



POLYTECH<sup>®</sup>  
MONTPELLIER



UNIVERSITÉ  
DE MONTPELLIER

---

## SOFTWARE ENGINEERING PROJECT

---

## TOOL SHARE APPLICATION

---



A Java application developed for people to share tools

By BARBOU Bryan

& GUYON Valentin

& ISSARTEL Solène

& LAUNOIS Remy

in Computer Science & Management 4

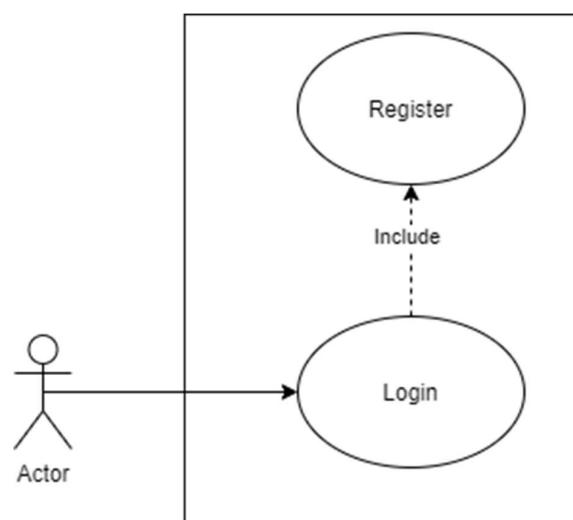
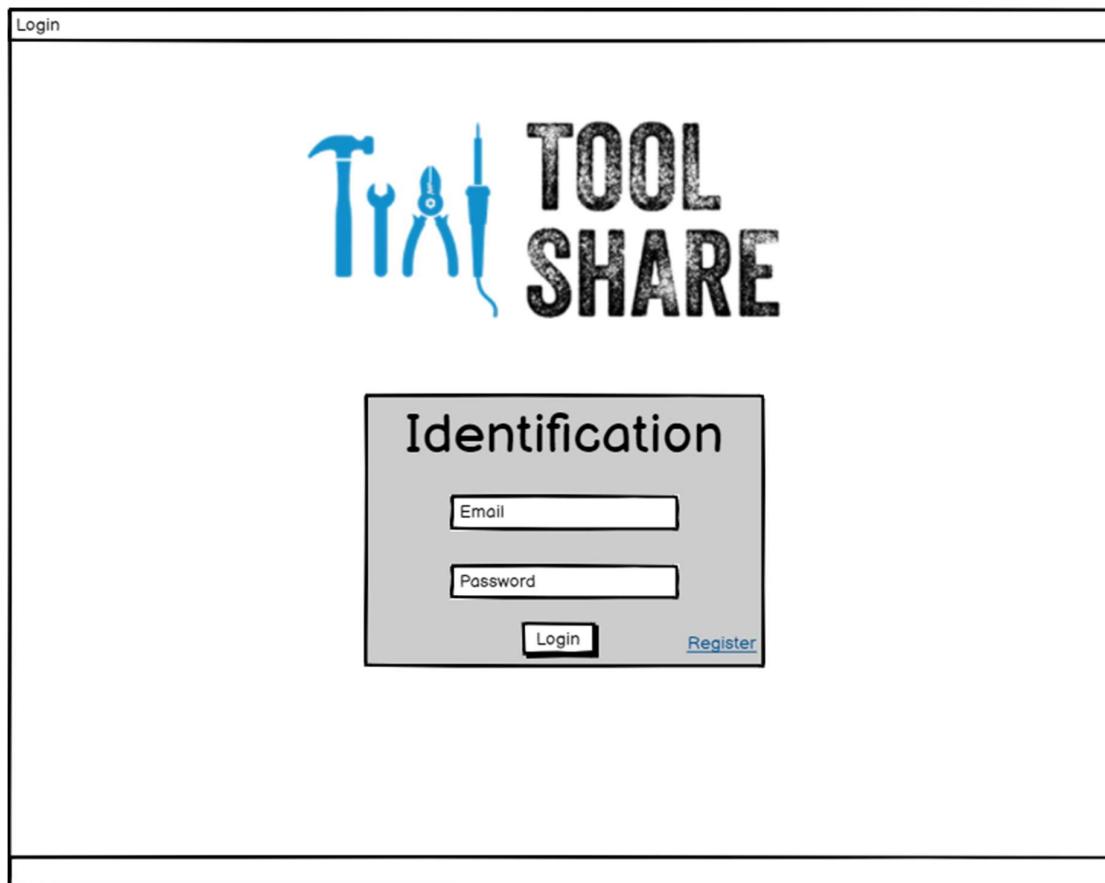
YEARS 2020-2021

Tutor: Mr. Stratulat

## Contents

1. Login .....	2
2. Account/User Management .....	5
3. Offer .....	12
4. Priority Offers.....	17
5. Evaluation system .....	21
6. FAQ .....	26
7. RESERVATION.....	31
8. Payment .....	36
9. Favorite System .....	40
10. History System.....	43
11. Category .....	46
Annex .....	51

## 1. Login



## 1.1 Brief Description

This use case describes how a user logs into the ToolShare app.

## 1.2 Flow of Events

### 1.2.1 Basic Flow

This use case starts when the actor wishes to Login to the ToolShare app.

1. The system requests that the actor enter his/her email and password
2. The actor enters his/her email and password.
3. The system validates the entered email and password and logs the actor into the system.

### 1.2.2 Alternative Flows

#### 1.2.2.1 Invalid Name/Password

If in the **Basic Flow**, the actor enters an invalid email and/or password, the system displays an error message. The actor can choose to either return to the beginning of the **Basic Flow** or cancel the login, at which point the use case ends.

## 1.3 Special Requirements

None.

## 1.4 Pre-Conditions

The actor is supposed to have an account.

## 1.5 Post-Conditions

If the use case was successful, the actor is now logged into the system. If not, the system state is unchanged.

## 1.6 Extension Points

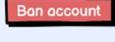
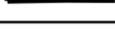
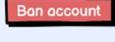
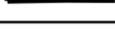
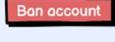
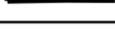
None.

