Architecture Design

# Airbnb Data Analysis

|  |  |
| --- | --- |
| **Written By** | Anugrah Mohan |
| **Document Version** | 1.0 |
| **Last Revised Date** | 20/06/2023 |

**DOCUMENT CONTROL**

## Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **AUTHOR** | **COMMENTS** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Reviews:**

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **REVIEWER** | **COMMENTS** |
|  |  |  |  |

**Approval Status:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VERSION** | **REVIEW**  **DATE** | **REVIEWED BY** |  | **APPROVED BY** | **COMMENTS** |
|  |  |  |  |  |  |

# Contents

1. [Introduction 04](#_TOC_250005)
   1. [What is Architecture Design Document? 04](#_TOC_250004)
   2. [Scope 04](#_TOC_250003)
2. [Architecture 05](#_TOC_250002)
   1. Jupyter Notebook Architecture 05
   2. Client – Server Model 05
   3. Notebook Document 05
   4. Kernel 05
   5. User Interface 05
   6. Cell Execution 05
   7. Extension and Libraries 06
   8. Collaboration and Sharing 06
3. Deployment 07
   1. [Deployment Architecture 0](#_TOC_250001)7
   2. [Data Loading and Preparation 0](#_TOC_250000)7
   3. Data Analysis and Visualization 07
   4. Results and Reporting 07
   5. Sharing and Collaboration ………………………………………………………………………. 07

# Introduction

## What is Architecture design document?

Any software needs the architectural design to represents the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.

Each style will describe a system category that consists of :

* A set of components (eg: a database, computational modules) that will perform a function required by the system.
* The set of connectors will help in coordination, communication, and cooperation between the components.
* Conditions that how components can be integrated to form the system.
* Semantic models that help the designer to understand the overall properties of the system.

## Scope

Architecture Design Document (ADD) is an architecture design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and then refined during architectural design work.

# 2. Architecture

**2.1 Jupyter Notebook Architecture:**

Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. It provides an interactive computing environment that supports multiple programming languages, including Python, R, Julia, and more. The architecture of Jupyter Notebook can be described as follows:

**2.2 Client-Server Model:**

Jupyter Notebook follows a client-server architecture. The main components are:

Client: The web browser used to access and interact with the Jupyter Notebook interface.

Jupyter Notebook Server: The backend server that handles the execution of code, manages the notebook documents, and communicates with the client.

**2.3 Notebook Document:**

A Jupyter Notebook document is a file that contains code, narrative text, and visualizations organized in a hierarchical structure.

The notebook document is stored with the .ipynb extension and is written in JSON (JavaScript Object Notation) format.

The document consists of cells, each containing code or markdown text.

**2.4 Kernels:**

Kernels are separate processes that execute the code within notebook cells.

Each notebook has an associated kernel, which can be based on different programming languages.

The kernel handles the code execution, variable storage, and output display.

Kernels communicate with the notebook server using the Zero MQ messaging protocol.

**2.5 User Interface:**

The Jupyter Notebook web interface is accessed through a web browser.

It provides an interactive environment for editing and executing code cells, managing notebooks, and displaying outputs.

The user interface allows users to create, open, save, and share notebook documents.

Code cells can be executed individually or collectively, and the outputs are displayed inline within the notebook interface.

The interface also provides options for creating and editing markdown cells for adding narrative text and documentation.

**2.6 Cell Execution:**

Code cells can be executed by the user through the interface.

When a code cell is executed, the associated kernel receives the code and processes it.

The kernel executes the code and sends the output back to the notebook interface for display.

Code execution can be asynchronous, allowing multiple cells to be executed simultaneously.

**2.7 Extensions and Libraries:**

Jupyter Notebook supports a wide range of extensions and libraries to enhance functionality and interactivity.

Extensions provide additional features such as code linting, code snippets, spell-checking, and more.

Libraries like Matplotlib, Pandas, NumPy, and others are commonly used within notebook cells for data analysis, visualization, and other tasks.

**2.8 Collaboration and Sharing:**

Jupyter Notebook allows for collaboration by enabling multiple users to work on the same notebook simultaneously.

Notebooks can be shared with others by exporting them to various formats, including HTML, PDF, and Markdown.

Shared notebooks can be viewed and interacted with, even without a Jupyter Notebook installation, using platforms like Jupyter nbviewer or JupyterHub.

**Conclusion**:

The architecture of Jupyter Notebook follows a client-server model, with the client interacting through a web browser and the server managing the execution of code and notebook documents. Kernels handle the code execution, and the user interface provides an interactive environment for editing, executing, and sharing notebooks. With its extensibility and support for multiple programming languages, Jupyter Notebook has become a popular tool for data analysis, research, and interactive computing.

## 3. Deployment:

The deployment section of the high-level design document for the Airbnb Data Analysis project outlines the approach for deploying the project's components and making the analysis accessible to stakeholders.

## 3.1 Deployment Architecture:

The Airbnb Data Analysis project will be deployed as a Jupyter Notebook-based solution. The Jupyter Notebook environment allows for interactive data analysis, code execution, and visualizations.

The project can be deployed on a local machine or a cloud-based platform that supports Jupyter Notebooks, such as Google Colab, Microsoft Azure Notebooks, or AWS SageMaker.

**3.2 Data Loading and Preparation:**

The project requires the Airbnb dataset to be loaded and preprocessed before analysis. The Jupyter Notebook should include the necessary code to load the dataset and perform data cleaning steps, such as handling missing values, data type conversions, and feature engineering if required.

Detailed documentation should be provided within the Notebook to guide users on how to load their own dataset or update the existing dataset path.

**3.3 Data Analysis and Visualization:**

The Jupyter Notebook should include sections that perform exploratory data analysis, statistical analysis, and data visualizations using libraries like Pandas, NumPy, Matplotlib, Seaborn, Plotly or folium.

Each analysis step should be well-documented and include clear explanations of the insights gained from the analysis.

**3.4 Results and Reporting:**

The project should provide a clear presentation of the analysis results and insights derived from the data.

The Jupyter Notebook should include visualizations, summary statistics, and any relevant findings that answer the research questions or address the problem statement.

The Notebook should also include markdown cells or narrative explanations to provide context and interpretation for the results.

**3.5 Sharing and Collaboration:**

To facilitate collaboration and knowledge sharing, the Jupyter Notebook can be shared with stakeholders and team members via online platforms like GitHub.

**6**