High-Level Design (HLD)

Flight Fare Prediction

**Contents**

Document Version Control**……………………………………………………………………….**

Abstract**…………………………………………………………………………………………..**

1. Introduction**…………………………………………………………………………………..**
   * 1.1 Why this HLD Document? **……………………………………………………...**
   * 1.2 Scope**…………………………………………………………………………….**
   * 1.3 Definitions**……………………………………………………………………….**
2. General Description**………………………………………………………………………….**
   * 2.1 Problem Perspective**……………………………………………………………..**
   * 2.2 Problem Statement**……………………………………………………………….**
   * 2.3 Proposed Solution**………………………………………………………………..**
   * 2.4 Further Improvements**……………………………………………………………**
   * 2.5 Technical Requirements**………………………………………………………….**
   * 2.6 Data Requirements**……………………………………………………………….**
   * 2.7 Tools Used**………………………………………………………………………..**
   * 2.8 Constraints**………………………………………………………………………..**
   * 2.9 Assumptions**………………………………………………………………………**
3. Design FLow**………………………………………………………………………………….**
   * 3.1 Modelling Process**………………………………………………………………..**
   * 3.2 Deployment Process**………………………………………………………………**
   * 3.3 Logging**……………………………………………………………………………**
   * 3.4 Error Handling**…………………………………………………………………….**
4. Performance Evaluation**……………………………………………………………………….**
   * 4.1 Reusability**………………………………………………………………………..**
   * 4.2 Application Compatibility**………………………………………………………...**
   * 4.3 Resource Utilization**………………………………………………………………**
   * 4.4 Deployment**……………………………………………………………………….**
5. Conclusion**……………………………………………………………………………………1**

**Abstract**

The aviation sector has undergone substantial transformations in recent times, owing to the dynamic shifts in the international market. These changes have exerted a profound influence on both the business landscape and customer preferences. Notably, governments across the globe have introduced diverse regulations and policies, significantly impacting airline companies. Consequently, the prices of flight tickets have become subject to substantial variations across different destinations. In light of these developments, our project aims to leverage data analysis techniques and machine learning algorithms to predict flight ticket prices accurately. By meticulously examining a multitude of factors, we endeavor to unravel the intricate dynamics that govern the pricing structures of air travel, thus empowering both businesses and customers to make informed decisions.

**1. Introduction**

**1.1 Why this HLD Document?**

The purpose of this High-Level Design (HLD) document is to provide project details and outline the Model Creation, Evaluation, and Deployment process. It describes the design aspects, user interface, hardware and software interfaces, performance requirements, and non-functional attributes such as security, reliability, maintainability, portability, reusability, and resource utilization.

**1.2 Scope**

The HLD document presents the system's structure, including database design, architectural design, application flow, and technology architecture. It uses non-technical terms to explain technical concepts, making it understandable to system administrators.

**1.3 Definitions**

| **Term** | **Description** |
| --- | --- |
| FFP | Flight Fare Prediction |
| Database | Collection of all the information used by the System |
| Jupyter Notebook | Interactive computational environment for code execution |
| AWS | building, running, and operating applications in the cloud Aplication |

**2. General Description**

**2.1 Problem Perspective**

The flight fare prediction system is a machine learning model that helps users predict the price of flight tickets and understand the cost of their journey.

**2.2 Problem Statement**

After implementing new rules, flight fare prices have changed between different locations. The goal of the system is to create a model that predicts flight fare prices based on user-provided inputs such as the date of journey, source, destination, and more.

**2.3 Proposed Solution**

To solve this problem, we have developed a user interface that takes input from the user to predict flight fare prices using our trained machine learning model. The input is processed, and the predicted value is communicated back to the user.

**2.4 Further Improvements**

We also analyzed the data used for training the machine learning model by considering different factors such as weekdays, seasons, or social events. By incorporating such information and predicting discounted flight fare prices, users can benefit while considering the business perspective of the airline industry.

**2.5 Technical Requirements**

No specialized hardware is required for this project. Users only need a device with web access and basic input capabilities. A server is required for the backend to run the necessary packages for processing inputs and predicting flight fare prices.

**2.6 Data Requirements**

The data requirements are based on the problem statement, and the dataset is available in .xlsx format on Kaggle. To provide a real-time experience, the data has been transformed into a Prophet database and exported as a CSV file.

**2.7 Tools Used**

* Python 3.9 as the programming language, utilizing frameworks like NumPy, Pandas, Scikit-learn, and other modules for model building.
* Jupyter Notebook as the IDE.
* Seaborn and Matplotlib for data visualizations.
* Prophet for data collection.
* HTML/CSS for front-end development.
* Flask for data and backend deployment.
* GitHub for version control.
* Heroku for deployment.

**2.8 Constraints**

The flight fare prediction solution should be user-friendly, automated, and require no user knowledge of the underlying operations.

**2.9 Assumptions**

The main objective of this project is to implement utility cases for a new dataset that provides users with the ability to predict flight fare prices. A machine learning model is employed to process user input and make predictions. It is assumed that all aspects of this project can work together as expected.

**3. Design Flow**

**3.1 Modelling Process**

[Insert design flow for the modeling process here]

**3.2 Deployment Process**

[Insert deployment process flow here]

**3.3 Logging**

Logging is performed whenever an error or exception occurs, recording the event, reason, and timestamp in the system log file. This facilitates debugging and error rectification.

**3.4 Error Handling**

When an error occurs, the reason is logged into the log file with a timestamp for proper handling and resolution.

**4. Performance Evaluation**

**4.1 Reusability**

The code and components used in this project should be designed for reusability without encountering any issues.

**4.2 Application Compatibility**

Different parts of the system communicate using Python as an interface. Each component has its own tasks, and Python ensures proper data transfer between them.

**4.3 Resource Utilization**

When a task is performed, it may utilize the available processing power until the process is completed.

**4.4 Deployment**

The model can be deployed using various cloud services such as Microsoft Azure, Amazon Web Services (AWS), Heroku, Google Cloud, etc.