Team Amazonian Prime

BRD + External Requirements

April. 4, 2023
Version 3.0

Document Information

Revision History

Date	Version	Status	Prepared by	Comments
Feb 1, 2023	1.0	Graded	Amazonian Prime	
Feb 11, 2023	2.0	Complete	Amazonian Prime	Revisions for Midterm Check-in: ● Updated use cases and requirements ○ Use case scenarios that involved entering emails were removed, since Google OAuth will handle it ○ Reworded certain use cases that were worded like user stories ○ Updated main scenario for checkout flows to remove item listings ○ Added use case for removing postings as a user ● Added missing requirements: ○ Payment/address validation ○ Order cancellation ○ Registering a base user ○ Search order history by ID ● Revised budget plans (removed AWS Educate credits.)
April 3, 2023	3.0	Complete	Amazonian Prime	Revisions for final submission: ● Added (DESCOPED) tag to requirements and use cases that were descoped: ○ Adding users as admin ○ Deleting orders as admin ○ Stretch goals such as geography based tax and shipping costs ○ Others ● Linked test cases from test document to requirements and updated test statuses

Document Control

Role	Name	E-mail	Telephone
Professor	Jerry Jim		
TA	Marie Salomon		
TA	Shijun Shen		
Project Sponsor	Peter Smith		
	Mahmoud Al Khatib	mahmoudalkhatib.ubc@gmail.com	
	Michael He	michaelhe17@gmail.com	
	Joshua Luong	joshualuong@hotmail.com	
	Tristan Martinuson	tmartinuson@gmail.com	
	Elaine Shi	elaineshi328@gmail.com	
	William Suryawidjaja	suryawidjajaw@gmail.com	

Approval

Role	Name	Signature	Sign-off Date
Amazonian Prime Member	Mahmoud Al Khatib	Mahmoud Al Khatib	April 4, 2023
Amazonian Prime Member	Michael He	Michael He	April 4, 2023
Amazonian Prime Member	Joshua Luong	Joshua Luong	April 4, 2023
Amazonian Prime Member	Tristan Martinuson	Tristan Martinuson	April 4, 2023
Amazonian Prime Member	Elaine Shi	Elaine Shi	April 4, 2023
Amazonian Prime Member	William Suryawidjaja	William Suryawidjaja	April 4, 2023

Table of Contents

Business Requirement Summary

Business Requirement Summary: Employee E-Commerce Application

This document is meant to illustrate what is going to be built within the project (opposed to how it is built). The target audience for this document are non-IT individuals and the project sponsor.

Objective: To create an e-commerce application for internal use by employees, aimed at increasing interaction and collaboration through buying and selling items with one another.

Scope: The application will allow employees to purchase and sell items with each other, such as office equipment, personal items, and second-hand goods. The application will also allow the marketing department to promote and sell discounted company merchandise.

Target Audience: All employees of the company.

Key Features:

- User profile creation (separate systems for buyer and seller)
- Listing and searching of items
- Purchasing (and tax calculation) of items
- Mock payment system
- Mock shipment system
- Admin panel for managing listings and users

Out of Scope Features:

- Geography based tax and shipping calculation

Timeline: The project timeline is estimated to be 12 weeks, including design, development, testing, and deployment.

Budget: The free-tier of AWS services.

Deliverables: A fully functional e-commerce application that is easy to use, secure, and accessible by all employees. The project team will also provide support to ensure that any bugs identified after the deployment of the application are addressed.

In summary, the employee E-Commerce Marketplace project is a valuable opportunity to improve the work environment and foster a sense of community and collaboration among employees. The project will deliver a secure and user-friendly platform for employees to buy and sell items, leading to increased employee morale and improved work relationships.

Project Statement and Description

Project Statement

The purpose of this project is to create an internal e-commerce marketplace for employees to buy and sell items, fostering a sense of community and collaboration amongst employees. This project aligns with the company's goal of promoting a positive and collaborative work environment.

The project aims to deliver a fully functional e-commerce application that is easy to use, secure, and accessible by all employees. The application will include a user profile creation system, a listing and purchasing system, a secure payment system, a shipment status system, and an admin panel for managing listings and users.

The successful completion of this project will result in a platform that facilitates employee interaction and collaboration, leading to improved employee morale and a more positive work environment.

Project Description

Working from home has become a norm for many people since the pandemic, but it has also brought negative impacts such as decreased productivity, isolation, and blurred boundaries between work and personal life. The absence of in-person interactions and face-to-face communication can lead to feelings of loneliness and disconnection from one's colleagues and workplace culture.