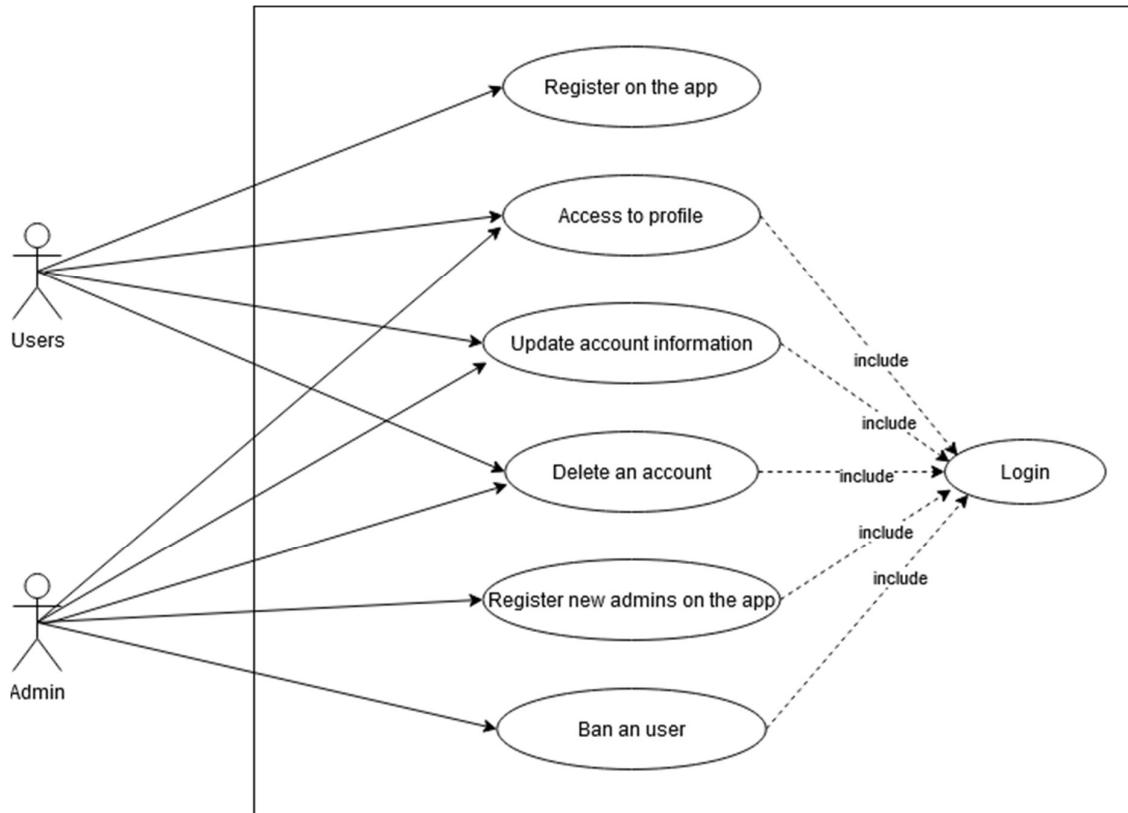
## 2. Account/User Management

<p>Register</p>  <p><b>Register</b></p> <p>Email <input type="text"/></p> <p>Firstname <input type="text"/></p> <p>Lastname <input type="text"/></p> <p>Accommodation Address (city) <input type="text"/></p> <p>Phone Number <input type="text"/></p> <p>Password <input type="text"/></p> <p><input type="button" value="Validate"/></p>	<p>Profile</p>  <p><b>Post an offer</b></p> <p><b>PROFILE</b></p> <p>Email xxxxx.xxx@xxxx.fr</p> <p>Firstname xxxxxxxx</p> <p>Lastname xxxxxxxx</p> <p>Accommodation Address (city) xxxxxxxx</p> <p>Phone Number 04xxxxxxxx</p> <p>Password xxxxxxxx</p> <p><input type="button" value="Edit profile"/></p> <p><input type="button" value="Change password"/></p> <p><input type="button" value="Delete account"/></p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Update account</p>  <p><b>Post an offer</b></p> <p><b>Update account</b></p> <p>Email <input type="text" value="xxxx.xxxx@xxxx.fr"/></p> <p>Firstname <input type="text" value="xxxx"/></p> <p>Lastname <input type="text" value="xxxxxx"/></p> <p>Accommodation Address (city) <input type="text" value="xxxxx"/></p> <p>Phone Number <input type="text" value="04xxxxxxxx"/></p> <p><input type="button" value="Validate"/></p>	<p>Change password</p>  <p><b>Post an offer</b></p> <p><b>Change password</b></p> <p>Current password <input type="text"/></p> <p>New password <input type="text"/></p> <p>Confirm password <input type="text"/></p> <p><input type="button" value="Validate"/></p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## &lt; View Admin &gt;

<p>Ban an user</p>  <p>Offers  Users list </p> <p>USERS LIST </p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 5px;">id_user: xx Name : xxxxxxxx xxxxxxx</td><td style="text-align: right; padding: 5px;"></td></tr> <tr><td style="padding: 5px;">id_user: xx Name : xxxxxxxx xxxxxxx</td><td style="text-align: right; padding: 5px;"></td></tr> <tr><td style="padding: 5px;">id_user: xx Name : xxxxxxxx xxxxxxx</td><td style="text-align: right; padding: 5px;"></td></tr> <tr><td style="padding: 5px;">id_user: xx Name : xxxxxxxx xxxxxxx</td><td style="text-align: right; padding: 5px;"></td></tr> <tr><td colspan="2" style="height: 40px;"></td></tr> </table>	id_user: xx Name : xxxxxxxx xxxxxxx		id_user: xx Name : xxxxxxxx xxxxxxx		id_user: xx Name : xxxxxxxx xxxxxxx		id_user: xx Name : xxxxxxxx xxxxxxx				<p>Register new admins</p>  <p>Offers  Users list </p> <p>Register an admin</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Email</td><td style="width: 150px; height: 30px;"></td></tr> <tr><td>Firstname</td><td style="width: 150px; height: 30px;"></td></tr> <tr><td>Lastname</td><td style="width: 150px; height: 30px;"></td></tr> <tr><td>Password (temporary)</td><td style="width: 150px; height: 30px;"></td></tr> <tr><td colspan="2" style="text-align: right; padding: 5px;"></td></tr> </table>	Email		Firstname		Lastname		Password (temporary)			
id_user: xx Name : xxxxxxxx xxxxxxx																					
id_user: xx Name : xxxxxxxx xxxxxxx																					
id_user: xx Name : xxxxxxxx xxxxxxx																					
id_user: xx Name : xxxxxxxx xxxxxxx																					
Email																					
Firstname																					
Lastname																					
Password (temporary)																					
																					



## 2.1 Brief Description

In this use case there are two types of actors: common users and administrators. The admins have additional rights like ban users accounts but they don't have others rights like post offers, comments or rates for examples.

This use case allows the different actors to manage their information and specifically to the administrators to manage users accounts. This includes registering, updating, banning, and deleting an account from the system.

## 2.2 Flow of Events

### 2.2.1 Basic Flow

This use case starts when an actor wants to register, change, and/or delete his account from the system. The flow depends on the role of the actor.

- If the actor is a common user then he can *register, modify, delete* his account (refer to **common flow** and **flow for common users**).
- If the actor is an admin then he can *register, modify, delete* his account and he can *ban* common users (refer to **common flow** and **flow for admins**).

### 2.2.2 Common flow

#### 2.2.2.1 Access to profile

When an actor wants to access his account information, he has the possibility to access the profile page.

1. The user requests the system to access his account information.
2. The system retrieves and displays the actor information.
3. The actor has access to all information concerning his account information and has the possibility to edit (update) his profile or change his password or delete his account.

#### **2.2.2.2 Update account information (apart from the password)**

1. The system requests that the actor goes to his profile and decides to modify information.
2. The system retrieves and displays the actor information.
3. The actor makes the desired changes to his information. This includes any of the information specified in the **Register on the app** sub-flow.
4. Once the actor updates the necessary information, the system updates the actor profile with the updated information.

