Requirements Document

Informal Description

Ezelectronics is a cutting-edge software solution tailored for electronics retailers, simplifying the management of products and enhancing the customer shopping experience through a dedicated online platform.

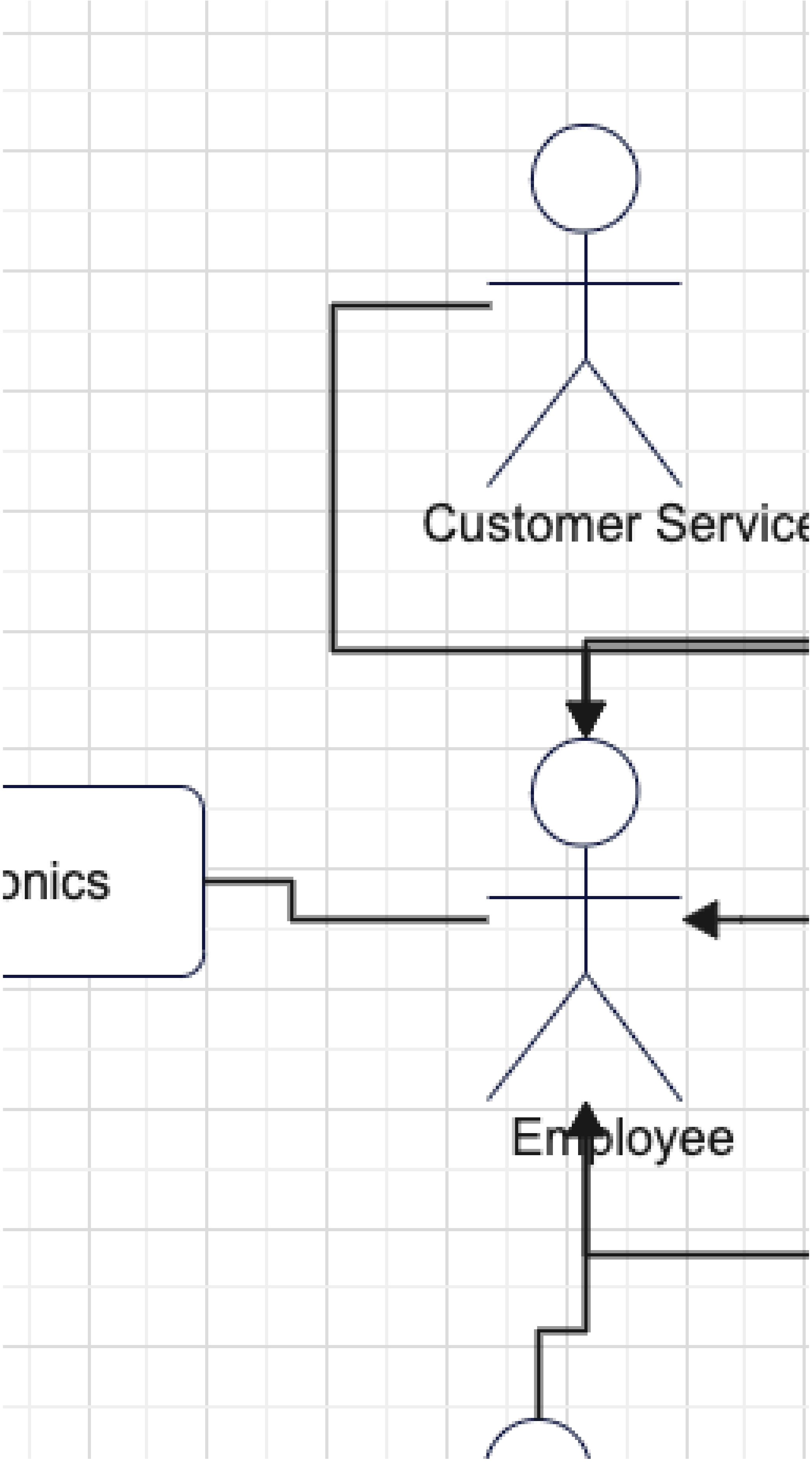
Store managers can efficiently oversee their inventory, seamlessly add new products, and effortlessly process purchases.