As such, this project aims to provide a secure and user-friendly platform for employees to buy and sell items within the company. The marketplace will be accessible by all employees and will provide an opportunity for them to communicate and transact, leading to increased employee morale and improved work relationships. The ultimate goal of the project is to promote a positive and collaborative work environment for employees.

The e-commerce marketplace will feature a user profile, listing & purchasing, payment and shipping status systems, and an admin panel. The project timeline is estimated to be 12 weeks, including design, development, testing, and deployment.

The budget for this project will leverage the free-tier of AWS services which include Amazon RDS for the database. This will cover costs for development, testing, and deployment, as well as any additional expenses such as hardware and software requirements. If there are more costs required, we will reevaluate our budget with the sponsor.

The deliverables for this project will be a fully functional e-commerce application that is easy to use, secure, and accessible by all employees. The project team will also provide support to ensure that any bugs identified after the deployment of the application are addressed.

Project Impacts

Impact to Other Systems

The use of AWS Lambda and AWS Step Function makes our solution highly scalable to accommodate more users in the future. Furthermore, our product would be able to easily integrate with other AWS services, giving it the ability to work seamlessly with the corporate directory or other software solutions at Amazon. AWS Lambda and Step Function are extremely cost efficient, which makes the product easy to maintain without sacrificing performance or quality. This project also has the potential to integrate with the AWS payment and shipping systems.

Impact to Groups & Individuals

The project aims to create a vibrant and more connected workspace for Amazonians by providing a platform for employees to sell second-hand items to each other. We give employees who originally don't know each other the opportunity to bond over a common topic, and different AWS teams to connect and discuss our service. These could foster collaboration and improve workplace culture. Employers offering discounted company merchandise could also boost employee satisfaction and pride. The management and HR teams will have full control of the product so they could monitor the employees' usage of it and make necessary adjustments or implement new functionalities. The relationship between HR, the management team, and other employees would improve as a result.

Additionally, our project promotes environmentally responsible practices by encouraging the reuse of old or unwanted items, thereby reducing global waste.

Business Constraints/ Assumptions/ Dependencies

Business Constraints

Some business constraints that may hinder our project workflow include but are not limited to: having a limited cost for AWS services, which may ultimately limit the potential to include other useful services (AWS, or others); having only 6 team members, a limited time, and a limit on the knowledge of technologies proposed to be used (E.g. AWS) puts constraints on the actualization of potential features. Communication with the sponsor may put constraints on our project, when we take into account the time it takes for a response and the time to implement all tasks.

Business Assumptions

- There exists a need for increased interactions and collaborations between Amazonians
- Amazonians are willing to try using this service to buy and sell items
- The users have the time and item availability to participate in second-hand exchanges
- The availability of management team to address any bugs or updates required after deployment
- The system is compatible with the environment of the company
- Mock data and the payment system can be successfully substituted
- Tools provided by AWS will function as intended

Business Dependencies

Business dependencies that the project is dependent on includes but is not limited to: The chosen technology stack (AWS Step Function, AWS Lambda, AWS RDS, React, Redux, etc.); some external dependencies may include a dependency on the sponsor that may be required for clarifications.

Functional Requirements

List of Use Cases

1. Register as a Buyer
2. Register as a Seller
3. View Order history and Shipping Status
4. Add Items to Shopping Cart
5. Remove Items from Shopping Cart
6. Checkout Items in Shopping cart
7. Post Product To Sell
8. Add User as Admin
9. Delete User as Admin
10. Delete Postings as Admin
11. Delete Order as Admin

Refer to the detailed tables in the Appendix.

Requirement Traceability Matrix (RTM)

*The scale used for the RTM Priority is from 1-5 where:

- 1 is the highest priority and included in the MVP
- 2 is lower priority than 1 but still included in the MVP
- 3 is in scope but not included in the MVP
- 4 is out of scope, but one that would be nice to have
- 5 is out of scope and a stretch goal that is ambitious

Req No.	Reference	Requirement Description	Type	Priority* (1-5)	MVP (Y/N)	Related Test Case	Test Status	Comments
1.0	use-case 1	Register user	Functional	1	Y	1	Complete	
1.1	use-case 1	Complete buyer profile	Functional	1	Y	2	Complete	
1.2	use-case 1	Integrate registration with Google OAuth	Functional	1	Y	1	Complete	
2.0	use-case 2	Complete seller profile	Functional	1	Y	3	Complete	
3.0	use-case 3	View order history	Functional	2	Y	7	Complete	
3.1	use-case 3	View shipping status	Functional	2	Y	7	Complete	
3.2	use-case 3	Search for orders by ID	Functional	4	N	7	Complete	
3.3	use-case 3 **(DESCOPED)**	Cancel order	Functional	4	N	N/A	N/A	
4.0	use-case 4	Add items to shopping cart	Functional	1	Y	4	Complete	