#### **2.2.2.3 Update account information (for the password)**

1. The system requests that the actor goes to his profile and decides to modify his password.
2. The system display a dialog box and asks the actor to enter his actual password and to enter a new one two times to confirm the modification.
3. The actor makes the desired changes to his password.
4. Once the actor updates the password information, the system updates it with the new password.

#### **2.2.2.4 Delete an account**

1. The system requests that the actor goes to his profile and decides to delete the account.
2. The system prompts the actor to confirm the deletion of his account by entering his password thanks to a dialog box.
3. If the actor confirms, the system deletes the account from the system.
4. Then, the actor will be redirected to the home page of the application.

### 2.2.3 Flow for common users

#### 2.2.3.1 Register on the app

The system requests that the actors enter all their information. This includes:

- First name (String)
- Last name (String)
- Mailing address (Email)
- Accommodation address (String)
- Phone number (String)
- Password (String)

1. Once the actor provides the requested information, the system generates and assigns a unique user id number to the actor. The actor is added to the system.

### 2.2.4 Flow for admins

#### 2.2.4.1 Register new admins on the app

The admin has access to a panel that allows him to create new admin accounts.

The system requests that the admin enter information concerning the new admin to register. This includes:

- Mailing address (Email)
- Firstname (String)
- Lastname (String)
- Password (String)

1. Once the admin provides the requested information, the system generates and assigns a unique user id number to the new admin. The new admin is added to the system.
2. All information concerning the new admin are sent to him by email.
2. Then the future admin will have to connect, change his password and add his information via his profile page.

#### 2.2.4.2 Ban a user

The admin has access to a panel that allows him to see the list of all user accounts. If someone's conduct is not good, the admin has the possibility to ban him.

1. The admin accesses the user panel and finds the concerned user thanks to his name.
2. The admin decides to ban him. The system prompts the admin to confirm the ban of the account thanks to a dialog box.
3. If the admin confirms, the system deletes the account from the system and prohibits the use of the user email for new account creation.

### 2.2.5 Alternative Flows

#### 2.2.5.1 User banned

If in the **Register on the app** sub-flow, a black listed common user (banned) tries to register, the system displays an error message.

#### 2.2.5.2 Update Password Cancelled

If in the **Update account information (for the password)** sub-flow, the actor decides not to update his password, the update is cancelled and the **Basic Flow** is re-started at the beginning.

#### 2.2.5.3 Delete Cancelled

If in the **Delete an account** sub-flow, the actor decides not to delete the account, the delete is cancelled and the **Basic Flow** is re-started at the beginning.

## 2.3 Special Requirements

None.

## 2.4 Pre-Conditions

The actors must be logged onto the system before this use case begins except for the **Register on the app** sub-flow.

## 2.5 Post-Conditions

If the use case was successful, the actor's information is added, updated, or deleted from the system. Otherwise, the system state is unchanged.

## 2.6 Extension Points

None.

### 3. Offer

The wireframe illustrates the user interface for managing offers on the Tool Share platform.

- My Offers:** A list of existing offers. Each item shows product name, price, location, category, and actions (See offer, edit, delete).
- Offer:** A detailed view of a specific offer. It includes the title, description, image placeholder, offer category, and state.
- Post an Offer:** A form for creating a new offer. Fields include Offer Title, Description, State (dropdown: none, excellent, good, ok, bad, broken), Category (dropdown: Hand Tools, Power Tools, Fasteners, Gardening Tools, or Add new category), Price, and a file upload area for images.
- Edit Offer:** A form for updating an existing offer. It mirrors the Post an Offer fields but includes the current offer's details (Offer Title, Description, State, Category, Price) for reference.
- Offer Form:** A detailed view of the offer creation form, showing the filled-in fields and three image attachments (image1.png, image2.png, image3.png) ready for validation.

&lt; View Admin &gt;

**Offers**

**Offers**

Post an offer

My reservations My offers Offers Favery History

Categories search Search

Product name: xxxxx Category: xxxxx Hand Tools Power Tools Fasteners Gardening Price: xx € Location date: 01/02/2020 See the offer

Product name: xxxxxxxx Category: xxxxxx Price: xx € Location date: 01/01/2020 See the offer

Product name: xxxxxxxx Category: xxxxxx Price: xx € Location date: 01/12/2019 See the offer

Product name: xxxxxxxx Category: xxxxxx Price: xx € Location date: 01/11/2019 See the offer

**Offers**

Post an offer

Offers Users list

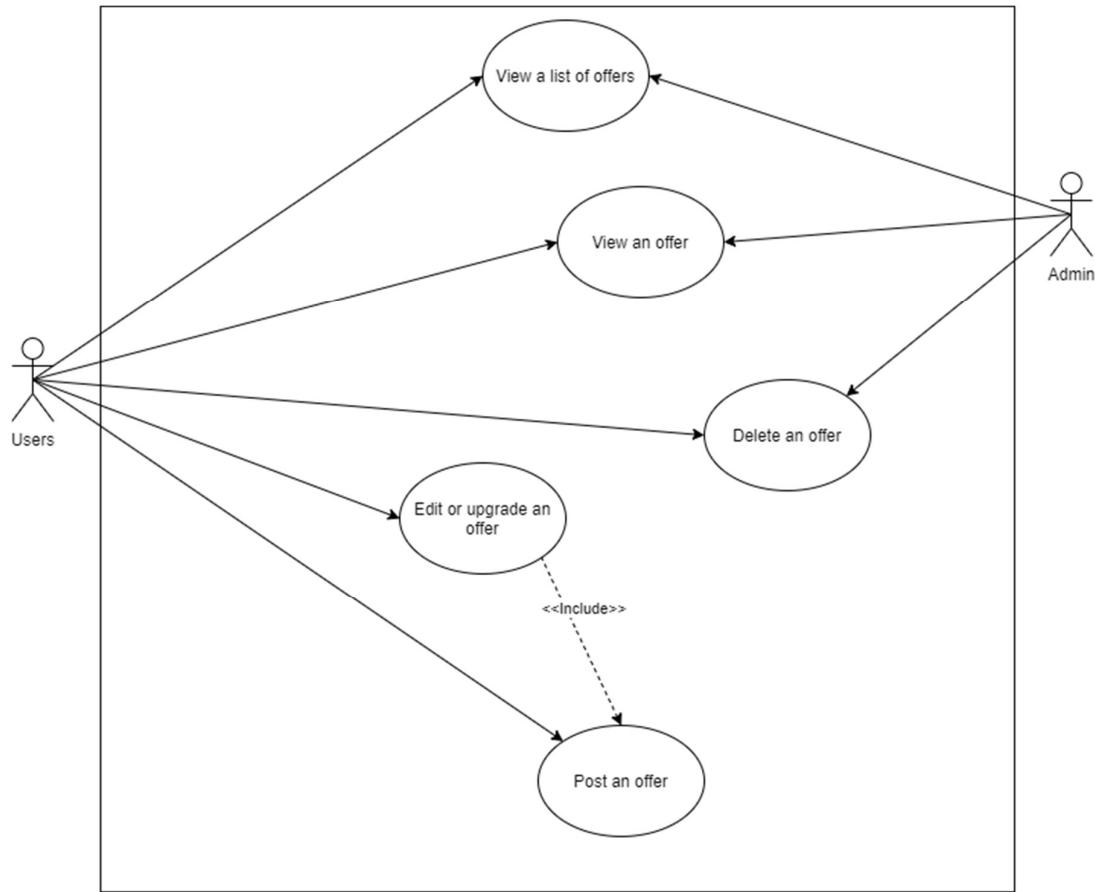
New Category Categories search Button

Product name: xxxxx Category: xxxxx Hand Tools Power Tool Fasteners Gardening Price: xx € Location date: 01/02/2020 See the offer

Product name: xxxxxxxx Category: xxxxxx Price: xx € Location date: 01/01/2020 See the offer

Product name: xxxxxxxx Category: xxxxxx Price: xx € Location date: 01/12/2019 See the offer

Product name: xxxxxxxx Category: xxxxxx Price: xx € Location date: 01/11/2019 See the offer



### 3.1 Brief Description

This use case describes how a user can create, view, update and delete an offer into the Tool Share app. The User will also be able to search and filter the offers to get a list.

