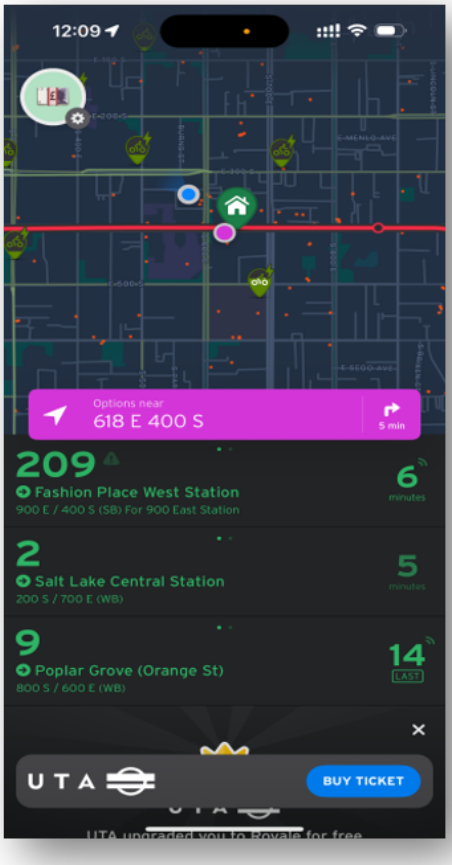
ABOUT TRANSIT

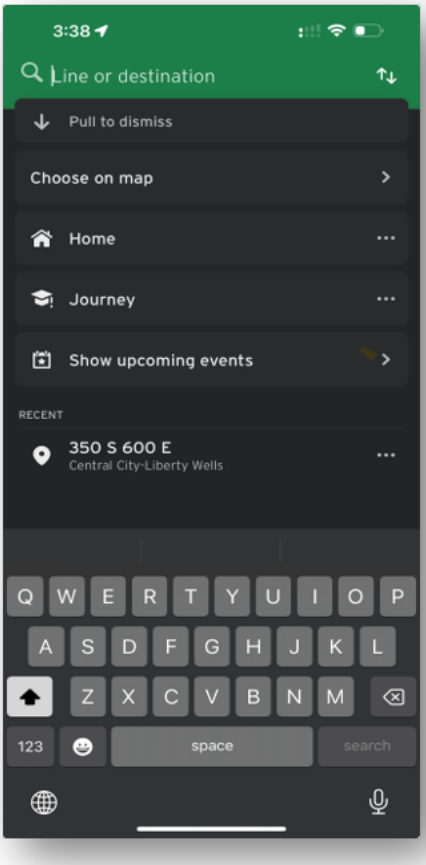
Urban transportation relies heavily on transit to give residents of cities cost-effective and efficient mobility options. Buses and trax are examples of transit systems that operate on fixed routes with predestined stops. The government or private sector may oversee these services, and payment methods include cash, tickets, and digital platforms. Modern technology allows passengers to receive real-time updates on arrivals and possible delays via digital displays and mobile apps. To ensure that all passengers are included, accessibility features like wheelchair ramps and audio announcements are essential.

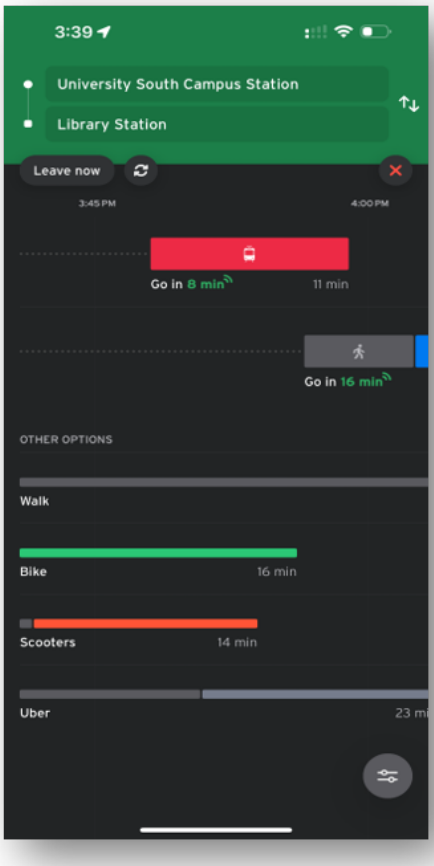
Additionally, public transportation is essential to the sustainability of the environment. Cities that are cleaner and easier to live in have reduced emissions and traffic congestion. Coordinating with other forms of transportation, like walking and cycling, makes travel more accessible. Using technological advancements and interacting with local communities are essential components of developing and improving transit systems. By prioritizing these components, cities can create robust, environmentally responsible transit systems that improve urban mobility and quality of life. To formulate innovative, sustainable cities that encourage accessibility, community, and sustainability, public transit is essential.

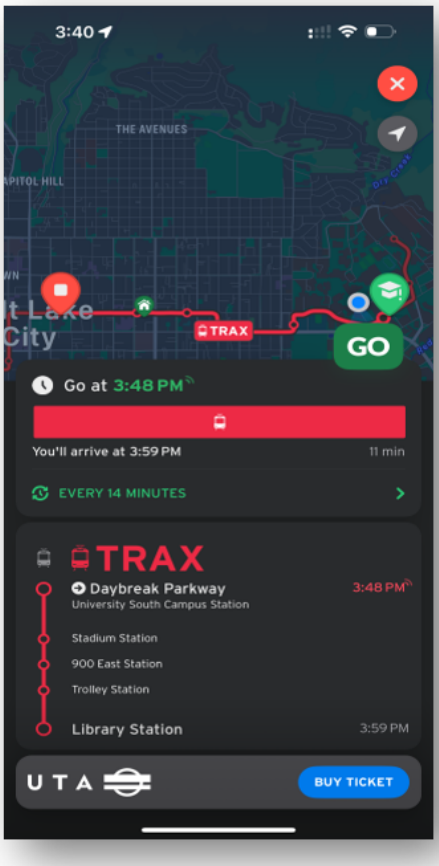
SNAPSHOTS OF TRANSIT APPLICATION

These are the snapshots of the transit application, which shows the application flow from the home page until the user sees the transport mode.