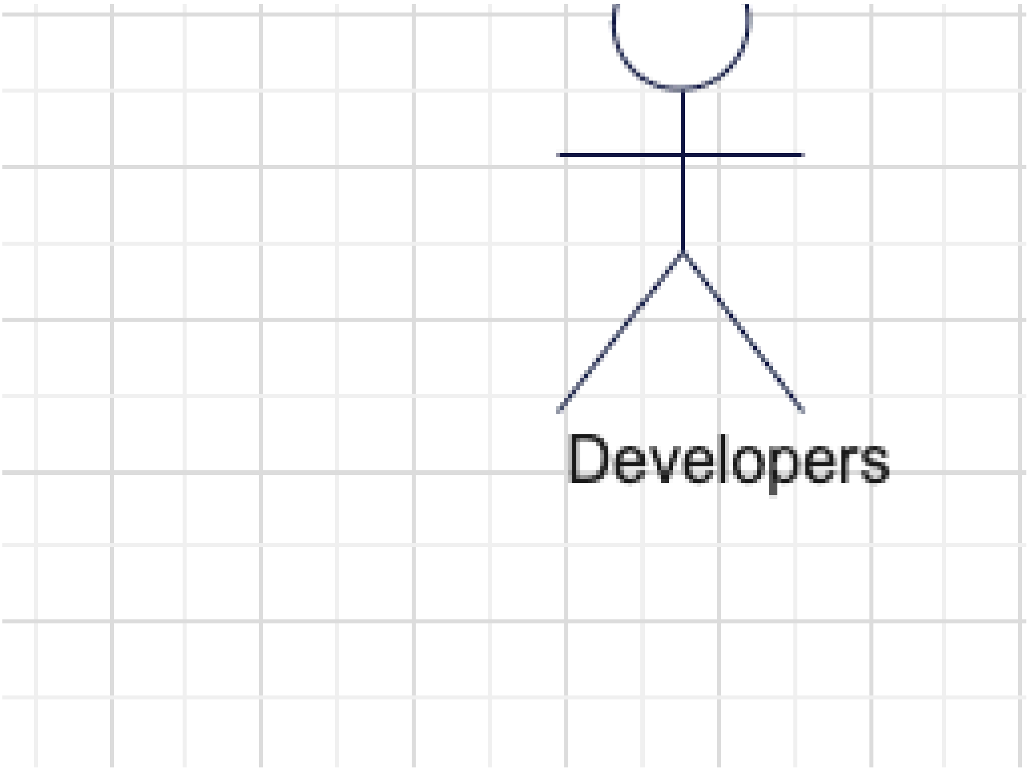
Customers benefit from a user-friendly interface, where they can browse a wide range of products, add items to their cart with ease, and conveniently track their purchase history.

Stakeholders

Stakeholder name	Description
Store Managers	to manage products, update inventory, and track sales.
Customers (Users - more specified, such as premium ones, valuable for business strategy)	who visit the dedicated website to browse and purchase electronics products more often.
Developers	The team responsible for building and maintaining the Ezelectronics software application, ensuring its functionality, security, and usability
IT Administrators	Oversee the installation, configuration, and maintenance of the Ezelectronics application within the store's infrastructure.
Sales Team	Utilizes the data and insights from Ezelectronics to optimize product offerings, pricing strategies, and promotional campaigns.
Marketing Team	Utilizes customer data and feedback from Ezelectronics to craft marketing strategies, including targeted promotions and advertising campaigns.
Customer Service	Assists customers with inquiries, issues, or feedback related to the Ezelectronics platform, and replies to reviews.

Context Diagram





Interfaces

Actor	Logical Interface	Physical Interface
Store Managers	Product management, inventory updates, sales tracking.	Web browser.
Customers (Users)	Browsing products, placing orders, providing feedback.	Web browser.
Developers	Software development, security, maintenance.	Development environment (IDE, Git & other tools).
IT Administrators	Installation, configuration, maintenance.	Server console (SSH), network cables & connection.
Sales Team	Data analysis, optimization.	Desktop/laptop (Excel, BI tools).
Marketing Team	Marketing strategy, customer segmentation.	Desktop/laptop (Email, CRM systems).
Customer Service	Customer support, issue resolution and feedbacks.	Desktop/laptop (Email, chat).

Stories & Personas

1-Persona: Alex;

Age: 24;

Role: customer;

Gender: M;

Level of education: PoliTo student;

Studies software engineering.

Country: Italy;

city: Turin;

Story: wants to buy a better laptop for better performance in his university carrier so he browses products, adds products to his cart, checks specifications and filters products by various criterias.