4.1	use-case 4	Search for items by name	Functional	2	Y	16	Complete	
4.2	use-case 4	Search for items by category	Functional	2	Y	16	Complete	
4.3	use-case 4	Shopping carts have persistence	Functional	2	Y	4	Complete	
5.0	use-case 5	Remove items from shopping cart	Functional	1	Y	5	Complete	
5.1	use-case 5	Adjust quantity of one item in shopping cart	Functional	4	N	4	Complete	
6.0	use-case 6	Calculate total costs at checkout	Functional	1	Y	6	Complete	Use one standard mock tax/shipping cost
6.1	use-case 6 **(DESCOPED)**	Calculate tax based on user's location	Functional	5	N	N/A	N/A	Stretch goal
6.2	use-case 6 **(DESCOPED)**	Calculate shipping costs based on user's location	Functional	5	N	N/A	N/A	Stretch goal
6.3	use-case 6	Add items to order history when purchased	Functional	1	Y	7	Complete	
6.4	use-case 6	Simulate shipping notifications to user's email on item purchase	Functional	3	N	N/A	Complete	
6.5	use-case 6	Simulate processing payment	Functional	3	N	6	Complete	
6.6	use-case 6	Modify shipment information at checkout	Functional	4	N	N/A	Complete	Common to e-commerce websites but not specified in scope
6.7	use-case 6	Modify payment information at checkout	Functional	4	N	N/A	Complete	Common to e-commerce websites but not

								specified in scope
6.8	use-case 6	Validate payment info	Functional	3	N	2	Complete	Payment inputs should be of the correct form
6.9	use-case 6	Validate address info	Functional	3	N	2	Complete	Address field inputs should be of the correct form
7.0	use-case 7	Implement posting products from seller supplied details	Functional	1	Y	8	Complete	
7.1	use-case 7	Mark product postings as used or new goods	Functional	3	N	8	Complete	
7.2	use-case 7	Handle post requests that contain product details and images	Functional	1	Y	8	Complete	
8.0	use-case 8	Remove posted product from seller's active listings	Functional	2	Y	N/A	Complete	
9.0	use-case 9 **(DESCOPED)**	Users with admin privileges are able to add users	Functional	2	Y	N/A	N/A	
9.1	use-case 9	Users with admin privileges are able to elevate the privilege levels of another user	Functional	1	Y	10	Complete	Also able to remove privileges from a user
10.0	use-case 10	Users with admin privileges are able to remove users	Functional	1	Y	9	Complete	
10.1	use-case 10	Remove listings associated with removed user	Functional	2	Y	9	Complete	

10.2	use-case 10	Users with admin privileges are able to search for users by id	Functional	3	N	N/A	Complete	
11.0	use-case 11	Users with admin privileges are able to remove postings	Functional	1	Y	N/A	Complete	
11.1	use-case 11	Users with admin privileges are able to search for postings by name	Functional	3	N	12	Complete	
12.0	use-case 12 **(DESCOPED)**	Users with admin privileges are able to remove orders	Functional	1	Y	N/A	N/A	
12.1	use-case 12	Users with admin privileges are able to search orders by id	Functional	3	N	16	Complete	
13.0	NA	Implement a database to store information	Non Functional	1	Y	N/A	N/A	
13.1	NA	Implement Google OAuth for authentication	Non Functional	1	Y	1	Complete	
13.2	NA **(DESCOPED)**	Use SCRAM for sensitive data	Non Functional	2	Y	N/A	N/A	
13.3	NA	Handle 10 concurrent users	Non Functional	3	N	15	Complete	
13.4	N/A	Response time for searches/filters should be under 4 seconds	Non Functional	3	N	16	Complete	
13.5	N/A	Develop installation guide and user manual	Non Functional	3	N	N/A	N/A	

Use Case Diagram

Refer to the diagrams in the appendix.

Non Functional Requirements

Backup Needs

Our team is using **Amazon RDS**. By default, Amazon RDS securely stores automated backups of our database instance in Amazon S3 for the duration specified by our team. In addition, we can take manual snapshots, which are backups of our instance that remain until we manually delete them.
We can also create a new instance from a database snapshot at any time. Although snapshots act as full backups, we are only charged for the additional storage space they use.

Security Needs

Everyone will be working under the same AWS account but working in different regions. Sensitive user data stored can potentially be encrypted using Salted Challenge Response Authentication Mechanism (SCRAM). Some other security needs may include secure admin access to allow for admin privileges within the app. Encryption of payments may not be a large security need if we are simulating payments (at the current scope).