This use case features two main actors: the user and the administrator.

### 3.2 Flow of Events

#### 3.2.1 Basic Flow

This use case starts when the user wishes to post a new offer, edit an offer, upgrade an offer to priority, delete an offer published previously or simply want to search / view a list of offers.

1. The system displays a list of offers from all categories and all places with the newest filter (by default).
2. The actor can browse those offers or do a research in order to **view another list of offers, view an offer, view his own offers, delete an offer, post an offer, edit an offer and upgrade an offer.**

#### 3.2.2 Common flow

This flow is for all the Actors.

##### 3.2.2.1 View a list of offers

1. The system requests the actor to select filter or/and categories and/or to type keywords in the search bar.
2. The actor can type keywords in the search bar or/and select one or more existing categories from the drop-down menu category or/and select a filter among: newest, most popular, best, closest.
3. The system displays a list of offers based on the actor entries.
4. The actor can browse the list of offers.

##### 3.2.2.2 View an offer

1. When the system displays a list of offers.
2. The actor can browse the list and click on the one he wishes to see more details.

3. The system displays the chosen offer.
4. The actor has access to all information related to this offer.

### 3.2.2.3 Delete an offer

1. The user can **view his own offers** and click on the “delete offer” button. For the administrator, he can access this button from the **view an offer** flow.
2. The system displays a dialog box to confirm the suppression.
3. The system deletes the offer and displays the actor's offers.

## 3.2.3 Flow for Users

### 3.2.3.1 Post an offer

1. The actor can click on the “add an offer” button.
2. The system display a form and request the actor to fill some entry such as :
  - Product Name (String)
  - Description (text)
  - State (between: excellent, good, ok, bad, broken)
  - Category (between the existing one or ask admin for a new one)
  - Rent or Sale
  - Price (if rent price is for the day)(Integer)
  - Picture (0 to 3)
  - Location (String)
  - Priority Offer (boolean)
3. The actor fills in the information and validates the form to create an offer or a priority offer.
4. The system validates the entries according to the input types, creates a new offer and displays the newly created offer or redirects the actor to the priority offer.

### 3.2.3.2 Edit an offer

1. The actor can **view his own offers** and click on the “edit offer” button.
2. The system displays a form prefilled with all the offer information.
3. The actor can edit the information and update the form.
4. The system updates the offer and displays the updated offer.

### 3.2.3.3 Upgrade an offer

1. The actor can **view his own offers** and click on the “upgrade offer” button.

2. The system confirms the action by displaying a message, the actor is debited an amount of money and the offer is turned into a priority offer for a month. It displays the upgraded offer.

### 3.2.3.5 View my offers

1. The actor can click on the “my offers” button.
2. The system displays the actor’s offers list.

## 3.2.4 Alternative Flows

### 3.2.4.1 Offer not found

If in the **View an offer**, **Edit an offer**, **Upgrade an offer** or **Delete an offer** flows, the offer with the specified id doesn’t exist. Then the system displays an error message with “Offer not found!”. The use case ends.

### 3.2.4.2 No offers found

If in the **View a list of offers** or **View my offers** flows no offers are found. The system displays an error message with “no offers found!”. The use case ends.

## 3.3 Special Requirements

None.

## 3.4 Pre-Conditions

The Tool Share user / admin must be logged onto the system before this use case begins.

## 3.5 Post-Conditions

If the use case was successful, the offer is added, updated, or deleted from the system. Otherwise, the system state is unchanged.

## 3.6 Extension Points

None.

## 4. Priority Offers

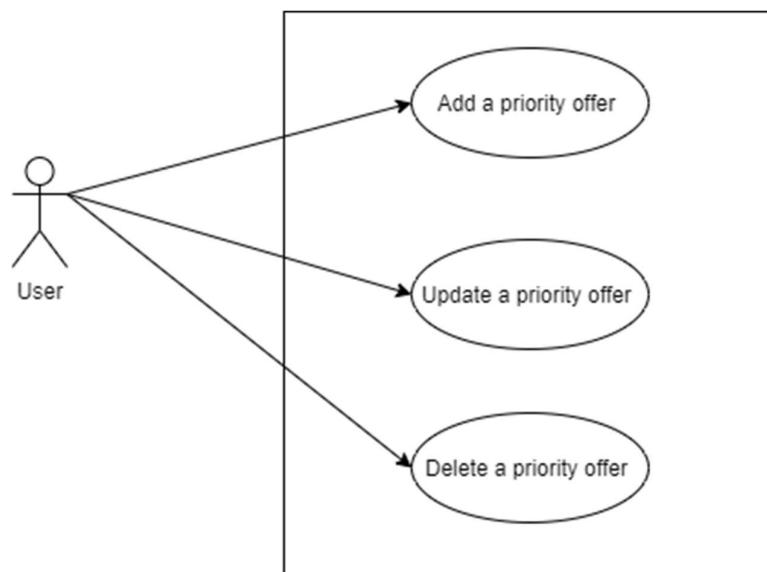
**My offers**

**TOOL SHARE** **Post an offer**

My reservations My offers Offers Favorites History

### MY OFFERS

Product name : xxxxxxxx xxxxxxxx	Price : xx € /per day	Category : xxxxxx	See offer	Edit	Delete
Location : xxxxxxxx					
Product name : xxxxxxxx xxxxxxxx	Price : xx € /per day	Category : xxxxxx	See offer	Edit	Delete
Location : xxxxxxxx					
Product name : xxxxxxxx xxxxxxxx	Price : xx € /per day	Category : xxxxxx	See offer	Edit	Delete
Location : xxxxxxxx					
Product name : xxxxxxxx xxxxxxxx	Price : xx € /per day	Category : xxxxxx	See offer	Edit	Delete
Location : xxxxxxxx					



## 4.1 Brief Description

This use case describes how an user can add, view, update and delete his priority offers into the Tool Share app. It allows his offers to show more often than a regular offer.

## 4.2 Flow of Events

### 4.2.1 Basic Flow (share by user and admin)

This use case starts when the user wishes to post a new offer and select “priority offer” or if he wants to edit an existing offer

1. The actor can browse his offers in order to edit an existing offer and put it in priority or create an offer and select to add it as a “priority offer”

### 4.2.2 Common flow (user & admins)

The flow is dedicated to users.