2-Persona: Leo;

Age: 46;

Role: manger;

Gender: M;

Level of education: postgraduate;

Owns electronics and appliances store;

country: Italy;

city: Turin;

Story: He wants to add products, view users, update product details.

Functional Requirements

ID	Description	API Method
FR1: User Authentication	Create a new session for user authentication	POST `/sessions`
FR2: User Logout	Log out the current user session	DELETE `/sessions/current`
FR3: Get Current Session Info	Get information about the current session	GET `/sessions/current`
FR4: User Registration	Register a new user	POST `/users`
FR5: View All Users	Get a list of all users	GET `/users`
FR6: View Users by Role	Get users by role	GET `/users/role/:role`
FR7: View Specific User	Get a specific user by username	GET `/users/:username`
FR8: Delete User	Delete a specific user by username	DELETE `/users/:username`
FR9: Delete All Users	Delete all users	DELETE `/users`
FR10: Add Product	Add a new product	`POST /products`
FR11: Record Product Arrivals	Record new product arrivals	POST `/products/arrivals`
FR12: Update Product Information	Update a product's information	PATCH `/products/:code`
FR13: View All Products	Get a list of all products	GET `/products`
FR14: View Specific Product	Get a specific product by code	GET `/products/:code`

FR15: View Products by Category	Get products by category	GET `/products/category/:category`
FR16: View Products by Model	Get products by model	GET `/products/model/:model`
FR17: Delete All Products	Delete all products	DELETE `/products`
FR18: Delete Specific Product	Delete a specific product	DELETE `/products/:code`
FR19: View Current Cart	Get current cart for user	GET `/carts`
FR20: Add Product to Cart	Add a product to the cart	POST `/carts`
FR21: Update Cart	Update products in the cart	PATCH `/carts`
FR22: View Purchase History	Get purchase history	GET `/carts/history`
FR23: Remove Product from Cart	Remove product from cart	DELETE `/carts/products/:productId`
FR24: Clear Current Cart	Clear current cart	DELETE `/carts/current`
FR25: Clear All Carts	Clear all carts	DELETE `/carts`
FR26: Get Rating	Rating fr each product.	
FR27: Get Review	Get review from customers	
FR28: Get Customer Address	Get address to ship the order	
FR29:		
FR30:		

Non-Functional Requirements

ID	Type (efficiency, reliability,)	Description	Refers to
NFR1	Usability	system is not too complex and there is few functions to use so it is very easy to learn Search bar allow user to find the product/user fastly Target search and ads with frequently viewed products.	All FR
NFR2	Efficiency	Minimal resource usage because the program is simple - Organised Database and DAO queries to optimise interaction with database.	All FR dao/db files
NFR3	Reliability	Defects visible by end user are already defined (cart not found, user not found product is sold)	Errors file
NFR4	Maintainability	Functions to add/modify/delete/getare already defined	All FR
NFR5	Portability	Expected to be portable across various environments, such as browsers on mobile	Erro

		phones, pc’s, laptops…. Typescript is a versatile and lightweight interface for web development, facilitating development lifecycle .	
NFR6	Security	user data carts are protected. Some function require logging in as manager other functions require logging in as costumer; Authentication is required to access and modify the different tables in the Database (Note: Passport.js is used for authentications). The session time is set to 1 DAY .	Routers File

Use Cases

Actors Involved	Customer, Manager
Precondition	Person doesn’t have a user account
Post condition	Creation of user account
Nominal Scenario	Scenario 1.1 (Customer sign up)
Variants	Scenario 1.2 (Manager sign up)
Exceptions	Scenario 1.3 (Username already exists);

Scenario 1.1

Use case 1, UC1: Registration:

Scenario 1.1	Customer creates an account
Precondition	Person doesn’t have user account
Post condition	Person has a new account
Step#	Description
1	Filling in username
2	Filling in Name
3	Filling in Surname
4	Filling in password
5	Choosing Customer role to create an account
6	Submitting inputs
7	All fields are being checked by system for not being empty
8	System checks if the username is not already taken

9	Person sees notification of successful creation of the account

Scenario 1.2	Manager creates an account
Precondition	Person doesn’t have user account
Post condition	Person has a new account
Step#	Description
1	Filling in username
2	Filling in Name
3	Filling in Surname
4	Filling in password
5	Choosing Manager role to create an account
6	Submitting inputs
7	All fields are being checked by system for not being empty
8	System checks if the username is not already taken
9	Person sees notification of the successful creation of the account

