ES – Practical Case 2

UABikes
Requirements specification

Version 1.0

Revision History

Date	Version	Description	Author
20/10/2022	1.0	Initial version	Florin Sofinet
15/12/2022	1.1	Second version	Florin Sofinet

Table of Contents

Requirements specification

1. Introduction

In this this document are described and specified the functionalities of UABikes application. The specification of functionalities are presented using use-case modeling followed by supplementary specification relevant to those use cases.

1.1 Scope

The scope of this document is part of the implementation of Practical Case 1, in the context of the curricular unit Software Engineering, from the Master in Informatics and Computer Engineering at DEETC at ISEL.

1.2 Definitions, Acronyms, and Abbreviations

See glossary (document GL-PRA-01).

1.3 References

Support documentation for the Software Engineering course

Luís Morgado, ISEL-DEETC, 2022.

1.4 Overview

The document has the following structure:

Section 1: Contains an overview of the document

Section 2: Use case organization

Section 3: Description of use cases

Section 4: Supplementary specification

2. Use case structure

2.1 General functionality

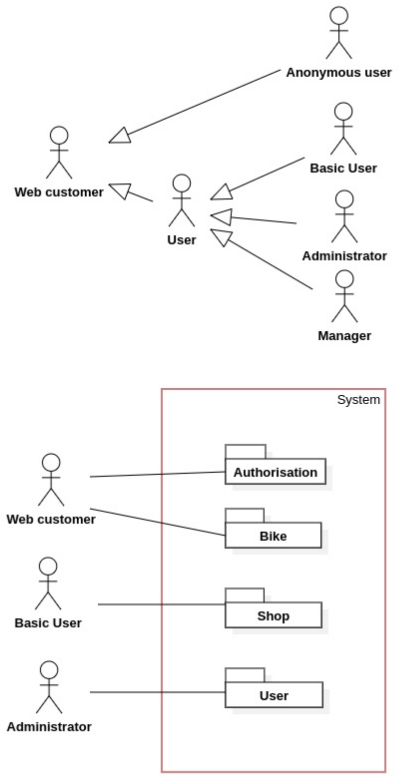

Figure 1: General functionality and actors extension

2.2 Functionality of Authentication

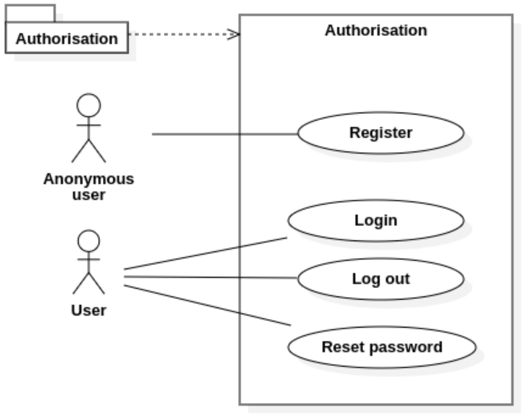

Figure 2: Functionality associated to Authorization

2.3 Functionality of Shop

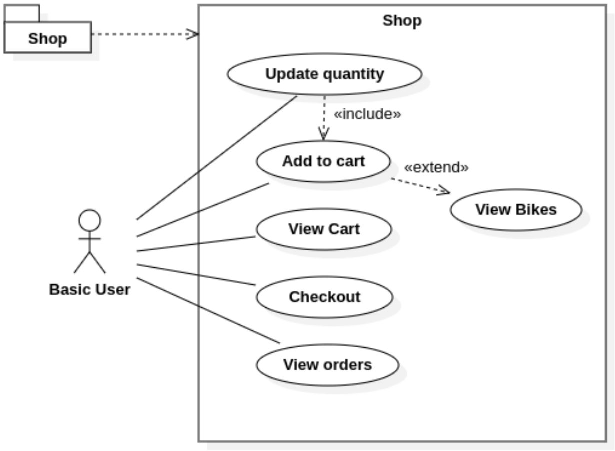

Figure 3: Functionality associated to Shop

2.4 Functionality of Bikes Management

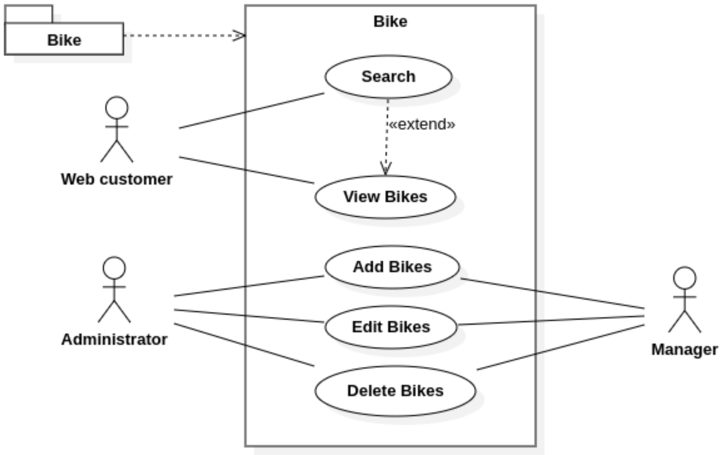

Figure 4: Functionality associated to Manage Bikes

2.5 Functionality of users management

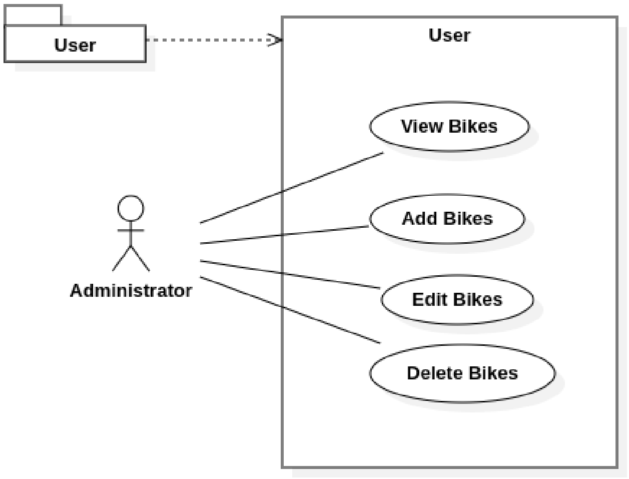

Figure 5: Functionality associated to Manage Users

3. Description of *Authorization* use cases

3.1 *Register* – Use Case

Description: This use case describes how the actor registers in the system.

Actors: Anonymous user

Main Flow:

1. The use case begins when the actor clicks in the Register tab in navigation menu.
2. The system shows the register page.
3. The actor fills the form (**see R1.1**) and clicks on register button.
4. The system saves the account.
5. The system notifies the actor (**see R1.2**) and the use case ends.

