1. Brainstorm Document –

Brainstorming Session

Mobile app
- 2 types of users who will use this app
    - 1 who gives rides out to other users
    - 1 who needs rides from other users

Opening screen - literally on click of the app icon
- 2 buttons at the center of the screen
    - Login, or
    - Signup

On click signup
- Username field
- Password field
- Button to designate whether you will be giving rides?
    - If the button is checked, car registration info is needed - make, model, license, etc…

On click login
- Username field
- Password field
- Submit button

After login
- Find a ride button
- Give a ride button
- Profile icon at the top right corner of screen
    - expands to a simple pop-up, no additional screens
- At the bottom: nav bar
    - View ride requests
    - View given ride history

Click find OR click give a ride
- Google Maps-esque screen
- From here - Point A field
- To here - Point B field
    - Optional - when to leave/arrive time fields can be inputted
- Possibly be able to edit/add stops, functionality that can be scaled later in the future

- Submit button

On click of the ride request/give ride submission button
- Driver info
- Car info
- Location info on where the driver and destinations are
- Maybe some additional profile info is displayed

Frontend: Kotlin
Backend: Kotlin
DB: SQLite
Docker/Kubernetes for DB access

Final Summary Paragraph

Our brainstorming session allowed us to converge on an outline of a ride-sharing mobile app designed to serve two specific types of users: riders and drivers. This app is different from other apps in the sense that drivers are to put in predetermined routes that they perform regularly in their lives, and the app will match riders with drivers who already drive a similar path to where the rider wishes to go (sort of similar to hitchhiking). A user signing up to be a driver will be required to fill out car registration details during a specialized signup process. The user flow prioritizes simplicity, beginning with a Login/Signup screen, followed by a main home screen offering 2 primary functions: to "Find a ride" or to "Give a ride." Both functions lead to a map interface for setting Point A and B locations. The app features a nav bar for viewing ride history/requests and an informational screen to display important driver, car, and location data upon match. The entire system is to be developed with a full-stack Kotlin approach for both frontend and backend components of the mobile app, using SQLite as the database and Docker/Kubernetes for managing access to the database. The scope of the app will primarily focus on serving Drexel students and the local UCity area, with options in the future the scale out many different components of the app if desired.

2. Brainstorm the Database –

Find Existing Data Sets

Student Performance - https://www.kaggle.com/datasets/ananta/student-performance-dataset

This dataset contains academic and behavior information about students. It's useful as it will help us understand student's lifestyle patterns and reliability which will make our mock data more realistic for students. And help for future features with creating timed-rides.

Student Entrepreneurship - https://www.kaggle.com/datasets/namanmanchanda/entrepreneurial-competency-in-university-students

This dataset measures entrepreneurial knowledge, personalities, and attitudes among university students. It's useful for us because we know how and which students are motivated and help us realistically decide which students are more likely to be riders for our application.

Ride-Sharing - https://www.kaggle.com/datasets/zahraatouq/ride-sharing-dataset

This dataset contains simulated trip data, along with ride requests, drive information, and trip duration. This is useful for usb because it gives us structural references for designing our own database and what we need to focus on getting correctly.

Cityride - https://www.kaggle.com/datasets/rishabhrajsharma/cityride-dataset-rides-data-drivers-data

This dataset contains information about riders, drivers, trip statistics in urban environments. This is helpful because it will give us more information on how drivers and riders interact which will allow us to create a better relational database.

NYC FHV - https://www.kaggle.com/datasets/jeffsinsel/nyc-fhvhv-data

This dataset includes millions of real rides from Uber, Lyft, and other services in NYC. This is very valuable because it focuses on large-scale and real-world mobility patterns which would help us benchmark realistic trip timing, ride density, and route distances.

Ola - https://www.kaggle.com/datasets/muhammadahmadmujahid/ola-dataset

This dataset contains ride-hailing data from Ola Cabs, including trip details, prices, and customer feedback. This highlights key ride metrics such as fare structures, cancellation rates, user rating, and all of this will help us fine-tune our own pricing, status tracking, and rating system.

