

Universidad Distrital Francisco José de Caldas



David Santiago Buitrago C 20221020085
Carlos Andres Pardo Angel 20231020041

First Project Delivery

Bogotá Abril 2024

Description of the project

Overview

The project aims to develop a mobile application similar to Mercado Libre, a popular e-commerce platform. This app will offer a marketplace where users can buy and sell a wide range of products. One of the key focuses of this project will be the design and implementation of a robust and scalable database to handle large volumes of data effectively.

Stakeholders

- **Primary Stakeholders**

Users: Individuals who will buy and sell products on the platform.

Developers: The team responsible for designing, developing, and maintaining the app.

Business Owners: The individuals or company funding the project and expecting a return on investment.

- **Secondary Stakeholders**

Suppliers: Entities providing products to be sold on the platform.

Payment Gateways: Services facilitating transactions between buyers and sellers.

Regulatory Bodies: Organizations overseeing e-commerce regulations and standards.

Business Model

1. **Revenue Streams**

- a. **Transaction Fees:** Charging a small percentage of each transaction made through the platform.
- b. **Premium Listings:** Offering paid options for sellers to highlight their products.
- c. **Advertising:** Displaying targeted ads to users based on their browsing and purchasing history.

2. **Value Proposition**

- a. **Convenience:** A user-friendly platform that simplifies buying and selling processes.
- b. **Variety:** A wide range of products and categories available in one place.
- c. **Security:** Secure payment gateways and buyer protection policies to ensure safe transactions.

3. **Market Strategy**

- a. **Target Audience:** Initially focusing on a specific demographic or geographic region before expanding globally.
- b. **Partnerships:** Collaborating with suppliers, delivery services, and other e-commerce platforms to enhance the platform's offerings.

Tools to Use

Development Tools

Programming Languages: Python

Frameworks: Django or Flask for backend development, React Native for cross-platform mobile development.

Version Control: Git and GitHub for code management and collaboration.

Database Tools

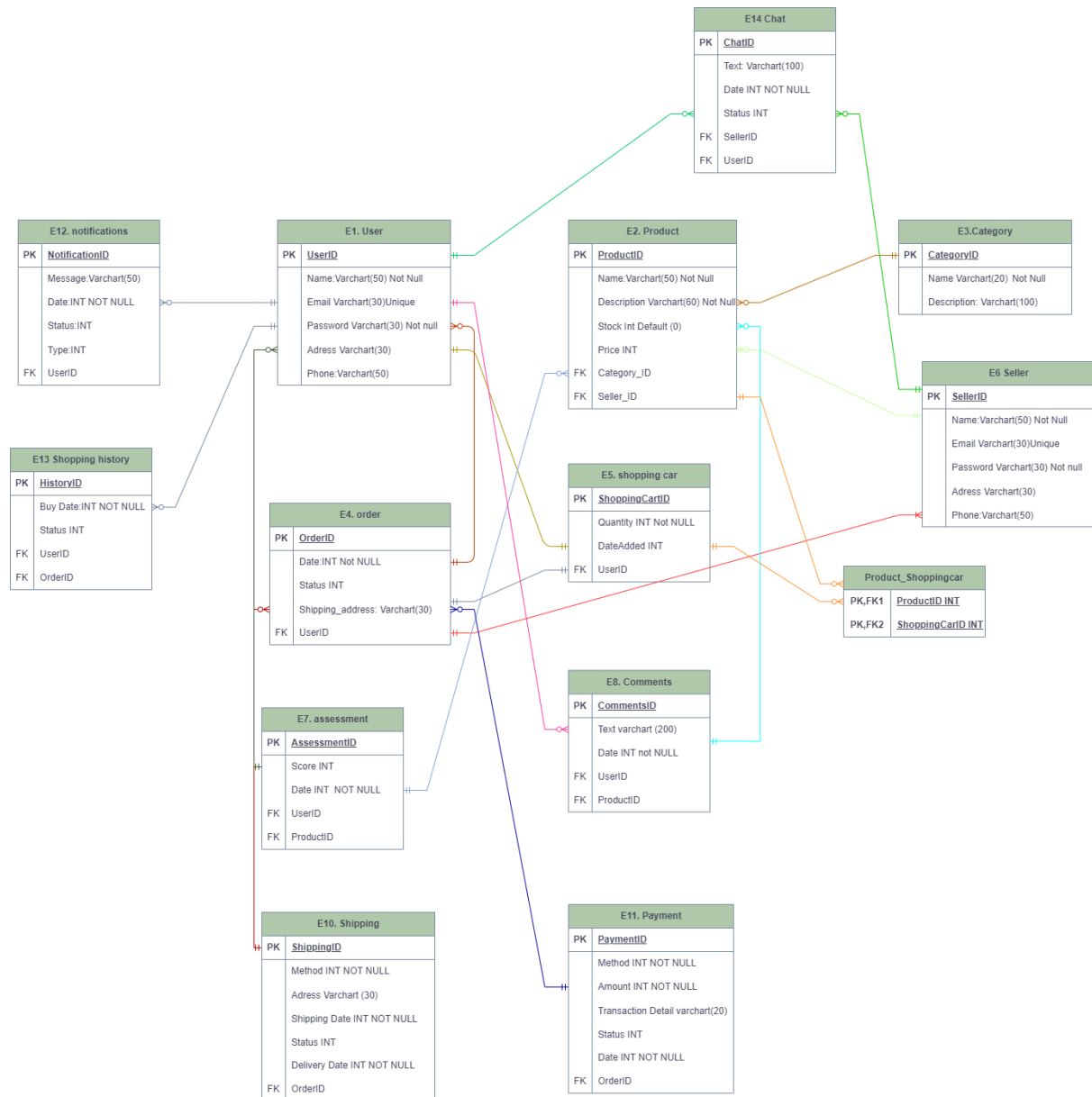
Database Management System: PostgreSQL or MongoDB for storing product listings, user profiles, transaction history, etc.

