

- **Description in sentences of a software engineering project that you will work on it with your team during the quarter.**

We are creating an application to help people improve their physical lives. We will sell athletic gear to inspire the public to get moving for at least a few minutes a day. Customers will be able to log into our application and pick any type of athletic gear or athletic equipment that is in stock. They can pick up items from a storage/Amazon locker or get it delivered to their home. Customer/Guest can edit their cart before FINAL checkout by removing items or adding items. This application will be an interface for the customer to avoid needing to go to a brick-and-mortar store. Customers or guests have the ability to cancel order(s) within 12 hours without penalty. Member will get an exception if logging on to our application with WRONG user/password combination

- **Functional and non-functional requirements**

- Functional Requirements:
- Non-member/Guest will be able to browse athletic gear and equipment without logging in
- Non-member/Guest will be able to add items to the generic cart without logging in as a member
- Non-member/Guest will be able to check out with items as a Guest
- Member will be able to browse athletic gear and equipment with and without logging in
- Member will be able to add items to their personal cart; Meaning logged into account
- Member can check previous orders in account
- Member will be able to check out in their personal account
- Member/Non-member will be able to filter by type of item, size, and gender specific

- **Nonfunctional Requirements**

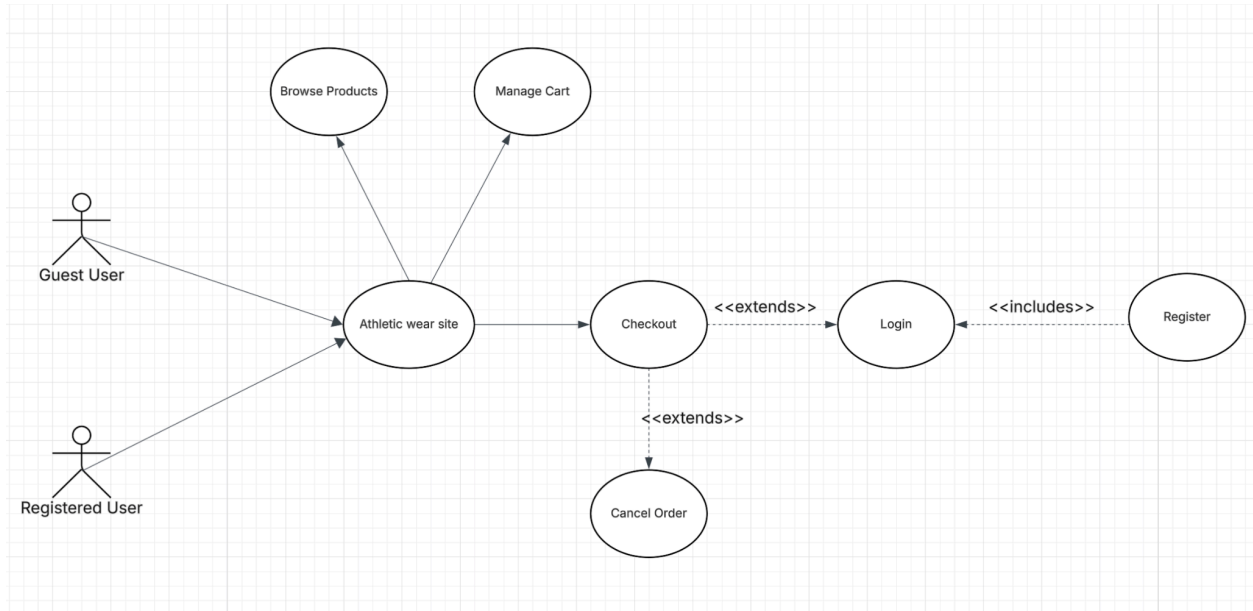
- Usability - Non-member and member will be able to choose language; Prices of items are bolded. Size of font is 12pt;
- Reliability - Non-member and member will not have to worry about our system being down due to our distributed system that our application sits on.
- Performance - When selecting items Products must load in <2 sec
- Supportability - Database backup system is in place to protect from data loss.