3:39
University South Campus Station
Library Station
Leave now
3:45 PM
4:00 PM
Go in 8 min
11 min
Go in 16 min
OTHER OPTIONS
Walk
Bike
16 min
Scooters
14 min
Uber


3:40
THE AVENUES
TRAX
GO
Go at 3:48 PM
You'll arrive at 3:59 PM
11 min
EVERY 14 MINUTES
TRAX
Daybreak Parkway
University South Campus Station
3:48 PM
Stadium Station
900 East Station
Trolley Station
Library Station
3:59 PM
UTA
BUY TICKET

PRODUCT AND SERVICES

BUS:

Buses are a popular means of transportation. When picking up or dropping off people, they travel along specified paths and specific stations. They are frequently utilized because they are inexpensive to ride and simple to locate. Buses these days are designed to serve everybody, including those who require special assistance or use wheelchairs. Additionally, they are becoming more environmentally friendly, new models run on electricity rather than gas. Buses are a vital component of the city's transportation network because they enable people to go places, they need to, like work or school.

@ Reference :-

TRAX (LIGHT RAIL):

Trax is a well-liked means of quickly transporting large numbers of people over longer distances. Because they're much faster than other options, they're crucial for journeys between cities. Some trax have a speed limit to travel very far. Additionally, trax are accessible to all, with features designed specifically for those with disabilities. They are also better for the environment because they consume less power and emit less pollution. Railways in more extensive transportation networks facilitate travel to many locations, enhance our quality of life, and maintain cleaner air.

@Reference:https://www.fox13now.com/news/local-news/gas-leak-in-downtown-salt-lake-city-halts-trax-service

TRANSACTIONAL DATABASES

The transactional database functions as an interface for all focused-on user operations through the complex system of the Transit application, serving a foundational and transformative role. The record data is in the database upon the user's account creation. The interface is made to be highly responsive, allowing users to easily modify stored data and react in real-time to new inputs while traveling through establishing their preferences, especially regarding location. Creating a dynamic, interactive experience that feels instinctively responsive to each user's needs.

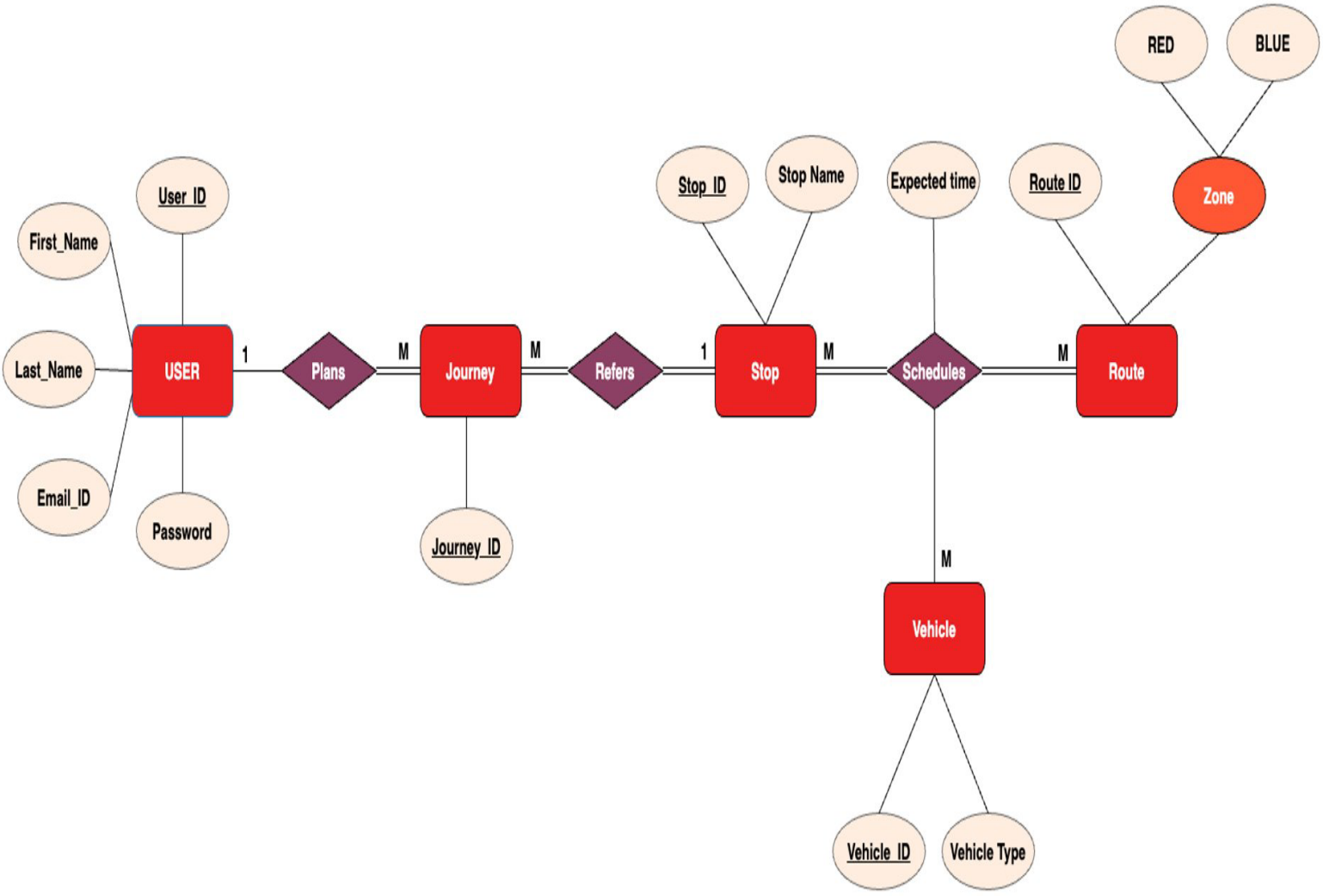
INITIAL REQUIREMENTS

This project is to develop a user-friendly, streamlined transit application database that will enable users to plan their trips more effectively. We aim to build an initial structure that considers how users engage with the application, focusing on the relationship between the application and its database. Simply put, we're re-engineering the database to serve as a strategic interaction that changes the entire user experience, from creating an account to arriving at their destination and showing the fastest route for their destination.