Scenario 1.3	Person fails to create an account due to existing username
Precondition	Person doesn’t have user account
Post condition	Person fails to create new account
Step#	Description
1	Filling in username
2	Filling in Name
3	Filling in Surname
4	Filling in password
5	Choosing role to create an account
6	Submitting inputs
7	All fields are being checked by system for not being empty
8	System checks if the username is not already taken
9	Person sees notification that username has been taken

Use Case 2, UC2: Log in

Actors Involved	Customer, Manager
Precondition	Person has existing account
Post condition	Person logs in to the system
Nominal Scenario	Scenario 2.1 (Person logs in)
Variants	
Exceptions	Scenario 2.2 (Username doesn't exist); Scenario 2.3 (Password is wrong)

Scenario 2.1	Person logs in
Precondition	Person has an existing Customer/Manager account
Post condition	Person logs in to the system
Step#	Description
1	Filling in username
2	Filling in password
3	Submitting inputs
4	All fields are being checked by system for not being empty
5	System checks if the username is in the database
6	System checks if the password is correct
7	Person sees notification that they're logged in

Scenario 2.2	Person fails to log in due to wrong username
Precondition	Person enters wrong username
Post condition	Person fails to log in
Step#	Description
1	Filling in username
2	Filling in password
3	Submitting inputs
4	All fields are being checked by system for not being empty
5	System checks if the username is in the database
6	System checks if the password is correct

7	Person sees notification that the username doesn’t exist

Scenario 2.3	Person fails to log in due to incorrect password
Precondition	Person enters incorrect password
Post condition	Person fails to log in
Step#	Description
1	Filling in username
2	Filling in password
3	Submitting inputs
4	All fields are being checked by system for not being empty
5	System checks if the username is in the database
6	System checks if the password is correct
7	Person sees notification that the password is incorrect

Use Case 3, UC3: Log out

Actors Involved	Customer, Manager
Precondition	Person is logged in as a Customer/Manager
Post condition	Person logs out
Nominal Scenario	Scenario 3.1 (Person logs out)
Variants	No variants
Exceptions	No exceptions

Scenario 3.1	Person logs out
Precondition	Person is logged in
Post condition	Person logs out
Step#	Description
1	Person does the log out action
2	Person sees notification about being logged out

Use case 4, UC4: Purchase

Actors Involved	Customer

Precondition	Person is logged in as a Customer; Cart is not empty
Post condition	Person makes a purchase as a Customer
Nominal Scenario	Scenario 4.1 (Person makes a payment)
Variants	Scenario 4.2 (Person removes a product from the cart): If the customer decides to remove a product from the cart before completing the purchase, they can do so using the appropriate endpoint Scenario 4.3 (Product Not Available): If any of the selected products are not available in the inventory or have already been sold before the purchase is completed, the system responds with a 404 error
Exceptions	New product is already in the cart

Scenario 4.1	Person logs out
Precondition	Person is logged in as a Customer; Cart is not empty
Post condition	Person makes a purchase as a Customer
Step#	Description
1	Customer selects products to purchase
2	Customer navigates to the checkout section to review the selected products
3	Customer sends a request to make the purchase
4	The system verifies that the customer is logged in and has selected valid products
5	The system calculates the total cost of the products in the cart
6	The system marks the products as sold
7	The system marks the products as sold
8	The system responds with a 200 status code, confirming the successful purchase transaction

Use case 5, UC5: Add Product to Cart

Actors Involved	Customer
Precondition	Customer is logged in and has selected a product to add to the cart
Post condition	Product is added to the customer's cart
Nominal Scenario	Scenario 5.1 (Add Product to Cart)

Variants	None
Exceptions	Scenario 5.2 (Product Already in Cart): If the selected product is already in the customer's cart, the system responds with a 409 error

Scenario 5.1	Add Product to Cart
Precondition	Customer selects a product to add to the cart
Post condition	Person makes a purchase as a Customer
Step#	Description
1	Customer selects a product to add to the cart
2	Customer sends a request to add the selected product to the cart
3	Customer sends a request to make the purchase
4	The system adds the product to the customer's cart
5	The system responds with a 200 status code

Use case 6, UC6: View Cart History

Actors Involved	Customer
Precondition	Customer is logged in and has completed purchases in the past
Post condition	Customer views the history of past cart transactions
Nominal Scenario	Scenario 6.1 (View Cart History)
Variants	None
Exceptions	None

