

Deadline 1

DBMS Project :
Defining Project Scope and Requirements

Shamik Sinha (2022468)

January 22, 2024

Project Scope

Traditional clothing and artifacts are integral to India's rich heritage, representing centuries-old craftsmanship and cultural significance. With the advent of the internet, people have become more aware about their own cultural heritage and the demand for these products has increased significantly. Yet, a missing link between customers and craftsmen hampers the industry. We aim to bridge this gap and elevate the accessibility of these cultural treasures.

The project envisions a user-friendly online retail store for traditional items, offering a diverse product catalog and seamless functionalities for buyers and sellers. Buyers enjoy a desirable shopping experience with customised carts and secure payments. Sellers benefit from efficient product management tools, real-time sales insights, and adaptability. Admins have control over inventory, while delivery agents use it as a central hub for order tracking and feedback.

The implementation will encompass essential back-end functionalities and an intuitive user interface, ensuring a seamless and satisfying shopping experience for users. The system would ideally support efficient searching through the catalog and be able to handle a large number of transactions.

Functional and Business Requirements

A customer should be able to login using a login ID or password if they were already registered, or they could make a new account by providing information like email address or phone number and setting up a password for themselves. They can browse through the catalogs available on the website and choose a product (and see its description) they would like to order. They can add the product to the cart if it is available in stock or they could add it to the wish-list. They can finally order the product by providing personal information like delivery address and payment information. They may need to provide details of their card/UPI account if they wish to pay digitally and not offline. After placing an order, customers can see the order status of their order. Customers can also leave a review for the product they bought if they wish to. They may also do the same for the delivery partner assigned to them. Customers can also see their history of all the ordered items and can mark an old order for return or cancel any existing orders if cancellation is allowed. Customers can use their E-wallet related to their accounts to receive compensation for any items they may have returned or to make payments for new items.

Shop owners need to register themselves on the platform. The shop owners can see the items which have been bought from their shops, and the revenue they have generated. They can also remove an item from the listing on the website or add a new item by requesting the web site administrator for the same. They may also choose to change the prices of the existing products.

The delivery agents can see the deliveries they have been assigned. They can see where the orders are to be picked up and where they need to be delivered. They can also see the amount of money they receive as compensation for the number of deliveries made successfully, using the E-wallet related to their account.

The web site administrator oversees all the products listed on the website. They can add or remove products. They may create discounts/special offers which may be availed by the customers.

Identifying Entity Sets and Relationship Sets

Entity Sets:

We identified the following entity sets as a part of the research done about how e-commerce platforms work and taking into consideration all the stakeholders and their needs.

- Customers
- Website Administrators
- Shop Owners
- Products
- Orders
- Product Review
- E-wallet
- Offers
- Catalogue
- Payment
- Cart
- Delivery Agents
- Wish-list
- Agent Review
- Order History

Relationship Sets:

Following are some of the relationship sets we identified. It is important to note, that the following list is not exhaustive.

- *order* **PlacedBy** *customer*
- *customer* **Leaves** *product review*
- *customer* **AddsTo** *cart*
- *customer* **Views** *order history*
- *customer* **Tracks** *delivery agent*
- *admin* **AddsItemTo** *catalogue*
- *admin* **RemovesItemFrom** *catalogue*
- *customer* **IsEligibleFor** *offers*
- *products* **AddedTo** *Cart*
- *delivery agent* **DeliversTo** *customer*
- *e-wallet* **BelongsTo** *customer*
- *catalogue* **Has** *products*
- *shop owner* **Checks** *e-wallet*

Overview of Features Available to Users

- **Customer**

- Create an account for logging in
- Browse the catalogue to search for products
- Add/Remove items from the cart or the wish-list
- Manage their wallet
- Track order status
- Leave review for products or delivery agents
- Check order history

- **Shop Owners**

- Create an account to sell products/Register themselves
- Add/Removes items they sell
- Change price of products they sell
- Manage their finance through wallet
- Manage/Check sales statistics

- **Website Administrator**

- Manage products available on the website
- Add/Remove products from the website
- Create/Remove discounts special offers
- Manage the catalogue

- **Delivery Agents**

- Manage deliveries assigned
- Ability to accept or reject new deliveries
- Manage and view their finance through wallet
- View reviews left by customers

Technical Requirements

We plan to make a full fledged project with both a functional front-end and back-end. The current selection of tools and technologies is as follows. However, it's worth noting that we remain open to adjustments in response to course guidelines or unforeseen developments during the development process. Adaptability is a key principle guiding our approach.

1 MySQL (Relational Database Management System)

MySQL, a free and open-source relational database management system, will be employed to handle diverse data storage and retrieval tasks, including bookings, payment management, parcel tracking, and dispatch management etc.

2 Django (Python-based Backend Framework)

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. We plan on using it for building the backend of our online retail store, enabling efficient product management, secure payments, and real-time insights for buyers, sellers, admins, and delivery agents.

3 React (JavaScript-based Frontend Library)

React, a JavaScript-based frontend library, empowers dynamic and interactive user interfaces. Its declarative syntax simplifies UI development, enabling efficient component-based architecture. We'll utilize React to craft a responsive and engaging user interface for our online retail store, ensuring a seamless and enjoyable shopping experience.

4 JavaScript

JavaScript, a versatile scripting language, is integral to our project. It will help us enhance the interactivity and responsiveness of our online retail store's front-end, ensuring dynamic features, smooth user interactions, and real-time updates.

5 Python3

Python 3, a powerful and user-friendly programming language, would form the backbone of our project. It facilitates efficient back-end development, data processing, and system integration.

6 Cascading Style Sheets

Cascading Style Sheets (CSS) would be pivotal in shaping our project's visual identity. It will be used to style the HTML elements, ensuring a cohesive and aesthetically pleasing design for our online retail store.

7 Hypertext Markup Language

Hypertext Markup Language (HTML) will be the fundamental structure of our project. It would define the content and layout of web pages for our online retail store.

Contributions

Shamik Sinha - 2022468