Users must create an account by providing basic personal information like first name, last name, email ID, and a new password. After signing up, users can use the application's primary feature, trip planning. Users must enter their current and desired locations for the database to function. The system sorts through various modes of transportation using pre-stored static data, determines the quickest route according to transport availability, and promptly displays this information to the user.

Moreover, the project optimizes the user experience. The main goal is to ensure that the database is carefully selected to provide the finest route options, enhancing the user experience overall and that the backend operations run easily.

CONCEPTUAL MODEL



The conceptual model consists of five main entities. User, Journey, stop, route and vehicle. The "USER" entity is represented by attributes like "FirstName," "LastName," "EmailID," and "Password", where each user is uniquely identified by a unique 'UserID', and has "1 to M" (one-to-many) relationship between "USER" and "Journey". This signifies that each user can plan multiple journeys, while each Journey should be planned by at least one user for a journey to exist.

The "Journey" entity serves as a link between the "user" and the "stops", and it is distinguished by a unique 'Journey ID'. A "M to 1" (many to one) relationship between "Journey" and "Stop". indicates that each route will include at least one stop to begin and conclude the journey, with several stops in between as they are part of the journey

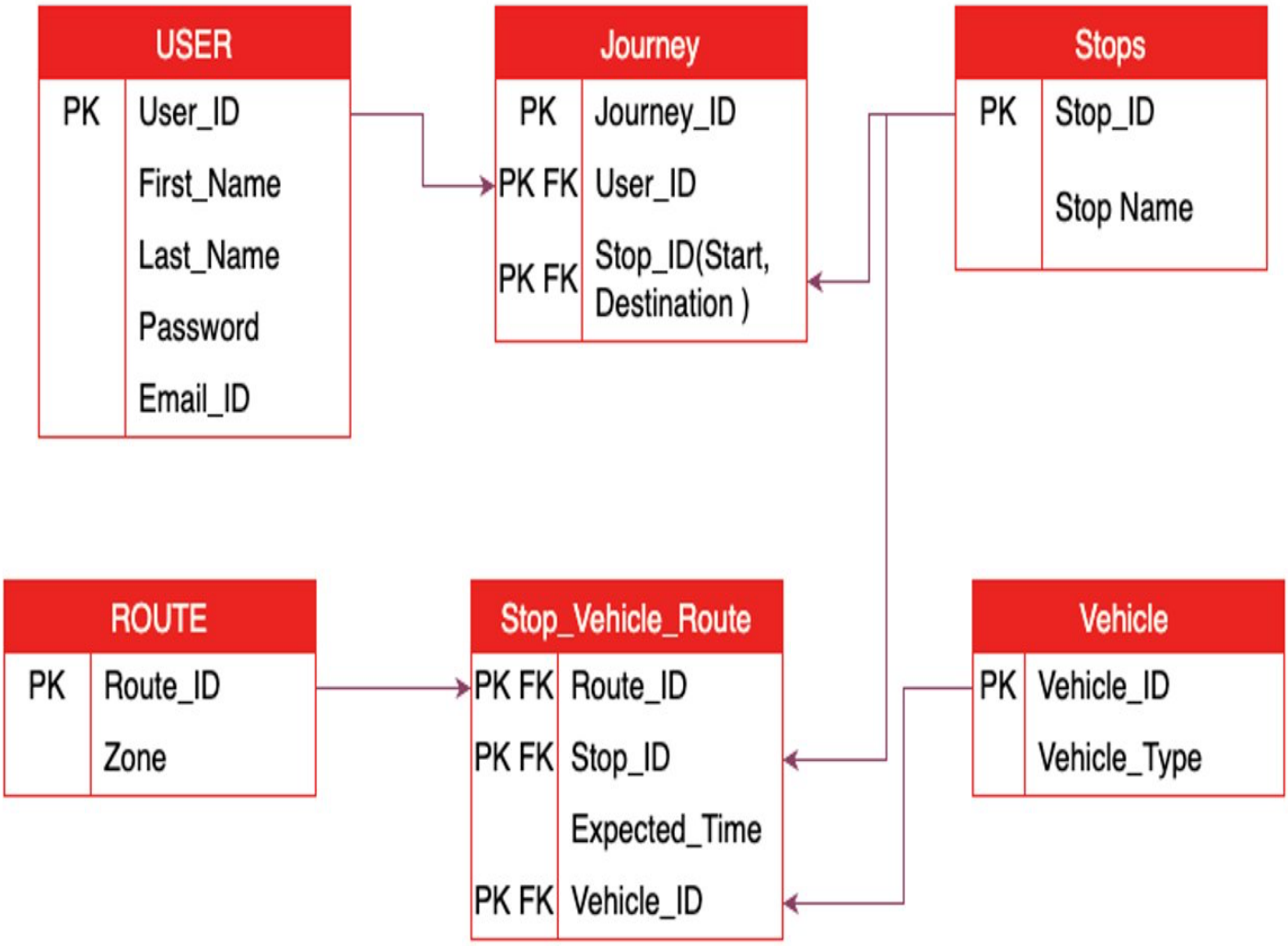
The “Stop” entity refers to the station where the user can wait for the vehicle to arrive at beginning or ending point. Where each stop is uniquely identified by a unique ‘Stop ID’ and has an attribute called Stop name for each stop ID, the stop entity has an “M to M” (Many to Many) relation with the Route entity where each stop can present in multiple routes as some times both routes can have common stops in between

to reach the destination as well as the route can have multiple stops in between start and end as stops are part of the route and a route is defined through stops.