Alternative flow 1 (Invalid email):

4. In step four if the actor types an invalid e-mail the system displays an error message (**see R1.3**).
5. The actor can try again by typing a correct e-mail.
6. The flow resumes at step 3.

Alternative flow 2 (Passwords do not match):

4. In step four if the actor fills passwords that do not match an error message is shown (**see requirement R1.4)**
5. The actor can try again by retyping the passwords.
6. The flow resumes at step 3.

3.2 *Login* – Use case

Actors: Basic user, Manager, Administrator

Description: This use case describes how the user sign in the system.

Main Flow – Login

1. The use case begins when the actor clicks in the Login icon in navbar.
2. The system displays the Login page.
3. The actor types the credentials in the form (**see R2.1**) and clicks in *Log In* button.
4. The system validates the credentials and logs the actor in the system.
5. The system displays the main page showing navigation bar related to type of the account (**see requirement R2.2, R2.3, R2.4)** and use case ends.

Post-condition: Actor is logged into the system.

Alternative Flow 1 (invalid credentials):

4. On step four the system cannot validate the credentials of the actor and an error message is displayed (**see R2.5)**.
5. The actor can try again by typing the credentials in the form (**see R2.1**).
6. The system resumes the flow at step 3.

3.3 *Reset password* – Use case

Description: This use case describes how the actor resets the password of his account.

Actors: Basic User, Manager, Administrator

Main Flow:

1. The use case begins when the actor clicks in the Reset Password button in Login Page.
2. The system shows the reset page.
3. The actor types the e-mail in the input box (**see R3.1**) and clicks send button.
4. The system notifies the actor that an e-mail was send to the email address (**see R3.2**) and the use case ends.

Post-condition: The system sends an email to the associated user to reset the password.

Alternative Flow 1 (account doesnt exist):

4. In step four if the actor typed an e-mail that doesnt exist an error message is displayed instead (**see R3.3**).
5. The actor can try again by typing in the input box (**see R3.1**) an e-mail that exists in the system.
6. The system resumes the flow at step 3.

