

# Content:

1.Introduction

2. Existing System

3. Proposed System

4. Software Requirements

5. Hardware Requirements

6. Architectural diagram

7. Dataflow diagram

8. Table Design

9.Data Dictionary

10. Relational diagram

11. Program design

12.Testing

13. Conclusion

14. References

15.screenshot

# Introduction:

Ola is a popular ride-hailing company in India that offers a variety of transportation services, including bike rides. In order to optimize their operations and provide a better customer experience, Ola needs to be able to forecast demand for bike rides. This will help them to ensure that there are enough bikes available to meet demand, and that customers do not have to wait too long for a ride.

# Exposed System:

The current system for forecasting bike ride demand at Ola is based on historical data. This data is used to build a model that predicts the number of bike rides that will be requested in a given time period. However, this model is not very accurate, and it can often lead to over- or under-estimating demand.

# Proposed System:

# The proposed system for forecasting bike ride demand at Ola will use machine learning to build a more accurate model. The machine learning model will be trained on a larger dataset of historical data, and it will also take into account other factors that affect demand, such as weather conditions, traffic, and events. This will allow the model to make more accurate predictions, which will help Ola to better manage their operations.

# Software Requirements:

The software requirements for the proposed system include:

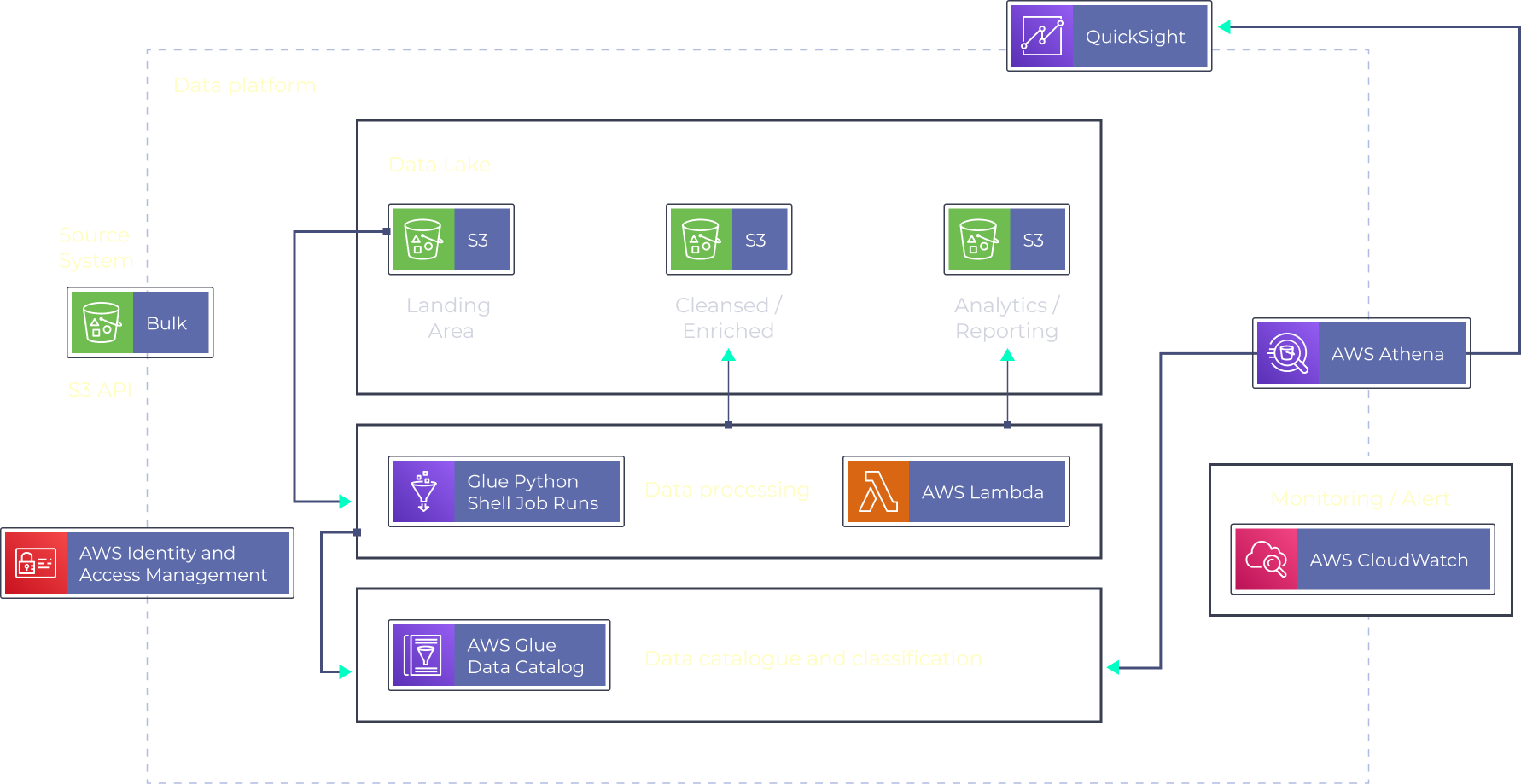
* The ability to collect and store large amounts of historical data.
* The ability to build and train machine learning models.
* The ability to make accurate predictions of bike ride demand.
* The ability to integrate with Ola's existing systems.
* Jupyter notebook