The entity "Route" is specified as a set path with many stops located between the start and finish of each route, and each route may share the same stops based on their path. A unique 'Route ID' identifies each route, and each route is grouped into a zone based on its location.

The entity "Vehicle" refers to the mode of transportation for the journey, with each vehicle uniquely identified by "Vehicle ID." It has a vehicle type attribute that allows you to select the desired vehicle. Instead of having a binary relationship between the vehicle and the stop and the vehicle and the bus, the vehicle entity forms a ternary relationship with the stop and route entities, ensuring that at a stop where the vehicle arrives, is always in the context of route avoiding redundancy.

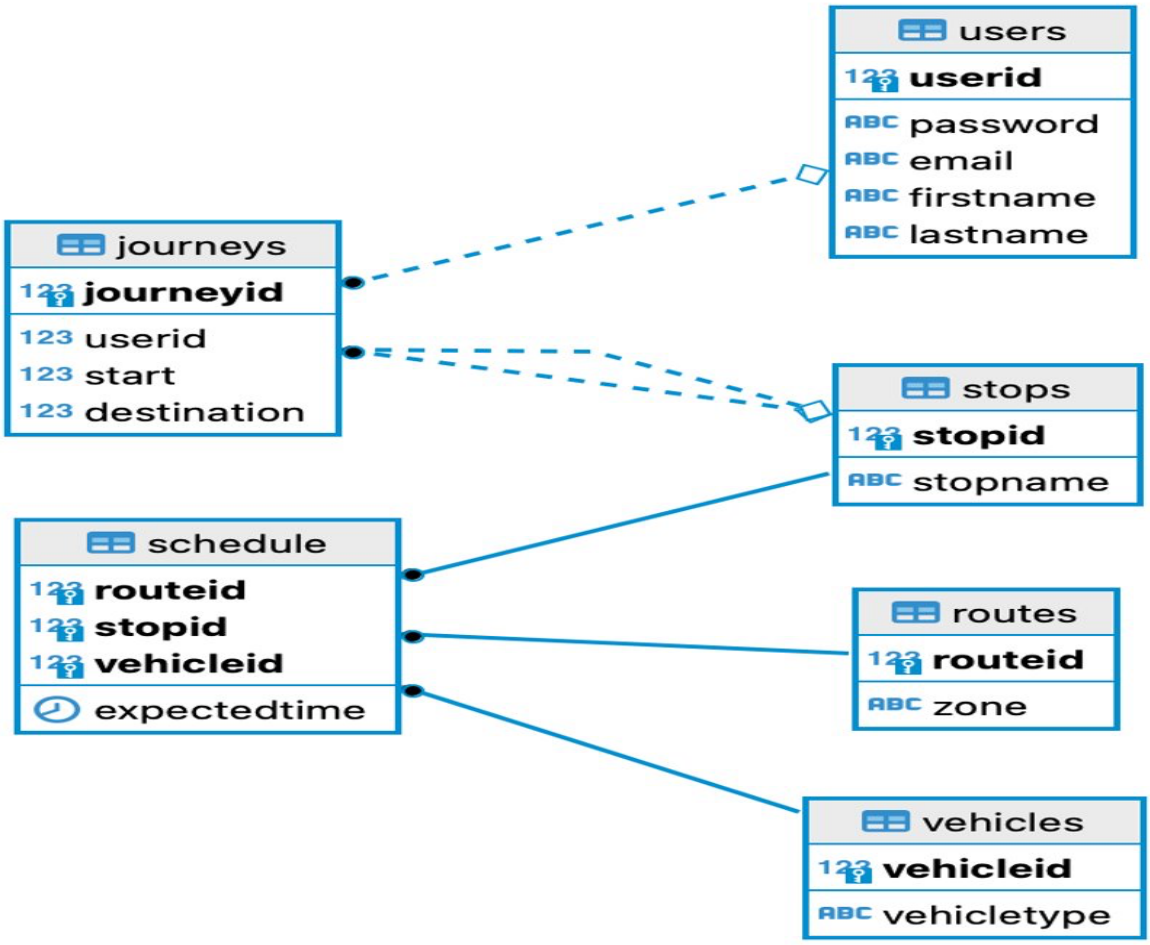
LOGICAL MODEL



The logical diagram has six tables: user, journey, stop, route, “StopVehicleroute”, and vehicle table. The user table consists of five attributes: "UserID," "FirstName," "LastName," "Password," and "EmailID." "UserID" acts as the Primary Key, a foreign Key to the journey table. In the journey table, we have three attributes, which are "JourneyID," "UserID," and "StopID." all three attributes act as primary Key but "UserID" and "StopID" are foreign Key for the stop table and user table. In the Stop table, there are two attributes, one of which is the primary Key, "StopID," And another is "Stopname."

The route table has two attributes: "RouteID," the primary Key, and the "zone" attribute. In the stopvehicleroute table, there are four attributes, of which three are both primary and foreign Key they are "RouteID," "StopID," and "VehicleID," and the fourth attribute is "ExpectedTime." And the final table in the logical diagram is the Vehicle table. This table consists of two. Attributes: one of them is the primary Key, that is, "VehicleID," and another attribute is "vehicletype."

PHYSICAL MODEL



In the physical model shown above, we are able to discern the breakdown of the logical model into meaningful data about the tables

present in our database. in this diagram "users" table, each user is uniquely identified by an integer "user-id." This table also stores textual details such as the user's password, email, first name, and last name.

The "journeys" table keeps track of user trips, with each journey having a unique "journeyid" and tracked by starting point and destination, both of which are textual. This table links back to the "users" table through the "user-id" attribute. Transit stops are stored in the "stops" table, identified by "StopID" and associated with a textual "StopName." Routes within the application are defined in the "routes" table, with each route having a unique integer "RouteID" and a textual descriptor called "zone." The "schedule" table connects the route, stop, and vehicle data, which includes data related to which vehicle will be at which stop and when, as indicated by the "expected time." the "vehicles" table lists vehicles, identifying a unique "VehicleID" for each and storing its "vehicle type," such as bus or train. These elements provide a relational database that stores the application's user, transit, and scheduling data.