Capacity Needs

Our solution is designed to accommodate a maximum of 500 users, but it must also be able to effectively handle 10 users simultaneously.

Cost Parameters + Constraints

Free-tier of AWS services,including Amazon RDS for the database. This will cover costs for development, testing, and deployment, as well as any additional expenses such as hardware and software requirements. If there are more costs required we will reevaluate our budget with the sponsor.

Additional Needs

Publication of a document or video that includes instructions on how to set up the system

Appendix

Glossary of Terms / Abbreviations

Terms / Abbreviations	Description
SaaS	Software as a Service, a method of software delivery and licensing in which software is accessed online via a subscription
React/Redux	React is a front-end JavaScript library for building user interfaces managed and based on modular components. Redux is a JavaScript library for managing and centralizing application state used in conjunction with React..
Lambda	Serverless compute service that allows code to be hosted and run without provisioning or managing servers. It automatically scales and monitors with the application.
Step Function	Service that enables coordination and initiation of distributed workflows and applications through the use of visual workflow diagrams
Aurora RDS	Relational database service provided by Amazon Web Services (AWS) designed to provide high availability, performance, and scalability for cloud-based applications
Google OAuth	Provides user authentication with a service without the need of maintaining user private information such as usernames, passwords and other sensitive details.
SCRAM	Salted Challenge Response Authentication Mechanism (SCRAM) is a family of modern, password-based challenge–response authentication mechanisms providing authentication of a user to a server
Amazon S3	Amazon S3 or Amazon Simple Storage Service is a service offered by Amazon Web Services that provides object storage through a web service interface

Detailed Use Cases

1. Register as a Buyer	
Description	A user needs to have a valid account on our platform to be able to participate in the transactions. Therefore, a new user will need to create an account documenting their personal information before being allowed to buy from the platform.
Primary Actor	Amazon employees
Preconditions	- The user must be on our website
Postconditions	- A new record will be created in our database containing the new buyer's account information
Main Scenario	1. The user accesses the Amazonian Prime website 2. The user clicks on the "Sign Up" button 3. The user registers an account using Google OAuth 4. The website displays a registration form that signs them up as a buyer: a. username (alias of user's choice) b. first and last name c. shipping address d. payment details (credit card) 5. The user fills in their information 6. The user clicks the "Sign Up" button at the end of this form 7. The system saves the user's account information as a record in the database

Alternate Scenario	4a) The user provides a username that’s unavailable or with format issues - The webpage displays an appropriate error message

2. Register as a Seller	
Description	A user needs to provide additional information and register under the seller module before they can sell on the platform. They do not need to create a new account.
Primary Actor	Amazon employees
Preconditions	- The user must be on our website - The user must be logged in to a buyer account that’s not already a seller
Postconditions	- The user will be able to participate in the buyer and the seller modules - The system will update the user’s account status to be a seller
Main Scenario	1. The user accesses the Amazonian Prime website 2. The user clicks on the “Sell” button 3. The website displays a questionnaire form with the following fields: a. banking information (deposits) 4. The user fills in their information 5. The user clicks the “Continue to sell” button at the end of this form 6. The system updates the user’s account information in the database
Alternate Scenario	3a) The user provides invalid information - The webpage displays an appropriate error message

3. View Order History and Shipping Status	
Description	A user would like to view my order history, so I can verify my orders, check what I have ordered in the past, track my orders, and contact the seller.
Actors	Amazon employees
Preconditions	- The user must be registered as a buyer - The user must be logged in - Must have placed an order to view shipping status
Postconditions	Order history will remain unchanged after viewing it
Main Scenario	1. User selects view orders button from the dashboard 2. User sees a table of their orders 3. User selects an order and app displays its shipment status
Alternate Scenarios	2) There are no orders placed, should preview no orders are placed

4. Add Items to Shopping Cart

Description	A user would like to browse the market and add item(s) to their shopping cart
Primary Actor	Amazon employees
Preconditions	- The user is registered into the system as a buyer - The user is logged in - The marketplace has posting(s) of item(s)
Postconditions	- The item(s) are added to the user’s shopping cart - The item(s) remain in the user’s shopping cart across sessions
Main Scenario	1. User has an item they would like to add to their shopping cart 2. User searches for item either by name or category 3. User sees and clicks on the desired item’s page 4. User clicks the “Add to shopping cart” button 5. The system adds the item to the user’s shopping cart 6. The system remembers which items are in the user’s shopping cart and persists across sessions
Alternate Scenario	4) The item is already in the user’s shopping cart and there is only one for sale - App notifies the user and the user’s shopping cart does not change 4) The item is already in the user’s shopping cart and there are multiple for sale - App increases the quantity of the item in the user’s shopping cart 4) The item has been purchased by another user - App notifies the user that the item is no longer available and the user’s shopping cart does not change 4) App is unresponsive and/or timeouts - App notifies the user that something has gone wrong and prompts them to try again *Note: Adding items to the shopping cart doesn’t reduce the inventory. It is only reduced once the user purchases the shopping cart items.

