**AMAZON E-COMM APPLICATION**

**Document Version:** 1.0

**Date:** 25/02/2024

**Prepared By:** ABHINAY KUMAR. NALBAND

TABLE OF CONTENTS

1. **Introduction**

* + Why this HLD?
  + Scope
  + Definitions
  + Overview
  + Purpose of this Document

2. **Features**

3. **Architecture Overview**

* + Technology Stack
  + Design Principles
  + Scalability Considerations
  + Activity Diagram

**4. Component Details**

* Authentication Service
* Restaurant Service
* Order Service
* Payment Gateway Service
* Review Service

**5. Data Storage**

* User Profiles
* Restaurant Data
* Orders and Transactions
* Reviews and Ratings

**6. Security Considerations**

* Data Encryption
* Authentication Mechanisms
* Authorization Policies
* Secure Payment Processing

**7. Deployment Strategy**

* Containerization with Docker
* Orchestration with Kubernetes
* CI/CD Pipeline

**8. Conclusion**

1. **Introduction**

* Why this HLD document?

This High-Level Design (HLD) document is created to provide a structured plan and overview of the architecture, features, and key components of a application. The HLD document plays a crucial role in guiding the development process, aligning stakeholders, mitigating risks, ensuring compliance and security, and ultimately, delivering a successful application that meets the needs and expectations of its users.

The HLD will:

* Present all of the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the architecture of the project
* List and describe the non-functional attributes like:

1. Scalability
2. Reliability
3. Security
4. Performance
5. Usability
6. Compliance
7. Maintainability
8. Monitoring and Logging

* Scope

This HLD documentation is comprehensive, covering various aspects essential for the successful design, development, and deployment of a application. It provides a holistic view of the application's architecture, features, data management, security, and deployment strategy, serving as a guide for the development team and stakeholders involved in the project.

* Definitions
* High Level Design (HLD) – A document that provides an overview of the architecture, features and key components of a software system.
* Microservices architecture – An architectural style that structures an application as a collection of loosely coupled services, each encapsulating a specific business function.
* Containerization – Process of encapsulating an application into containers which can be deployed across different environments.
* Continuous Integration/ Continuous Deployment (CI/CD) – These pipelines automate the build, test and deployment processes, enabling the rapid and reliable software delivery.
* Relational Databases – Organizes data into tables (rows and columns) with pre-defined relation between them.
* NoSQL Database - A non-relational database that provides flexible data models and scalability for handling large volumes of unstructured or semi-structured data.
* Authentication – The process of verifying the identity of users or systems accessing a software application.
* Authorization – The process of granting or denying access to specific resources or functionalities based on authenticated user’s permissions.
* Payment Gateway Integration – The integration of third-party payment processing services into a software application to facilitate secure and convenient payment transactions.
* Overview of Application:-
* Amazon is an advanced e-commerce platform offering a seamless shopping experience across diverse product categories. With a user-friendly interface, it provides easy navigation, personalized recommendations, and secure payment options. Amazon prioritizes security, partnering with trusted payment gateways for safe transactions. From fashion to electronics, users can explore a wide range of products from reputable brands. Bazaar aims to revolutionize online retail by delivering convenience, reliability, and quality service to modern consumers.
* Purpose of this Document
* This High-Level Design (HLD) document specifies the implementation, including inter-component dependencies, and provides sufficient design detail that any product based on this HLD will satisfy the product requirements.

1. **Requirements:**

* Search for products.
* Recommendations on UserHomepage.
* Place Order.
* Check Order Status.
* Write / View Product Review.

1. **Assumptions:-**
2. User Profile Creation is available.
3. Product onboard is Provided.
4. Payment Service is Available.
5. **Non Functional Requirements:**

* Low latency(Recommendation & Search)
* High Consistency(Placing Order, order Status, and Payments)

1. **Estimation:**

* Active users:- 300m monthly active users.
* Total products: 10m.

1. **Database design:**
2. **User db – Sql**
   1. user id(primary key)
   2. Username - string
   3. Password- string
   4. Firstname –string
   5. Lastname –string
   6. Email – string
   7. Loginlasttime- datetime
   8. Created account –datetime
3. **Address Db- Sql**
4. Address Db- Sql
5. AddressId- primary Key
6. User ID –Foreign Key
7. Effective date: Date
8. Address: String
9. Address line1 : String
10. Address line2: String
11. City : String
12. Country: String
13. Zip: Alphanumeric

**C. Product table**

* If we use sql database for product information, we will be wasting lot of space, so we will be using nosql.(Document Db- Dynamic Db or Mongo Db)

{

“Id”:”124568”,

“Category”:”Cloth”,

“item”:”Tshirt”,

“Gender”:”Female”,

“Size”:”S”

}

**D. Review DataBase:**

{

“Key”: “23444322134”,

“Rating”:4,

“Description”:”Product review”,

“Images”:[{

“AttachedId”:”73994”

},{

“AttachedId”:”73239”}]

}

**E. OrderDB(NoSql)**

{

“ordered”:”75545”,

“userid”:”4343267”,

“addressid”:”54324”,

“items”:[{

“itemid”:”439087”,

“quantity”:2,

“price”:60,

“currency”:usd},

{

“itemid”:”439327”,

“quantity”:1,

“price”:100,

“currency”:usd}],

“Totalamount”:220,

“taxes”:10,

“totalamt”:230}

**}**

**API’S:**

1. Get Recommendation(UserId):-

Will return a list of products recommendations.

