SPORTS JERSEY E-SHOPPER

MULUKEN DENEKE | FINAL PRESENTATION | SENIOR PROJECT

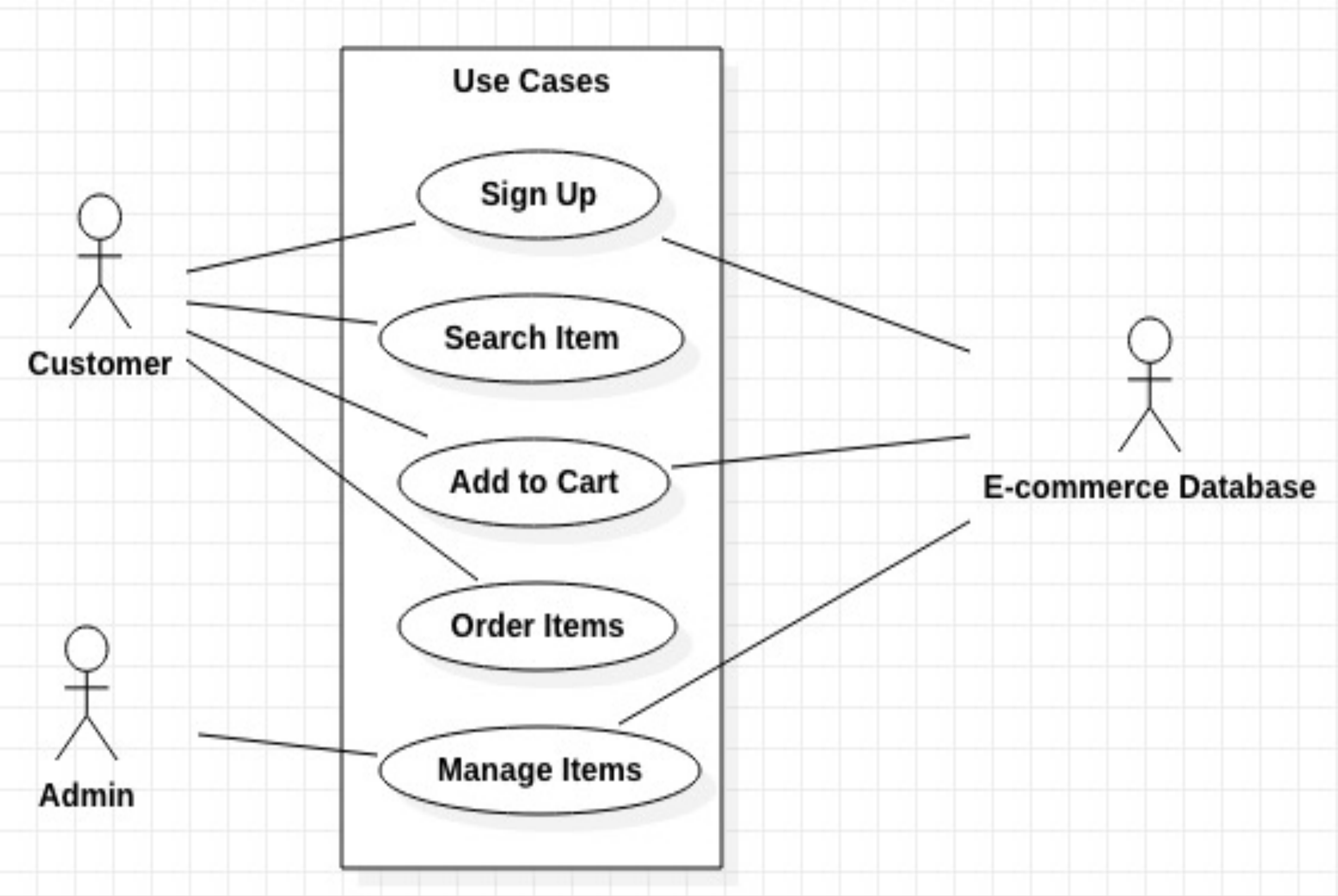
PROBLEM STATEMENT

- The problem of searching and purchasing multiple sport jerseys from multiple websites affects customers.
- The impact of which is time consuming, overcostly in money, inconvenient, hard to search and find.
- A successful solution would be one website which has multi-sport jerseys available for online shoppers.

