# High Level Design (HLD)

# Analyzing Swiggy : Bangalore delivery outlet data

Revision Number: 1.0

Last date of revision: 24/06/2024

Bysani Sai Santhosh

## Document Version Control

| Date Issued | Version | Description | Author |
|---|---|---|---|
| **24th June 2024** | 1.0 | First Version of Complete HLD | Bysani Sai Santhosh |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Contents

## Abstract

This paper describes the High-Level Design (HLD) for a project to analyses Swiggy's Bangalore delivery outlet data. The goal of this project is to use ETL (Extract-Transform-Load) on the dataset to extract critical insights, metrics, and correlations that are important in the online food ordering business. This market encompasses a wide range of products, including ready-to-eat restaurant meals, individually produced food, and groceries delivered to clients' doors. The project's goal is to unearth important information that can be used to guide corporate choices, improve customer satisfaction, and optimize operations.

The procedure entails taking data from the specified source, converting it to ensure correctness and consistency, and loading it into an appropriate format for analysis. The project intends to provide clear, actionable insights by utilizing Python for data processing and analysis, as well as Tableau or Power BI for visualization. Delivery times, order volumes, customer happiness, revenue trends, and geographical insights will be among the key parameters evaluated.

The findings will be provided in the form of detailed visualizations and dashboards, allowing stakeholders to have a clear understanding of the data. Furthermore, the project will follow excellent coding practices, ensuring that the code is modular, maintainable, and version-controlled using GitHub. The goal is to produce a detailed, informative report and distribute the findings on professional sites such as LinkedIn, showcasing the project's influence and value in the food delivery business.

# Introduction

## 1.1 Why this High-Level Design Document?

This HLD paper is meant to provide an overview of the project's design and architectural elements. It describes the scope, tools used, design details, Key Performance Indicators (KPIs), and deployment methodologies. This document acts as a guide to guarantee that the project accomplishes its objectives and is carried out efficiently.

The HLD will:

• Showcase every design element and provide a detailed definition of it;

• Explain the user interface that is being used;

• Describe the hardware and software interfaces;

• Describe the performance requirements.

Include the project's architecture and design features. Enumerate and describe the non-functional characteristics, such as application compatibility, security, dependability, maintenance, portability, and reusability.

o Usage of resources o Serviceability

• Showcase every design element and provide a detailed definition of it;

• Explain the user interface that is being used;

• Describe the hardware and software interfaces;

• Describe the performance requirements.

• Provide a list and description of the non-functional characteristics, such as:

o Security o Dependability o Maintainability o Portability o Reusability o Application compatibility o Resource

## 1.2 Scope

Extracting data from the supplied dataset.

Transforming and cleaning data to guarantee its correctness and uniformity.

Loading processed data into an appropriate format for analysis.

Data mining is used to identify essential indicators and relevant correlations.

Presenting the findings through visualisations and dashboards.

Maintaining the code on a public GitHub repository.

# 2 General Description

## 2.1 Product Perspective & Problem Statement

The online food ordering business includes a wide range of food goods, from ready-to-eat meals made by restaurants to groceries delivered to your house. Understanding the market's dynamics is critical for firms seeking to improve their offerings and consumer happiness. This project will analyses Swiggy's Bangalore delivery data to gain important insights that can be used to make business choices.

## 2.2 Tools used

Business Intelligence tools and libraries works such as

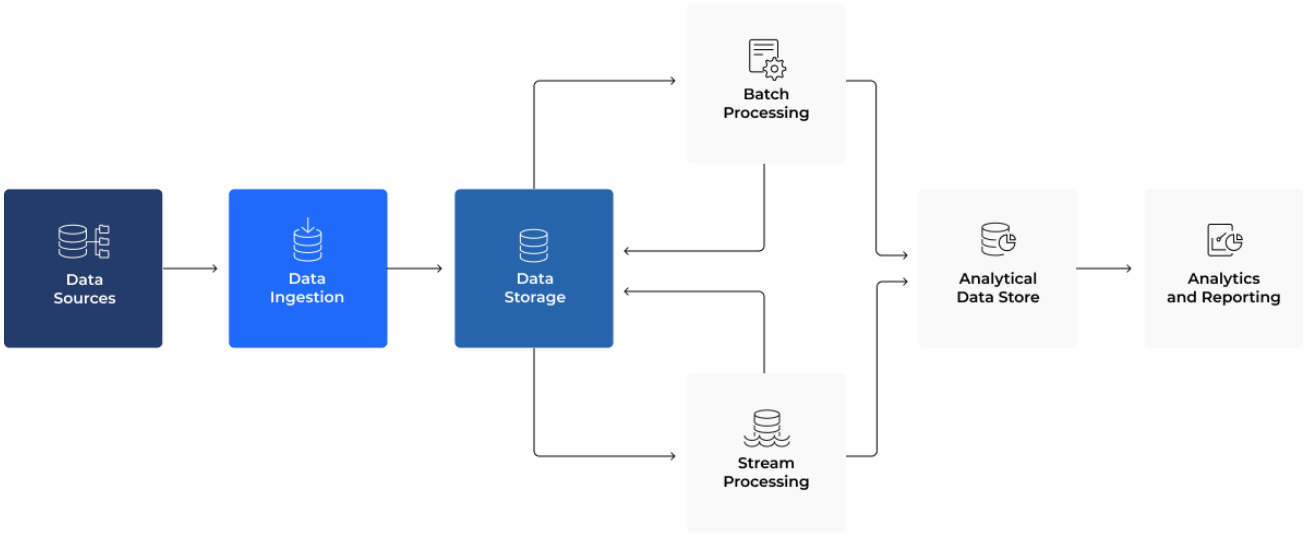
- Python: Used for data extraction, transformation, and loading (ETL).
- Pandas with NumPy: Tools for data manipulation and analysis.
- Matplotlib and Seaborn are tools for visualizing data.
- Tableau or Power BI are tools for producing interactive dashboards and visualizations.
- GitHub: Used for version control and code maintenance.

