Project 2- Design & Implement a Relational Database
Database & Design
Raneem Belbisi

Business Requirement:

Introduction:
This project aims to create a bike-sharing system that allows residences and tourists to access bicycles for short-trips across the Bay Area. This is important as it is a sustainable transportation solution, promotes healthy living and exercise, provides accessible and affordable transportation as well as provides valuable data on consumer patterns that can be used for urban planning and traffic management within the area.

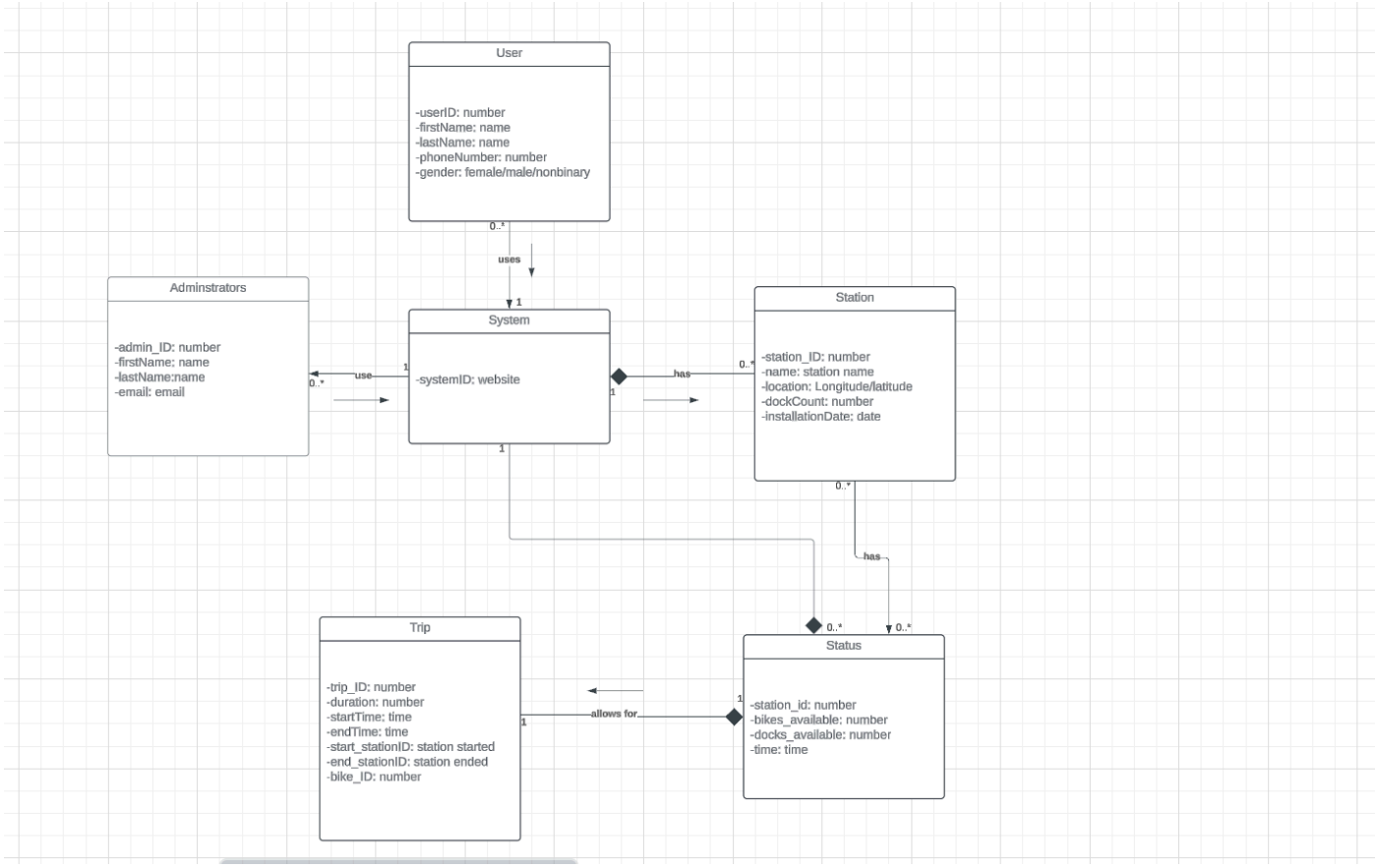
Business Rules:
1) Each bike has a unique ID and belongs to a specific station.
2) Stations have unique identifiers and are located at distinct geographical points.
3) Reservations have a start and end time.
4) Stations must have name and location.
5) Stations must have a maximum number of bikes it can hold.
6) Stations have multiple bikes that can be rented and their status is reported.
7) Administrators can view usage analytics reports.