INTRODUCTION

- Shop All is an e-commerce website that provides the most convenient way to buy sport jerseys.
- Shop All has four of the most popular sports (baseball, basketball, Soccer and Football) jerseys all available for purchase on a single website.
- Shop All customers are able to save time, money and energy as well as avoid online scammer websites whenever they purchase a jersey.

USE-CASE DIAGRAM



USE-CASE DESCRIPTION

CUSTOMER ORDERS ITEMS

- Brief Description
  - This use case allows the customer to order items.
- Actors
  - Customer
- Preconditions
  - Customer is at the home page
- Postconditions
  - System updates inventory in database
- Business Rules
  - User address and Credit card must be valid
  - Ordered items size and amount must exist in inventory
- Nonfunctional requirements - None

FLOW OF EVENTS

User Action	System Response
1. Customer searches an item	1. List of items is displayed
2. Customer selects an item	2. System displays items description
3. Customer selects size, and amount then submits order	3. System adds the items to the Order List and displays payment form
4. Customer fills payment information including shipping address	4. System process payment information and notifies the customer of successful payment

USE-CASE DESCRIPTION

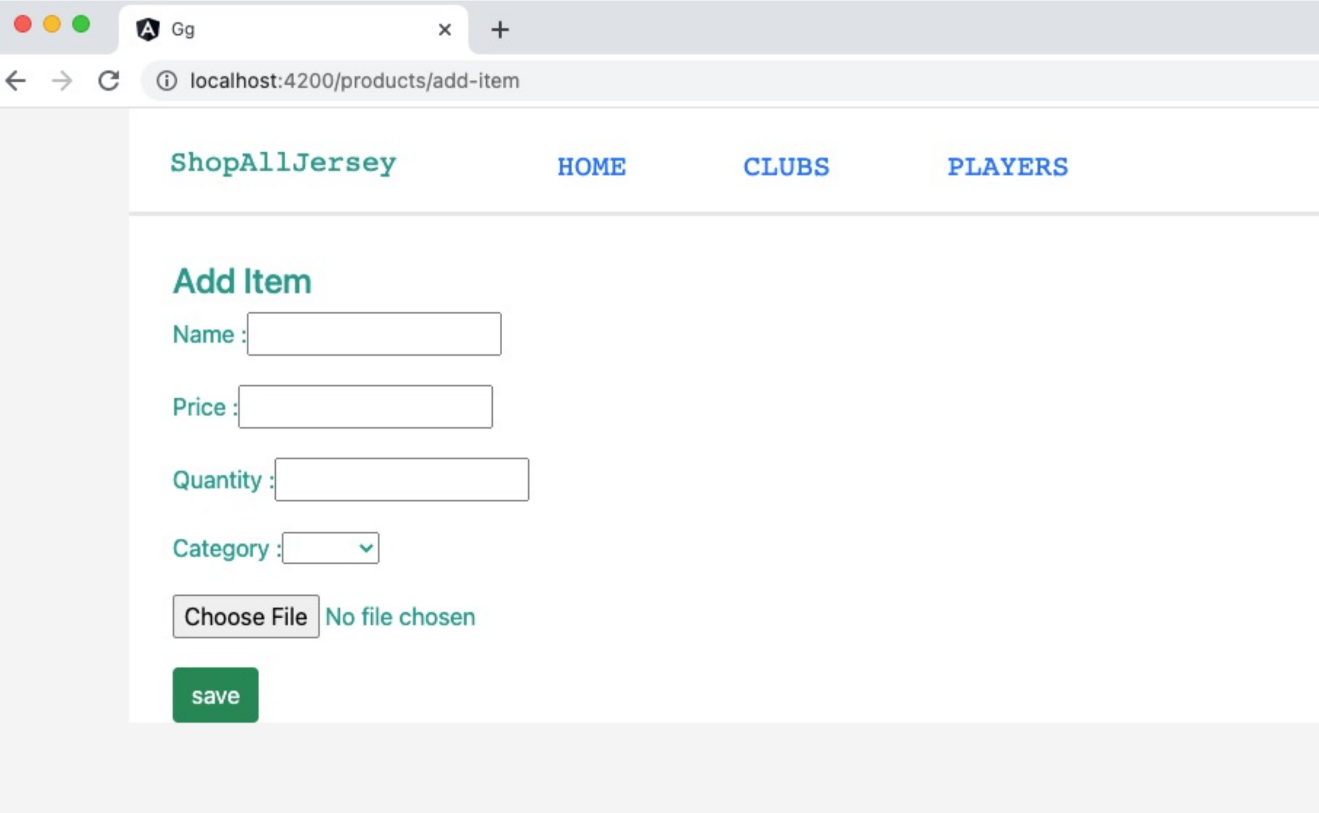
SUPER USER ADDS ITEMS

- Brief Description
  - This use case allows a super user to add items.
- Actors
  - A Super User
- Preconditions
  - A super user is at the home page
- Postconditions
  - System adds items in the right category and updates inventory in database
- Business Rules
  - Super User inputs valid item information
  - Added items must exist and match inventory
- Nonfunctional requirements - None

FLOW OF EVENTS

User Action	System Response
1. Super User heads over to add-item page	1. System displays add-item page
2. Super user fills the Name, Price, Quantity, selects a Category and then attaches an image of the item	2. System displays items description
3. Super user clicks the save button to add item	3. System adds the items to right category
4. Super user checks if the item is added under the right category.	4. System loads item in the right added category.

ADD ITEM FORM



Gg

localhost:4200/products/add-item

Add Item

Name :

Price :

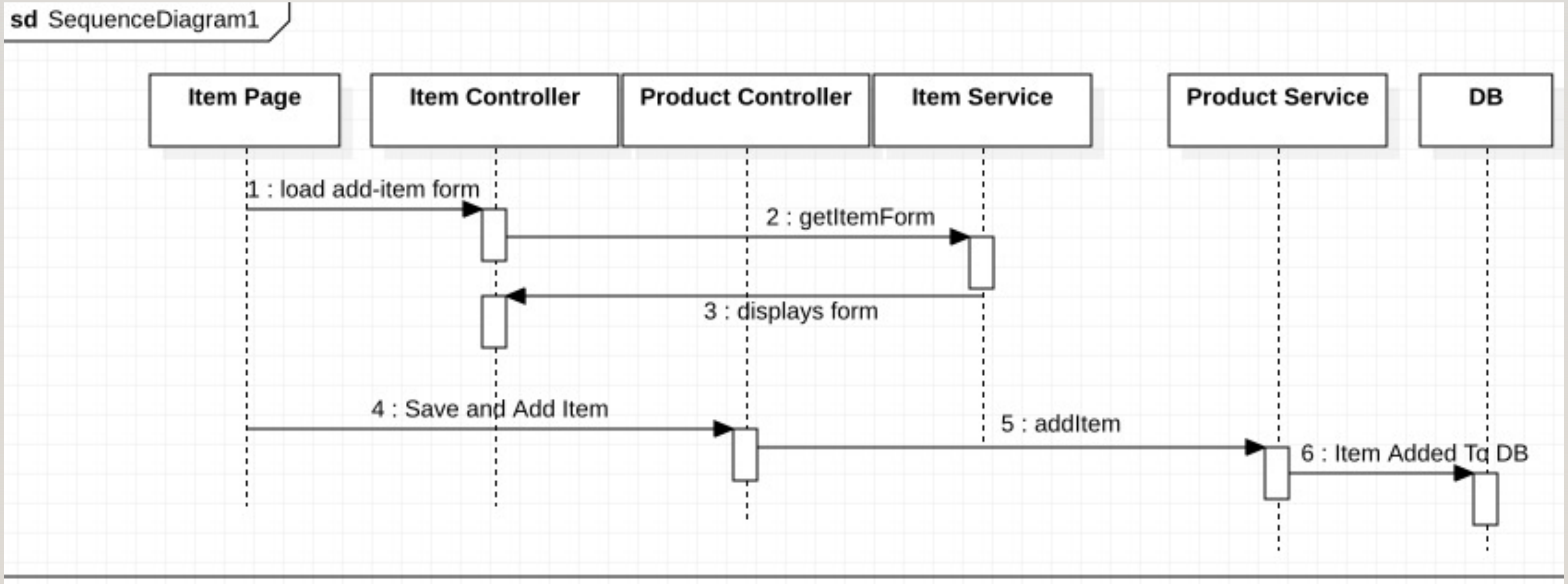
Quantity :