SQL CODE

```
CREATE SCHEMA TRANSIT
GO
DROP TABLE IF EXISTS [TRANSIT].[Routes]
DROP TABLE IF EXISTS [TRANSIT].[Stops]
DROP TABLE IF EXISTS [TRANSIT].[Users]
DROP TABLE IF EXISTS [TRANSIT].[Vehicles]
DROP TABLE IF EXISTS [TRANSIT].[Journeys]
DROP TABLE IF EXISTS [TRANSIT].[Schedule]

CREATE TABLE TRANSIT.Users (
    UserID INT NOT NULL,
    Password VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL,
    FirstName VARCHAR(100) NOT NULL,
    LastName VARCHAR(100) NOT NULL,
    CONSTRAINT userPK PRIMARY KEY (UserID)
);

CREATE TABLE TRANSIT.Stops (
    StopID INT NOT NULL,
    StopName VARCHAR(100) NOT NULL,
    CONSTRAINT StoppK PRIMARY KEY (StopID)
);

CREATE TABLE TRANSIT.Routes (
    RouteID INT NOT NULL,
    Zone VARCHAR(100),
    CONSTRAINT routepK PRIMARY KEY (RouteID)
);

CREATE TABLE TRANSIT.Vehicles (
    VehicleID INT NOT NULL,
    VehicleType VARCHAR(10) NOT NULL,
    CONSTRAINT vehiclepK PRIMARY KEY (VehicleID)
);

CREATE TABLE TRANSIT.Journeys (
    JourneyID INT NOT NULL,
    UserID INT,
```

```
    Start INT,
    Destination INT,
    CONSTRAINT journeyFK PRIMARY KEY (JourneyID),
    CONSTRAINT userFK FOREIGN KEY (UserID) REFERENCES TRANSIT.Users(UserID),
    CONSTRAINT stopFK FOREIGN KEY (Start) REFERENCES  TRANSIT.Stops(StopID),
    CONSTRAINT stopdFK FOREIGN KEY (Destination) REFERENCES TRANSIT.Stops(StopID)
);

CREATE TABLE TRANSIT.Schedule (
    RouteID INT,
    StopID INT,
    VehicleID INT,
    ExpectedTime TIME,
    CONSTRAINT schedulepK PRIMARY KEY (RouteID, StopID, VehicleID),
    CONSTRAINT routeFK FOREIGN KEY (RouteID) REFERENCES TRANSIT.Routes(RouteID),
    CONSTRAINT stopsFK FOREIGN KEY (StopID) REFERENCES TRANSIT.Stops(StopID),
    CONSTRAINT vechileFK FOREIGN KEY (VehicleID) REFERENCES TRANSIT.Vehicles(VehicleID
)
);


--User table
INSERT INTO [TRANSIT].[Users] (UserID, Password, Email,FirstName,LastName)
VALUES
(1, 'password1', 'raju.b@example.com', 'raju','m'),
(2, 'password2', 'sssgkraja@example.com', 'Raja',' Sirum'),
(3, 'password3', 'akesh.a@example.com', 'Akesh',' anumalasetty'),
(4, 'password4', 'akilesh.t@example.com', 'Akilesh',' Thuniki'),
(5, 'password5', 'fnu.adarsh@example.com', 'Adarsh',' FNU'),
(6, 'password6', 'dheeraj.w@example.com', 'dheeraj',' yata'),
(7, 'password7', 'abhinay.m@example.com', 'abhinay','fnu'),
(8, 'password8', 'charith.j@example.com', 'charith','gopavaram'),
(9, 'password9', 'harsha.b@example.com', 'harsha','g'),
(10, 'password10', 'varun.o@example.com', 'varun',' gk');


--Trax Stops
INSERT INTO  [TRANSIT].[Stops] (StopID, StopName)
VALUES
(101, 'U. Of U. Medical Center Station'),
(102, 'Fort Douglas Station'),
(103, 'University South Campus Station'),
(104, 'Stadium Station'),
(105, '900 East Station'),
```

```
(106, 'Trolley Station'),
(107, 'Library Station'),
(108, 'Courthouse Station'),
(109, '600 South Station'),
(110, '900 South Station'),
(111, 'Ballpark Station'),
(112, 'Central Pointe Station'),
(113, 'Millcreek Station'),
(114, 'Meadowbrook Station'),
(115, 'Murray North Station'),
(116, 'Murray Central Station'),
(117, 'Fashion Place West Station'),
(118, 'Bingham Junction Station'),
(119, 'Historic Gardner Station'),
(120, 'West Jordan City Center Station'),
(121, '2700 W Sugar Factory Rd Station'),
(122, 'Jordan Valley Station'),
(123, '4800 W Old Bingham Hwy Station'),
(124, '5600 W Old Bingham Hwy Station'),
(125, 'South Jordan Parkway Station'),
(126, 'Daybreak Parkway Station');


--Bus stops
INSERT INTO [TRANSIT].[Stops] (StopID, StopName)
VALUES
(142, 'U Union Building'),
(143, 'Sunnyside Ave & Guardsman Way'),
(144, '1300 S & 1500 E'),
(145, '900 E & 1700 S'),
(146, 'State & 1700 S'),
(147, 'Central Pointe Station');

--Vehicle
INSERT INTO [TRANSIT].[Vehicles] (VehicleID, VehicleType)
VALUES
(801, 'Trax'),  -- Trax vehicle with ID 801
(802, 'Trax'),  -- Trax vehicle with ID 802
(803, 'Trax'),  -- Trax vehicle with ID 803
(17, 'Bus'), --Bus vehicle with ID 17
(170, 'Bus');
```

```
--Route
INSERT INTO [TRANSIT].[Routes] (RouteID, Zone)
VALUES
(401, 'Red'),
(17, Null);

--Journey
INSERT INTO [TRANSIT].[Journeys] (JourneyID, UserID, Start, Destination)
VALUES
    (301, 1, 101, 105),
    (302, 2, 102, 106),
    (303, 3, 119, 121),
    (304, 4, 126, 120),
    (305, 5, 124, 115),
    (306, 6, 110, 114),
    (307, 7, 143, 144),
    (308, 8, 145, 142),
    (309, 9, 143, 147),
    (310, 10, 144, 142);

--Schedule
INSERT INTO [TRANSIT].[Schedule] (RouteID, StopID, VehicleID, ExpectedTime) VALUES
(401, 101, 801, '08:00:00'),
(401, 102, 801, '08:15:00'),
(401, 103, 801, '08:30:00'),
(401, 104, 801, '08:45:00'),
(401, 105, 801, '09:00:00'),
(401, 106, 801, '09:15:00'),
(401, 107, 801, '09:30:00'),
(401, 108, 801, '09:45:00'),
(401, 109, 801, '10:00:00'),
(401, 110, 801, '10:15:00'),
(401, 111, 801, '10:30:00'),
(401, 112, 801, '10:45:00'),
(401, 113, 801, '11:00:00'),
(401, 114, 801, '11:15:00'),
(401, 115, 801, '11:30:00'),
(401, 116, 801, '11:45:00'),
(401, 117, 801, '12:00:00'),
(401, 118, 801, '12:15:00'),
```

```
(401, 119, 801, '12:30:00'),
(401, 120, 801, '12:45:00'),
(401, 121, 801, '13:00:00'),
(401, 122, 801, '13:15:00'),
(401, 123, 801, '13:30:00'),
(401, 124, 801, '13:45:00'),
(401, 125, 801, '14:00:00'),
(401, 126, 801, '14:15:00');

INSERT INTO [TRANSIT].[Schedule] (RouteID, StopID, VehicleID, ExpectedTime) VALUES
(401, 126, 802, '08:00:00'),
(401, 125, 802, '08:15:00'),
(401, 124, 802, '08:30:00'),
(401, 123, 802, '08:45:00'),
(401, 122, 802, '09:00:00'),
(401, 121, 802, '09:15:00'),
(401, 120, 802, '09:30:00'),
(401, 119, 802, '09:45:00'),
(401, 118, 802, '10:00:00'),
(401, 117, 802, '10:15:00'),
(401, 116, 802, '10:30:00'),
(401, 115, 802, '10:45:00'),
(401, 114, 802, '11:00:00'),
(401, 113, 802, '11:15:00'),
(401, 112, 802, '11:30:00'),
(401, 111, 802, '11:45:00'),
(401, 110, 802, '12:00:00'),
(401, 109, 802, '12:15:00'),
(401, 108, 802, '12:30:00'),
(401, 107, 802, '12:45:00'),
(401, 106, 802, '13:00:00'),
(401, 105, 802, '13:15:00'),
(401, 104, 802, '13:30:00'),
(401, 103, 802, '13:45:00'),
(401, 102, 802, '14:00:00'),
(401, 101, 802, '14:15:00');

INSERT INTO [TRANSIT].[Schedule] (RouteID, StopID, VehicleID, ExpectedTime) VALUES
(401, 101, 803, '11:00:00'),
(401, 102, 803, '11:15:00'),
(401, 103, 803, '11:30:00'),
(401, 104, 803, '11:45:00'),
(401, 105, 803, '12:00:00'),
```

```
(401, 106, 803, '12:15:00'),
(401, 107, 803, '12:30:00'),
(401, 108, 803, '12:45:00'),
(401, 109, 803, '13:00:00'),
(401, 110, 803, '13:15:00'),
(401, 111, 803, '13:30:00'),
(401, 112, 803, '13:45:00'),
(401, 113, 803, '14:00:00'),
(401, 114, 803, '14:15:00'),
(401, 115, 803, '14:30:00'),
(401, 116, 803, '14:45:00'),
(401, 117, 803, '15:00:00'),
(401, 118, 803, '15:15:00'),
(401, 119, 803, '15:30:00'),
(401, 120, 803, '15:45:00'),
(401, 121, 803, '16:00:00'),
(401, 122, 803, '16:15:00'),
(401, 123, 803, '16:30:00'),
(401, 124, 803, '16:45:00'),
(401, 125, 803, '17:00:00'),
(401, 126, 803, '17:15:00');


INSERT INTO [TRANSIT].[Schedule] (RouteID, StopID, VehicleID, ExpectedTime)
VALUES
(17, 142, 17, '10:00:00'),
(17, 143, 17, '10:30:00'),
(17, 144, 17, '11:00:00'),
(17, 145, 17, '11:30:00'),
(17, 146, 17, '12:00:00'),
(17, 147, 17, '12:30:00');


INSERT INTO [TRANSIT].[Schedule] (RouteID, StopID, VehicleID, ExpectedTime)
VALUES
(17, 147, 170, '12:00:00'),
(17, 146, 170, '12:30:00'),
(17, 145, 170, '13:00:00'),
(17, 144, 170, '13:30:00'),
(17, 143, 170, '14:00:00'),
(17, 142, 170, '14:30:00');
```