### 4.2.3 Flow for Users (user)

Whether the user selects a priority offer when he is creating his offer or if he decides to edit an existing offer in order to put it as a priority offer, the flow “**Add Priority Offer**” is executed.

If the user wants to edit his priority offer for another week for exemple, then the flow “**Update Priority Offer**” is executed.

If the user has an existing priority offer that he does not want to view online anymore, he can decide to delete it, then the flow “**Delete Priority Offer**” is executed.

#### 4.2.3.1 Add Priority Offer

1. The user selects for how long he wants to promote his offer.
2. The system displays how much it will cost for this period.

3. The user decides to validate or not the cost for this offer.  
If validated the flow “**Payment**” is executed
4. Once the payment validated, the system adds this offer to the list of priority offers

#### **4.2.3.2 Update Priority Offer**

1. The user selects for how long he wants to promote his existing priority offer.
2. The system displays how much it will cost for this period.
3. The user decides to validate or not the cost for this offer.  
If validated the flow “**Payment**” is executed
4. Once the payment validated, the system update this offer

#### **4.2.3.3 Delete Priority Offer**

1. The user selects which offer he would like to delete.
2. The system asks for validation.
3. If the user validates, both the priority offer and the offer are deleted from the system.

#### **4.2.4 Flow for Admins (admin)**

None

#### **4.2.5 Alternative Flows**

##### **4.2.5.1 Add or Update or Delete Cancelled**

If the user decides to not validate when asked, the action is then aborted and the flow ends.

### **4.3 Special Requirements**

None.

### **4.4 Pre-Conditions**

The Tool Share user must be logged onto the system before this use case begins.

## 4.5 Post-Conditions

If the use case was successful, the priority offer is added, updated, or deleted from the system. Otherwise, the system state is unchanged.

## 4.6 Extension Points

None.

## 5. Evaluation system

**History**

**Rate**

**HISTORY**

Product name : xxxxxxxx xxxxxxxx      Price paid: xx €  
Category : xxxxxx      Location date : 01/02/2020      Rate      See offerer     

Product name : xxxxxxxx xxxxxxxx      Price paid: xx €  
Category : xxxxxx      Location date : 01/01/2020      Rate      See offerer     

Product name : xxxxxxxx xxxxxxxx      Price paid: xx €  
Category : xxxxxx      Location date : 01/12/2019      Rate      See offerer     

Product name : xxxxxxxx xxxxxxxx      Price paid: xx €  
Category : xxxxxx      Location date : 01/11/2019      Rate      See offerer     

**Offer Title**

Rate offer : ★★☆★☆

**Validate**

**Payment**

**Offer**

**Offer Title**

Do you want to let a comment ?      Yes      No

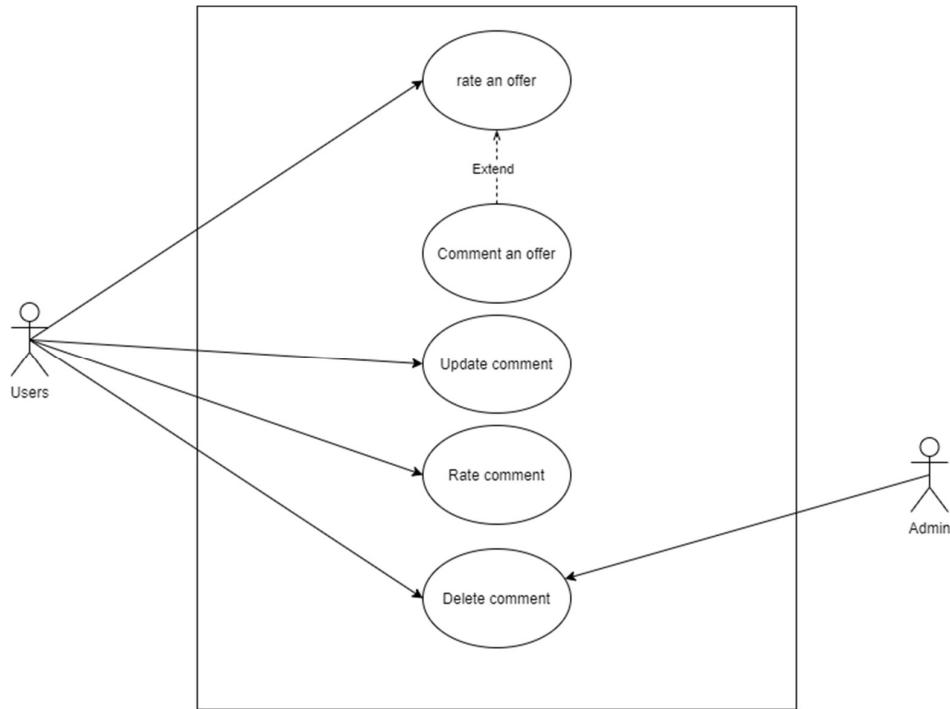
Good offer !

**Validate**

To rent/to sale : Offer Title      xx €

**Offer Comments**

25	3
10	5
0	0



## 5.1 Brief Description

In this use case there are two types of actors: simple users and administrators. The admins have additional rights.

This use case allows a user to rate an offer and comment (after having rate). This includes adding a rate and the possibility of creating, changing and deleting a comment. They can also like or dislike comment

## 5.2 Flow of Events

### 5.2.1 Basic Flow

This use case starts when an user wants to add a note/comment on an offer (an offer he has already booked or he is using.) or update, delete a comment. If the actor is a common user, he can rate the offer, giving the possibility to comment and delete comments he posted (**Common flow** and **Flow for users**). Admin can only update and delete comments (**Common flow**).

## 5.2.2 Common flow

### 5.2.2.1 Delete.

1. The actor chose to delete a comment of an offer.
2. The system requires a validation, if the admin answer yes the comment is delete

### 5.2.2.2 Update.

1. The actor chose to update a comment of an offer. He write the new comment
2. The system requires a validation, if the admin answer yes the comment is update

## 5.2.3 Flow for users

1. The user wants to note the offer (From zero to five stars), he goes on his history to rate an offer (follow **Rate offer** flow). If he wants to like/dislike a comment , he goes on page of the offer.( Follow **Rate comment** flow)

### 5.2.3.1 Rate offer

1. Once the user has rated the offer, the system requests if he wants to comment.
2. If the actor wants to give comment the subflow **Write Comment** is executed, else the evaluation is saved and the flow ends.

### 5.2.3.2 Write comment

1. The user writes a comment and can post
2. The system asks for validation ( yes or no)
3. If he says yes for the post : the comment is uploaded with the evaluation and the flow ends, else he can rewrite the comment.

### 5.2.3.3 Rate comment

1. When the system displays an offer. The actor can scroll down to the comments section in order to view the comments related to this offer. Then he can click on like or dislike.
2. The system validates the actor vote and updates the question/answer score.

### 5.2.4 Alternative Flows

#### 5.2.4.1 Rate canceled

If in the flow user, the user decides not to upload his rate he come-back on the menu.

