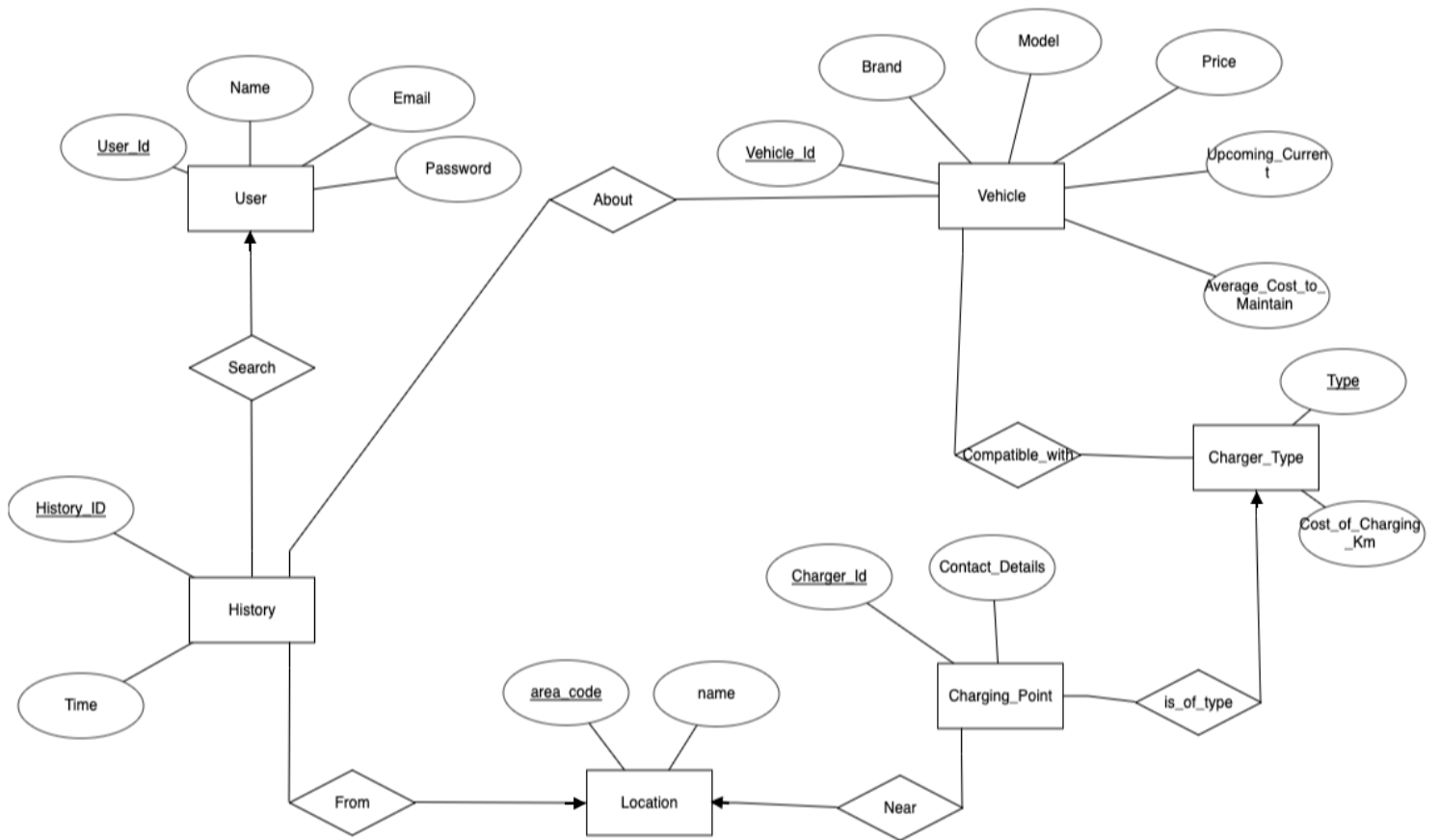


ER DIAGRAM FOR OUR DESIGN



ASSUMPTIONS

1. We store the user information in the User Table. One user can make many searches. Each search is stored in the history database and has one user who makes that search.
2. We also assume that each search is made about a vehicle. Many searches can be made about multiple vehicles, hence the entities have a many-many relationship.
3. We also plan to show a user the available charging points in the nearby location from where the search was made (functionality to find nearby charging stations). Hence each charging point is linked to a particular location (linked with its area code). Every search is also linked to a specific location from which the search was made.
4. We maintain a separate entity set called charger type that maintains the different types of EV chargers. Multiple vehicles can be connected to multiple chargers.
5. Each charging point is of a particular charger type, which is determined by the is_of_type relationship.

LOGICAL DESIGN (RELATIONAL SCHEMA)

- TABLE User (
 UserID [PK] INT,
 Name VARCHAR,
 Email VARCHAR
 Password VARCHAR
);
- TABLE Vehicle (
 ID [PK] INT,
 Brand VARCHAR,
 Model VARCHAR,
 Price (Range) Real,
 Upcoming/Current VARCHAR,
 Average_cost_to_maintain REAL,
);
- TABLE History (
 History_ID [PK] INT,
 Time TIME,
 Customer_ID [FK to User(User_ID)] INT,
 Area_code [FK to Location(Area_code)] INT
);
- TABLE Charging_Point (
 ID [PK] INT,
 Contact_Details VARCHAR,
 Area_code [FK to Location(Area_code)] INT
 Type [FK to Charger_Type(Type)] VARCHAR,
);
- TABLE location (
 Area_code [PK] IN,
 Name_of_area VARCHAR
);

- TABLE Charger_Type (
Type [PK] VARCHAR,
Cost_of_Charging_Km REAL
);

RELATIONSHIP TABLES

- ABOUT (HistoryID, VehicleID): to account for many-many relationship between history and vehicles
- Comptable_with(VehicleID, Type): to account for many-many relationship between vehicle and Charger_Type