3.4 *Log out* – Use case

Description: This use case describes how the actor logs out from the system.

Actors: Basic User, Manager, Administrator

Pre-condition: The actor is logged in.

Main Flow

1. The use case begins when the actor clicks in the Log out icon in navigation bar.
2. The system logs out the actor.
3. The actor is redirected to the main page and use case ends.

Post-condition: The actor is logged out of the system.

4. Description of *Shop* use cases

4.1 *Add to cart* – Use case

Description: This use case describes how the actor adds bikes to the cart.

Actors: Basic User

Pre-condition: The actors is logged in the system.

Main Flow

1. The use case begins when the actor clicks on add to cart button (**see R6.2**) from bikes box in main page.
2. The system updates the cart.

Alternative flow 1 (the bike is out of stock)

1. In step one the bicycle is out of stock.
2. The system displays the add button as disabled (**see R6.2**) and use case ends.

4.2 *Delete from cart* – Use case

Description: This use case describes how the actor deletes bikes from the cart.

Actors: Basic User

Pre-condition: The actors is logged in the system and has bikes in the cart.

Main Flow

1. The use starts when the actor clicks on cart icon **(see R2.3)** in the navbar.
2. The system displays the cart page.
3. The actor clicks on delete from cart button (**see R5.3**) of bike box.
4. The system removes the cart item, updates the total value and use case ends.

4.3 *View Cart* – Use case

Description: This use case describes how the actor visualizes the cart page

Actors: Basic User

Pre-condition: The actors is logged in the system.

Main Flow:

1. The use starts when the actor clicks on cart icon (**see R2.3**) in the navbar.

2. The system displays the cart page and use case ends.

4.4 *Update cart quantity* – Use case

Description: This use case describes how the actor can update bike quantity in the cart.

Actors: Basic User

Pre-condition: The actor is logged in the system and has bikes in the cart.

Main Flow:

1. The use case starts when the actor clicks on cart icon **(see R2.3)** in the navbar.
2. The system displays the cart page.
3. The actor clicks on plus button (**see R5.1**) of bike box.
4. The system updates the total value, quantity number (**see R5.2)** and use case ends.

Alternative Flow 1 (the actor decreases quantity in the cart):

3. In step three the actor clicks on minus button (**see R5.1**) of bike box.
4. The system updates the total value, quantity number **(see R5.2)** and use case ends.

4.5 *Checkout* – Use case

Description: This use case describes how the actor places an order to buy or rent a bike.

Actors: Basic User

Pre-condition: The actors is logged in the system and has at least one bike in the cart.

Main Flow

1. The use case begins when the actor clicks on the checkout button from cart page (**see R5.4**).
2. The system shows the checkout page that contains a form with data to be filled (**see R7.1, R7.2**).
3. The actors fills in the data required (**see R7.1**) and chooses an option to acquire the bike (**see R7.2**)
4. The system updates the cost according to option selected.
5. The actor clicks on Finalise order button.
6. The system saves the order and displays a table with order details and use case ends (**see R7.3**).

Alternative Flow 1 (the actor selects the rent option):

1. In step three the actor selects rent option.
2. The system updates total value.
3. The flow is resumed from step 4.

4.6 *View orders* – Use case

Description: This use case describes how the actor checks orders history.

Actors: Basic User

Pre-condition: The actors is logged in the system.

Main Flow

1. The use case begins when the actor clicks on the order icon from navbar (**see R2.3**).
2. The system shows a table (**see R7.3**) with orders and use case ends.

Alternative Flow 1:

1. In step two an empty page is shown instead if there are no orders and use case ends.

5. Description of *Manage bikes* use cases

5.1 *Add bikes* – Use case

Description: This use case describes how the actor adds new bikes to the system.

Actors: Manager, Administrator

Pre-condition: The actor is logged in.

Main Flow

1. The use case begins when the actor clicks on the add bike button (**see R8.1**).
2. The system displays the page to add bicycles (**see R9**).
3. The actor fills the form (**see R9.1**) of bicycles and clicks on add button (**see R9.2**).
4. The system saves the bike in the system.
5. The system notifies the actor that the bicycles was sucessfully saved and the use case ends (**see R9.3**).

Post condition: The bike is stored in the system.

Alternative Flow 1 (the system fails to save the bike):

3. In step four the system fails to save the bike in the system.
4. The system notifies the actor showing an error message (**see R9.4**) and use case ends .