#### 5.2.4.1 Comment canceled

If in the sub flow write comment, the user decides not to upload his comment he goes back to the flow user.

#### 5.2.4.3 Comment not found

If the actor wants to delete a comment but when he valids the comment is not found the system displays an error message with “no offers found !”. The use case ends.

## 5.3 Special requirements

None

## 5.4 Pre-Conditions

The Actor must be logged onto the system before this use case begins.

To execute the common flow the Actor must be logged with an admin account or a user account.

To execute the user flow the Actor must be logged with an user account and not admin account.

## 5.5 Post-Conditions

If the use case was successful, the rate/comment are now added, deleted and visible into the system. If not, the system state is unchanged.

## 5.6 Extension Points

None.

## 6. FAQ

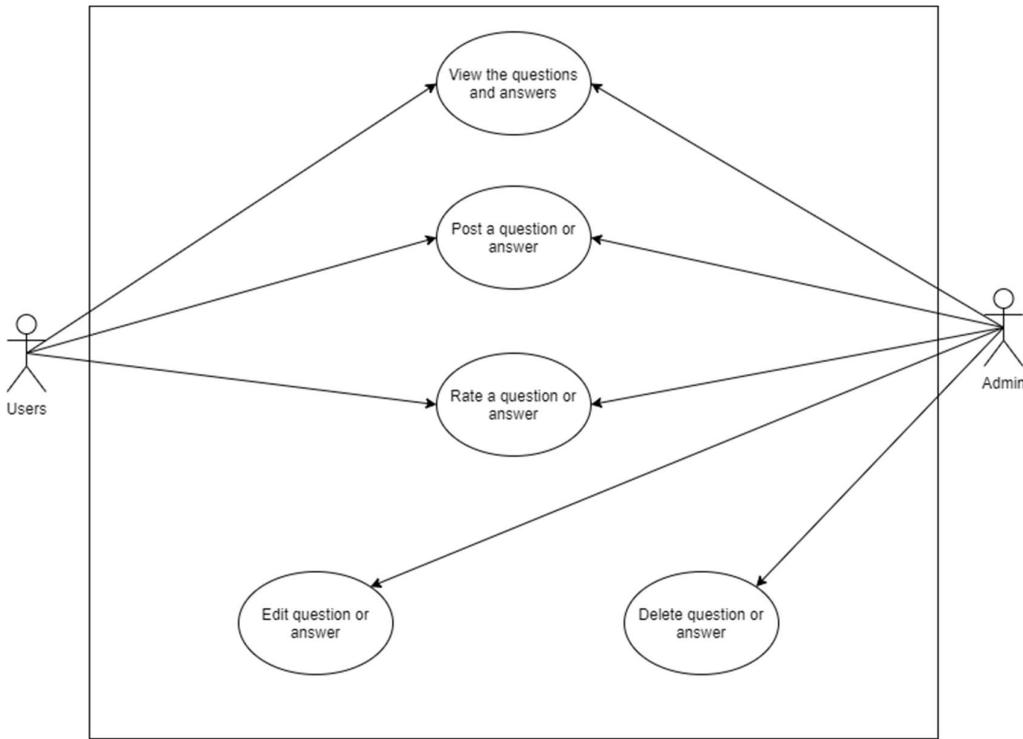


**Post question**

This screenshot shows the 'Post question' form. It includes the 'TOOL SHARE' logo, a 'Post an offer' button, and a navigation bar with 'My reservations', 'My offers', 'Offers', 'Favory', and 'History'. Below the navigation is a header bar with the text 'To rent/to sale : Offer Title' and 'xx €'. The main content area is a large text input field labeled 'Question ?' with a 'Validate' button at the bottom.

**Post answer**

This screenshot shows the 'Post answer' form. It includes the 'TOOL SHARE' logo, a 'Post an offer' button, and a navigation bar with 'My reservations', 'My offers', 'Offers', 'Favory', and 'History'. Below the navigation is a header bar with the text 'To rent/to sale : Offer Title' and 'xx €'. The main content area is a large text input field labeled 'Answer' with a 'Validate' button at the bottom.



## 6.1 Brief Description

This use case describes how an actor can create, view, update, delete and rate questions into the Tool Share app. The actor will be able to answer those questions and other actors will also be able to rate the answer.

This use case features two main actors: the user and the administrator.

## 6.2 Flow of Events

### 6.2.1 Basic Flow

This use case starts when the actor wishes to post a new question/answer, edit, delete, rate or simply want to view a question/answer related to an offer.

1. When the system displays a list of offers.
2. The actor can browse the list and click on the one he wishes to see more details.
3. The system displays the chosen offer.

4. The actor has access to all information related to this offer. He can **view the questions/answer, post a question/answer, rate a question/answer, delete a question/answer and update a question/answer**.

## 6.2.2 Common flow

### 6.2.2.1 View the questions/answers

1. When the system displays an offer.
2. The actor can scroll down to the FAQ section in order to view the questions/answers related to this offer.

### 6.2.2.2 Post a question/answer

1. When the system displays an offer.
2. The actor can scroll down to the FAQ section in order to view the questions/answers related to this offer. Then he can click on the “ask a question” button or “answer this question”.
3. The system displays a form and requests the actor to fill some entries such as:
  - o Question (or answer if he clicked “answer this question”)
4. The actor fills in the information and validates the form.
5. The system validates the data then displays the FAQ section of the offer to the actor to see the newly created question/answer.

### 6.2.2.3 Rate a question/answer

1. When the system displays an offer.
2. The actor can scroll down to the FAQ section in order to view the questions/answers related to this offer. Then he can click on ↑ or ↓. Those buttons are present next to every question/answer.
3. The system validates the actor vote and updates the question/answer score.

## 6.2.4 Flow for Admins

### 6.2.4.1. Edit a question/answer

1. When the system displays an offer.
2. The actor can scroll down to the FAQ section in order to view the questions/answers related to this offer. Then he can click “edit this question” or “edit this answer” in order to correct any spelling mistake or bad language.

3. The system displays a form prefilled with the old information and requests the actor to edit some entries such as:
  - o Question (or answer if he clicked “edit this answer”)
4. The actor can edit the information and validate the form.
5. The system validates the entries, updates the question/answer and displays the FAQ section related to the offer.

#### 6.2.2.4 Delete a question/answer

1. When the system displays an offer.
2. The actor can scroll down to the FAQ section in order to view the questions/answers related to this offer. Then he can click “delete this question” or “delete this answer”.
3. The system displays a warning message.
4. The actor validates the suppression.
5. The system validates the suppression, deletes the question/answer and displays the offer’s FAQ section. If the deleted question had any answers they are deleted as well.

### 6.2.5 Alternative Flows

#### 6.2.5.1 Question/answer not found

If in the **rate a question/answer, edit a question/answer or delete a question/answer** flows, the question/answer with the specified id doesn’t exist. Then the system displays an error message with “Question (or Answer) not found !”. The use case ends.

#### 6.2.5.2 No Questions/answers found

If in the **view the questions/answers** flow no questions are found. The system displays an error message with “There is no question... Ask one !”. The use case ends.

And if in the **view the questions/answers** flow no answers are found for a given question. The system displays an error message with “There is no answer to that question yet !”. The use case ends.