Data Warehousing: Amazon Redshift or Google BigQuery for analytics and reporting.

user stories

1. As a registered user, I want to be able to log in to the platform to access my account and manage it.
2. As a registered user, I want to be able to register on the platform by providing my personal and contact information to create a new account.
3. As a user, I want to be able to search for products by categories, keywords, or specific features to quickly and efficiently find what I need.
4. As a user, I want to see a list of products with images, detailed descriptions, prices, and ratings to make an informed decision about my purchase.
5. As a user, I want to be able to add products to my shopping cart and then proceed to the checkout process to complete the purchase.
7. As a site administrator, I want to be able to add, edit, and delete products from the catalog to keep the information up-to-date and accurate.
8. As a site administrator, I want to be able to manage user accounts, including the ability to approve registrations, reset passwords, and manage permissions.
9. As a site administrator, I want to be able to view sales reports, performance metrics, and inventory data to make informed business decisions.
10. As a user, I want to be able to leave reviews and ratings on the products I have purchased to help other users in their purchasing decisions.

3. CONCEPTUAL MODEL



3. RELATIONAL ALGEBRA

1. Buscar un usuario específico por su nombre de usuario y email:
 $\sigma (\text{UserID} = \text{'user_id'} \wedge \text{Email} = \text{'email'}) (\text{User})$
2. Display relevant data for a product with a specific name and within a price range:
 $\pi (\text{product_name}, \text{description}, \text{price}, \text{stock}, \text{seller}) (\sigma (\text{price} \geq \text{min_price} \wedge \text{price} \leq \text{max_price}) (\sigma (\text{product_name} = \text{'specific_name'}) (\text{product})))$
3. Display relevant data of a product through its category:
 $\pi (\text{product_name}, \text{description}, \text{price}, \text{stock}) (\text{Products} \bowtie \sigma (\text{Category_Name} = \text{'category_name'}) (\text{Categories}))$
4. View the date and status of the purchase history of a user:
 $\pi (\text{buy_date}, \text{status}) (\text{ShoppingHistory})$

5. View all orders of a seller by their shipping status:
 $\pi(\text{date, shipping_address, status, delivery_date, method}) (\sigma(\text{status} = \text{"entered_status"}) (\text{Shipping}) \bowtie (\text{Seller} \bowtie \text{Order}))$
6. Display the shopping cart list of a specific user:
 $\pi(\text{date_added, quantity, name, price}) ((\text{Product}) \bowtie ((\text{Product_ShoppingCart}) \bowtie (\text{User} \bowtie \text{ShoppingCart})))$
7. Display comments for a product:
 $\pi(\text{name, text, date}) (\text{Comments} \bowtie \sigma(\text{product_name} = \text{"product_name"}) (\text{Product}))$

5.

Strategy to Generate a Dummy Database:

- **Generate Random Data:**
 - Use programming languages like Python to generate random data for each entity. Libraries like Faker can be helpful for creating realistic names, addresses, emails, etc.
 - Maintain Relationships: Ensure that the generated data maintains the relationships between entities. For example, a product should have a valid category and be associated with a real vendor.
- **Data Consistency:**
 - Validate the generated data to ensure consistency. For instance, the stock of a product should not be negative, and the price should be realistic.
- **Scale the Data:**
 - Depending on your testing requirements, scale the data to create a large dataset. This will help you test the performance of your queries under different load conditions.
- **Data Injection:**
 - Use database scripts or ORM (Object-Relational Mapping) tools to inject the generated data into the database.
- **GitHub:** GitHub hosts a variety of open-source projects that may include sample datasets for testing purposes. You can search for repositories related to your domain and see if they provide any sample data.
- **Yelp Open Datasets:** This dataset comprises a portion of Yelp's business, reviews, and user data. It was originally compiled for the Yelp Dataset Challenge, providing students with an opportunity to delve into Yelp's data for research or analysis and to share their findings. In the latest dataset, you'll find details about businesses located in 8 metropolitan areas across the United States and Canada.