5.2 *View Bikes* – Use case

Description: This use case describes how the actor visualises bikes in the system.

Actors: Manager, Administrator, Basic User, Anonymous user

Main Flow

1. The use case begins when the actor clicks on the home icon in the navbar (see **R2.2**, **R2.3**, **R2.4**).
2. The system displays the main page with bikes and the use case ends.

Alternative Flow 1 (system failed to load bikes):

1. In the step two the system fails to load the bikes

2. A message is shown instead of bikes (**see R13**) and use case ends.

5.3 *Edit Bikes* – Use case

Description: This use case describes how the actor can update information related to bikes.

Actors: Manager, Administrator

Pre-condition: The actor is logged in the system

Main Flow

1. The use case begins when the actor clicks on edit bike button (**see R6.3**) from the bikes.
2. The system displays the edit page with the form(**see R10.1**) filled of selected bike.
3. The actor types data in the form and clicks on edit button (**see R10.2**).
4. The system notifies the actor that changes were updated successfully and use case ends.

Post condition: The bike is updated in the system.

5.4 *Delete Bikes* – Use case

Description: This use case describes how the actor deletes bikes from the system.

Actors: Manager, Administrator

Pre-condition: The actor is logged in the system

Main Flow

1. The use case begins when the actor clicks on delete bike button (**see R6.4**) from the bike container.
2. The system displays a dialog (**see R4**) asking the actor if he confirms the deletion.
3. The actor clicks on yes button.
4. The system deletes the respective bike and use case ends.

Post condition: The bike is removed from the system.

Alternative Flow 1 (the actor clicks on cancel):

1. In step three the actor clicks on cancel button.
2. The system does not delete the bike and use case ends.

5.5 *Search Bikes* – Use case

Description: This use case describes how the actor searches bikes in the system.

Actors: Manager, Administrator, Basic User, Anonymous User

Main Flow

1. The use case begins when the actor types on search input (**see R8.2**).
2. The system displays bikes according to search information typed by the actor and the use case ends.

Alternative flow 1 (the system cannot find matching case):

1. In step 2 the system doesn't find any matching case.
2. The system doesn't display anything instead and use case ends.

6. Description of *Manage users* use cases

6.1 *Add users* – Use Case

Description: This use case describes how the actor adds users in the system.

Actors: Administrator

Pre-condition: The actor is logged in.

Main Flow

1. The use case begins when the actor clicks on the admin icon from navigation bar.
2. The system displays the administrator page.
3. The actor fills the form (**see R11.1**) displayed and clicks on add user button (**see R11.2**)
4. The system stores the user in the system
5. The system notifies the actor (**see R11.3**) and use case ends.

Post condition: A user is added in the system.

6.2 *View Users* – Use case

Description: This use case describes how the actor visualises users in the system.

Actors: Administrator

Main Flow

1. The use case begins when the actor clicks on the admin icon in the navbar.

2. The system displays the admin page that contains rows of users and the use case ends.

6.3 *Edit Users* – Use case

Description: This use case describes how the actor can edit user data.

Actors: Administrator

Pre-condition: The actor is logged in the system

Main Flow

1. The use case begins when the actor clicks on the admin icon in the navbar.

2. The system displays the admin page.

3. The actor clicks on the edit icon (**see R11.4**) from the user it wants to edit.

4. The system shows edit page with form filled (**see R12**) from user selected.

5. The actor changes data in the form.

6. The actor clicks on edit user button (**see R12.2**).

7. The system updates the user data in system

8. The system notifies the actor above the form that the action was completed successfully and the use case ends.

Post-condition: User data is updated in the system.

6.4 *Delete Users* – Use case

Description: This use case describes how the actor can delete users.

Actors: Administrator

Pre-condition: The actor is logged in the system

Main Flow

1. The use case begins when the actor clicks on the admin icon in the navbar.

2. The system displays the admin page.

3. The actor clicks on the trash icon (**see R11.5**) from an user.

4. The system shows a dialog(**see R4**) to confirm the action.

5. The actor clicks on yes button.

6. The system deletes the user from the system

7. The system notifies the actor that the action was completed and the use case ends.

Post-condition: The user is deleted from the system.

Alternative Flow 1 (the actor clicks on cancel button):

1. In step 5 the actor clicks on cancel
2. The system does not perform the delete action.
3. The system hides the dialog and use case ends.