Scenario 6.1	View Cart History
Precondition	Customer is logged in and has completed purchases in the past
Post condition	Customer views the history of past cart transactions
Step#	Description
1	Customer navigates to the cart history section
2	Customer sends a request to view the cart history

3	Customer sends a request to make the purchase
4	The system retrieves the history of past cart transactions for the logged-in customer
5	The system responds with the list of past cart transactions

Use case 7, UC7: Create New Product

Actors Involved	Manager
Precondition	Manager is logged in and has the authority to add new products to the inventory.
Post condition	New product is successfully added to the inventory
Nominal Scenario	Scenario 7.1 (Add New Product)
Variants	Scenario 5.2 (Invalid Product Information): If the manager provides invalid or incomplete information for the new product, the system responds with an error message prompting the manager to correct the information; Scenario 5.3 (Duplicate Product Code): If the provided product code already exists in the inventory, the system responds with an error message, indicating that the product code must be unique
Exceptions	None

Scenario 7.1	Add New Product
Precondition	Manager is logged in and has the authority to add new products to the inventory
Post condition	New product is successfully added to the inventory
Step#	Description
1	Manager navigates to the "Add Product" section of the application
2	Manager provides the necessary details for the new product, including code, selling price, model, category, and optional details
3	Manager submits the product information
4	The system validates the provided information, ensuring that all mandatory fields are filled and that the selling price is greater than zero
5	The system checks if the provided product code is unique and not already existing in the inventory.
6	If the validation passes, the system adds the new product to the inventory

7	The system responds with a confirmation message, indicating the successful addition of the new product

Use case 8, UC8: Delete Product

Actors Involved	Manager
Precondition	Manager is logged in and has the authority to manage products, including deleting them from the inventory
Post condition	The specified product is successfully removed from the inventory
Nominal Scenario	Scenario 8.1 (Delete Product)
Variants	Scenario 6.2 (Cancel Deletion): If the manager decides to cancel the deletion action after confirmation, the system aborts the deletion process and returns to the product list view; Scenario 6.3 (Invalid Product Selection): If the manager attempts to delete a product that does not exist or is not accessible, the system responds with an error message, indicating that the specified product cannot be deleted
Exceptions	None

Scenario 8.1	Add New Product
Precondition	Manager is logged in and has the authority to add new products to the inventory
Post condition	New product is successfully added to the inventory
Step#	Description
1	Manager navigates to the "Manage Products" section of the application
2	Manager selects the option to view the list of products
3	Manager identifies the product to be deleted from the list
4	Manager selects the option to delete the identified product
5	The system prompts the manager to confirm the deletion action
6	Manager confirms the deletion of the product
7	The system removes the specified product from the inventory
8	The system responds with a confirmation message, indicating the successful deletion of the product

Use case 9, UC9: Register Arrived Products

Actors Involved	Manager
Precondition	Manager is logged in and has the authority to manage products, including registering arrivals of new products
Post condition	The newly arrived products are successfully registered in the inventory
Nominal Scenario	Scenario 9.1 (Register Arrived Products)
Variants	Scenario 9.2 (Invalid Information): If the manager provides incomplete or invalid information about the arrived products, the system prompts for corrections and prevents registration until all requiqed fields are properly filled Scenario 9.3 (Duplicate Product Code): If the manager attempts to register products with codes that already exist in the inventory, the system responds with an error message, indicating that the provided product codes are already in use
Exceptions	None

Scenario 9.1	Register Arrived Products
Precondition	Manager is logged in and has the authority to add new products to the inventory
Post condition	New product is successfully added to the inventory
Step#	Description
1	Manager navigates to the "Manage Products" section of the application.
2	Manager selects the option to register new arrivals
3	Manager provides the necessary information about the arrived products, such as model, category, quantity, selling price, and optional details
4	The system validates the provided information, ensuring that all required fields are filled and the data formats are correct
5	If validation is successful, the system registers the newly arrived products in the inventory, assigning them unique codes and recording the arrival date
6	The system responds with a confirmation message, indicating the successful registration of the arrived products

