<B-Bay>



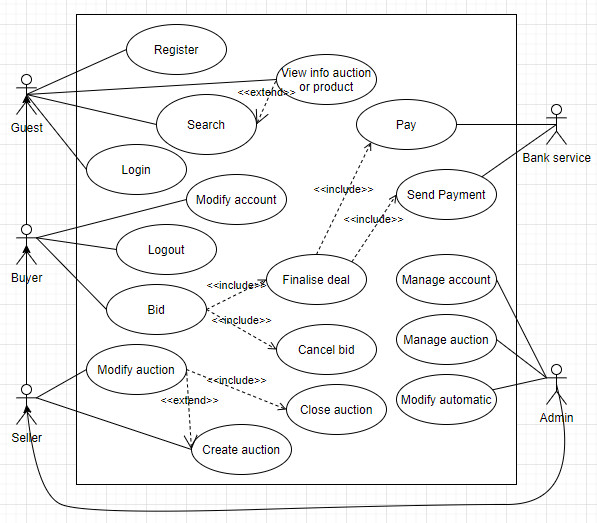
**Table of Contents**

1.Use case model

2.Use-case Specifications  
 2.1 Use-case: post auction products  
 2.2 Use-case: Search products  
 2.3 Use-case: Bid  
3.List architecture desgin and reason

4.Content of architecture desgin   
 4.1 Class diagram  
 4.2 Squence diagram  
 4.2.1 : post auction products  
 4.2.2 : Search products  
 4.2.3 : Bid

**1.Use case model**



**2. Use-case Specifications**

**2.1 Use-case: post auction products**

|  |  |
| --- | --- |
| Use case Name  (Tên use-case) | Post auction products |
| Brief description  (Mô tả) | This use case describes how the users post the products for auction |
| Actors  (Actors) | Sellers |
| Basic Flow  (Luồng cơ bản) | 1. At the homepage, the user click on the button “create auction” 2. System display the text field for user input data about product 3. User click “Create” and system will check validate 4. After check validate.System add new products for the auction and show the mess”Successfully” |
| Alternative Flows  (Các luồng thay thế) | **Alternative flow 1: User cannot create auction**   1. From #2 of the basic flow, user check validate again 2. Continue step #2 in the basic flow |
| Pre-conditions  (Điều kiện đầu vào) | User goes to homepage |
| Post-conditions  (Điều kiện đầu ra) | The user successfully adds new products for the auction |

**2.2 Use-case: Search product**

|  |  |
| --- | --- |
| Use case Name  (Tên use-case) | Search product |
| Brief description  (Mô tả) | This use case describes how the Guest find product they want |
| Actors  (Actors) | Guest |
| Basic Flow  (Luồng cơ bản) | 1. At the homepage, the user enters keywords on the ‘Search’ field 2. Users clicks on ‘Search’ button to start searching a product 3. System displays the products found |
| Alternative Flows  (Các luồng thay thế) | **Alternative flow 1: User cannot find products searched**   1. From #1 of the basic flow, user enters another term 2. Continue step #2 in the basic flow |
| Pre-conditions  (Điều kiện đầu vào) | User goes to homepage |
| Post-conditions  (Điều kiện đầu ra) | Search successfully and show product |

**2.3 Use-case: Bid**

|  |  |
| --- | --- |
| Use case Name  (Tên use-case) | Bid |
| Brief description  (Mô tả) | This use case describes how the Buyer auction the product |
| Actors  (Actors) | Buyer |
| Basic Flow  (Luồng cơ bản) | 1. At the homepage, the user enters keywords on the ‘Search’ field 2. Users clicks on ‘Search’ button to start searching a product 3. System displays the products found 4. User click “View profile” button on the expected product 5. User click “Bid” button for auction 6. System record the auction of user 7. System displays the quantity of all products in the list auction of user |
| Alternative Flows  (Các luồng thay thế) | **Alternative flow 1: User cannot find products**   1. From #1 of the basic flow, user enters another term 2. Continue step #2 in the basic flow   **Alternative flow 2: Another user bid higher**   1. From #5 of the basic flow, user do it again 2. Continue step #6 in the basic flow |
| Pre-conditions  (Điều kiện đầu vào) | User goes to homepage |
| Post-conditions  (Điều kiện đầu ra) | Bid successfully |

**3. Evaluation architecture style**

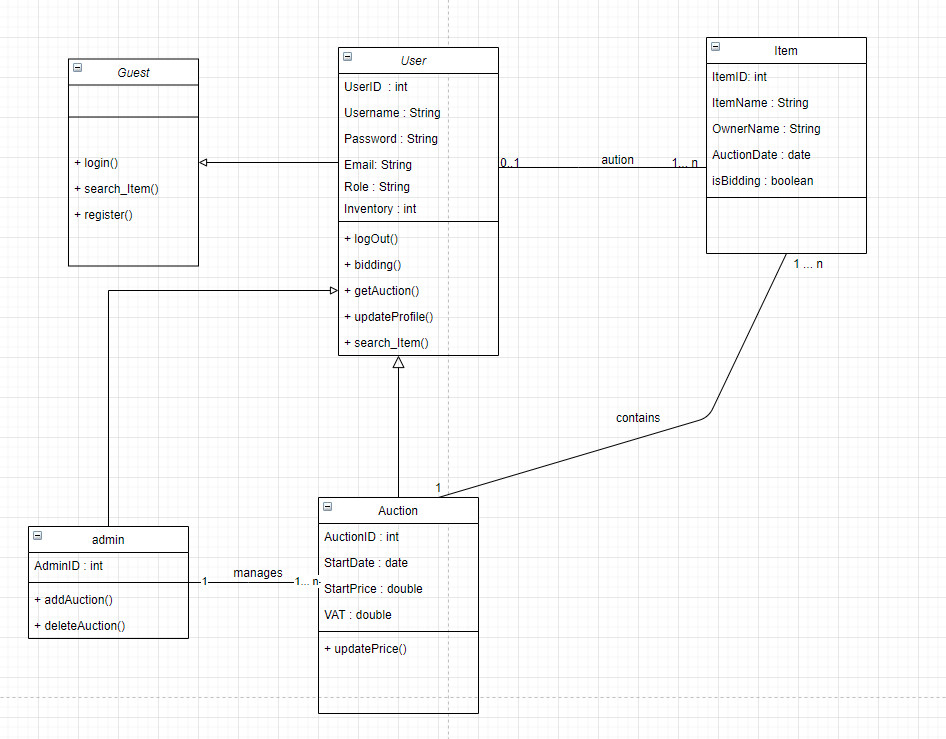
We choose MVC architecture for post auction products and search products because:

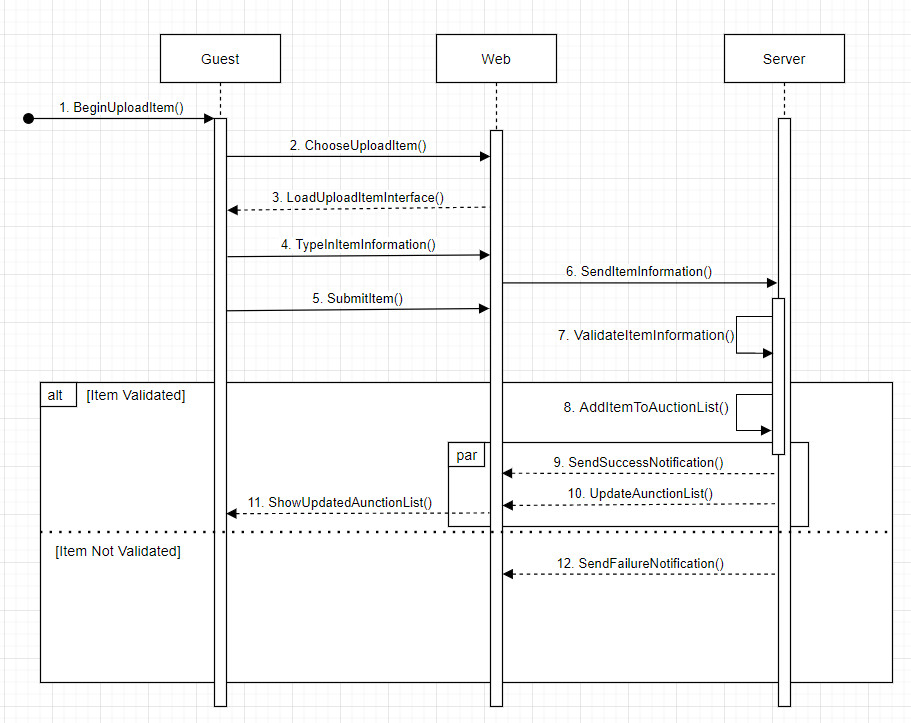
* Multiple views synchronized with same data model.
* Easy to change or plug in new interface views, allowing updating of interface views with new technologies without overhauling the rest of the system.
* Very effective for developments if graphics, programming, and database development professionals are working in a team in a designed project.

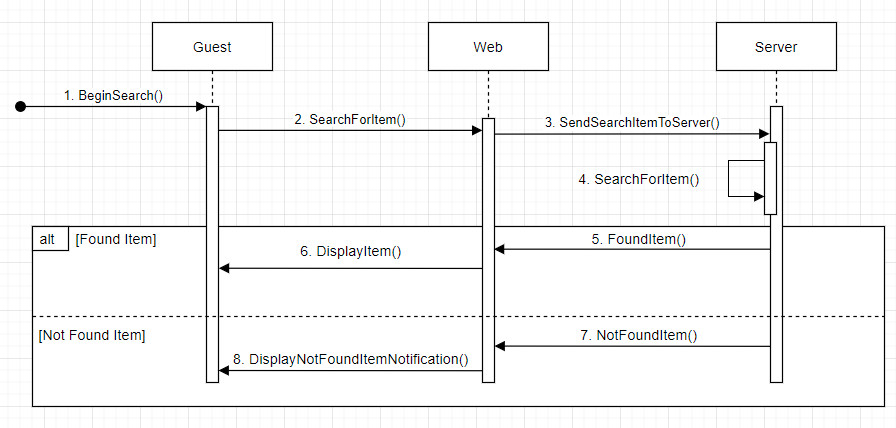
We choose Asynchronous architecture for bid because:

* Anonymity: provides high degree of anonymity between message producer and consumer. The message consumer does not know who produced the message (user independence), where the producer lives on the network (location independence), or when the message was produced (time independence).
* Concurrency: supports concurrency both among consumers and between producer and consumers.
* Scalability and reliability of message delivery: Reliability mechanisms include: control level setting of message acknowledgement; message persistence setting without loss; message priority level setting; and message expiration setting.
* Supports loose coupling between message producers and consumers, and between legacy systems and modern systems for int gration development.

**4.Content of architecture desgin**

**4.1 Class diagram**

**4.2 Squence diagram** **4.2.1** **post auction products (MVC Architecture)** 

* + 1. **Search (MVC Architecture)**

**4.2.3 Bid (Asynchronous Architecture)**