#### 6.2.5.3 Questions/answers creation denied

If in the **post a question/answer** flow the actor posting a question is the one who created the offer. The system displays an error message with “You can’t ask questions to yourself !”. The use case ends.

And if in the **post a question/answer** flow the actor posting an answer is not the one who created the offer or not an administrator. The system displays an error message with “You can’t answer questions !”. The use case ends.

## 6.3 Special Requirements

None.

## 6.4 Pre-Conditions

The Actor must be logged onto the system before this use case begins.

To execute the common flow the Actor must be logged with an admin account or a user account.

To execute the Admin flow the Actor must be logged with an admin account and not user account.

## 6.5 Post-Conditions

If the use case was successful, the question/answer is added, updated, rated, or deleted from the system. Otherwise, the system state is unchanged.

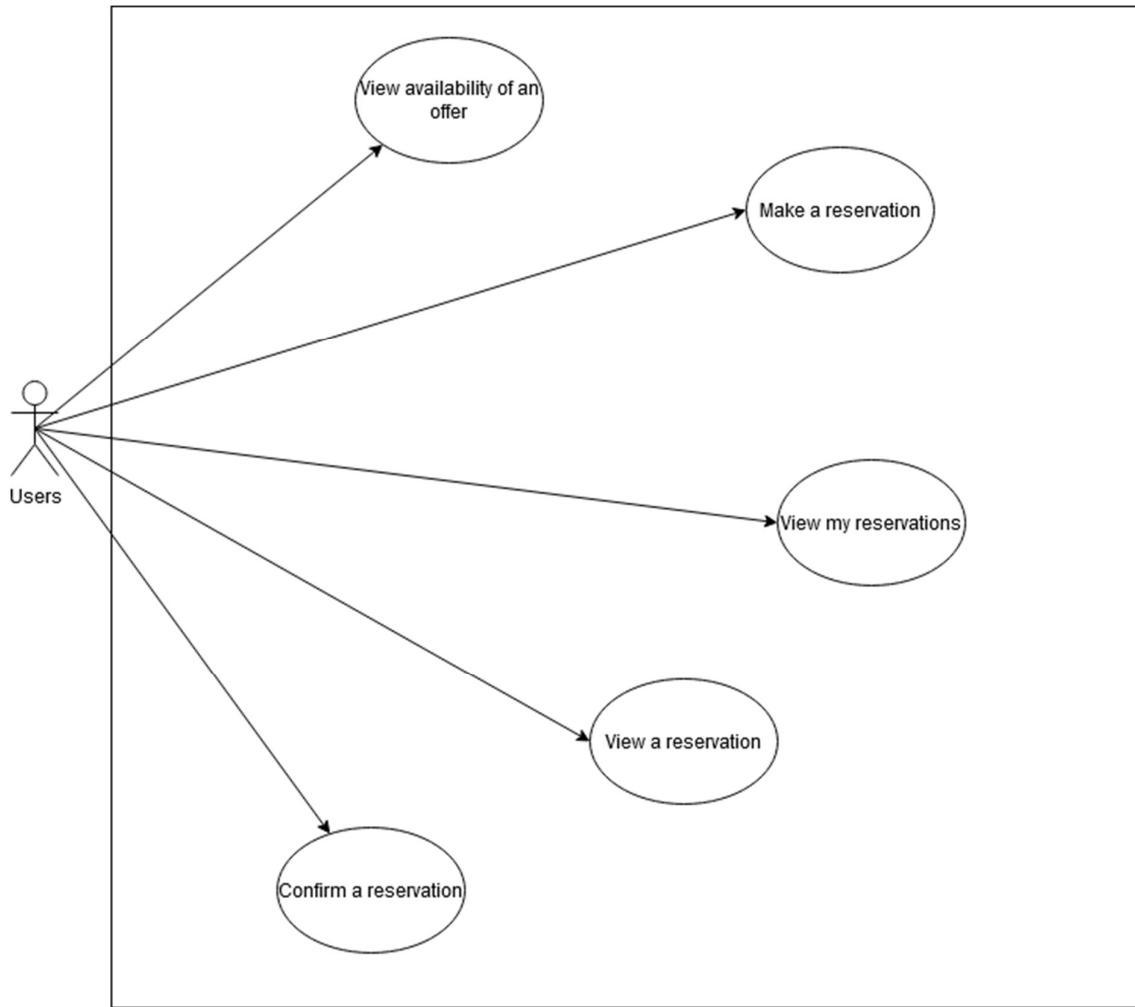
## 6.6 Extension Points

None.

## 7. RESERVATION

<p><b>Reservation</b></p> <p>To rent/to sale : Offer Title xx €</p> <p><b>Offer Reservation</b></p> <p>NOVEMBER 2020</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>S</td><td>M</td><td>T</td><td>W</td><td>T</td><td>F</td><td>S</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td></tr> <tr><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td></tr> <tr><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td></tr> <tr><td>29</td><td>30</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr> </table> <p><a href="#">Book it now!</a></p>	S	M	T	W	T	F	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	<p><b>Post reservation</b></p> <p>To rent/to sale : Offer Title xx €</p> <p>2020</p> <p>Wed Nov 18</p> <p>NOVEMBER 2020</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>S</td><td>M</td><td>T</td><td>W</td><td>T</td><td>F</td><td>S</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td></tr> <tr><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td></tr> <tr><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td></tr> <tr><td>29</td><td>30</td><td></td><td></td><td></td><td></td><td></td></tr> </table> <p><a href="#">CANCEL</a> <a href="#">OK</a></p>	S	M	T	W	T	F	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30					
S	M	T	W	T	F	S																																																																																						
1	2	3	4	5	6	7																																																																																						
8	9	10	11	12	13	14																																																																																						
15	16	17	18	19	20	21																																																																																						
22	23	24	25	26	27	28																																																																																						
29	30	1	2	3	4	5																																																																																						
6	7	8	9	10	11	12																																																																																						
S	M	T	W	T	F	S																																																																																						
1	2	3	4	5	6	7																																																																																						
8	9	10	11	12	13	14																																																																																						
15	16	17	18	19	20	21																																																																																						
22	23	24	25	26	27	28																																																																																						
29	30																																																																																											

<p><b>Post reservation</b></p> <p><b>MY RESERVATIONS</b></p> <p>Sent Received</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td><td></td></tr> </table>	Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>			Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>			Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>			Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>			Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>			<p><b>Post reservation</b></p> <p><b>MY RESERVATIONS</b></p> <p>Sent Received</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td></tr> <tr><td>Product name : xxxxxxxx xxxxxxxx</td><td>Price : xx €</td><td>Start date : xx/xx/xxxx</td><td>End date : xx/xx/xxxx</td><td><a href="#">See offer</a></td><td></td></tr> </table>	Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>		Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>		Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>		Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>		Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>		Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>	
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				
Product name : xxxxxxxx xxxxxxxx	Price : xx €	Start date : xx/xx/xxxx	End date : xx/xx/xxxx	<a href="#">See offer</a>																																																																				



## 7.1 Brief Description

This use case describes how an actor can book an offer, view his reservations, cancel one and view the availability of an offer into the Tool Share app. The base unit for a reservation is a day, you can't book an offer for less than a day and if you want it for 2,5 days, you will have to book it for 3 days. although there is no limit.

