Design Document

The UberEats Order Delivery/Pickup System

Course: ITEC 4010

GROUP MEMBERS

Onyinyechi Onyeagucha 216883456

Jordan Bayon 216424897

Ahmet Yavas 217274978

Design Strategy

For our chosen design strategy, we will be making use of a combination of in-house system development, outsourced system(i.e. a consulting firm) development and existing system software packages. In-house system development will allow us to create custom solutions for various unique needs. It will be used to develop the user interface that allows customers to make changes to their orders before they are received by the restaurant. Additionally, we will use it to create a feature for the restaurant to confirm the accuracy of their order. The use of in-house development will allow greater control over the development process, allowing us to provide solutions to these problems found in the initial system.

Outsourced system development from things like consulting firms will provide us with access to any specialized expertise that may not be available to us in-house. Specifically, a consulting firm could help develop the coding system to be implemented for use between drivers and customers in order to ensure that the order only goes to the customer. Additionally, the consulting firm will assist us in the development of the 24/7 response team needed to handle customer requests. The use of outsourcing system development will allow us to save time and resources while getting access to specialized expertise.

Finally, existing system software packages will provide solutions like allowing us to purchase an existing chatbot software package to respond faster to and handle customer requests. We will also make use of an existing software package that incorporates traffic data and other delivery factors in order to improve the accuracy of the delivery wait times. The use of existing software packages will provide us with quick solutions to these problems without us having to invest significant resources into their development.

The current UberEats system has an integrated system with its Uber app and we will also integrate that into our system. Data analytics to gain insight into the behaviour of customers and improve their experience on the app will be one of the things we strategize into our system. Alongside this, ensuring that the security of the system and the privacy of customer data is maintained will be a top priority, our design strategy will also include cybersecurity measures such as encryption, secure data storage, and access control. This will help to prevent data breaches which could put our customers in vulnerable positions and help to protect the reputation of the business making use of our newly designed system.

Architecture Design

Description

Up until February 13, 2023 Uber as well as other Uber products such as Uber Eats have used their own **server-based architecture** (aka, host-based architecture) which included a backend service, a frontend service, and a single database as well as programs such as Python and SQLAlchemy to use for data access logic.

Since then, Uber decided to move to **cloud computing architecture** by moving some of their critical workload processes to Oracle's cloud infrastructure in order to help with pricing, performance, flexibility, and security to help on the business side of things. Uber also signed a deal with Google to move some of their applications and on-premise data center infrastructures to the Google cloud in order to help with customer service.

Implementation

In addition to the type of architecture needed for the UberEats delivery system the non-functional requirements are essential to know when designing delivery system:

- **Operational Requirements:**
    - The system will be able to work on any web browser, as well as be available on Google play store and Apple app store.
    - System must work on both desktop and on mobile devices running Android 5.0 and up and on mobile devices running IOS 8 and up.
    - System must have different UI (user interfaces) for desktop and mobile devices to account for each devices screen space
    - System must include a chatbot and consistent customer service representatives
    - System must use Google Maps API to track location and track delivery ETAs
    - The system must use Uber's **Dispatch system** (Dispatch optimization/DISCO) to assign drivers to customer orders, reduce ETA, and give drivers the shortest routes for deliveries.
    - System must use Uber's open source library - **Ringpop**
        - Helps to scale the dispatch system
        - Assigns work to drivers through consistent hashing
        - Allows each server to know the job that each driver is doing at the moment
        - Helps to distribute loads evenly when a server/node is added or removed
    - System must be able to read data about driver profiles to determine the safety of the driver and to give incentives to drivers to encourage good, reliable behavior.