Use case 10, UC10: User Management

Actors Involved	Manager

Precondition	Manager is logged in and has administrative privileges to manage users
Post condition	User management actions (such as creation, retrieval, update, and deletion) are successfully executed based on the manager's actions
Nominal Scenario	Scenario 10.1 (Manage Users)
Variants	cenario 10.2 (Invalid User Information): If the manager provides incomplete or invalid information for user creation or update, the system prompts for corrections and prevents the action until all required fields are properly filled Scenario 10.3 (Existing Username): If the manager attempts to create a new user with a username that already exists in the system, the system responds with an error message, indicating that the username is already in use Scenario 10.4 (User Deletion Confirmation): When deleting a user, the system asks for confirmation to prevent accidental deletions
Exceptions	None

Scenario 10.1	Register Arrived Products
Precondition	Manager is logged in and has the authority to add new products to the inventory
Post condition	New product is successfully added to the inventory
Step#	Description
1	Manager navigates to the "Manage Products" section of the application
2	Manager selects the option to create a new user
3	Manager fills in the required information for user creation, including username, name, surname, password, and role
4	The system validates the provided information, ensuring all fields are filled and the username is unique
5	If validation is successful, the system creates a new user account with the provided details
6	Manager receives a notification confirming the successful creation of the user account

Scenario 10.1	Create New User
Precondition	Manager is logged in and has the authority to add new products to the inventory

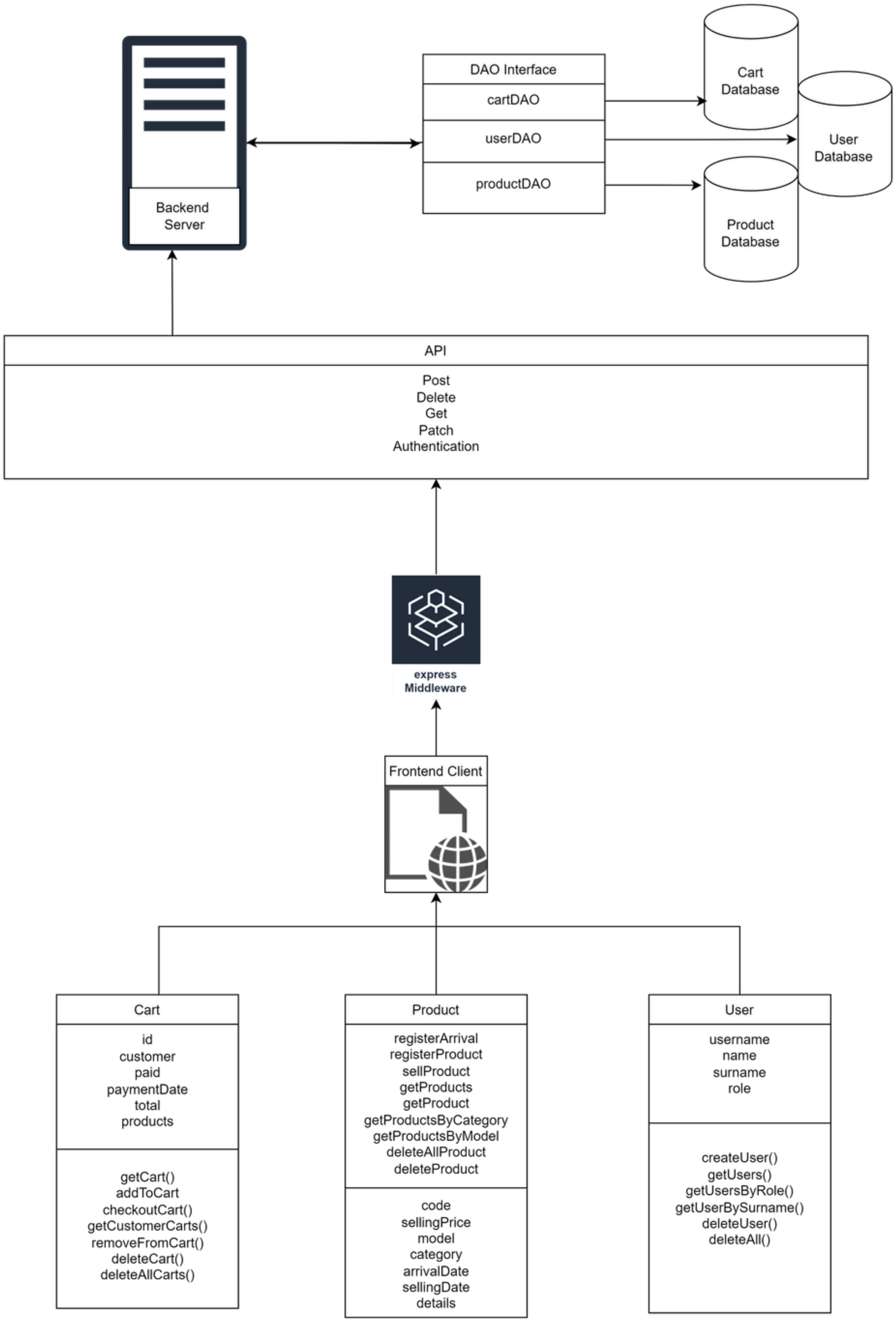
Post condition	New product is successfully added to the inventory
Step#	Description
1	Manager navigates to the "Manage Products" section of the application
2	Manager selects the option to create a new user
3	Manager fills in the required information for user creation, including username, name, surname, password, and role
4	The system validates the provided information, ensuring all fields are filled and the username is unique
5	If validation is successful, the system creates a new user account with the provided details
6	Manager receives a notification confirming the successful creation of the user account