1. Search(Search string, userId)

Returns a list of products.

1. Addto cart(UserID, Product Id,amount,quanty)

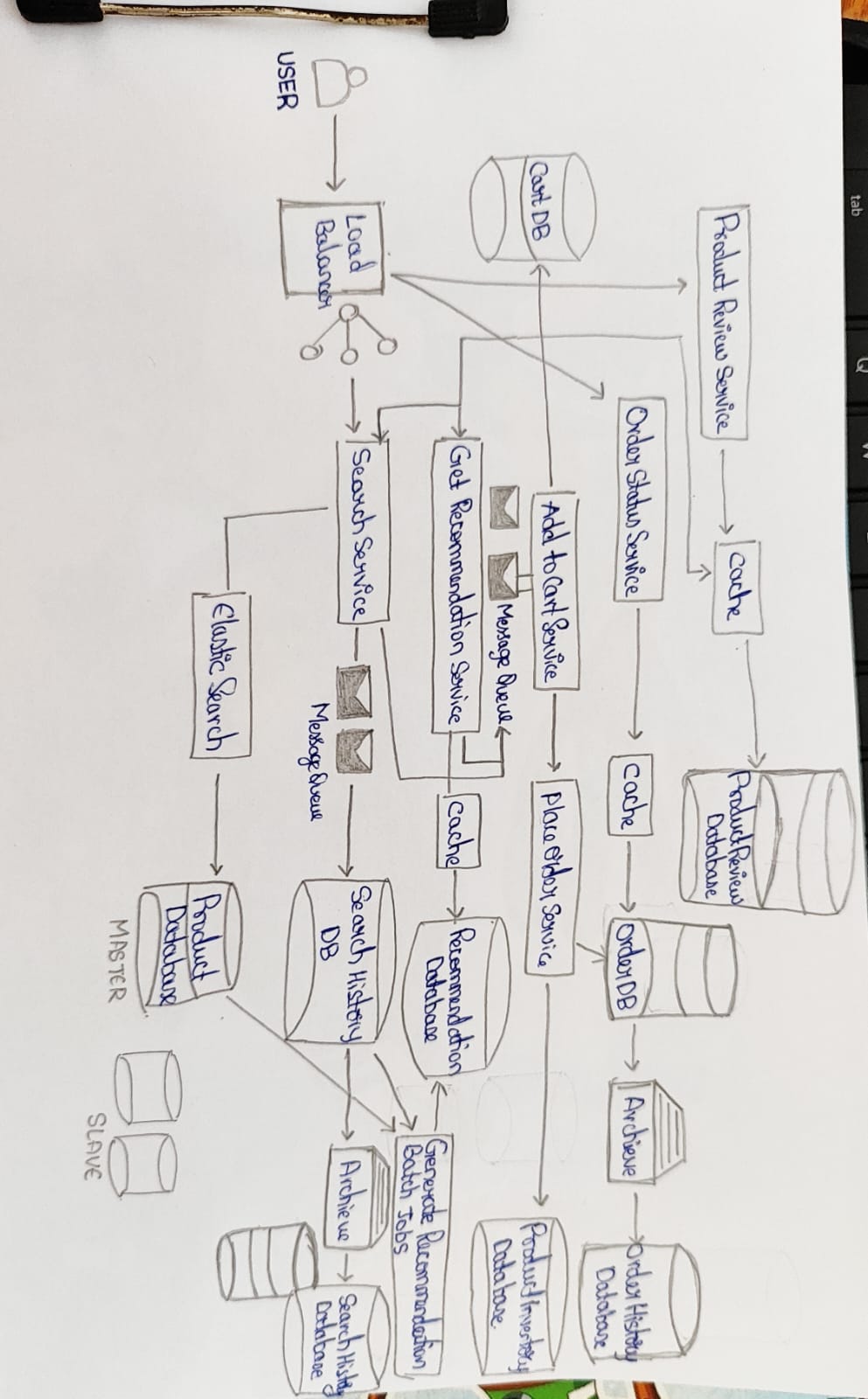
Return a Boolean (true/false)

1. Place order(UserID, OrderID, AddressId,payment Status)

Returns Boolean

1. CheckorderStatus(orderId)- returns status.

**System Design:-**

****