**Glossary:**

|  |  |
| --- | --- |
| **Subject** | **Explain** |
| **Member information** | * ID * Email * Username * Password * Name * Date of birth * Address |
| **Valid** | No **Member** exists such that it has the same Email, Username or ID as the provided information. |
| **Login information** | * Username * Email * Password |
| **Valid** | A **Member** exists such that a provided the Username, Email and Password match with their **Member information**. |
| **Shipping information** | * Name * ID * Address |
| **System/Market System** | Represents everything that isn’t controlled by **users**. Responsible for transferring information, presenting options and so on |
| **User** | A person who is visiting the **System**. |
| **Registration/Sign-up** | A prerequisite for **Login**. A process in which a **Guest User** can provide **Valid Member Information** and request that the **System** will save saidMember Information. |
| **Login** | A process in which a **Member** can get access to additional functionalities such as selling **Products** or **founding a store.** Moreover, the **System** will save a Member’s **Shopping cart** even after **Logout**.  In order to log in a **User** must provide **Login information**. |
| **Logout** | The opposite of **Login**. A process in which a **Logged in Member** switches state to a **Guest** |
| **Founding a Store** | A process in which a **Logged in Seller** can create a new **Store** and become into a **Store Founder.** |
| **Appoint Store owner** | A process in which a **Store Owner** can appoint a **Member** (who isn’t already one of the store owners) to become one of the Store owners. |
| **Appoint Store Manager** | A process in which a **Store Owner** can appoint a **Member** (who isn’t already one of the Store Owners or **Store Managers**) to become one of the Store Managers. |
| **Guest** | A **user** labeled as Guest isn’t **logged in** to the system. That mean that it’s a **user** with only the base functionalities of the **system** such and buying **products**, **logging in** or **signing up** |
| **Member** | A member is a **user** who is **registered** to the **System** in the past. Upon **login**, a member will get access to more functionalities such as **founding** **a** **store** or **managing** a store. |
| **Buyer** | Any **User** that is not **logged in** as **Admin** |
| **Seller** | A **Member** who sells products |
| **Store Founder** | A **Seller** who **founded a** **store**. Has additional functionalities such as manage **inventory**, appoint additional **owners**, appoint **managers** or manage **store policies** |
| **Store Owner** | A **Seller** who owns a **store**. Has access to special functionalities such as manage **inventory**, appoint additional **owners**, appoint **managers** or manage **store policies**.  A store owner will always have a single appointer who appointed them to be an owner. |
| **Store manager** | A **Member** that was appointed by a **Store owner** to manage a **store**. Has additional functionalities based on the permissions given to him by his appointer |
| **Admin** | A **User** that manages the **System**. Has access to System wide functionalities such as viewing all **transactions** in the system |
| **Product** | An item of value that is sold in a **store**. Can be bought and sold. |
| **Basket** | A basket represents a group of **products** that came from the same **Seller or Store**. A basket contains products |
| **Shopping cart** | Each **user** has a shopping cart in which he can place his chosen **products** (whether digital or physical). A shopping cart is divided to **Baskets**. A user can view and edit his shopping cart. |
| **Transaction/Purchase information** | Represents a single sale between a **buyer** and at least one **Seller or Store**. Saves information about **products** andtheir amounts and prices |
| **Store** | A location (physical or otherwise) in which a **Seller** can sell **products**. |
| **Store policy** | The combination of a **Discount Policy** and a **Purchase Policy** for a specific **Store** |
| **Discount** | A reduction in the cost of a **Product** set by the **Seller** |
| **Visible Discount** | A **Discount** that is visible to all **Buyers.** Consist of a percentage and a duration, for the duration the **Product** will cost (percentage) less |
| **Conditional Discount** | A **Discount** that is visible to all **Buyers.** Consists of a percentage, duration and condition. For the duration, if the condition is met, the **Product** will cost (percentage) less. |
| **Hidden Discount** | A **Discount** that consists of a duration, percentage and a code. For the duration, during **Checkout** a **Buyer** can enter the code and the **Product** will cost (percentage) less |
| **Discount Policy** | Information the represents conditions and parameters under which a **Buyer** will get a **Discount** for a **Product**. |
| **Purchase Policy** | Represents information chosen by a **Seller** regarding which **Buyers** can buy **products** from this seller, what are the **Buying options** for each product and what are the conditions or parameters for **purchase**. |
| **Buying option** | Different options for **Purchasing** a **Product** set by a **Seller** |
| **Direct Purchase** | A **Buyer** must pay the listed price and **purchases** the **Product** immediately. |
| **Offer-Purchase** | A **Buyer** must offer a sum for the **Product.** Afterwards the **Seller** can accept the offer (in which case the Buyer must pay the sum to purchase the Product immediately similar to **Direct Purchase**), Decline or counter-offer (the buyer then must pay the counter offer sum in order to purchase the product). In this form a buyer must get the seller’s permission in order to purchase the product |
| **Auction Purchase** | Has a starting price and a duration. For the duration, a **Buyer** can offer a sum (higher than the current price) in which case his offer will become the current price. At the end of the duration the buyer with highest offer must pay the sum they offered and **Purchase** the **Product.** |
| **Lottery Purchase** | Has a Product-price and a duration. For the duration a **Buyer** can buy “chances” to win the product for any amount he wishes as long as the total amount from all the buyers does not exceed the product-price.  If the product-price is achieved, a buyer will be selected at random (based on the chances they bought) and they will get the **Product** otherwise if the duration ended and the target price was not achieved each buyer will get the money they payed back. |
| **Store Inventory** | A representation of the offered **products** in a specific **Store** |
| **Checkout** | A state in which a **Buyer** can choose to **Purchase** every **Product** in their **Shopping cart**. |
| **Purchase** | the process in which a **Buyer** pays for a **Product** sold by a **Seller** |
| **Enter/Visit the System** | The process of connecting to the System and becoming a **Guest User** (either by physically walking into a market, opening up a site online and so on) |
| **Product search information** | * Product name * Product ID * Product Category * Product’s keywords |
| **System database** | A representation of all the data in the **System** (could be physical like books or non-physical such as an online database) |
| **Total price** | Represents the total price a **Buyer** must pay for all the **Products** in their **Shopping Cart** |
| **Payment information** | * Credit card number * Expiration date * ID * Name * Security code |
| **User Status** | The current state of a **User** (**Guest**, **Buyer**, **Admin** and so on) |
| **Personal Purchase/Transaction history** | A sub-set of all the **Transactions** in the System. In this case those are all transactions in which a specific **User** was the **Buyer** |
| **Sub-Manager** | A **Store Manager** that has an **appointer**. The Appointee is the Sub-Manager of the Appointer |
| **Store Purchase/Transaction History** | A sub-set of all the **Transactions** in the System. In this case those are all transactions in which a specific **Store** was the **Seller** |