Uber Fare - https://www.kaggle.com/datasets/yasserh/uber-fares-dataset

This dataset contains real Uber trips data with pickup and drop-off coordinates, trip distances, and fare amounts. It helps our project because it will give us a better pricing model, especially when setting a base price and a price per mile.

Uber Boston - https://www.kaggle.com/datasets/brllrb/uber-and-lyft-dataset-boston-ma

This dataset provides real ride information about both services Uber and Lyft in the Boston area which prices, distances, and times of day. This will help our project because it will help us analyze realistic fare patterns and demand fluctuations.

Uber Rides - https://www.kaggle.com/datasets/stantyan/uber-rides

This dataset contains tons of person records of hundreds of Uber trips, including timestamps, location, and ride durations. This will help us tremendously because it gave us realistic trip frequency and timing patterns.

Uber - https://www.kaggle.com/datasets/yashdevladdha/uber-ride-analytics-dashboard

This dataset contains information about Uber trips, including trip count, locations, and user activity trends. This is valuable for us because it will help us understand the overall ride distribution and user engagement patterns,allowing us to design synthetic data that realistically will be reflecting peak hours, route popularity, and ride frequency within our system.

Single Student Uber Driver - https://www.kaggle.com/datasets/zusmani/uberdrives

This dataset contains the personal trip logs of a single Uber customer throughout 2016. This is useful for us to be able to more accurately model and simulate user data, especially regarding our app's focus on matching individuals on predetermined routes.

Lyft Costs - https://www.kaggle.com/datasets/ravi72munde/uber-lyft-cab-prices

This dataset contains information about simulated ride price estimates and corresponding weather data collected in real-time. This is useful because the data directly relates to pricing, distance, and time-based factors critical to a ride-sharing service.

University Ranks - https://www.kaggle.com/datasets/mylesoneill/world-university-rankings

This dataset is a compilation of information about university rankings across the world. This is useful for defining the scope of our app's user base and knowing how to create realistic synthetic user data.

Car Data - https://www.kaggle.com/code/mohaiminul101/car-price-prediction

This dataset contains information used to estimate a car's selling price based on various features. This could be useful for our ride-sharing app in terms of supporting the driver onboarding process and possible expansions to include future pricing features.

Ride-Sharing Platform - https://www.kaggle.com/datasets/adnananam/ride-sharing-platform-data

This dataset contains simulated data for users, drivers, vehicles, rides, and ratings representing a ride-sharing platform. This is useful for us to be able to simulate how data gets processed in a real ride-sharing environment.

Create Synthetic Data

Mock Data on How We Can Make it

Rating - Mock Data

rating_id	match_id	from_user_id	to_user_id	stars	comment	created_at
1	1	2	12	5	Friendly	2025-10-13T15:30:00
2	5	8	8	3	Great ride	2025-10-13T18:45:00
3	8	9	4	5	Great ride	2025-10-13T04:45:00

From_user_id != to_user_id

Matches - Mock Data

match_id	request_id	offer_id	seats_booked	price_total	state	matched_at
1	3	3	1	11.71	completed	2025-10-13T13:30:00
2	4	7	1	7.95	cancelled	2025-10-13T06:00:00
3	8	16	2	14.77	confirmed	2025-10-15T02:00:00
4	9	16	1	15.16	no_show	2025-10-13T08:00:00
5	15	2	1	11.01	completed	2025-10-13T16:45:00
6	21	3	1	10.23	no_show	2025-10-13T23:45:00

Ride Offer - Mock Data

offer_id	driver_id	vehicle_id	origin_location_id	dest_location_id	depart_at	seats_available	price_base	price_per_mile	status	created_at
1	5	3	3	8	2025-10-16T01:45:00	2	4.17	0.72	closed	2025-10-15T01:45:00
2	1	6	6	10	2025-10-15T18:45:00	1	5.31	0.86	closed	2025-10-14T18:45:00
3	12	1	2	3	2025-10-14T13:00:00	2	4.63	1.13	open	2025-10-13T13:00:00
4	6	2	3	8	2025-10-14T17:45:00	3	8.52	0.9	closed	2025-10-13T17:45:00

