System Design Document for Airline Delay Prediction

Author: Li Guan, Robin Sah, Hengchuan Zhang

1. INTRODUCTION

1.1 Propose and Scope

Our team is dedicated to helping people choose better flights by predicting airline delays. In this project, our team uses big data tools to build a data lake, data warehouse to store data. Using machine learning to process data by building a model to predict the airline delay based on historical data. The final outcome should be an application that is able to access our data warehouse. When users typed the date, start airport and destination, the application can tell users the possibilities of airline delay within the time range.

1.2 Overview of System

This use case diagram provides a comprehensive visualization of the interactions between users, administrators, and the airline delay prediction system, encompassing both data management and predictive functionalities. Users can search their airlines with the airline ID, date and airports. Then the application will query the database within that range and the database contains all the processed airline data with a delay prediction. Finally, the database will send the prediction results back to users from the application we designed.

Actors and their Use Cases:

User/Customer:

- Search Airline: Enables the user to query specific airline details or flight services.
- Search Date: Facilitates the user in filtering flights based on specific dates.
- Search Airports: Provides the user with the capability to look for flights related to particular airports.
- View Prediction: Empowers the user to view the predictive results related to flight delays.
- Filter Results: Allows the user to refine and streamline search results or predictions based on certain criteria.
- Alternate Flights: As an extended functionality from viewing predictions, users can also access and explore alternative flight options.

Administrator:

- Collect Dataset: This use case is dedicated to the collection and aggregation of raw flight data by the administrator.
- Upload Dataset: Admins can upload the gathered data to the system.
- Clean Dataset: Involves the preprocessing, cleaning, and standardization of the uploaded dataset to ensure data integrity and quality.
- Analyze Dataset: Represents the action of conducting various analyses on the dataset to derive insights and patterns.
- Update Database: Allows the administrator to commit refined or new data to the primary system database.

Cloud Services:

This actor symbolizes the cloud-based infrastructure and services that facilitate data backups and potentially other cloud-specific operations.

