**Application’s Glossary**

|  |  |  |
| --- | --- | --- |
| **Term** | **Explanation** | **Comment** |
| **Administrator** | A registered user of the system, which has access to some specific tools, such as: opening/closing stores, monitoring user’s actions, viewing order’s history, etc. | An admin can be appointed by another admin only. |
| **Bucket** | A list of all the products which a buyer wants to purchase - from a SPECIFIC store, that he chose items from. | A bucket contains products that were chosen by the buyer from the same store. |
| **Cart** | A list of all the products which a buyer wants to purchase (from all of the stores that he chose from). | A buyer can choose different products from many stores and pay for all of them in the same purchase. |
| **Contact information** | The email address and the phone number of a specific Member. | Each member should fill this mandatory information when checking out with the his cart. |
| **Database Module** | An external system that stores all of the market’s information, such as: user’s information, store’s information – including products, stock, purchases history, etc. |  |
| **Delivery Module** | An external system that checks the availability of a delivery for the desired products that has been purchased by the buyer – and informs the market about it. |  |
| **Employment** | The connection between a seller and a store:  A seller may be a store owner or a store manager. | A store may have a lot of employees, of different roles. |
| **Founder** | The first owner of a specific store.  Has all of the store’s management permissions | Hence, the Member who has opened the store. |
| **Guest** | A buyer, who isn’t logged-in into the system (or even not registered to the system). | Any user who isn’t logged-in nor registered, is considered as a guest. |
| **Item** | A product which the user chooses from a specific store (which he wants to purchase). | A specific product which the user chooses from a store – turns into an item which the user may checkout with and pay for. |
| **Manager** | An employee of the store (a seller) who has some additional permissions, in addition to the basic seller. The permissions may be set by an owner of the store – by defining management permissions for this employee when hiring him. | A manager may be ‘promoted’ – meaning that he may have his permissions upgraded, or ‘demoted’. |
| **Market** | The whole system is called a ‘market’ – which has stores, users and orders. |  |
| **Management permissions** | Each Store Manager (Manager) and Store Owner may have different permissions that has been set for him, in a specific store, by his employer (one of the owners of this store). | The permissions may vary from the stock management of the store, and to managing the employees of the store. |
| **Default managing permissions** | The most basic permissions that may be set for a Store Manager or a Store Owner. |  |
| **Member** | A guest that has logged in with his unique password. | Anyone can become a member by registering to the system. |
| **Notification** | A general component which can be shared between users of the system.  A notification may be a popup notification (if the other user is logged-in) or a ‘regular’ message (which he can view when he loggs-in). | For example: a buyer may contact a seller via a message, or a seller may contact a buyer by sending him a popup notification (if he is logged-in currently). |
| **Popup Notification** | A notification that is being received when performing a check-out, or being promoted, etc. Will be displayed IFF the receiving user is being currenty logged-in. |  |
| **Message** | A general notification - if the receiving user is not currently logged-in, he will receive the message when he logges-in. |  |
| **Owner** | An owner of a specific store.  Has access to performing some actions, in regard to the store’s management, such as: Appointing new sellers, store managers, stock management, etc. | A store may have many owners. |
| **Product** | A unique item (per store), which can be bought by a buyer - if it has a sufficient amount of quantity.  Each product belongs to one and only store or a seller (until it is bought). | Products are being sold in a store. |
| **Purchase Policy** | A ‘contract’ between the store and the buyers of the store – such as: payment methods, products return policy, etc. | Each store must have a purchase policy set for it when being opened. |
| **Purchase Strategy** | May be a restriction of the buying policy, such as ‘forcing’ the buyer to buy at least 3 products from the store, or buying at most 5 products from the store. May also be in the form of a discount for some products of the store. |  |
| **Payment System** | An external system that charges the money from the buyer and informs the market that it has ben successfully done (or not). |  |
| **Store** | A virtual store component that must have a founder, purchase policy, purchase strategy and products (available stock). The store may also have managers and owners. The store sells its products to the buyers of the system. Each store may have a lot of products. | A store cannot be opened without having a founder, and neither be active without one. |
| **Stock** | The items in a specific store, that are available to be purchased. | Each store has its own stock. |
| **User** | A person who is interacting with the system; he can be either a member or a guest. | Any User is a potential buyer – and is considered as one. |

**Use cases**

1) Use case: **Assigning a member as an owner of a store**.