Category :

Choose File No file chosen

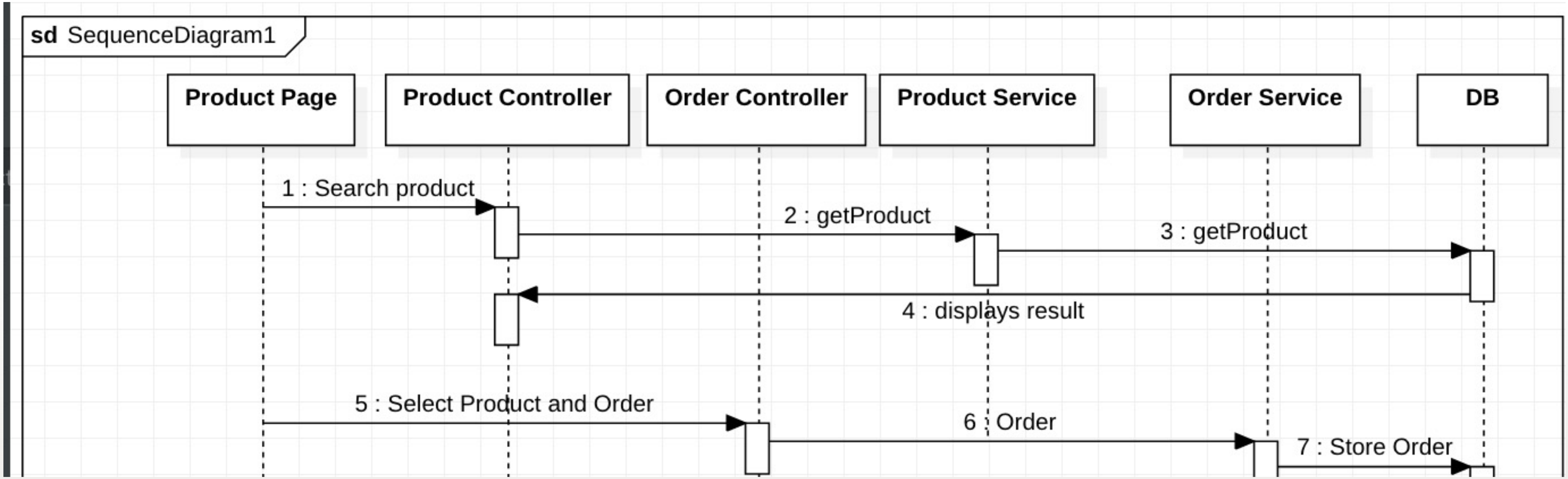
save

ADD ITEM
SEQUENCE DIAGRAM



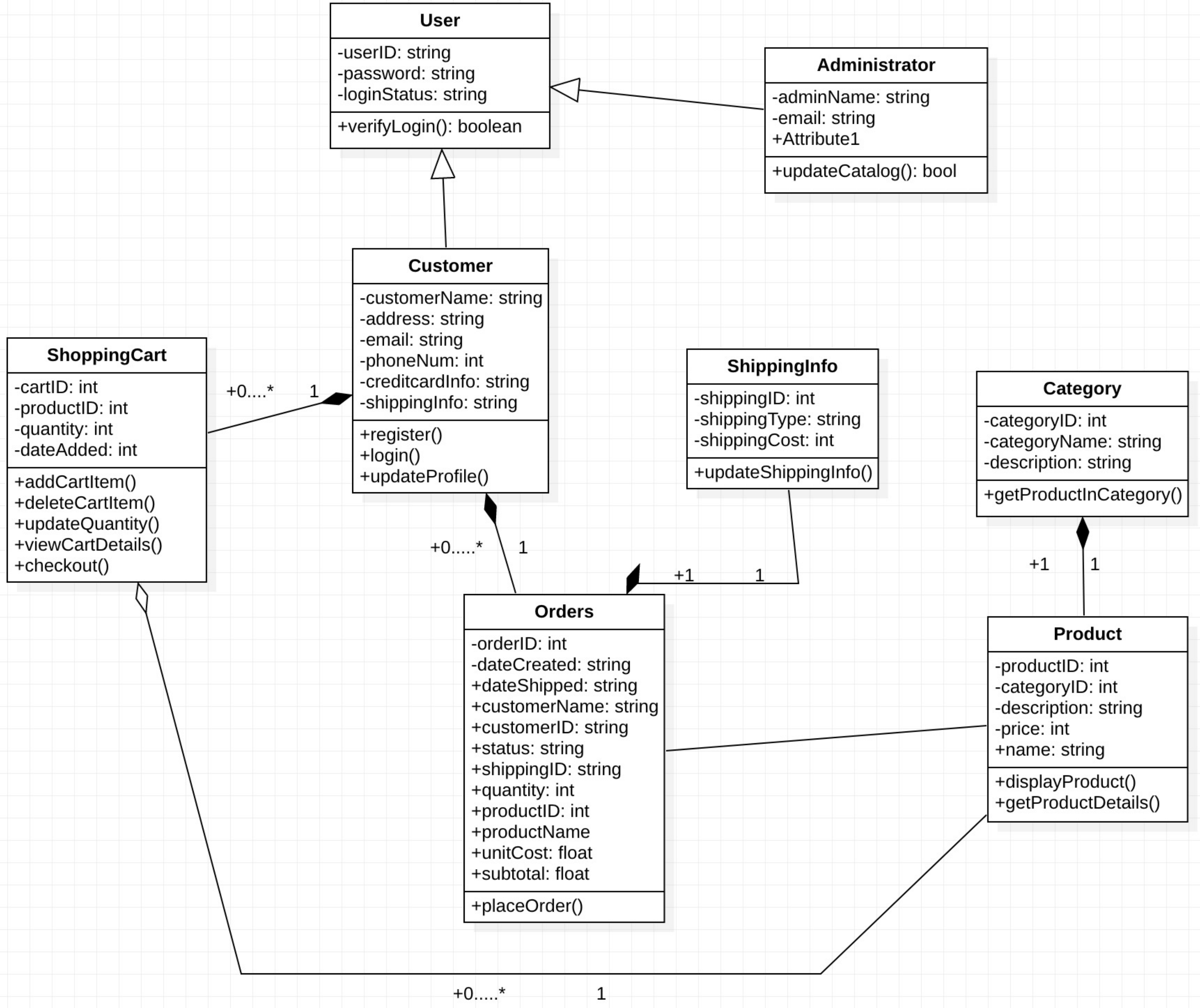
KEY ABSTRACTIONS

Register	Administrator
Customer	Item
Category	Inventory
Cart	Order
Payment	E-Commerce Database



ORDER ITEM SEQUENCE DIAGRAM

Class Diagram



User
-userID: string
-password: string
-loginStatus: string
+verifyLogin(): boolean

Administrator
-adminName: string
-email: string
+Attribute1
+updateCatalog(): bool

Customer
-customerName: string
-address: string
-email: string
-phoneNum: int
-creditcardInfo: string
-shippingInfo: string
+register()
+login()
+updateProfile()

ShoppingCart
-cartID: int
-productID: int
-quantity: int
-dateAdded: int
+addCartItem()
+deleteCartItem()
+updateQuantity()
+viewCartDetails()
+checkout()