Nouns:
1) Users
2) Station
3) Administrations
4) Trips
5) Status
6) System

Verbs:
1) Register
2) Record
3) Reserve
4) View

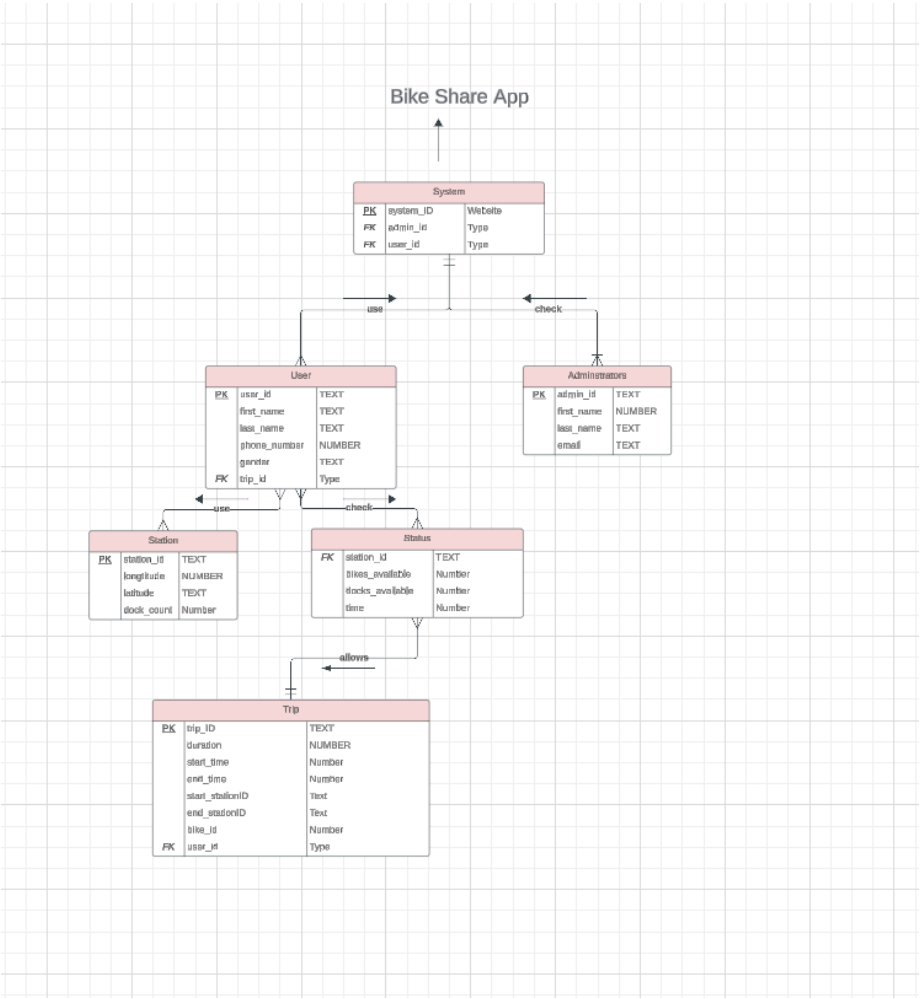
UML Class Diagram:

Link:https://lucid.app/lucidchart/invitations/accept/inv_8bb94a9e-7709-4259-889e-ade6e55ec3fb



Logical Model: Hierarchy

Link:https://lucid.app/lucidchart/invitations/accept/inv_8bb94a9e-7709-4259-889e-ade6e55ec3fb



Relational Schema:

User(user_id, first_name, last_name, phone_number, gender, *trip_id*)

System (system_id, *admin_id, user_id*)

Administrators (admin_id, first_name, last_name, email)

Station (station_id, longitude, latitude, dock_count)

Status (*station_id*, bikes_available, docks_available, time)