- **Glossary**

- 1. API (Application Programming Interface) - A set of rules and conventions that allow different software applications to communicate with each other and share data or functionality.
- 2. IDE (Integrated Development Environment) - A software application that provides tools and features to developers for writing, testing, and debugging code in a single integrated environment.
- 3. UI (User Interface) - The graphical or visual elements of a software application that users interact with, including buttons, menus, forms, and screens.
- 4. Application URL - the user friendly domain name that customers will access
- 5. App Store - using the application URL to display items that the store have in stock for athletic gear and equipment
- 6. Frontend - The client side of the application that users interact with on mobile phones or web browsers. UI development, display of items and data
- 7. Backend - server side of an application that handles data storage, processing, and retrieval
- 8. Database - A structured collection of data that users can analyze, manipulate, and access through an application or platform
- 9. Java: An object-oriented programming language. This language will be used for the back end
- 10. JavaScript: It's a versatile, high-level, and dynamic programming language primarily used to add interactivity to websites.
- 11. JSON- JavaScript Object Notation — a lightweight data-interchange format that can be analyzed by both development teams and machines. JSON is independent of all programming languages and able to function outside of Java.
- 12. REST: Stands for Representational State Transfer — a software architectural style for building web services. RESTful web services use HTTP requests to perform actions (such as create, read, update, and delete) on the server.

- **Initial Use Case Model, consisting of a Use Case diagram and Use Case descriptions.**

Use Case Diagram:



Use Case Descriptions:

Use Case ID:	RetailApp.001		
Use Case Name:	Browse Products		
Created By:	Hari Hara Sudhan	Last Updated By:	
Date Created:	April 22, 2025	Date Last Updated:	April 22, 2025
Actor:	Guest User, Registered User		
Description:	User browses available athletic products		
Preconditions:	The application is loaded		
Postconditions:	Products displayed to the user based on categories		
Priority:	Essential		
Frequency of Use:	Frequent		
Flow of Events:	<ol style="list-style-type: none">1. User lands on the product listing page.2. System displays products by category.		
Alternative Workflow:			
Exceptions:			
Includes:			
Special Requirements:	Products must load in < 2 sec.		
Assumptions:			
Notes and Issues:	Requires filter options for customers		

Use Case ID:	RetailApp.002		
Use Case Name:	Manage Cart		
Created By:		Last Updated By:	
Date Created:	April 22, 2025	Date Last Updated:	April 22, 2024
Actor:	Guest User, Registered User		
Description:	User adds selected products to cart		
Preconditions:	Chosen product is in stock		
Postconditions:	Shopping Cart gets updated		
Priority:	Essential		
Frequency of Use:	Frequent		
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks "Add to Cart". 2. System updates cart. 		
Alternative Workflow:	Product is already in cart: <ol style="list-style-type: none"> 1. User clicks “Add to Cart”. 2. Cart increments item by 1 that is already in cart. 		
Exceptions:			
Includes:	Confirms the availability of the product		
Special Requirements:			
Assumptions:			
Notes and Issues:			

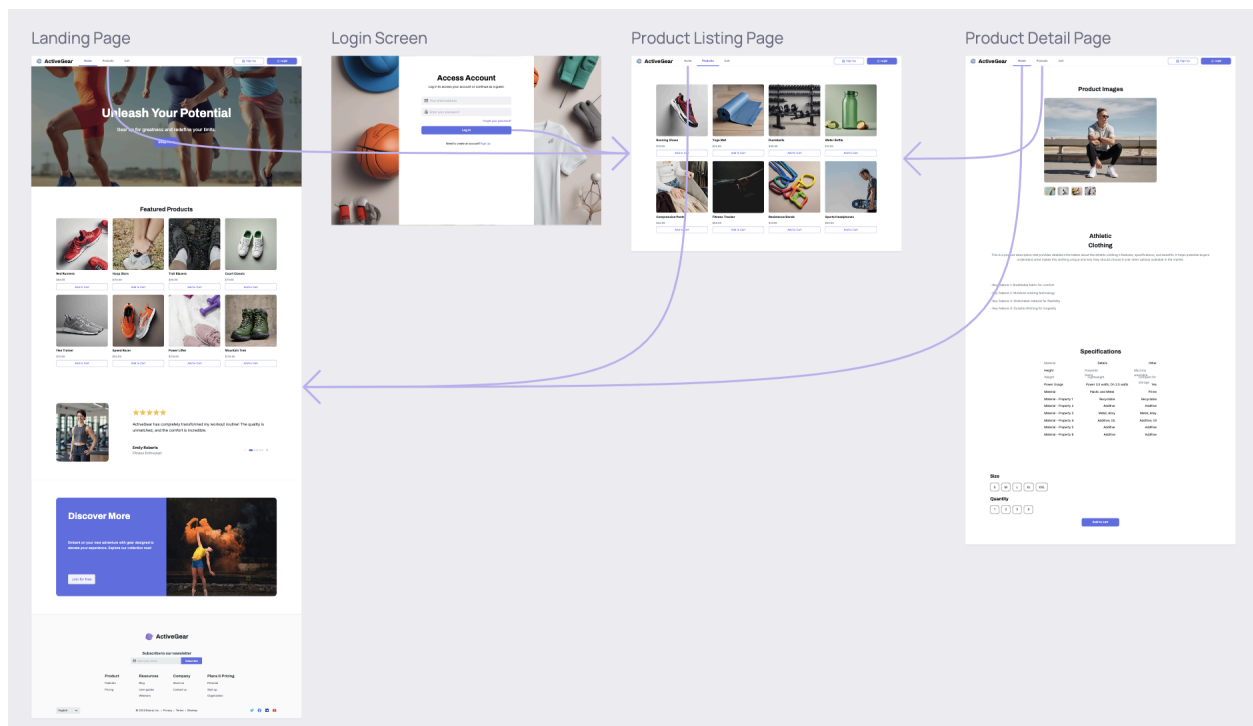
Use Case ID:	RetailApp.003		
Use Case Name:	Login		
Created By:		Last Updated By:	
Date Created:	April 22, 2025	Date Last Updated:	April 22, 2025
Actor:	Guest User, Registered User		
Description:	User Logs in to their account		
Preconditions:	User has already registered account		
Postconditions:	User is authenticated		
Priority:	Essential		
Frequency of Use:	Frequent		
Flow of Events:	<ol style="list-style-type: none"> 1. The User enters credentials 2. The system verifies and logs in 		
Alternative Workflow:	Reset password, create account		
Exceptions:	User not found		
Includes:	Verifies the credentials with the user registration details		
Special Requirements:			
Assumptions:			
Notes and Issues:			

Use Case ID:	RetailApp.004		
Use Case Name:	Checkout		
Created By:		Last Updated By:	
Date Created:	April 22, 2025	Date Last Updated:	May 16, 2025
Actor:	Guest User, Registered User		
Description:	User completes the item purchase by entering information and submitting payment.		
Preconditions:	Items are added to the cart		
Postconditions:	Order is placed successfully and user is notified		
Priority:	Essential		
Frequency of Use:	Frequent		
Flow of Events:	<ol style="list-style-type: none"> 1. User goes to the checkout page 2. User enters billing address and payment details 3. System confirms the availability of chosen items 4. Payment processing is completed by the system 5. Order details are stored and displays a confirmation message to the user. 		
Alternative Workflow:	<ol style="list-style-type: none"> 1. In case of payment failure, user is prompted to retry the payment 2. If chosen products are not in stock, the user is notified of the status and checkout process is stopped. 		
Exceptions:	Payment method not supported; Entered billing address is not valid.		
Includes:			
Special Requirements:	Confirmation message should include the necessary order details		
Assumptions:	Items are added to the cart		
Notes and Issues:			

Use Case ID:	RetailApp.005		
Use Case Name:	Register		
Created By:		Last Updated By:	
Date Created:	April 22, 2025	Date Last Updated:	April 22, 2025
Actor:	Guest User		
Description:	Allows a guest to create a new user account by entering personal details.		
Preconditions:	Entered email is not previously registered		
Postconditions:	a new account is created and stored in the database.		
Priority:	Essential		
Frequency of Use:	Occasional		
Flow of Events:	<ol style="list-style-type: none"> 1. Guest User enters required information 2. System validates the information 3. New account is created and user can login anytime 		
Alternative Workflow:	If email already exists in the system, the user is asked to log in instead.		
Exceptions:			
Includes:			
Special Requirements:	Password must be at least 10 characters		
Assumptions:	User has access to internet		
Notes and Issues:			

Use Case ID:	RetailApp.006		
Use Case Name:	Cancel Order		
Created By:		Last Updated By:	
Date Created:	April 22, 2025	Date Last Updated:	April 22, 2025
Actor:	Guest, Registered Member		
Description:	Users can cancel order within 12 hours; remove items		
Preconditions:	Purchase had been made		
Postconditions:	Within 12 hours		
Priority:			
Frequency of Use:	Moderate		
Flow of Events:	<ol style="list-style-type: none"> 1. User goes to 'My orders' section 2. Selects an order that is active 3. Clicks on 'cancel order' 4. Confirm to cancel the order 		
Alternative Workflow:	Option to not cancel the item(s)/order		
Exceptions:	Order is not active, no previous orders on the account		
Includes:			
Special Requirements:			
Assumptions:			
Notes and Issues:			

- User Interface Mockup



The mockup above visually represents how this application would appear to the end user. From the landing page where all other pages can be readily accessed, which includes the login screen for a user to either make a new account or log into their existing account. From there, the user is directed to the product listing page, where all available items will be displayed with their price and an add to cart option. A user can then get more information on a specific product through the product detail page, which is accessible by just clicking on any product on the listing page.