- **Performance Requirements:**
    - Viewing speed of restaurants and menus such as filter and search menus must be under 300 milliseconds
    - ETAs and delivery times must be updated in real time and accurate
    - Databases should be horizontally scaled to account for the large amount of customers, servers, and restaurants being added daily.
    - Availability is a high priority as customers and drivers need to know their ETAs constantly and have to be changed and updated constantly due to outside factors (traffic, restaurant delays, etc..)
    - Logs must be kept after every order to ensure that the delivery went to the right person at the right address
    - Customer service representatives must be available 24/7
    - Restaurant information must be updated every 30 minutes to customer and drivers to account for restaurant close times and menu changes to avoid customer to restaurant confusion
- **Security Requirements:**
    - Customer, driver, and order information will be encrypted to ensure secure ordering
    - Customers and drivers will not be able to change any information to pricing and restaurant information
    - Drivers will not be able to read any customer information other than restaurant address, delivery address, and tipping information.
    - Customer service representatives must be able to read customer information, customer order history, and driver information to handle customer service requests.
    - Drivers must prove that they have completed all safety training before being hired
    - Customers and drivers must ensure that codes given to them match up. Codes will be encrypted upon being given to ensure no one else is able to receive them
    - Chatbot will only be able to contact customers and drivers upon their request
- **Cultural & Political Requirements:**
    - System must operate in multiple languages
    - System must accommodate for the different metric systems and currencies
    - Chatbot must be able to send messages in multiple languages
    - Chatbot must assign customer representatives that speak the same language as customer who made the request
    - System must accept different forms of payment and card types depending on the country
    - Driver and vehicle requirements must conform to country requirements on top Uber safety training requirements
    - GPS must accommodate for different countries traffic signs and rules

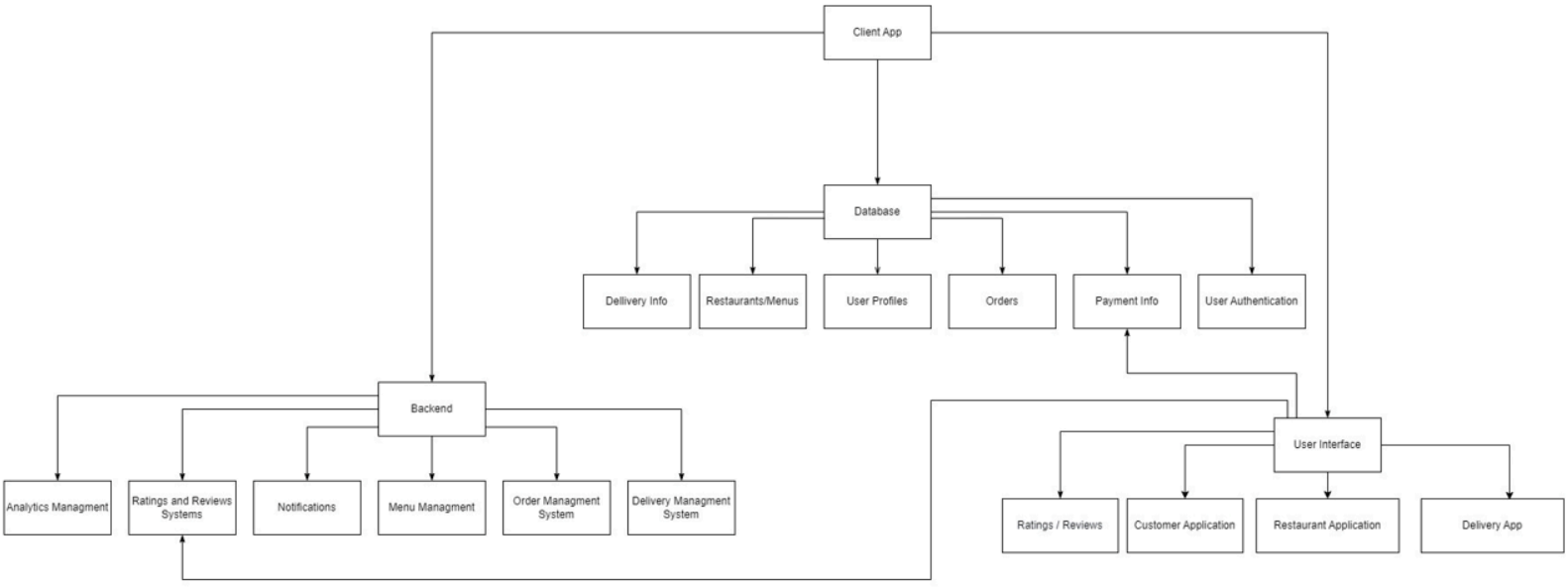
Database And File Specifications

Uber Eats data storage is designed to handle massive volumes of data while ensuring high scalability and availability to the users.

