**Use Case Model Document**

SEGY eShopper

**Team members:**

|  |  |  |
| --- | --- | --- |
| 110995 | Eden Teclemariam | Zere |
| 611145 | Gebrehiywet Fissaha | Tsehaye |
| 611098 | Simon Abraham | Weldemichael |
| 110930 | Yohannes Baye | Mulualem |

**Date: Thursday, May 21, 2020**

Table of Contents

[Introduction 4](#_Toc40901997)

[1. use case diagram 4](#_Toc40901998)

[2. use case description 5](#_Toc40901999)

List of Tables

[Table 1 Admin Registration 4](#_Toc40942203)

[Table 2 Seller Registration 5](#_Toc40942204)

[Table 3 Buyer Registration 5](#_Toc40942205)

[Table 4 Product Category 6](#_Toc40942206)

[Table 5 Shopping Cart 7](#_Toc40942207)

[Table 6 Payment 8](#_Toc40942208)

[Table 7 Approve Seller 8](#_Toc40942209)

[Table 8 Checkout 9](#_Toc40942210)

[Table 9 Manage Order 10](#_Toc40942211)

List of Figures

[Figure 1 Use Case Diagram 4](#_Toc40902072)

# Introduction

SEGY eShopper application will have three actors. Admin, Seller and Buyer. This Use Case and Use Case Description document has developed for SEGY eShopper firm’s new system for implementing an online ecommerce system that automates branch store offices.