# 3 Design Details

## 3.1 Functional Architecture

This project's functional architecture consists of multiple stages:

- Data Extraction: Use the supplied URL to get the dataset.
- Data Transformation: Clean and preprocess the data to remove missing numbers, outliers, and inconsistencies.
- Data Loading: Save the processed data in a format appropriate for analysis (e.g., CSV, database).
- Data Analysis: Use exploratory data analysis (EDA) to better understand the data distribution and correlations between variables.
- Data Visualization: Create visualizations to convey important insights and measurements.
- Reporting: Put the findings together into a complete report and make dashboards for simple comprehension.



## 3.2 Optimization

Swiggy's massive delivery data requires optimization to be handled efficiently.
Key regions include:

**Data Processing**:
Pandas and NumPy are efficient libraries for performing vectorized computations.
Memory management involves optimizing data types and processing data in pieces to conserve memory.

**Modular code:**
**Reusability**: Create modular, reusable functions to improve readability and maintainability.
Parallel Processing: Use multiprocessing or Dask to accelerate data activities.

**Data transformation:**
**Optimized Algorithms**: Use efficient sorting, filtering, and hash tables to perform rapid operations.
**Indexing**: Use indexes to accelerate data access and manipulation.

**Visualization**:
**Performance**: Optimize visualizations using appropriate chart types and data aggregation.
**Interactivity**: Implement efficient interactive features in dashboards.

**Database Management:**
**Efficient Storage**: Normalize databases to reduce redundancy, use columnar storage if beneficial.
**Query optimisation** include creating optimised queries, indexing commonly used information, and planning queries to reduce bottlenecks.

These solutions provide efficient data handling, shorter processing times, and high-performance visualisations

# 4 KPIs

Dashboards will be used to show and signal key performance indicators and other related metrics with the help of Power BI.


As the system begins to collect historical/periodic data for a user, dashboards will be incorporated to present charts over time showing progress on different indicators or parameters.

## 4.1 KPIs (Key Performance Indicators)

Key indicators displaying a summary of the Swiggy Bangalore outlet data and its relationship with different metrics. The key performance indicators (KPIs) for this project are:

- Delivery Time Analysis: Average delivery time, peak delivery hours, and factors influencing delivery time.
- Order Volume: The number of orders placed each day, week, and month.
- Customer Satisfaction: Ratings and Reviews Analysis.
- Revenue includes total revenue, revenue per order, and revenue trends over time.
- Popular Items: The most often requested items and their influence on income.
- Geographical insights include delivery hotspots and locations with strong demand.

# 5 Deployment

Deployment entails providing extensive documentation for the project codebase on GitHub, as well as ensuring that the code is modular, maintainable, and version-controlled. The processed data and analytical findings will be stored in a secure database designed for efficient querying. Power BI will be used to build visualizations, including interactive dashboards that stakeholders may view. The research will be recorded with full reports and posted on professional forums such as LinkedIn to

showcase its findings and effect. This approach ensures transparency, accessibility, and professional visibility while enabling efficient data analysis and visualization for informed decision-making.

Power BI On-premises (Power BI Report Server)

Pros:

- Full control over hardware and software.
- Infrastructure and data stay behind the firewall.
- Integration with on-premise data sources.
- Customisable security settings

Cons:

- Need specialised administrators to handle hardware and software.
- Needs considerable infrastructural investment.
- Limited to on-premises data unless extra configuration for cloud access.
- Regular maintenance and upgrades are required.

Power BI in the public cloud (IaaS)

Pros:

- Complete control of software on controlled hardware
- Scalable infrastructure, spin up and down as needed.
- Streamlined integration with cloud-based data sources
- Access from anywhere with an internet connection.

Cons:

- Need dedicated administrators to manage software
- Dependence on cloud provider for infrastructure reliability
- Additional security measures required for sensitive data
- Potentially higher ongoing costs compared to on-premises

Power BI service (SaaS)

Pros:

- Fully hosted solution, with no hardware or software upkeep.
- Quick to deploy and scale.
- Users may easily access content from anywhere.

- Frequent upgrades and new features are instantly accessible
- Built-in collaborative features.

Cons:

- Customization is limited as compared to on-premises alternatives.
- Data residency issues for sensitive information
- Dependent on internet connectivity for access.
- Certain sophisticated features may need extra licenses or services.

Depending on your organizational roles and responsibilities, Power BI deployment should be coordinated by a systems administrator and the designated Power BI Administrator in collaboration with the appropriate IT roles. For Power BI On-Premises (Power BI Report Server), installation requires coordination with IT to ensure proper hardware setup and network configuration. For Power BI Service (SaaS), integration with existing technology and configuration of tenant settings will be necessary.

To begin, conduct a Data & Analytics Survey within business teams to identify and prioritize data use cases, audience size, and user roles. This information will guide your deployment strategy, influencing decisions on sizing, installation, and configuration of your Power BI Report Server, or the integration and configuration of your Power BI Service.

Administrators must also plan for the client software installation, which includes Power BI Desktop for report development, Power BI Mobile for mobile access, and Power BI Gateway for secure data transfer between on-premises data sources and Power BI Service. Proper planning and coordination with IT will ensure a smooth deployment and integration process, allowing users to leverage Power BI’s capabilities effect