7. Supplementary Requirements

Ref.	**Description**	**Category**
R1	The register form has the following:	
R1.1	The data of form: a) Username b) Name c) Email d) Password e) Repeat password	
R1.2	After the system saved the account the actor is notified through a green notification that the account was created.	
R1.3	The system shows a red notification when an invalid e-mail is inserted in the register form.	
R1.4	The system shows a red notification when passwords do not match in the register form.	
R2	The login form contains the following:	
R2.1	Data of form: a) Username b) Password	
R2.2	The navbar displays home and logout icon for manager type account.	
R2.3	The navbar displays home, logout, orders and cart button for basic user type account.	
R2.4	The navbar displays home, logout and manage users button for administrator type account.	
R2.5	The system shows bellow the login button a red notification when the password or username is incorrect.	
R3	The reset-password form contains the following:	
R3.1	Input box where the user types email to reset the password.	
R3.2	The system shows a green notification that includes a message when the it successfully sends an email to the respective account.	
R3.3	The system shows a red notification when e-mail inserted is not found in the system.	

R4	The system shows a dialog, a box that contains a message and 2 action buttons respectively yes and cancel. The message asks the user if he proceeds with respective action.	
R5	The cart page contains the following:	
R5.1	Plus and minus buttons inside cart item that are visible only when the actor is in cart page.	
R5.2	Number between **R5.1** buttons that represents the current quantity of the respective cart item.	
R5.3	A button that has an **x** inside used to delete cart items, visible only to basic users when they are in cart page.	
R5.4	Checkout button situated in right-bottom of cart page and is clickable when user has at least 1 cart item with quantity > 0.	
R6	The bike container has the following:	
R6.1	Information section that contains: a) Model b) Image c) Details d) Stock e) Price.	
R6.2	Add button that appears in the bikes box visible only when type of actor is basic user and is in bikes page. It is clickable when the stock is greater then 0. Is disabled and not clickable when the stock is 0.	
R6.3	Edit button, visible only to managers and administrators.	
R6.4	Delete from system button, visible only to managers and administrators.	
R7	The checkout form has:	
R7.1	Data of form: a) Name b) Address c) Telephone d) Zip Code	
R7.2	Radio button that has 2 options to select: a) Buy b) Rent	
R7.3	The system displays a table with following fields: 1. Transaction (code used for the payment) 2. Order state 3. Acquire Type 4. Items 5. Total	

	6. Address 7. User	
R8	The main page contains:	
R8.1	Add bike button that is visible only to managers and administrators, appears at the right of Input box(see R8.2) under navbar.	
R8.2	Input checkbox used to search bikes, appears under navbar in left corner.	
R9	Add bikes page contains:	
R9.1	Data form: a) Model b) Image Source c) Stock d) Details e) Electric f) Price.	
R9.2	If the user doesn’t add a value in the form and clicks on edit button, a tooltip appears above the empty input and the edit action is canceled.	
R9.3	The system shows a green notification above the form when the bike was successfully added in the system.	
R9.4	The system shows a red notification above the form when the bikes failed to be added in the system.	
R10	Edit bikes page contains:	
R10.1	See R9.1	
R10.2	Edit button that is gray and not clickable (disabled) initially. It becomes green and clickable (active) only when the form (**see R10.1**) is filled.	
R11	Administrator page has:	
R11.1	Data form: a) Username b) Name c) Email d) Password e) Repeat Password f) User type	
R11.2	Add user button that is gray and not clickable (disabled) initially. It becomes green and clickable (active) only when the form (**see R10.1**) is filled.	
R11.3	The system shows a green notification above the form when the user was successfully added in the system.	
R11.4	Edit icon placed in the right of each user from admin page.	
R11.5	Delete icon placed in the right of each user from admin page.	

R12	Edit user page contains:	
R12.1	Data form: a) Username b) Name c) Email d) Password e) Repeat Password f) User type	
R12.2	Edit user button that is gray and not clickable (disabled) initially. It becomes green and clickable (active) only when the form (**see R12.1**) is filled.	
R13	Bikes page displays: “Bikes failed to load!”.	

Note for the second version:
1. This version now includes a stakeholders requirement which is the order history use case. This was asked by stakeholders for more brevity on the orders that were placed. Therefore we added this new case.
2. Forms were updated for R12.1, R11.1 and R7 forms.