# use case diagram

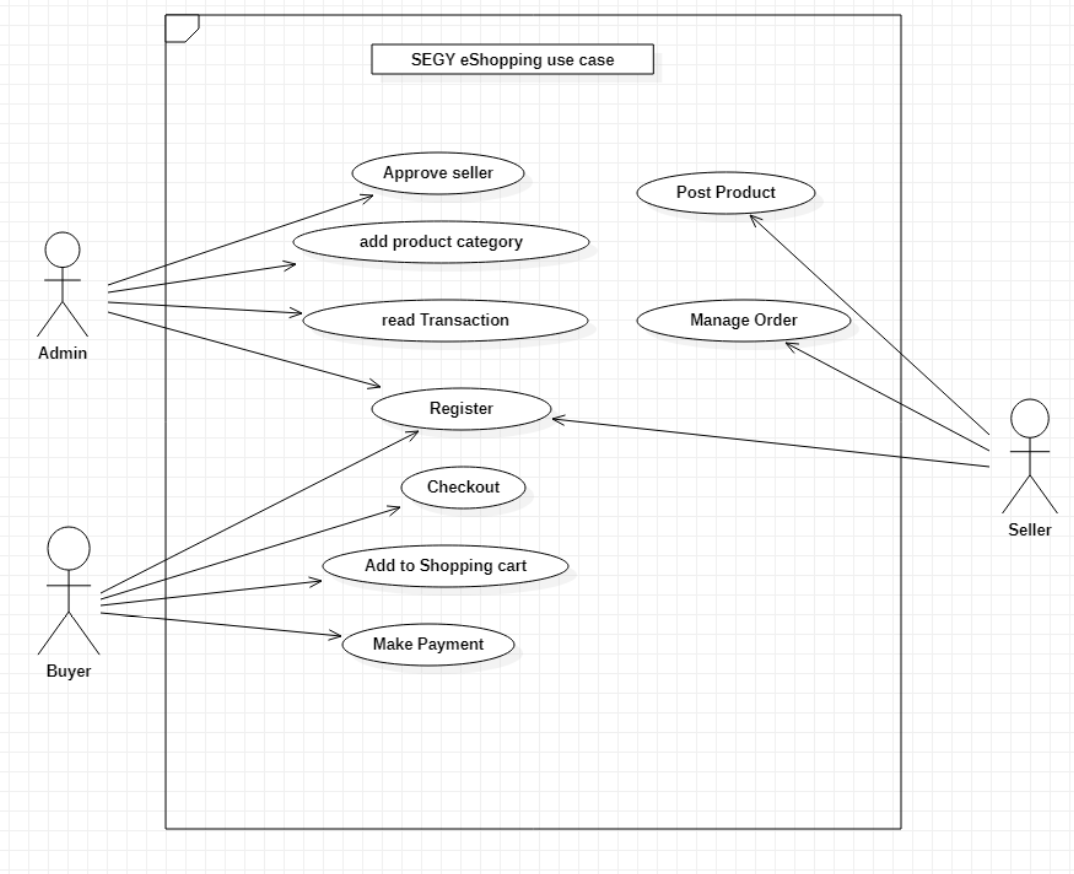


Figure 1 Use Case Diagram

# use case description

Table 1 Admin Registration

|  |  |
| --- | --- |
| **Use Case Number:** | UC01 |
| **Use Case Name:** | **Admin Registration** |
| **Description:** | This use case allows an actor to create his/her account to use the system as an admin. |
| **Actor:** | Admin |
| **Pre-condition:** | 1. The user must open valid admin registration form |
| **Flow of Events** | |
| 1. The user opens admin registration form 2. System displays admin registration form 3. The user fills out the form and requests to save the details 4. The system verifies [Alternative Flow A] [Alternative Flow B] 5. The system displays success message 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If the actor email already exists   * 1. The system returns the message indicating a duplicate entry exists   2. Use case resumes at step 3   Alternative Flow B: If the actor detail is not valid  4.1. The system returns the message indicating a validation error  4.2. Use case resumes at step 3 | |
| **Post-condition** | 1. The actors account detail must be persisted on database |

Table 2 Seller Registration

|  |  |
| --- | --- |
| **Use Case Number:** | UC02 |
| **Use Case Name:** | **Seller Registration** |
| **Description:** | This use case allows a user to create his/her account to use the system as a seller. |
| **Actor:** | Seller |
| **Pre-condition:** | 1. The user must open valid seller registration form |
| **Flow of Events** | |
| 1. The user opens seller registration form 2. System displays seller registration form 3. The user fills out the form and requests to save the details 4. The system verifies [Alternative Flow A] [Alternative Flow B] 5. The system displays success message 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If the actor email already exists   * 1. The system returns the message indicating a duplicate entry exists   2. Use case resumes at step 3   Alternative Flow B: If the actor detail is not validated  4.1. The system returns the message indicating a validation error  4.2. Use case resumes at step 3 | |
| **Post-condition** | 1. The sellers account detail must be persisted on database 2. The registered seller must be approved to be activated. |

Table 3 Buyer Registration

|  |  |
| --- | --- |
| **Use Case Number:** | UC03 |
| **Use Case Name:** | **Buyer Registration** |
| **Description:** | This use case allows a user to create his/her account to use the system as a buyer. |
| **Actor:** | Buyer |
| **Pre-condition:** | 1. The user must open valid buyer registration form |
| **Flow of Events** | |
| 1. The user opens buyer registration form 2. System displays buyer registration form 3. The user fills out the form and requests to save the details 4. The system verifies [Alternative Flow A] [Alternative Flow B] 5. The system displays success message 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If the actor email already exists   * 1. The system returns the message indicating a duplicate entry exists   2. Use case resumes at step 3   Alternative Flow B: If the actor detail is not validated  4.1. The system returns the message indicating a validation error  4.2. Use case resumes at step 3 | |
| **Post-condition** | 1. The actor account detail must be persisted on database |

Table 4 Product Category

|  |  |
| --- | --- |
| **Use Case Number:** | UC04 |
| **Use Case Name:** | **Product Category** |
| **Description:** | This use case allows an admin to create product category. |
| **Actor:** | Admin |
| **Pre-condition:** | 1. Admin must login to the system 2. The user must open product category registration form |
| **Flow of Events** | |
| 1. The admin opens product category registration form 2. System displays product category registration form 3. The admin fills out the form and requests to save the details 4. The system verifies [Alternative Flow A] [Alternative Flow B] 5. The system displays success message 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If the product already   * 1. The system returns the message indicating a duplicate entry exists   2. Use case resumes at step 3   Alternative Flow B: If the product category detail is not validated  4.1. The system returns the message indicating a validation error  4.2. Use case resumes at step 3 | |
| **Post-condition** | 1. The product category detail must be persisted on database |