5. Remove Items From Shopping Cart	
Description	A user would like to remove item(s) from their shopping cart
Primary Actor	Amazon employees
Preconditions	- The user is registered into the system as a buyer - The user is logged in - The user has item(s) in their shopping cart
Postconditions	- The item(s) are removed from the user’s shopping cart
Main Scenario	1. User has an item they would like to remove from their shopping cart 2. User clicks on their shopping cart 3. App displays item(s) in their shopping cart along with a “Remove from cart” button for each one 4. User identifies the item they would like to remove 5. User clicks on the “Remove from cart” button 6. The system removes the item from the user’s shopping cart

Alternate Scenario	4) User has multiple quantities of one item in their cart and would like to change the quantity - User accesses the quantity dropdown to select the desired amount 5) App is unresponsive and/or timeouts - App notifies the user that something has gone wrong and prompts them to try again

6. Checkout Items in Shopping Cart	
Description	A user would like to purchase the item(s) in their shopping cart
Primary Actor	Amazon employees
Preconditions	- The user is registered into the system as a buyer - The user is logged in - The user has item(s) in their shopping cart
Postconditions	- The item(s) are removed from the user’s shopping cart - App adds item(s) to the user’s order history - The posting(s) for the item(s) are removed
Main Scenario	1. User has item(s) in their shopping cart that they would like to purchase 2. User clicks on their shopping cart 3. App displays items in their shopping cart along with a “Checkout” button 4. User clicks on “Checkout” button 5. App displays checkout page with calculated shipping costs and taxes based on user’s registered information 6. User clicks “Place your order” button 7. Reduce the quantity of the item(s) available, if zero, take down listing(s) 8. App sends mock shipping notifications to user’s email within 2 minutes a. “Order on its way” email b. “Delivery arrived” email
Alternate Scenario	4, 6) An item in the shopping cart has been purchased by another user - App notifies the user that the item is no longer available and the item is removed from the user’s shopping cart. App will prompt the user to place their order again 5) User wishes to change their shipping address before purchase - User will click “Modify” button on the checkout page near shipping information, new shipping cost will be calculated 5) User wishes to change payment method - User will click “Modify” button on the checkout page near payment information 4, 6) App is unresponsive and/or timeouts - App notifies the user that something has gone wrong and prompts them to try again

7. Post Product to Sell	
Description	A seller would like to post a product that they would like to sell with its corresponding description, price, quantity and pictures of the product.
Actors	Amazon employees
Preconditions	- The user is registered into the system as a seller - The user is logged in
Postconditions	The product is updated in the system with its details stored in the system database
Main Scenario	**1.** Seller has a product they would like to sell **2.** Seller creates a new listing on App requiring the following information: **a.** Product name **b.** Product description **c.** Price **d.** Quantity **e.** Pictures **3.** Seller fills in required information for their product **4.** Seller submits product posting request **5.** App creates a listing containing the user's product details
Alternate Scenarios	3) Seller is missing some required information - App highlights which information is missing for the product on submission and prevents the creation of the new listing 3) Seller submits an image in the wrong format - App highlights the incorrect file and requests the user to submit an image in the supported file types 4) Seller submits a product post request but the App is unresponsive and/or timeouts - App does not create a new listing of the product which further requires the user to submit a new listing with the same details

8. Remove Posted Product	
Description	A seller would like to remove their listing
Actors	Amazon employees
Preconditions	- The user is registered into the system as a seller - The user is logged in - The user has previously added a listing, and it has not been sold yet
Postconditions	The product is delete in the system
Main Scenario	**6.** Seller has a product they would like to delete **7.** Seller navigates to “My Listings” page 8. Seller scrolls/ searches for the listing they want to delete 9. Seller clicks “Delete listing” 10. App asks for confirmation to delete an active listing **11.** Seller agrees to delete, and the listing is deleted from the App

Alternate Scenarios	3) User clicks out of the confirmation modal - The App will not delete the listing

9. Add User as Admin (DESCOPED)	
Description	The administrator would like to add a new user into the database
Actors	Administrator
Preconditions	- The user has admin privileges - The user is signed in
Postconditions	A new user is created with the specified fields
Main Scenario	1. Administrator selects “Add/ Remove users” 2. Administrator selects “Add a new user” 3. App displays a new user form with the following fields: a. username b. first and last name c. shipping address d. payment details e. provide admin privileges f. email address (to link with Google OAuth) 4. Administrator enters the corresponding fields 5. Administrator saves the user 6. App stores the newly created user into the database
Alternate Scenarios	3f) If a user with specified email already exists, then the user isn’t added and the app throws an error. 4) If any of the fields are left blank, then the app throws an error.