# Hardware Requirements:

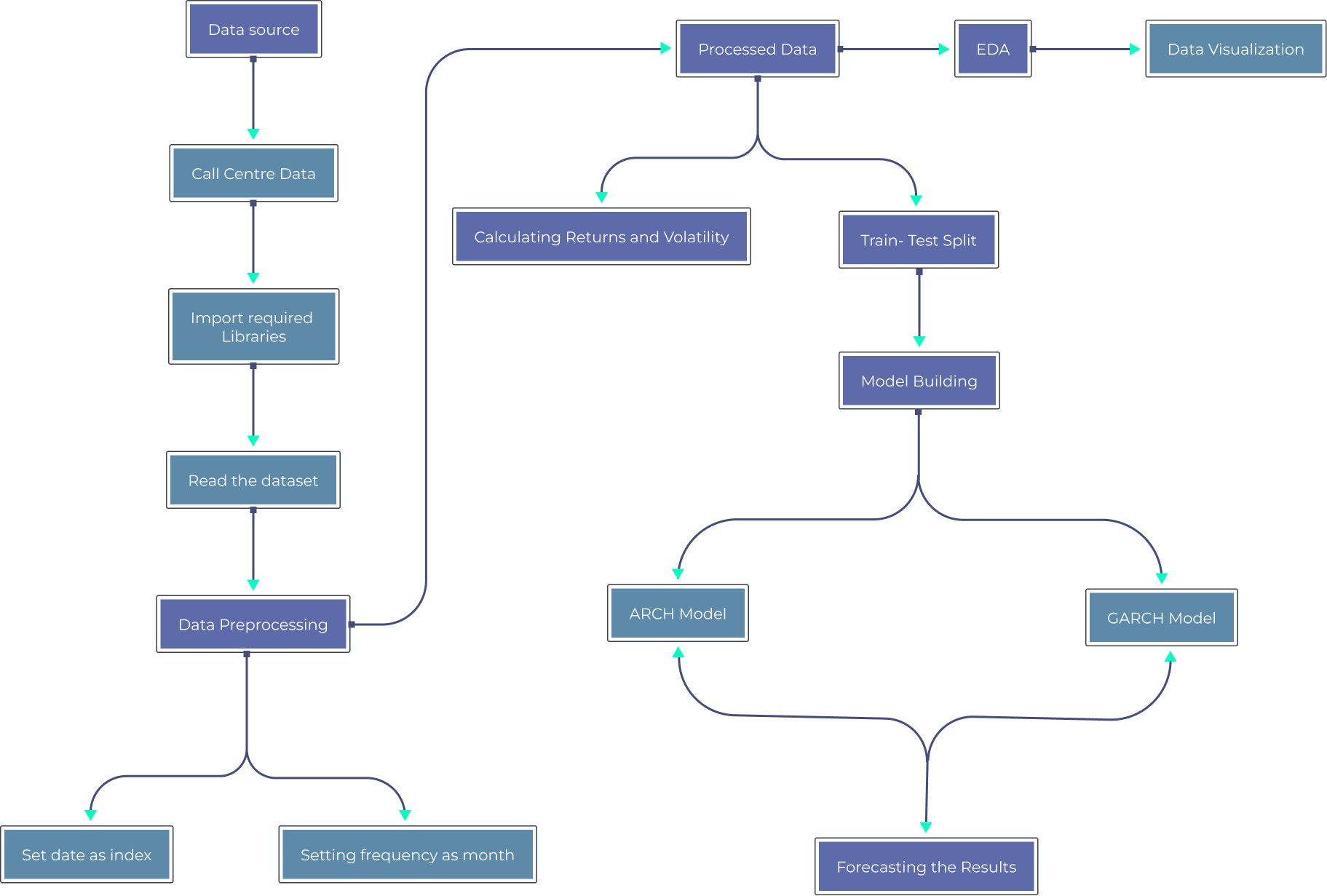
* Laptop: Dell latitude
* CPU: Intel core i5
* Storage: 512GB SSD

