

ElectroBase Management System

EBMS - *an Online Electronics Retail Store*, built as a course project for
CSE202: Fundamentals of Database Management Systems

Deadline 1: *Defining Project Scope and Requirements*

Divyajeet Singh (2021529) Mehar Khurana (2021541)

January 25, 2023

Project Scope

Electronics has always been a booming industry. With the advent of the internet, the industry has seen a massive shift towards online retail. However, it is difficult to keep track of the technical requirements of a store. It is difficult to keep all stakeholders in the loop and keep them involved and updated.

This is where we come in with **EBMS**, i.e. the **ElectroBase Management System**. EBMS is an online retail platform for electronics. It aims to provide a common platform for suppliers, store managers, customers, and delivery agents.

- It is an easy solution for the **customers**, as it aims to provide a diverse catalogue of products to customers. The customers get to choose from a wide range of categories, make changes to their cart, and make secure payments with a method of their choice.
- The **suppliers** get to keep track of their products and change their description, price, etc. They can also keep track of their sales statistics and make changes to their products as and when required.
- The database managers (**admins**) get assisted in monitoring the transactions and managing the inventory. Based on their requirements, they can add deals or combos on the available products or remove categories from their store.
- EBMS provides a platform for **delivery agents** to keep track of all orders that have been assigned to them. They can set their activity/inactivity status and view the feedback given to them.

The primary focus of the project is to design an efficient backend. We aim to create a system that is smooth and easy to use for the customers and easy to manage for the suppliers. The system should support efficient searching through the catalogue and should be able to handle a large number of transactions.

The backend will be built using MySQL, along with Python and Django, and will be hosted on a server. The frontend will be built using HTML, CSS, and ReactJS. By the end of the semester, we plan to host this project on a public server and make it accessible publicly.

TL; DR

The aim of this project is to bring to life an integrated online retail store for electronics. The project will bring all stakeholders on a common platform and will ensure a smooth and easy-to-use experience for the customers.

Technical Requirements

Tech Stack

We plan for EBMS to be a full-stack project with a backend and a frontend. According to the requirements, we plan to use the following tools and technologies:

- MySQL Database
- Python-3
- Django Framework
- HTML
- CSS
- ReactJS

Entities, Stakeholders, Relations, & Constraints

The following entities will be used in the project:

1. **Customers:** Customers are the primary stakeholders interacting with the system. Customers have to register and log in first.
2. **Admin:** Admins are the store managers. They are the stakeholders responsible for managing the inventory and maintaining the store (database).
3. **Suppliers:** Suppliers are responsible for maintaining the products in the store.
4. **Delivery Agents:** Delivery Agents will be assigned to each Order and are responsible for delivering the orders.
5. **Reviews:** Reviews are the feedback given by Customers to Products and Delivery Agents.
6. **Products**
7. **Orders**

To effectively manage the database, we will be using the following (non-exhaustive list of) relationships among the data:

1. **Cart:** Product, Customer
2. **Sells:** Supplier, Product
3. **Sold to:** Product, Supplier, Customer
4. **Delivered:** Customer, Delivery Agent
5. **Reviewed:** Product, Review, Customer

Constraints on the cardinalities of entity sets limit the relationships and associations among data. The following are the cardinality-constraints of the relations:

1. **Many-to-One:** Product, Supplier
2. **Existential:** Product, Supplier
3. **One-to-Many:** Delivery Agent, Order
4. **Existential:** Order, Customer
5. **Existential:** Review, Product Sold to Customer

Since some data must remain private while other data must be accessible to all, we must implement access-control constraints. The following points show which stakeholders can access which data:

1. **Admin:** All data except for Customer Passwords
2. **Customer:** Personal records from the Customer entity set, their past and current Orders, and all Products and Reviews
3. **Supplier:** Their Product catalogue, personal records, sales statistics, and Customer Reviews
4. **Delivery Agent:** Personal records, current Orders, and Feedback from past Orders

Functional Requirements

All stakeholders except the **Admins** will need to create an account and log in. The features covered in this section also cover the functional requirements of the project:

1. As a **Customer:**
 - Add balance to wallet
 - Browse and search/sort/filter for products
 - Manage (add/remove) items in their cart
 - Place an order (checkout cart)
 - Confirm/Authenticate transaction
 - View and search for previous orders

2. As an **Admin**:

- Add/Delete categories
- Delete products
- Add/Modify discounts
- Create deals/combos
- View transaction history
- Appoint other Admins

3. As a **Supplier**:

- View products currently on sale
- Add/Discontinue a product
- Change price of a product
- Change quantity of a product
- View sales statistics

4. As a **Delivery Agent**:

- Confirm a delivery
- View the address of the order
- View the ETA of the order
- View feedback (which will also be stored as a review)
- Choose current activity status

Footnotes

This is the first draft of the project proposal. Some details, like the tech stack and the functional requirements, are currently tentative and are subject to change.