Uber Eats uses a combination of various technologies to store and access its data. Uber uses Apache Cassandra to store large amounts of data across its data centers. Apache Cassandra is a distributed NoSQL database that stores data/ information about UberEats' users (customers), drivers, trips, restaurants, menus, and details about many daily transactions and requests.

Data warehouses are also used by UberEats, Amazon Web Services(AWS), along with Amazon Redshift, is used for UberEats cloud computing, where AWS is used to store a large amount of UberEats data, and Amazon Redshift is used in running large datasets of queries in the data storage (back end).

UberEats personalizes user recommendations based on customer order history and menu preferences. With the help of machine learning algorithms, UberEats recommends customer products depending on the users' criteria and recommends more relatable and better products to its customers.



Program Design

User Interface

Customer Application: allows the user to browse through the application categories, restaurants, and menus and filter the user's preferences.
Restaurant Application: it handles the restaurant data where the menu items/ store product information are updated (product price, availability) and also holds the order fulfillment.
Delivery App: this component helps the delivery drivers to access navigation and update the delivery status.
Ratings / Reviews: this component is where the review and ratings of stores/restaurants, delivery drivers and menu items are inputted by the user.

Backend

Order Management Systems: This handles changes in orders, such as cancelling and placing or updating anything within the orders.
Menu Management Systems: This component allows the restaurant to update their store item's prices, descriptions, stock amount or any deals on menu items.
Delivery Management Systems: handles the delivery part of the orders. It uses advanced algorithms to allocate drivers near restaurants, assign them with the order and determine the EAT of the order to its requested address.

Notification: this component sends updates on the order status to the user. It also sends notifications of recommendations about the present deals on the items the customer has recently bought or searched upon.
Ratings / Reviews Systems: This component includes the user saved or submitted review or rating about the restaurant/store or a product on the menu of the restaurant/store. It makes it visible for other users to see the store rating upon browsing menus or restaurant catalogues.
Data Insights/ Analytics: is responsible for determining customer behaviour and, with the help of using algorithms, determining better and more accurate product/ restaurant recommendations on the customer search history, order history and menu preferences.

Database

User Profile: this includes details about the user's address, phone number, email address, restaurants and drivers.
User Authentication: this is used to authorize the user. This can be a password, face ID or Touch ID to authorize access for the user to log in to one's account.
Restaurants / Menus: this includes information about the registered menus and restaurants/stores.
Delivery Information: Contains data on delivery drivers, including the type of vehicle they use (bike or car etc.) and if the driver is available for deliveries at that moment.
Payment Info: this component securely charges or refunds orders to the saved payment method account.
Orders: is responsible for updating the customer with the order status, such as processing, in a delivery vehicle, and delivered. It also keeps the order status updated in case of any cancellations on changes that take place.

References

Dissanayake, K. (2021, August 18). Uber architecture and system design. Medium. Retrieved April 9, 2023, from https://medium.com/nerd-for-tech/uber-architecture-and-system-design-e8ac26690dfc The linked article discusses the architecture and system design of the Uber platform. It begins by providing an overview of Uber's business model and then delves into the technical details of its architecture. The article explains how Uber's platform is built using a microservices architecture and how the platform is divided into various services, such as the dispatch service, payment service, and rating service. It also discusses how Uber's technology stack includes various technologies such as Node.js, Python, Cassandra, and Hadoop. The article goes on to explain how Uber uses real-time data processing and stream processing to handle the massive amount of data generated by its platform. It also talks about how Uber uses various techniques to ensure the reliability and availability of its platform, such as using redundant systems and failover mechanisms.