10. Delete User as Admin	
Description	The administrator would like to remove a user from the database
Actors	Administrator
Preconditions	- The user is registered as an administrator - The user is signed in
Postconditions	The specified user is removed from the database
Main Scenario	1. Administrator selects “Add/ Remove users” 2. Administrator selects a user from the list of users 3. App displays the user’s information 4. Administrator selects “Remove user” 5. App asks for confirmation by typing in “DELETE” 6. Administrator types the confirmation message 7. The app deletes any active postings associated with the user. 8. The app deletes the user from the database
Alternate Scenarios	5) If the administrator doesn’t input “DELETE” as the confirmation message, the app repeats the request for confirmation.

	7) If there are sold listings associated with the user, do not delete these postings, or associate them with a special “Deleted User” user ID so that the buyers of these previous postings can still see the orders in their order history

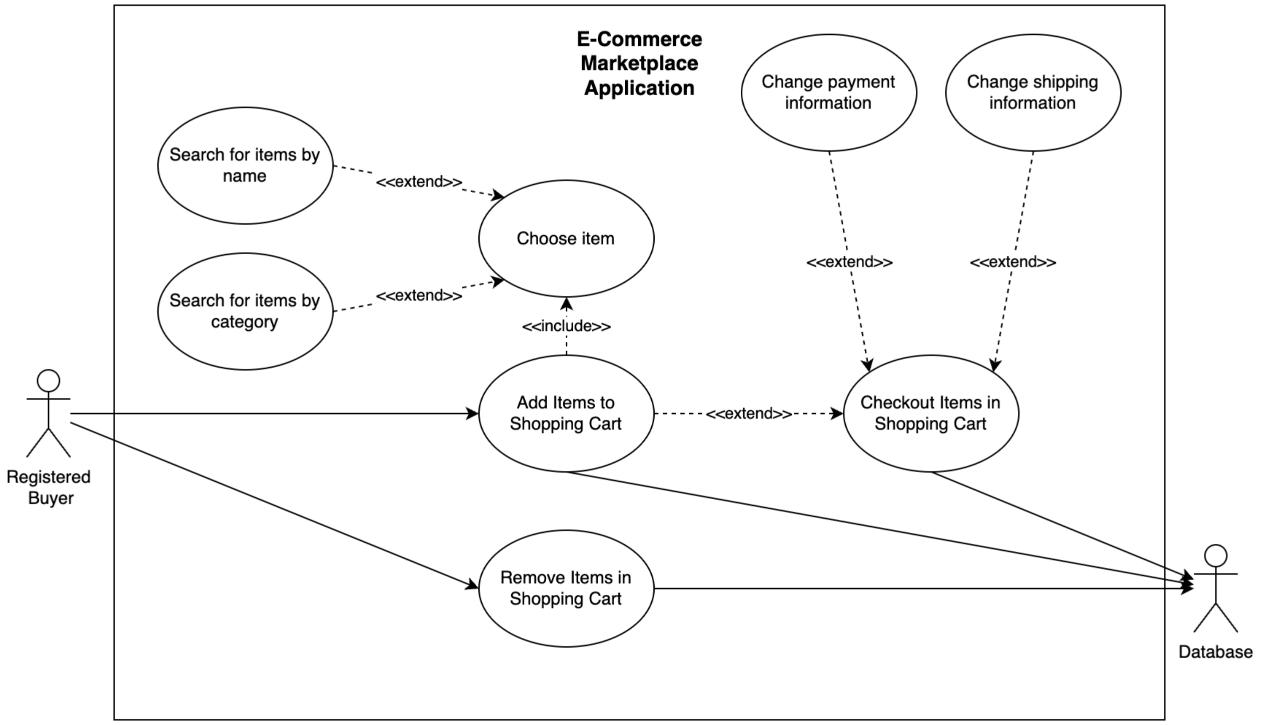
11. Delete Posting as Admin	
Description	The administrator would like to remove a posting from the marketplace
Actors	Administrator
Preconditions	- The user is registered as an administrator - The user is signed in
Postconditions	The specified posting is removed from the marketplace
Main Scenario	1. Administrator selects “Delete postings” 2. Administrator selects a posting from the list of postings 3. App displays the posting’s details 4. Administrator selects “Remove posting” 5. App asks for confirmation by typing in “DELETE” 6. Administrator types the confirmation message 7. The app deletes the posting from the marketplace
Alternate Scenarios	5) If the administrator doesn’t input “DELETE” as the confirmation message, the app repeats the request for confirmation.

12. Delete Order as Admin (DESCOPED)	
Description	The administrator would like to remove an **uncompleted** order from the database
Actors	Administrator
Preconditions	- The user is registered as an administrator - The user is signed in
Postconditions	The specified order is removed from the database
Main Scenario	1. Administrator selects “Delete orders” 2. Administrator searches for the order by order ID 3. App displays the order’s details 4. Administrator selects “Remove order” 5. App asks for confirmation by typing in “DELETE” 6. Administrator types the confirmation message 7. The app deletes the order from the database
Alternate Scenarios	5) If the user doesn’t input “DELETE” as the confirmation message, the app repeats the request for confirmation.

Use Case Diagrams

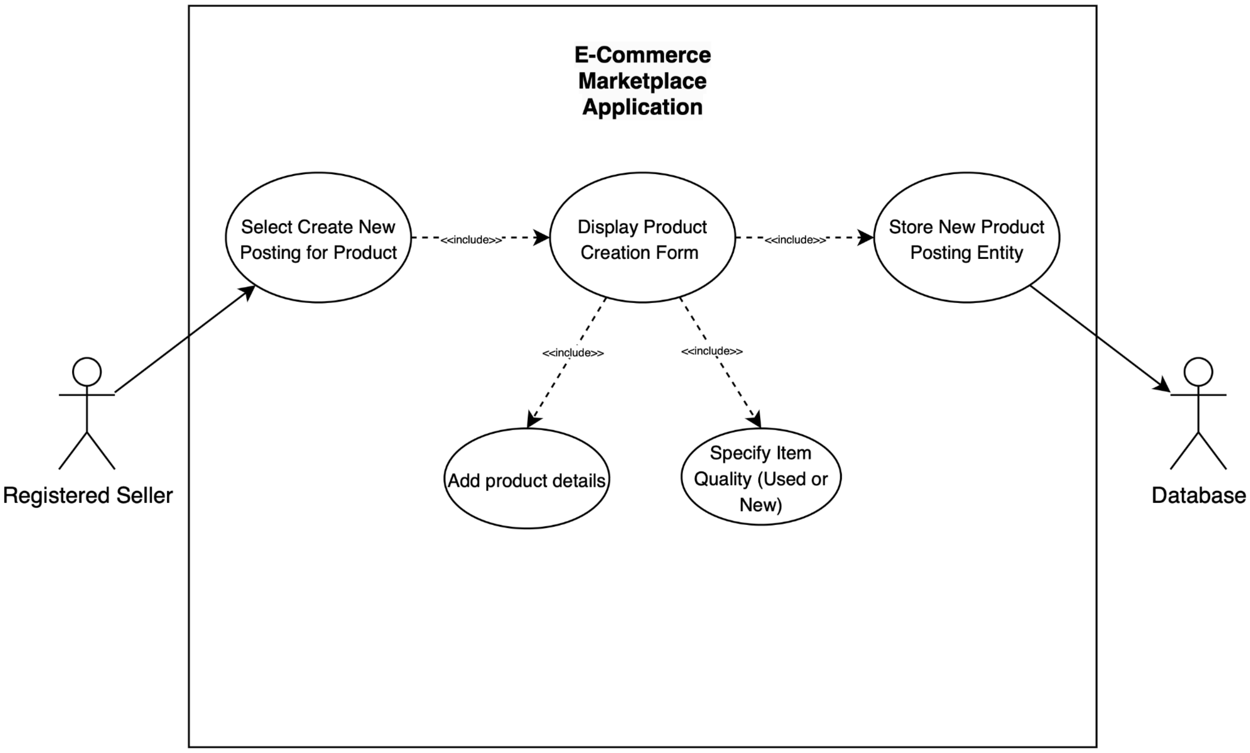
Legend: The box represents the system, the ovals represent an action/ use case and the arrows represent the relationship between them. The solid line represents association relationships and the dashed lines represent either include or extend relationships.

Marketplace Workflow


E-Commerce Marketplace Application
Change payment information
Change shipping information
Search for items by name
<<extend>>
Choose item
<<extend>>
<<extend>>
Search for items by category
<<extend>>
<<include>>
Add Items to Shopping Cart
<<extend>>
Checkout Items in Shopping Cart
Registered Buyer
Remove Items in Shopping Cart
Database

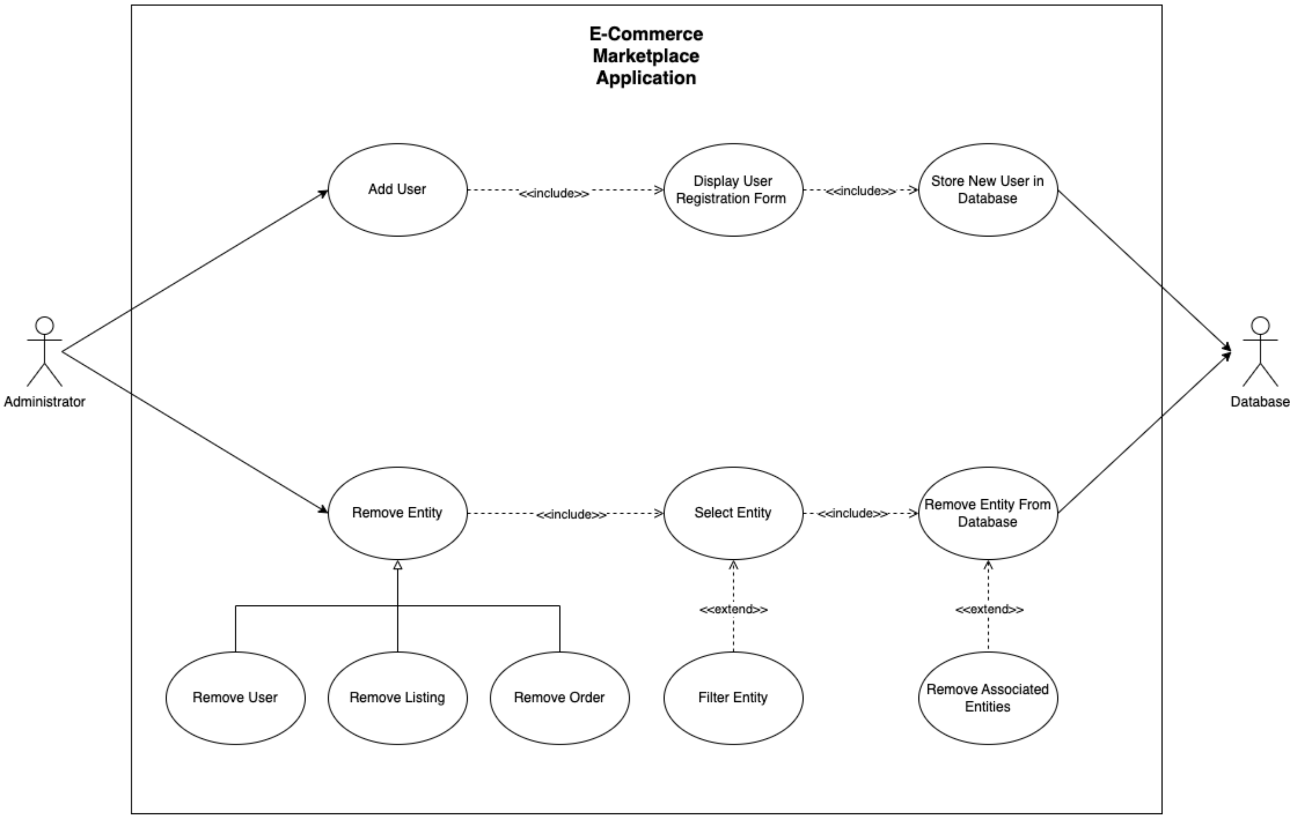
- A registered buyer can add or remove items from their shopping cart, and it will be updated in the database
- To add an item to their shopping cart, a user will need to choose their item first, and they can do this by browsing or searching for the item by name or category
- A registered buyer can then checkout their items in their shopping cart, which simulates shipping and payment processing and removes the postings from the database
    - They can optionally change their payment or shipping information at checkout

Registered Seller Marketplace Workflow


E-Commerce Marketplace Application
Select Create New Posting for Product
<<include>>
Display Product Creation Form
<<include>>
Store New Product Posting Entity
<<include>>
<<include>>
Add product details
Specify Item Quality (Used or New)
Registered Seller
Database

- A registered seller can create a new listing for a product they would like to sell with details such as the product name, description, quantity, picture and quality
- To post a product, the seller will fill out a product creation form and will specify the quality of the product defining either new or used.
- Upon completion of filling out the form, the posting is created as an entity and is stored in the database as a product posting entity.

Administrator Workflow



- An administrator can add a user with provided details such as name, shipping address, payment details, etc. and it will be updated in the database.
- An administrator can also remove a user/ listing/ order. The “remove entity” is a generalization of the three different cases.
- The administrator would specify the user/ listing/ order by browsing by name or order ID respectively.