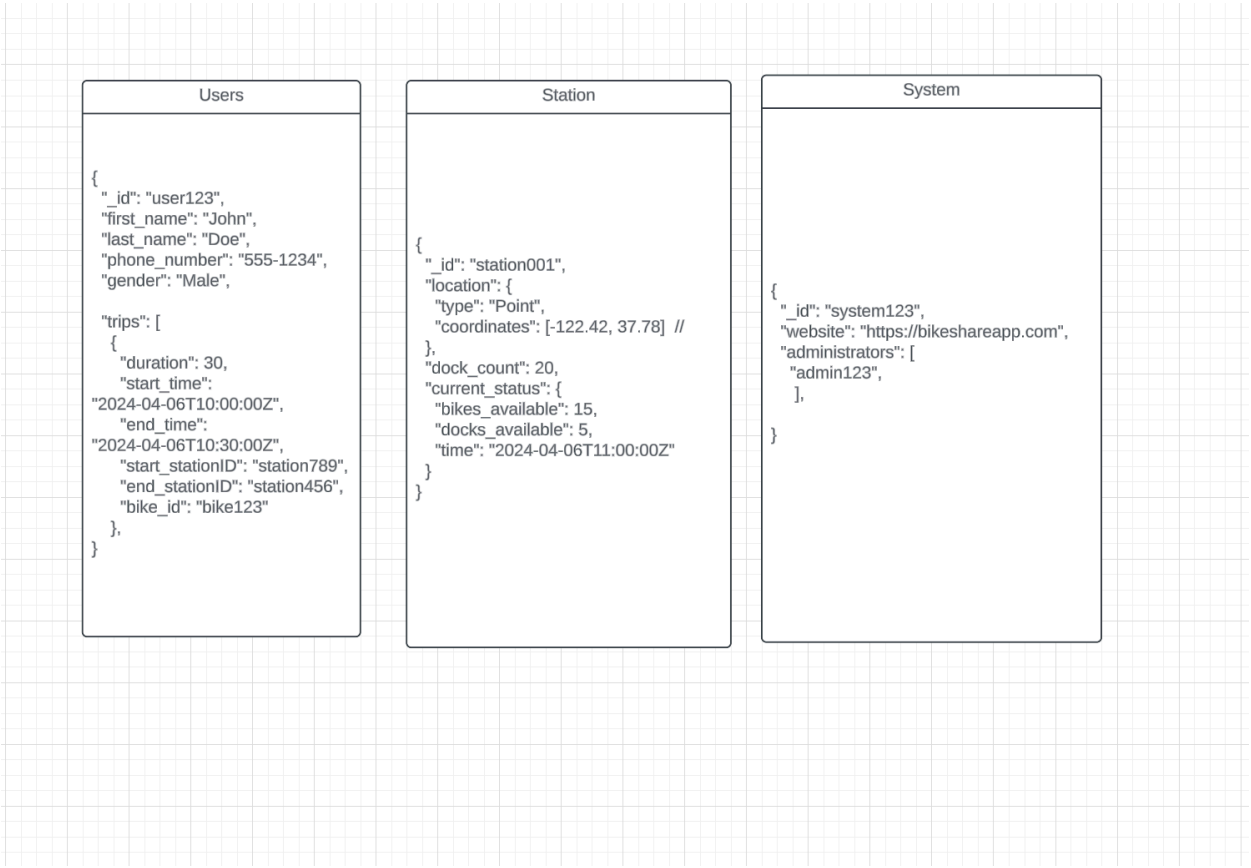
Trip (trip_id, duration, start_time, end_time, start_stationID, end_stationID, bike_id, *user_id*)

Proof that my Relational Schema is a BNF:

1) Each table has a superkey or has an attribute that is a superkey.
2) A superkey is a set of attributes that can uniquely identify each tuple in the table.

Main Collections (Documents/Tables):

Link:https://lucid.app/lucidchart/invitations/accept/inv_8bb94a9e-7709-4259-889e-ade6e55ec3fb



Users: This collection includes personal information about users and embeds their trips.
Stations: Represents the physical stations with current statuses. Status of bicycles in stations is embedded.
System: This collection holds overarching data for the bike-sharing system, including references or embedded documents for administrators.

README file (also on GitHub):

Attached to this Repository is a file called Project 2 which contains:
1) 5 files called Station.json, Users.json, Admin.json, Status.json, and Trips.json.
2) 1) All my queries named Query1.js, Query2.js, Query 3.js, Query4.js, Query5.js.
3) It also has 5 csv files which contain all 5 query results.

I did these queries by using mongo compass by following these steps:
1) In Compass, click on the "Databases" tab, then "Create Database" to set up a new database.
2) With the database and collection created, you should see options to insert and import data within the collection view.
3) Click on the "Add Data" dropdown and select "Import File". Choose the JSON file you want to import.
This imports the data and allows for queries to be created.

For the data collected, I used 4 tables from SF Bay Area Bikeshare from Kaggle and created two tables 'users' and 'admins' from Mackaroo. I joined them together on the db browser, created json files for each table then uploaded them into MongoDB Compass to complete my project.