Savitz, E. J. (2023, February 14). Uber signs cloud computing deals with Oracle and google. Barron's. Retrieved April 9, 2023, from https://www.barrons.com/articles/uber-stock-price-alphabet-oracle-81d07881 The linked article from Barron's discusses the potential impact of a recent court decision that ruled Uber drivers in the UK should be classified as workers rather than as self-employed. The ruling could lead to significant changes for Uber's business model and labor practices, as it could require the company to provide additional benefits and protections for its drivers. The article also discusses the potential involvement of tech giants Alphabet and Oracle in Uber's future, with speculation that one or both of these companies could acquire or invest in Uber. The article notes that such a move could provide benefits for both Uber and the acquiring company, as well as potentially leading to increased competition with other ride-sharing services.

Srivastava, S., & Choudhary, P. (2022, August 11). Streaming real-time analytics with Redis, AWS Fargate, and Dash Framework. Uber Blog. Retrieved April 10, 2023, from https://www.uber.com/en-CA/blog/streaming-real-time-analytics/ This linked article discusses how Uber uses streaming real-time analytics to process billions of events per minute. The article explains how Uber processes real-time data from its diverse set of products and services, including ride-sharing, Uber Eats, and Freight. The article goes on to discuss the challenges faced by Uber in processing real-time data, such as high velocity and volume of data, complex data sources, and real-time processing requirements. The article then describes the architecture of Uber's streaming platform, which includes a set of services and technologies that work together to collect, process, and analyze real-time data. The article provides insights into the technologies used by Uber, including Apache Kafka, Apache Samza, and Apache Hadoop. The article also explains how Uber uses machine learning algorithms to process and analyze real-time data, which helps improve the accuracy of its services and products.

Tinnus, F. (2008). How food delivery brands are providing connected experiences in an uncertain time. Amazon. Retrieved April 10, 2023, from https://aws.amazon.com/blogs/industries/how-food-delivery-brands-are-providing-connected-experiences-in-an-uncertain-time/ This linked article discusses how food delivery brands are leveraging technology to provide seamless and connected experiences to customers during the uncertain times of the COVID-19 pandemic. The article highlights the importance of technology in the food delivery industry, particularly in the areas of personalization, delivery logistics, and payment processing. The article also touches upon

the use of artificial intelligence (AI) and machine learning (ML) to improve the customer experience and streamline operations. Overall, the article provides valuable insights into how food delivery brands are adapting to the new normal and leveraging technology to provide connected experiences to customers.

Upadhyay, A. (2023, January 12). System design of Uber App - Uber System Architecture. GeeksforGeeks. Retrieved April 9, 2023, from https://www.geeksforgeeks.org/system-design-of-uber-app-uber-system-architecture/ The link provided is a website called GeeksforGeeks, and the article discusses the system design of the Uber app and the overall architecture of the Uber system. The article provides an overview of the various components of the Uber system, including the mobile app, server-side components, and the dispatching and routing systems. The article also discusses the challenges faced by Uber in scaling its system to handle millions of users and trips. It provides a detailed and technical explanation of the various components of the Uber system and how they work together to provide a seamless user experience.

Xu, H., Kale, S., & Du, S. (2022, September 7). MySQL to myrocks migration in Uber's distributed datastores. Uber Blog. Retrieved April 10, 2023, from https://www.uber.com/en-CA/blog/mysql-to-myrocks-migration-in-uber-distributed-datastores/#:~:text=Uber%20uses%20MySQL%20as%20the,and%20Docstore%2C%20our%20distributed%20databases The linked article describes Uber's transition from using MySQL to MyRocks as its distributed database management system. MyRocks is a storage engine for MySQL that utilizes RocksDB as its storage backend, which offers better compression and improved I/O performance compared to traditional MySQL storage engines. The article discusses the challenges Uber faced in migrating their databases to MyRocks, including dealing with data inconsistencies and tuning performance to maximize the benefits of MyRocks. It also outlines the benefits that Uber has seen from the migration, including reduced storage costs, improved I/O performance, and lower latency.