-Actors: **current** store owners

-Precondition: the member must be registered, and not a manager of this store.

-Input Parameters: user identifier to assign as manager, store identifier.

-Actions:

1. A store owner types the member and store identifiers to the system.
2. The store owner selects managing permission for the new manager.
3. A message is being sent to all of the **current** store owners, in order to approve or reject the employeement of this member.
4. If all of the current store owners have approved his employeement, he will be promoted to a store manager. Otherwise (hence, at least one member has rejected his promotion) he won’t be promoted. Either way, the member and the store owners will be notified via a message about the outcome.

2) Use case: **A member signs in.**

-Actors: member of the system.

-Precondition: the member is registered and not already signed-in in the system.

-Input Parameters: username, password

-Actions:

1. The member types In username and password.
2. The system validates the data:
   1. if data is correct the member will be signed in
   2. if data is incorrect, the member will receive an error message

3) Use case: **A guest adds a product from a store to the cart**.

-Actors: guest

-Precondition: The product must be in the store stock.

-Input Parameters: product id, quantity.

-Actions:

1. The guest chooses a product from a given products list.
2. The guest selects the "add to cart" option.
3. The guest selects purchase strategy , s.a. quantity, etc.

4.1 if the quantity is bigger than the stock then the system alerts.

1. The mediator accesses the guest’s cart via user component and adds the product to the corresponding store basket.

5.1. if it's the first product from this store , then initiate the basket.

1. The mediator will update number of products (the store’s quantity).
2. The cart updates cart's total cost (including sales and discounts).

4) Use case: **A guest checks out**.

-Actors: guest

-Precondition: The cart contains at least one product.

-Input Parameters: None

-Actions:

1. The guest chooses the option check-out.
2. The system asks the guest to type in the address and contact information.
3. The guest types the address and contact information.
4. The system shows the final price (including the delivery and sales) and the purchase options.
5. The guest selects the desired option and fills in the required details.
6. The system will send to the external delivery system the information about the delivery.

6.1 if the external delivery system throws an error, the system will inform the guest about the error and will disapprove the checkout.

1. The system validates that the guest can afford the cart via external payment system.

8.1 if the external payment system throws an error, the system will inform the guest about the error and will disapprove the checkout.

1. the system confirms the deal – if everything went smoothely.
2. the system will access communication component and will send a receipt using message notification.

5) Use case: **A member checks out**.

-Actors: member

-Precondition: The cart contains at least one product.

-Input Parameters: None

-Actions:

1. The member chooses the option check-out.
2. the system shows member‘s address and contact information.
3. The member can edit the address and contact information for this purchase.
4. The system shows the final price (including the delivery and sales) and the purchase options.
5. The member selects the desired option and fills in the required details.
6. The system will send to the external delivery system the information about the delivery.

6.1 if the external delivery system throws an error, the system will inform the member about the error and will disapprove the checkout.

1. The system validates that the member can afford the cart via external payment system.

7.1 if the external payment system throws an error, the system will inform the member about the error and will disapprove the checkout.

1. the system confirms the deal – if everything went smoothely.
2. the system will access communication component and will send a receipt using message notification.

6) Use case: **member-A assigning member-B as a manager of the store**.

-Actors: member-A.

-Precondition: member-A is an owner of the store, member-B is not an owner of the store, nor a manager of the store, and also a registered user in the system.

-Input Parameters: member-B identifier, Store identifier.

-Actions:

1. member-A types member-B and store identifiers to the system.
2. The userRepo promotes the memberto be a manager of the store with the appropriate permissions.

7) Use case: **owner changes manager managing permissions.**

-Actors: owner.

-Precondition: the owner is logged in, and he is the assigning owner of this manager.

-Input Parameters: manager identifier, permissions.

-Actions:

1. The owner selects "employees" option.
2. The owner selects desired manager, by typing his user name.
3. The owner selects "change permissions" option.
4. The owner selects desired permissions.
5. The userRepo updates the manager's permissions.

8) Use case: **store employee assigning a new sale on a product.**

-Actors: store employee.

-Precondition: The user is really a store employee (of this store) and the product belongs to this store.

-Input Parameters: product identifier , sale percent.

-Actions:

1. The store employee selects "sales" option.
2. The system presents the sales list.
3. The seller selects "add new sale".
4. The system applies the sale according to the sale percent.
5. If there was an error – it will be presented to the store employee and the operation will abort. Otherwise a success message will be displayed.