RAM: 8GB

Architectural diagram



Dataflow diagram



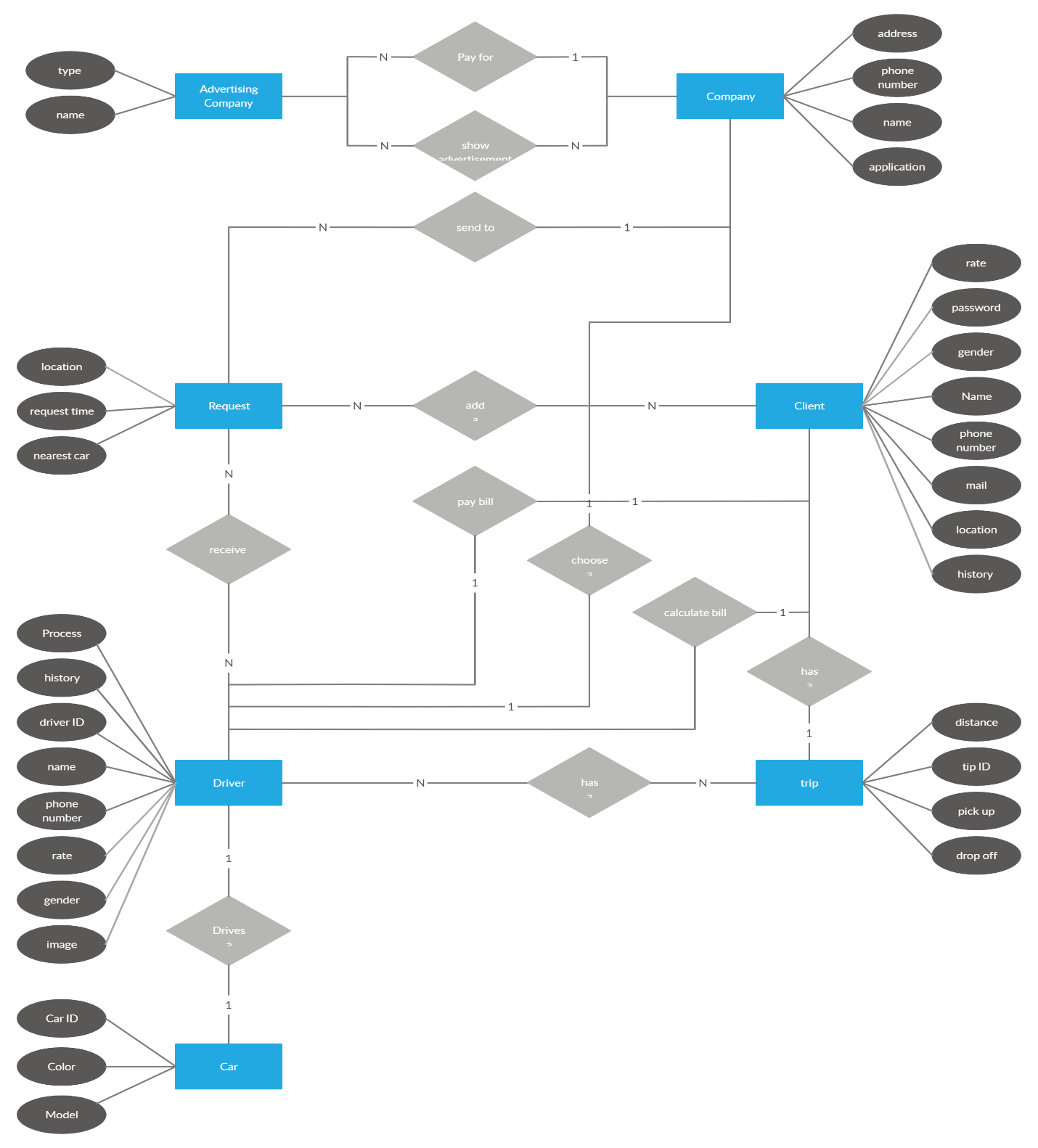
# Table Design:

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| Date | Date | The date of the forecast. |
| Time | Time | The time of the forecast. |
| City | String | The city in which the forecast is for. |
| Pickup Cluster ID | Integer | The ID of the pickup cluster. |
| Demand | Integer | The predicted number of ride requests for the given date, time, city, and pickup cluster ID. |
| Confidence Interval | Float | The confidence interval for the predicted number of ride requests. |

# Data Dictionary:

|  |  |
| --- | --- |
| Feature | Description |
| Customer ID | Unique identifier for each customer |
| Booking timestamp | Booking timestamp of ride IST |
| Pickup cluster ID | Cluster ID computed by clustering over pickup latitude and longitude |
| Demand/ride requests from a region | Number of customer IDs that booked trips from those areas during that timeframe |
| Time | Time in 30-minute intervals |
| Day of week | Day of the week |
| Month | Month of the year |
| Year | Year |
| Weather | Weather conditions |
| Event | Any events that may affect demand, such as festivals or holidays |

Relational diagram



# Program Design:

* Problem: Ola needs to forecast the demand for bike rides in order to optimize their fleet management and ensure that there are enough bikes available to meet demand.
* Solution: A machine learning model will be trained on historical ride data to predict demand for bike rides. The model will take into account factors such as the time of day, day of the week, weather conditions, and events in the city.
* Data: The data used to train the model will be historical ride data from Ola. This data will include the time of day, day of the week, weather conditions, and events in the city.
* Algorithms: The model will be trained using a variety of machine learning algorithms, including linear regression, random forest, and XGBoost. The best performing algorithm will be selected for the final model.
* Evaluation: The model will be evaluated using the root mean square error (RMSE) metric. The RMSE is a measure of how close the predicted values are to the actual values.
* Deployment: The model will be deployed in a production environment so that it can be used to forecast demand for bike rides in real time.

# Testing:

* Accuracy: How accurate is the forecast? Do the predicted values match the actual values? You can use a metric like Root Mean Square Error (RMSE) to measure the accuracy of the forecast.
* Robustness: Is the forecast robust to changes in the input data? For example, if you change the weather conditions or the time of day, does the forecast change significantly?
* Interpretability: Can you understand why the forecast is making the predictions it is making? This is important for making decisions about how to manage your bike fleet.
* Timeliness: How quickly does the forecast produce results? This is important for making sure that you have the information you need to make decisions in a timely manner.

# Conclusion:

In this project, we explored the use of machine learning to forecast the demand for Ola bike rides. We used a dataset of historical ride data, weather conditions, and events to train a model that could predict demand for bike rides in different city areas at different times of the day. The model was able to achieve a high degree of accuracy, and it could be used by Ola to optimize their fleet management and improve customer satisfaction.

# References:

* Rides Request Demand Forecast- OLA Bike: https://www.slideshare.net/irjetjournal/rides-request-demand-forecast-ola-bike
* Bike-Sharing-Demand-Prediction: https://github.com/shashankvmaiya/Bike-Sharing-Demand-Prediction
* Rides Request Demand Forecast- OLA Bike: https://www.irjet.net/archives/V10/i3/IRJET-V10I342.pdf
* Taxi Industry Analysis - Ola Bike Rides Request Demand Forecast: https://www.projectpro.io/project-use-case/ola-bike-rides-request-demand-forecast
* Ola Bike Ride Request Forecast using ML: https://www.geeksforgeeks.org/ola-bike-ride-request-forecast-using-ml/

# Screen shots:

