User Requirements specifications



Content

Introduction

Functional Requirements 1

Non-Functional Requirements 2

Use cases 3

**Introduction**

The following sections will detail the user requirements, functional requirements, non-functional requirements, and use cases necessary to create a successful online bike shop for Cycle Universe.

**Functional requirements**

* 1. The website should have a homepage that displays the latest bikes in the store.
* 2. Users should be able to filter bikes by kind and search term.
* 3. The website should have detailed pages for every bike with necessary descriptions and high-quality images.
* 4. Registered users should be able to add items to their shopping cart and view them later.
* 5. The desktop application should allow the store employees to add new bikes to the system.

**Non-Functional requirements**

* The webapp should load quickly to ensure a good user experience.
* The webapp should be easy to navigate and user-friendly.
* The page should be secure and protect user data.
* The desktop application should be reliable and performant.

**Use cases**

UC-1: Browse bikes without registering

Actor: User

Precondition: User is not registered

Main Success Scenario:

1. Actor visits the store page
2. System displays all of the available bikes
3. Actor selects a bike to view more details
4. System displays the detailed bike page with specifications and images

Extensions: None

UC-2: Register an account

Actor: User

Precondition: The user does not have an existing account

Main Success Scenario:

1. Actor goes to sign up page
2. System prompts the actor to fill in registration information
3. Actor fills in the required information and submits the form
4. System confirms the registration and shows the login form Extensions:

4a: Actor enters invalid information

.1: System displays an error message and prompts the actor to correct the information

.2: Return to MSS step 3

UC-3: Filter bikes by type

Actor: User

Precondition: Actor has already accessed the store page

Main Success Scenario:

1. Actor selects a type of bike from the filter options

2. System updates the displayed bikes to match the selected filter

3. Actor combine multiple types of bikes from the filter options

4. System updates the displayed bikes to match the selected filter

Extensions: None

UC-4: View detailed bike page

Actor: User

Precondition: Actor has a device with internet // change

Main Success Scenario:

1. Actor visits the website
2. System displays the homepage with 3 of the latest bikes
3. Actor selects a bike to view more details
4. System displays the detailed bike page with specifications and images

Extensions: None

UC-5: Add new bike to inventory

Actor: Store Manager

Precondition: Actor is logged in //select from predefined data

Main Success Scenario:

1. Actor selects the add bike option
2. System prompts the Actor to fill in the required information for the new bike
3. Actor fills in the information and submits the form
4. System adds the new bike to the inventory

Extensions:

3a: Store Manager enters invalid information

.1: System displays an error message and prompts the Actor to correct the information

.2: Return to MSS step 3

UC-6: Add Items to the shopping cart

Actor: User

Precondition: Actor is logged in

Main Success Scenario:

1. Actor goes to store page

2. System displays all the bikes in the shop

3. Actor adds a bike to the shopping cart

4. System displays actor’s shopping cart with the new item in it

UC-07: Browse Bikes using search bar

Actor: User

Precondition: Actor has already accessed the store page

Main Success Scenario:

1. Actor searches a keyword in the search bar

2. System displays the search result

Extensions:

1a: No results had been found from user search

.1: System shows that no result was found

.2: Return to MSS step 1

UC-08: Checkout

Actor: User

Precondition: Actor has already added items to the cart

Main Success Scenario:

1. Actor selects the proceed to payment option from the cart

2. System displays the checkout page with the total price of the cart and prompts the user to fill in the shipping and payment information

3. Actor fills in the information and submits the form

4. System register user order and shows the order history page

Extensions:

1a: Actor has more than one order placed and selects the proceed to payment option

.1: System shows the checkout page with the shipping information already filled in and prompts the user to fill in the payment information

.2: Return to MSS step 3

3a: Actor fills the form with invalid information

.1: System prompts the user to fill in valid information

.2: Return to MSS step 3

UC-09: Return Item

Actor: User

Precondition: Actor has placed an order

Main Success Scenario:

1. Actor selects an item he wants to return from the order history page

2. System shows a return page with the item to return and prompts the user to fill in the reason of return and a comment

3. Actor fills in the information and submits the form

4. System register user return request and changes the status of the item

5. Shop manager accepts the return from desktop app

6. System gives the option to the user to downland the return label

UC-10: Use coupon code at checkout

Actor: User

Precondition: Actor is already on the checkout page

Main Success Scenario:

1. Actor fills in the coupon field with a valid coupon code

2. System adds the coupon to the cart and shows the discount and the coupon applied

Extensions:

1a: Actor fills the coupon field with an invalid coupon code

.1: System shows a message that coupon code is invalid

.2: Return to MSS step 1