Glossary (Demo)

1. **User**: An individual registered within the EZElectronics system, typically identified by a unique username and associated with a name, surname, and role.
2. **Customer**: A type of user within the EZElectronics system who interacts with the platform primarily to browse and purchase products.
3. **Manager**: A type of user within the EZElectronics system who has administrative privileges, including managing users, products, and inventory.
4. **Product List**: A collection of products available within the EZElectronics system, typically displayed to users for browsing and selection.
5. **Inventory**: A database or record system containing information about available products, including details such as quantity, model, category, and arrival date.
6. **Carts History**: A record of past carts associated with a specific user, documenting previously purchased items and transactions.
7. **Cart**: A temporary storage container within the EZElectronics system where users can add and manage selected products before proceeding to checkout.
8. **Product Record**: An entry or instance within the inventory database containing detailed information about a specific product, including its unique code, model, category, and availability status.
9. **Product**: The physical or digital item available for purchase within the EZElectronics system, typically represented by its model, category, and other descriptive attributes.
10. **Authentication**:
    - Authenticator: A class responsible for user authentication using Passport.js.
    - Auth Middleware: Middleware functions for checking user authentication status and roles.
11. **User Management**:
    - User Class: Represents user data.
    - User DAO: Handles database operations related to users.
    - User Errors: Custom error classes for user-related errors.
    - User Routes: Express routes for user registration and login.
12. **Cart Management**:
    - Cart Class: Represents cart data.
    - Cart DAO: Handles database operations related to carts.
    - Cart Errors: Custom error classes for cart-related errors.
    - Cart Controller: Controller for cart operations.
    - Cart Routes: Express routes for cart management.

13. **Product Management**:
    - Product Class: Represents product data.
    - Product DAO: Handles database operations related to products.
    - Product Errors: Custom error classes for product-related errors.
    - Product Controller: Controller for product operations.
    - Product Routes: Express routes for product management.
14. **Error Handling**:
    - ErrorHandler: Centralized error handling middleware.
    - Custom Error Classes: Custom error classes for different types of errors in each module.
15. **Express Server**:
    - Express App: Configuration and initialization of the Express server.
    - Main Routes: Mounting of routes for authentication, user management, cart management, and product management.
16. **Database**:
    - SQLite3 Database: Connection setup and configuration.
17. **Validation**:
    - Express Validator: Middleware for request validation in routes.
18. **Utility Functions**:
    - Utility Module: Contains utility functions used across the application.
19. **Testing**:
    - Test Scripts: Provided test scripts in package.json for running tests.
    - Test Database Setup: Configuration for using a separate test database.
20. **Documentation**:
    - Requirement Document Template: Including sections for authentication, user management, cart management, product management, error handling, and testing.

Deployment Diagram


Backend Server
DAO Interface
cartDAO
userDAO
productDAO
Cart Database
User Database
Product Database
API
Post
Delete
Get
Patch
Authentication
express Middleware
Frontend Client
Cart
id
customer
paid
paymentDate
total
products
getCart()
addToCart
checkoutCart()
getCustomerCarts()
removeFromCart()
deleteCart()
deleteAllCarts()
Product
registerArrival
registerProduct
sellProduct
getProducts
getProduct
getProductsByCategory
getProductsByModel
deleteAllProduct
deleteProduct
code
sellingPrice
model
category
arrivalDate
sellingDate
details
User
username
name
surname
role
createUser()
getUsers()
getUsersByRole()
getUserBySurname()
deleteUser()
deleteAll()