User v 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**

* The website should have a homepage that displays the latest bikes in the store.
* Users should be able to filter bikes by brand and kind.
* The website should have detailed pages for every bike with necessary descriptions and high-quality images.
* Registered users should be able to add items to their shopping cart and view them later.
* The desktop application should allow the store to manage inventory by performing CRUD operations.

**Non-Functional requirements**

* The website should load quickly to ensure a good user experience.
* The webpage 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**

Use case: Browse bikes without registering

Actor: User

Main Success Scenario:

1. Actor visits the store page
2. System displays a list of the available bikes
3. Actor selects a bike to view more details
4. System displays the detailed bike page with specifications and images
5. Actor goes back to the homepage or continues browsing bikes

Extensions: None

Use case: Register an account and add items to shopping cart

Actor: User

Main Success Scenario:

1. Actor goes to the registration form
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 logs in the actor
5. Actor browses bikes and selects a bike to add to their shopping cart
6. System shows a detailed bike page
7. Actor adds the bike to the shopping cart
8. System adds the selected bike to the shopping cart and displays the updated cart

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

Use case: Filter bikes by brand and kind Actor:

User: user

Main Success Scenario:

1. Actor goes to the store app.
2. System displays a page with all the bikes in the shop.
3. Actor selects a brand and/or kind from the filter options
4. System updates the displayed bikes to match the selected filter

Extensions: None

Use case: View detailed bike page

Actor: User

Main Success Scenario:

1. Actor visits the website
2. System displays the homepage with the latest bikes
3. Actor selects a bike to view more details
4. System displays the detailed bike page with specifications and images
5. Actor goes back to the homepage or continues browsing bikes

Extensions: None

Use case: Add new bike to inventory

Actor: Store Manager

Main Success Scenario:

1. Actor opens the desktop application by completing the login information
2. System displays the inventory management interface
3. Actor selects the add bike option
4. System prompts the Store Manager to fill in the required information for the new bike
5. Actor fills in the information and submits the form
6. System adds the new bike to the inventory and updates the web page to display it

Extensions:

4a: Store Manager enters invalid information

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

.2: Return to MSS step 4