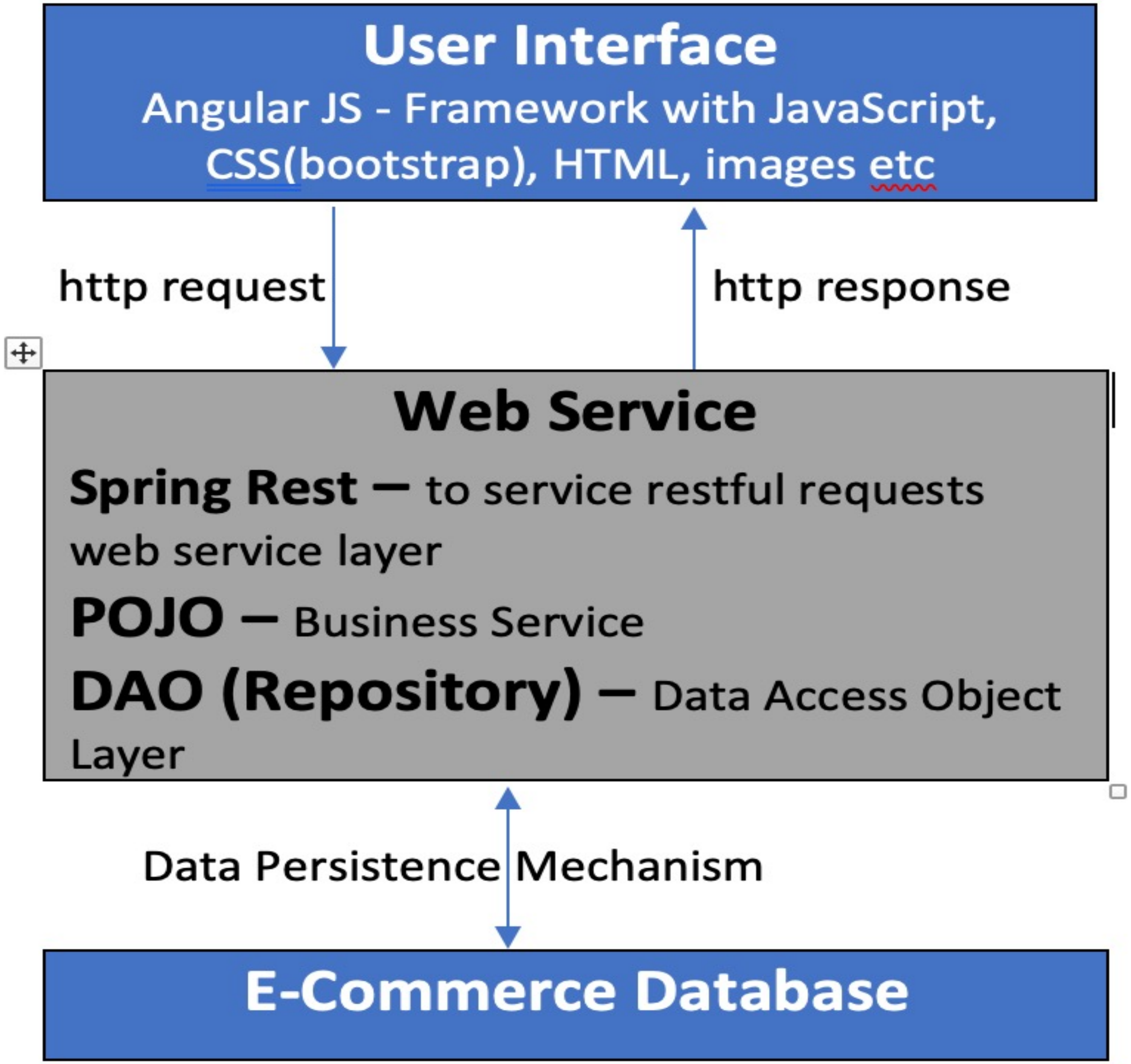
ShippingInfo
-shippingID: int
-shippingType: string
-shippingCost: int
+updateShippingInfo()

Category
-categoryID: int
-categoryName: string
-description: string
+getProductInCategory()

Orders
-orderID: int
-dateCreated: string
+dateShipped: string
+customerName: string
+customerID: string
+status: string
+shippingID: string
+quantity: int
+productID: int
+productName
+unitCost: float
+subtotal: float
+placeOrder()

Product
-productID: int
-categoryID: int
-description: string
-price: int
+name: string
+displayProduct()
+getProductDetails()

SYSTEM LAYERED ARCHITECTURE



User Interface

Angular JS - Framework with JavaScript, CSS(bootstrap), HTML, images etc

http request

http response

Web Service

Spring Rest – to service restful requests web service layer

POJO – Business Service

DAO (Repository) – Data Access Object Layer

Data Persistence Mechanism

E-Commerce Database

ANGULAR BASICS

- Angular has component-based approach to developing web applications.
- Every Component can be assigned a tag or a selector which is how the user can use the component.
- Assign a name to a component and assign a selector.
- A selector is what a consumer can use to call and render components
- Root component – holds the main components that needs to be displayed in the page.
- Components can have subcomponents.

DEMO SHOPALL

http://localhost:4200/products/home