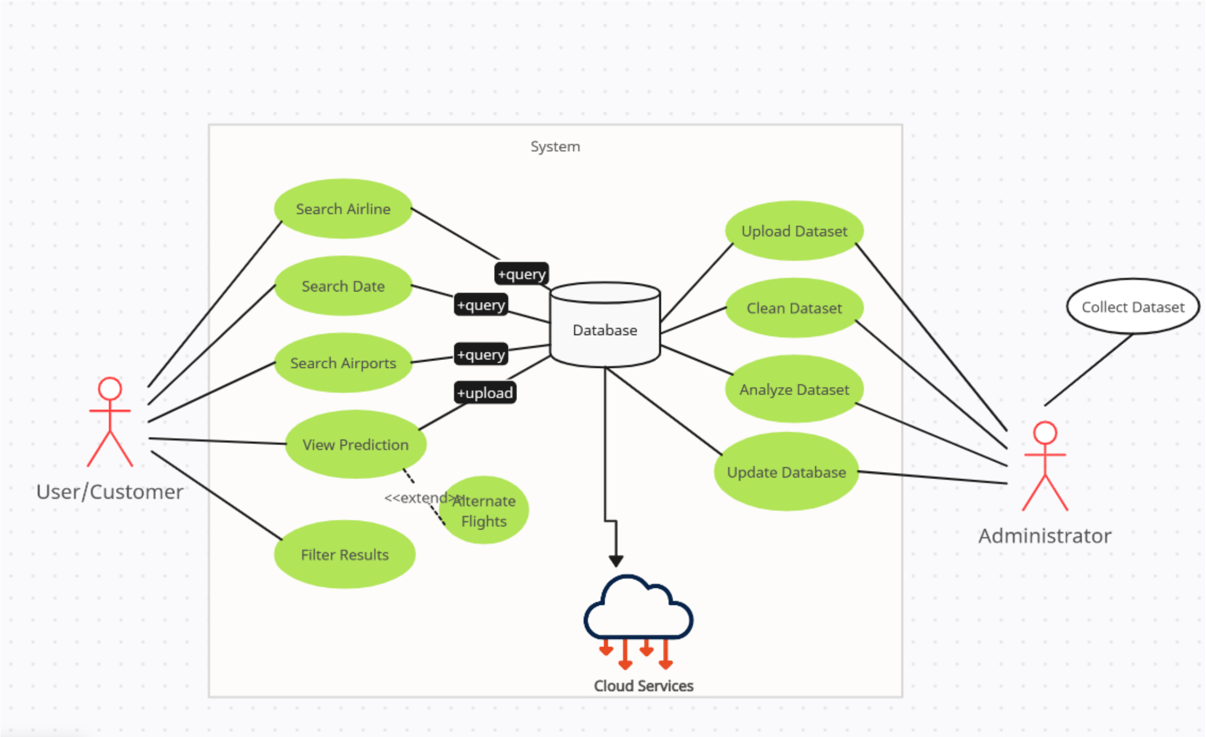


Fig.1 Use-Case Diagram for Airline Delay Prediction

Arrows originating from the actors showcase the possible interactions they can have with the system's functionalities. For instance, users can 'query' the system to search for flights based on specific criteria. The diagram efficiently encapsulates the holistic workflow of the airline delay prediction project, highlighting both the front-end user interactions and back-end administrative tasks, with a special emphasis on data handling and processing.

2. DATA FLOW

The diagram below shows the data flow diagram for our project. The starting point is the historical data set which contains the history data of airline delay. In the data ingestion part, we will design a new schema from the old data set. The new schema should only contain data, time, start airport, destination airport, weather, travel distance and time of delay. Those data will be stored in Amazon S3(D1) as our data lake.

With the help of Amazon Athena, our team can use Amazon S3 to do machine learning and export the prediction result to another Amazon S3(D2) as a data warehouse. After training our model, we will send our model to the real-time data processing.

Our team will use the Airport API to get real time data from specific airports. For example, we can get the data from JFK airport in the following week and store it in another Amazon S3(D3) as a data lake. Next, we use the trained model to predict the real time airline and store those results in the final data warehouse(D4). Finally, the application will query the data warehouse(D4) and replay users the result.