Table 5 Shopping Cart

|  |  |
| --- | --- |
| **Use Case Number:** | UC05 |
| **Use Case Name:** | **Shopping Cart** |
| **Description:** | This use case allows a buyer to create shopping cart |
| **Actor:** | Buyer |
| **Pre-condition:** | 1. Buyer must login to the system |
| **Flow of Events** | |
| 1. The buyer choose a product 2. System displays product detail 3. The buyer selects a product to his/her cart and selection detail saved to database 4. The system verifies selection detail [Alternative Flow A] 5. The system displays success message 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If the product is out of stock   * 1. The system returns the message indicating an out of stock message   2. Use case resumes at step 3 | |
| **Post-condition** | 1. The product category detail must be persisted on database |

Table 6 Payment

|  |  |
| --- | --- |
| **Use Case Number:** | UC06 |
| **Use Case Name:** | **Payment** |
| **Description:** | This use case allows a buyer to pay total price of items on shipping cart |
| **Actor:** | Buyer |
| **Pre-condition:** | 1. Buyer must login to the system 2. The buyer must select product items 3. The buyer must checkout selected items on shipping cart |
| **Flow of Events** | |
| 1. The buyer clicks checkout shipping cart items 2. System displays payment details 3. The buyer fills add payment details such as billing and shipping address and the detail is saved to database 4. The system verifies payment detail [Alternative Flow A] 5. The system displays payment success message 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If the payment has validation problem   * 1. The system returns the message indicating payment detail validation problem   2. Use case resumes at step 3 | |
| **Post-condition** | 1. Payment detail must be persisted on database 2. Order flow ends 3. Shipping cart items cleared |

Table 7 Approve Seller

|  |  |
| --- | --- |
| **Use Case Number:** | UC07 |
| **Use Case Name:** | **Approve Seller** |
| **Description:** | This use case allows an admin to approve or disapprove seller account |
| **Actor:** | Admin |
| **Pre-condition:** | 1. Admin must login to the system |
| **Flow of Events** | |
| 1. The admin clicks registered sellers list 2. System displays list of registered sellers 3. The admin change status of the seller to approved or disabled 4. The system verifies sellers status [Alternative Flow A] 5. The system displays success message 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If the status update has problem   * 1. The system returns the message indicating the problem   2. Use case resumes at step 3 | |
| **Post-condition** | 1. Seller status must be persisted on database |

Table 8 Checkout

|  |  |
| --- | --- |
| **Use Case Number:** | UC08 |
| **Use Case Name:** | **Checkout** |
| **Description:** | This use case allows a buyer to check out the items he put in the shopping cart and proceed to payment |
| **Actor:** | Buyer |
| **Pre-condition:** | 1. Buyer must login to the system 2. The buyer must select product items and put in to shopping cart |
| **Flow of Events** | |
| 1. The buyer clicks shipping cart 2. System displays items selected and put in shopping cart 3. The buyer confirms to checkout for payment and the detail is saved to database 4. The system verifies [Alternative Flow A] 5. The system displays payment page to proceed with payment 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If checkout to payment has problem   * 1. The system returns message indicating checkout problem   2. Use case resumes at step 3 | |
| **Post-condition** | 1. Checkout detail must be persisted on database |

Table 9 Manage Order

|  |  |
| --- | --- |
| **Use Case Number:** | UC09 |
| **Use Case Name:** | **Manage Order** |
| **Description:** | This use case allows a seller to cancel and update ordered items of his products |
| **Actor:** | Seller |
| **Pre-condition:** | 1. Seller must login to the system 2. Any buyer must order product item of the seller 3. Ordered product must be purchased by a buyer |
| **Flow of Events** | |
| 1. The seller clicks manage order 2. System displays order management page 3. The buyer updates order status of his own product and the detail is saved to database 4. The system verifies payment detail [Alternative Flow A] 5. The system displays success message 6. End use case | |
| **Alternative Flow of Events** | |
| Alternative Flow A: If the manage order has problem   * 1. The system returns the message indicating order update problem   2. Use case resumes at step 3 | |
| **Post-condition** | * 1. Updated order status must be persisted on the database   2. The system must send notification to the buyer |