REQUIREMENTS REVIEW

Category	Description	Status
User account	The User should be able to create a new account and add a personal information	Completed
User account	The User should be able to get a unique user ID	Completed
Navigating	User should be able to plan a journey	Completed
Navigating	Users can select start, Destination, and a unique journey ID should be generated.	Completed
Scheduling	Users will be given the option to select the mode of transport to travel.	Completed
Scheduling	Users can select either a trax or bus to reach the destination.	Completed
Scheduling	According to the User's selection of mode of transport & the start and destination route should be assigned.	Completed
Scheduling	According to the User's selection & route, the expected time will be shown to reach the destination.	Completed
Scheduling	Users should see the stops between the start and destination of the route.	Completed
Scheduling	Users will be given an option to choose the fastest mode of transport by the expected time to reach the destination.	Completed

FEATURES

FEATURE 1:

```
--Showing stops in between for User 1

SELECT s.StopID, s.StopName

FROM [TRANSIT].[Stops] s, [TRANSIT].[Journeys] j
WHERE j.JourneyID = 301
AND (
    (s.StopID >= j.Start AND s.StopID <= j.Destination)
    OR
    (s.StopID <= j.Start AND s.StopID >= j.Destination)
)
ORDER BY s.StopID;
```

Output: -

JourneyStopsList
U. Of U. Medical Center Station
Fort Douglas Station
University South Campus Station
Stadium Station
900 East Station

When a user organizes a travel by selecting start and finish locations, the app displays all the stops that are placed in between the start and end points, allowing the user to obtain an overview of what all stops are there in between his journey.

FEATURE 2:

```
--User travelling in forward direction (having single route in database)

WITH VehiclesForJourney AS (
    SELECT DISTINCT s.VehicleID
    FROM [TRANSIT].[Journeys] j
    INNER JOIN [TRANSIT].[Schedule] s ON j.Start = s.StopID
    WHERE j.JourneyID = 301
)

SELECT
    r.Zone,
    v.VehicleType AS VehicleTy,
    s.VehicleID AS VehicleNo,
    st.StopName AS Stops,
    s.ExpectedTime
FROM [TRANSIT].[Journeys] j
INNER JOIN VehiclesForJourney vfj ON j.JourneyID = 301
INNER JOIN [TRANSIT].[Schedule] s ON vfj.VehicleID = s.VehicleID
INNER JOIN [TRANSIT].[Routes] r ON s.RouteID = r.RouteID
INNER JOIN [TRANSIT].[Stops] st ON s.StopID = st.StopID
INNER JOIN [TRANSIT].[Vehicles] v ON s.VehicleID = v.VehicleID
WHERE
    j.JourneyID = 301
    AND (
        s.ExpectedTime BETWEEN
        (SELECT ExpectedTime FROM [TRANSIT].[Schedule] WHERE StopID = j.Start AND Vehi
cleID = s.VehicleID)
        AND
        (SELECT ExpectedTime FROM [TRANSIT].[Schedule] WHERE StopID = j.Destination AN
D VehicleID = s.VehicleID)
    )
ORDER BY s.ExpectedTime;
```

Output: -

Results Messages

Search to filter items...

Zone	VehicleTy	Vehicle_No	Stops	ExpectedTime
Red	Trax	801	Fort Douglas Station	08:15:00
Red	Trax	801	University South Campus Station	08:30:00
Red	Trax	801	Stadium Station	08:45:00
Red	Trax	801	900 East Station	09:00:00
Red	Trax	801	Trolley Station	09:15:00
Red	Trax	803	Fort Douglas Station	11:15:00
Red	Trax	803	University South Campus Station	11:30:00
Red	Trax	803	Stadium Station	11:45:00
Red	Trax	803	900 East Station	12:00:00
Red	Trax	803	Trolley Station	12:15:00

The feature two states that when a user has planned a journey by providing his start and destination points. User will get travel options which showcase the vehicles running in that route along with zone information. We would also be providing the anticipated time for the journey along with the timelines for individual stops present through the journey.

Feature 3:

```
---Vehicle added: BUS

WITH VehiclesForJourney AS (
    SELECT DISTINCT s.VehicleID
    FROM [TRANSIT].[Journeys] j
    INNER JOIN [TRANSIT].[Schedule] s ON j.Start = s.StopID
    WHERE j.JourneyID = 310
)

SELECT
    r.Zone,
    v.VehicleType AS VehicleTy,
    s.VehicleID AS VehicleN,
    st.StopName AS Stops,
    s.ExpectedTime

FROM [TRANSIT].[Journeys] j
INNER JOIN VehiclesForJourney vfj ON j.JourneyID = 310
INNER JOIN [TRANSIT].[Schedule] s ON vfj.VehicleID = s.VehicleID
INNER JOIN [TRANSIT].[Routes] r ON s.RouteID = r.RouteID
INNER JOIN [TRANSIT].[Stops] st ON s.StopID = st.StopID
INNER JOIN [TRANSIT].[Vehicles] v ON s.VehicleID = v.VehicleID

WHERE
    j.JourneyID = 310
    AND (
        s.ExpectedTime BETWEEN
        (SELECT ExpectedTime FROM [TRANSIT].[Schedule] WHERE StopID = j.Start AND Vehi
cleID = s.VehicleID)
        AND
        (SELECT ExpectedTime FROM [TRANSIT].[Schedule] WHERE StopID = j.Destination AN
D VehicleID = s.VehicleID)
    )

ORDER BY s.ExpectedTime;
```

Output: -

Results Messages

Search to filter items...

VehicleTy	Vehicle_No	Stops	ExpectedTime
Bus	17	Sunnyside Ave & Guardsman Way	10:30:00
Bus	17	1300 S & 1500 E	11:00:00
Bus	17	900 E & 1700 S	11:30:00
Bus	17	State & 1700 S	12:00:00
Bus	17	Central Pointe Station	12:30:00

Based on the user 9 declared start and end points, the database showcases the available transport mode in that route with anticipated timelines., which is a bus in this scenario.

FEATURE 4:

```
--User travelling in counter direction (having single route in database)

WITH VehiclesForJourney AS (
    SELECT DISTINCT s.VehicleID
    FROM [TRANSIT].[Journeys] j
    INNER JOIN [TRANSIT].[Schedule] s ON j.Start = s.StopID
    WHERE j.JourneyID = 304
)

SELECT
    r.Zone,
    v.VehicleType AS VehicleTy,
    s.VehicleID AS VehicleNo,
    st.StopName AS Stops,
    s.ExpectedTime

FROM [TRANSIT].[Journeys] j
INNER JOIN VehiclesForJourney vfj ON j.JourneyID = 304
INNER JOIN [TRANSIT].[Schedule] s ON vfj.VehicleID = s.VehicleID
INNER JOIN [TRANSIT].[Routes] r ON s.RouteID = r.RouteID
INNER JOIN [TRANSIT].[Stops] st ON s.StopID = st.StopID
INNER JOIN [TRANSIT].[Vehicles] v ON s.VehicleID = v.VehicleID

WHERE
    j.JourneyID = 304
    AND (
        s.ExpectedTime BETWEEN
        (SELECT ExpectedTime FROM [TRANSIT].[Schedule] WHERE StopID = j.Start AND Vehi
cleID = s.VehicleID)
        AND
        (SELECT ExpectedTime FROM [TRANSIT].[Schedule] WHERE StopID = j.Destination AN
D VehicleID = s.VehicleID)
    )

ORDER BY s.ExpectedTime;
```

Output: -

Results | Messages

Search to filter items...

Zone	VehicleTy	Vehicle_No	Stops	ExpectedTime
Red	Trax	802	Daybreak Parkway Station	08:00:00
Red	Trax	802	South Jordan Parkway Station	08:15:00
Red	Trax	802	5600 W Old Bingham Hwy Station	08:30:00
Red	Trax	802	4800 W Old Bingham Hwy Station	08:45:00
Red	Trax	802	Jordan Valley Station	09:00:00
Red	Trax	802	2700 W Sugar Factory Rd Station	09:15:00
Red	Trax	802	West Jordan City Center Station	09:30:00

Feature four is the enhanced version of feature two, where the user is traveling in the counter direction in the same route. Here, we have assigned a vehicle in the counter direction as per the scheduled table as the user gets details of vehicles running in that direction and provides the expected time for a journey along with the timelines for individual stops present through the journey.

APPENDIX

USER

UserID	Password	Email	FirstName	LastName
1	password1	raju.b@example.com	raju	m
2	password2	sssgkraja@example.com	Raja	Sirum
3	password3	akesh.a@example.com	Akesh	anumalasetty
4	password4	akilesh.t@example.com	Akilesh	Thuniki
5	password5	fnu.adarsh@example.com	Adarsh	FNU
6	password6	dheeraj.w@example.com	dheeraj	yata
7	password7	abhinay.m@example.com	abhinay	fnu
8	password8	charith.j@example.com	charith	gopavaram
9	password9	harsha.b@example.com	harsha	g
10	password10	varun.o@example.com	varun	gk

The user table stores the information about the user for their sign-up.

When the user creates their account in the application, they must provide details like first name, last name, email ID, and password. The system generates a unique ID, i.e., UserID, which is the primary key for the user table.

ROUTE

RouteID	Zone
17	
401	Red

The Route table stores the information about the route and Zones Using “RouteID” as the primary key. Stops are assigned to a particular route according to the schedule table. Routes are classified through zones.

STOPS

StopID	StopName
101	U. Of U. Medical Center Station
102	Fort Douglas Station
103	University South Campus Station
104	Stadium Station
105	900 East Station
106	Trolley Station
107	Library Station
108	Courthouse Station
109	600 South Station
110	900 South Station
111	Ballpark Station
112	Central Pointe Station
113	Millcreek Station

The Stops table stores the information of Stops using “StopID”, which is unique and acts as the primary key for the table.

VEHICLE

VehicleID	VehicleType
17	Bus
170	Bus
801	Trax
802	Trax
803	Trax
804	Trax
805	Trax
806	Trax

The vehicle table consists of the vehicle number as “vehicleID,” which uniquely acts as a primary key to the table and is classified through vehicle type, maybe trax or bus.

JOURNEY

JourneyID	UserID	Start	Destination
301	1	101	105
302	2	102	106
303	3	119	121
304	4	126	120
305	5	124	115
306	6	110	114
307	7	143	144
308	8	145	142
309	9	143	147
310	10	144	142

The journey table consists of four columns, which are “JourneyID”, which acts as a primary key. UserID is another column in the table, which is the primary and foreign key for the table. The start and destination columns are the columns in the table.

SCHEDULE

RouteID	StopID	VehicleID	ExpectedTime
17	142	17	10:00:00
17	142	170	14:30:00
17	143	17	10:30:00
17	143	170	14:00:00
17	144	17	11:00:00
17	144	170	13:30:00
17	145	17	11:30:00
17	145	170	13:00:00
17	146	17	12:00:00
17	146	170	12:30:00
17	147	17	12:30:00
17	147	170	12:00:00
401	101	801	08:00:00

This relation table is formed in ternary relation between route, stops, and vehicles tables with expected time as a relational attribute. RouteID, StopID, and VehicleID are foreign keys, making a composite primary key for the schedule table.

TIME TRACKING DETAILS

Date	Team Member	Total Hours	Description of work
09/05/23	Akesh Adarsh Raja	1.5	Project approach and title decision
09/09/23	Akesh Akilesh Adarsh	1	Formation of business reequipment and identifying the entities
09/10/23	Akesh Akilesh Raja	1.5	Designing the conceptual & logical model
09/11/23	Akesh Akilesh Adarsh Raja	2	Finalizing the conceptual and logical model
09/18/23	Akesh Akilesh Adarsh Raja	1.5	Structural of physical model and start working on summary report.
09/21/23	Akilesh Adarsh Raja	2	Creating the sql database and inserting the data
09/27/23	Akesh Akilesh Adarsh Raja	2	Creating the sql database and inserting the data

10/05/23	Akesh Akilesh Raja	2	Inserting the data and testing the code.
10/08/23	Akesh Akilesh Adarsh Raja	1.5	Started the report
10/10/23	Akesh Akilesh Adarsh Raja	2	Preparing report
10/17/23	Akesh Akilesh Adarsh Raja	1.5	Finalizing the report
10/19/23	Akesh Adarsh	2	Starting the PPT
10/22/23	Akesh Akilesh Adarsh Raja	2	Finalizing the PPT

TIME-TRACKING SUMMARY

Team Member	Total Time	Additional comments
Akesh Anumalasetty	20.5	-
Akilesh Thuniki	19	-
Adarsh FNU	19	-
Raja Sirum	19.5	-