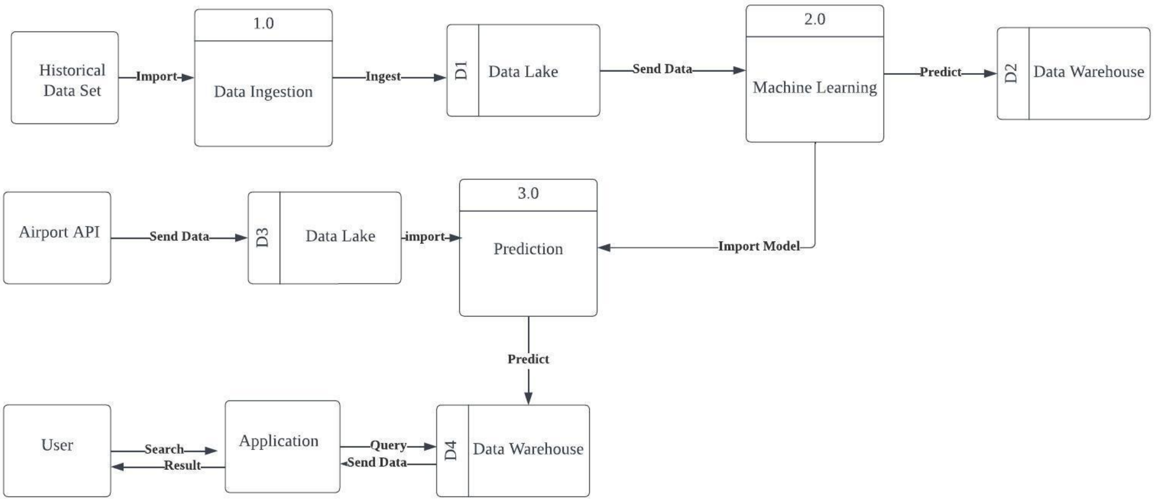
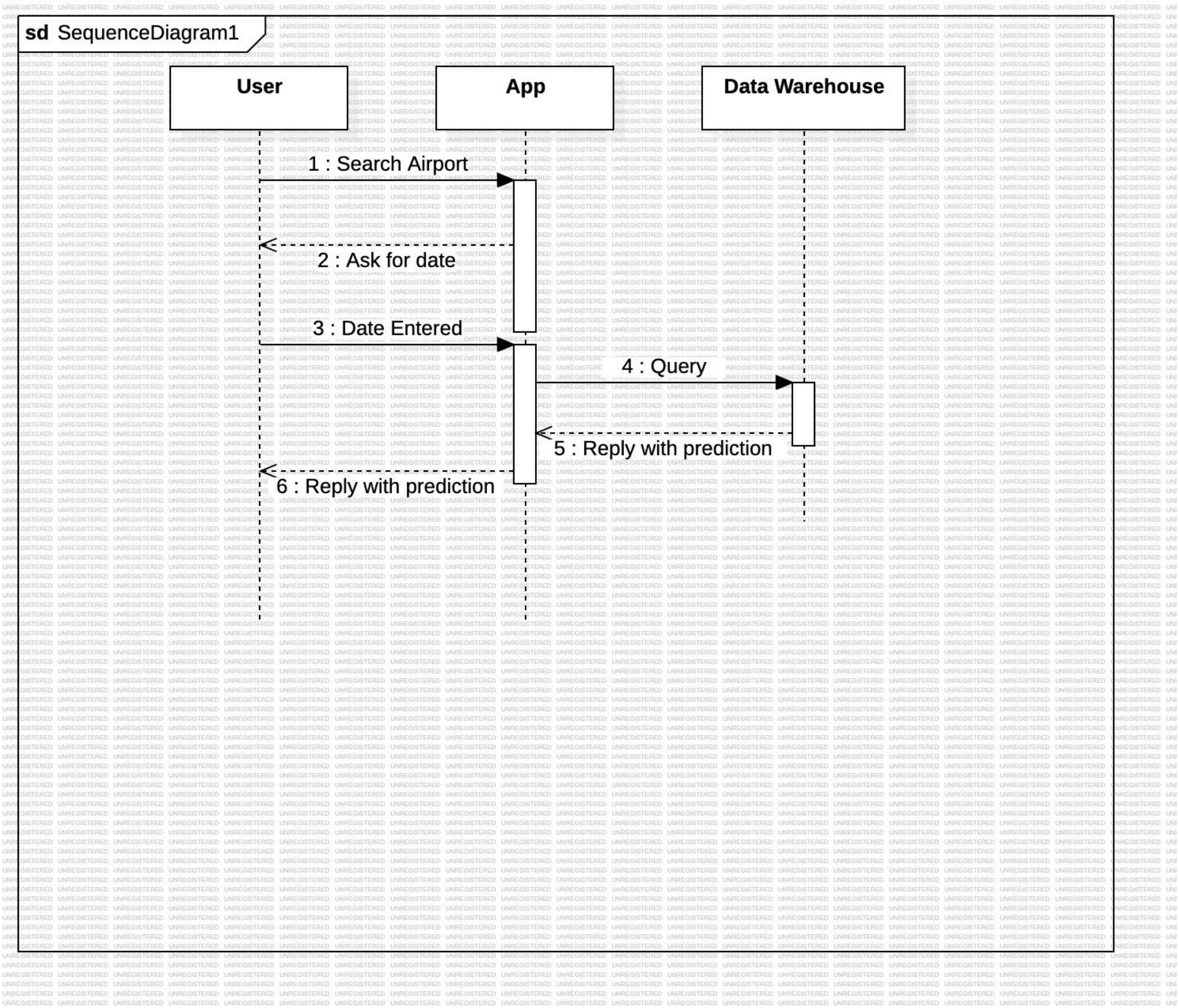


Fig.2 Data Flow Diagram for Airline Delay Prediction

So, the system boasts a meticulous divide between historical data processing, real-time data assimilation, and predictive operations. By leveraging cloud services like Amazon S3 and Athena, it ensures scalability, real-time processing, and efficient data storage. The architecture seamlessly integrates data transformation, machine learning, and real-time prediction to offer users accurate delay predictions, incorporating the strengths of archival data with the dynamism of live flight information.

3. Application

The diagram below is the sequence diagram from users to data warehouse. Our project will limit the choice of airports and date to users. Therefore, the application will not query the database unless both airport and date are valid. We want to reduce the times of query to database due to the cost. After all, the Amazon S3 charges us each time we query the database.



4. PROJECT SCHEDULE

- Data Collection and Ingestion - This phase encompasses the gathering of historical flight data and integrating real-time data using the Airport API. The data is then transformed and stored in a dedicated Data Lake.
- Model Development and Training - Features are selected and optimized, and suitable machine learning models are identified and trained. This phase also includes the refinement of the model through hyperparameter tuning.
- Real-time Data Integration and Prediction - Here, the trained model is integrated with the real-time data pipeline, and large-scale predictions are conducted, with results stored in the final Data Warehouse.

- Maintenance and Monitoring - The project transitions into this stage emphasizing continuous model performance evaluation and system health checks.