Ride Request - Mock Data

request_id	rider_id	pickup_location_id	dropoff_location_id	earliest_pickup	latest_pickup	seats_needed	status	created_at
1	11	7	10	2025-10-14T19:05:00	2025-10-14T19:45:00	1	cancelled	2025-10-13T19:15:00
2	10	6	8	2025-10-14T13:15:00	2025-10-14T13:25:00	1	open	2025-10-13T13:15:00
3	7	1	9	2025-10-14T07:20:00	2025-10-14T07:40:00	1	matched	2025-10-13T07:30:00

4	13	6	7	2025-10-13T23:50:00	2025-10-14T00:30:00	2	matched	2025-10-13T00:00:00
5	10	7	2	2025-10-15T21:45:00	2025-10-15T21:55:00	1	open	2025-10-14T21:45:00
6	11	3	2	2025-10-15T03:00:00	2025-10-15T03:10:00	1	cancelled	2025-10-14T03:00:00

Location - Mock Data

location_id	name	address
1	Drexel Main Building	3141 Chestnut St
2	Korman Center	3220-26 Woodland Walk
3	University Crossings	3175 JFK Blvd
4	30th Street Station	2955 Market St
5	Queen Lane Campus	2900 Queen Ln
6	Vidas Athletic Complex	43rd & Powelton
7	Cira Green	129 S 30th St
8	Wawa 34th Market	3400 Market St

Users - Mock Data

user_id	first_name	last_name	email	phone	role	rating_avg	created_at
1	Abdul	Bookwala	abdul.bookwala1@example.edu	8569861234	driver	4.72	2025-10-02T00:00:00
2	Quincy	Lu	quincy.lu2@example.edu	856-9862334	driver	4.67	2025-10-09T00:00:00
3	Ame	Shabuse	ame.shabuse3@example.edu	8569861334	rider	4.55	2025-10-02T00:00:00
4	Ame	Lu	ame.lu4@example.edu	8569831234	both	4.54	2025-10-10T00:00:00
5	Kennan	Shajid	kennan.shajid5@example.edu	8569861234	both	4.7	2025-10-01T00:00:00

Vehicles - Mock Data

icl	user_id	make	model	color	plate	seats_total	yr
1	2	Tesla	Model Y	Blue	NJ-7909	7	2025
2	5	Tesla	Model 3	Gray	NJ-6737	5	2024
3	6	Toyota	Camry	Silver	NJ-8767	4	2015

4	8	Chevrolet	Malibu	Red	NJ-8301	5	2021
5	9	Toyota	Camry	Blue	NJ-6823	4	2024
6	14	Honda	Civic	White	NJ-7519	5	2022

Explanation of Data Sources and Rationale

Our dataset will combine information from the real-world public datasets and the synthetic data we created to simulate Drexel-specific rideshare activities. As a group, we reviewed several Kaggle datasets and found really useful ones such as Uber Ride Analytics, Lyft Cab Prices, and Car Price Prediction. These will all help with price ranges that are competitive and help both parties, getting vehicle details, and for trip characteristics. These sources will help us design the best rates, per-mile pricing, and driver vehicle data while staying manageable within a smaller scope.

However, because our app is meant for Drexel students, no public dataset directly represents that population. Because university student data is private. So we generated synthetic user, ride, and match data modeled after typical campus travel routes (e.g., 30th Street Station, Main Building, Cira Green). This is an example of how we would approach this when designing our database for the application. This synthetic data we generated allows full control of relationships between users, vehicles, rides, and ratings while maintaining realistic values inspired by real-world datasets. This will overall keep our app realistic and allow us to only focus on helping University students. As we build out the app, we can use helpful APIs such as SheerID that only accept university emails. This approach ensures our database can demonstrate all key functionalities (requests, offers, matches, ratings) without using personal or sensitive information, and keeps us protected.

3. ER Diagram using Chen Notation –