This use case features two main actors: the user and the administrator.

## 7.2 Flow of Events

### 7.2.1 Basic Flow

This use case starts when the actor wishes to make or cancel a reservation, to view his reservations or the availability of an offer.

1. When the system displays an offer.
2. The actor can scroll down to the booking section of the offer. From there he can **view the availability of the offer** and **make a reservation** for this offer.  
or
2. The actor can click on the “My reservations” button in order to **view my reservations**, **view a reservation**, **confirm a reservation** and **cancel a reservation**.

### 7.2.2 Flow for Users

#### 7.2.2.1 View the availability of the offer (All the reservations for a given offer)

1. When the actor is in the booking section of an offer.
2. The system displays a calendar of the current month. The shaded days represent the offer being already booked on that day.
3. The actor can browse the months, years and days of the calendar.

#### 7.2.2.2 Make a reservation

1. When the actor is in the booking section of an offer, he can click the “book this offer” button.
2. The system displays a form in the shape of a calendar of the current month. The shaded days represent the offer being already booked on that day.
3. The actor can browse the months, years and days of the calendar. He has to select two dates: the beginning and the ending of the reservation. Then he validates the form.
4. The system registers the reservation and sends a notification to the offer owner. It redirects the user to the use case payment with all the information.

#### 7.2.2.3 View my reservations (The one I received AND I made)

1. When the actor clicks on the “My reservations” button.

2. The system displays the list of reservations of the actor. By default, it shows the reservations he made; to see the reservation he got he must toggle the “Received” button.
3. The actor can choose to display only a certain type of reservation thanks to checkboxes:
  - cancelled
  - done
  - ongoing
  - upcoming
4. The system updates the display to show the right reservations.

#### 7.2.2.4 View a reservation

1. When the user is doing the **view my reservations** flow.
2. He can click on any reservations in order to have more information.
3. The system displays the given reservation.

#### 7.2.2.5 Confirm a reservation

1. When the actor is doing the **view my reservations** flow, if the “Received” button is toggled.
2. He can click on the “Confirm reservation?” button which appears next to every upcoming reservation.
3. The system displays a form with two buttons: Validate or Reject.
4. The actor validates the reservation or not.
5. The system notifies the user who made the reservation of the confirmation or not by the offer owner.

This flow can also be executed from the **view a reservation** flow, if the reservation is upcoming.

#### 7.2.2.6 Cancel a reservation

1. When the actor is doing the **view my reservations** flow, if the “Received” button is not toggled.
2. He can click on the “Cancel reservation” button which appears next to every upcoming reservation.
3. The system displays a warning message.
4. The actor enters a reason for the cancellation and validates it.
5. The system notifies the offer owner of the cancellation of the reservation by the user.

This flow can also be executed from the **view a reservation** flow, if the reservation is upcoming.

### 7.2.3 Alternative Flows

#### 7.2.3.1 Reservation not found

If in the **view a reservation**, **confirm a reservation** and **cancel a reservation** flows, the reservation with the specified id doesn't exist. Then the system displays an error message with "Reservation not found ! ". The use case ends.

#### 7.2.3.2 No reservations found

If in the **view the availability of the offer** or **view my reservations** flows no reservation are found . The system displays an error message with "no reservations found !". The use case ends.

## 7.3 Special Requirements

None.

## 7.4 Pre-Conditions

The Actor must be logged onto the system before this use case begins.

To execute the common flow the Actor must be logged with an admin account or a user account.

To execute the Admin flow the Actor must be logged with an admin account and not user account.

## 7.5 Post-Conditions

If the use case was successful, the question/answer is added, updated, rated, or deleted from the system. Otherwise, the system state is unchanged.

## 7.6 Extension Points

None.

## 8. Payment

Payment

**TOOL SHARE**  [Post an offer](#)

My reservations  My offers  Offers  Favery  History 

**Payment**

You have to pay xx€ for ... click

Choose your payment method

-----  
Paypal  
CB

Validate

Payment

**TOOL SHARE**  [Post an offer](#)

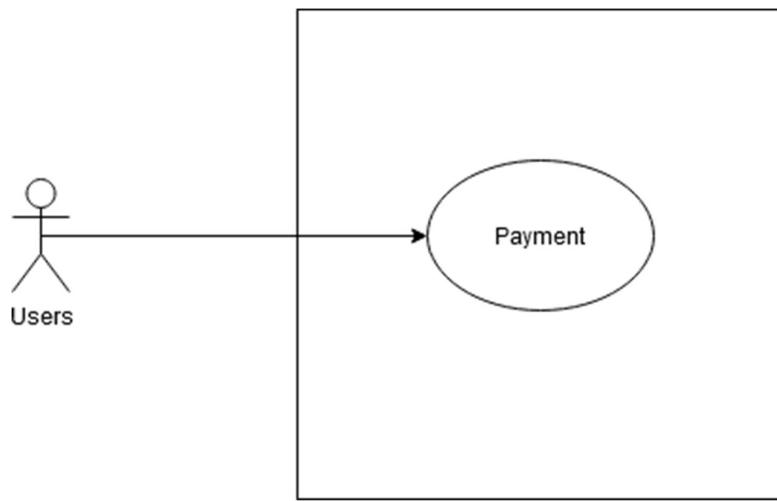
My reservations  My offers  Offers  Favery  History 

**Paypal**

account

password

Connect



## 8.1 Brief Description

This use case concerns only the users.

Is used to pay for rental fees, from the purchase of an offer to the reservation and if necessary, to pay additional fees if the object has been returned broken or late. It is also used to pay if the user wants to post a sponsored offer. He only deals with the moment he has to pay, not with the conditions and requests of the user.

## 8.2 Flow of Events

### 8.2.1 Basic Flow (share by user and admin)

This use case starts when a user needs to pay for a reservation, or if he has been returned the offer broken or late and the fees for sponsored offers. He has multiple choices to choose the method of payment

### 8.2.2 Common flow (user & admins)

None

### 8.2.3 Flow for Users (user)

1. The system show to the user how much he needs to pay with a summary of what he's paying for and ask him how he will pay

2. The user pays with the support he wants. The system redirects him and performs actions as appropriate (**Reservation payment, Broken/late payment or sponsored offer.**).

#### **8.2.3.1 Reservation payment**

If the user pays an offer, the reservation flow is succeeded and he is redirected on a summary of his command.

#### **8.2.3.2 Broken/late payment**

If the user pays for a problem the state of his reservation ends and he is redirected on the list of offers.

#### **8.2.3.3 Sponsored offer**

If the user pays for an additional fee on an offer, the offer is add/update with the status "sponsored". The user is then redirected to his offer.

### **8.2.3 Alternative Flows**

#### **8.2.3.1 Payment refused**

If the method of payment blocks the transaction, the user is redirected at the beginning of the **payment flow**.

## **8.3 Special requirements**

None

## **8.4 Pre-Conditions**

The user must be logged onto the system before this use case begins.

## **8.4 Post-Conditions**

If the use case was successful, the payment is saved.

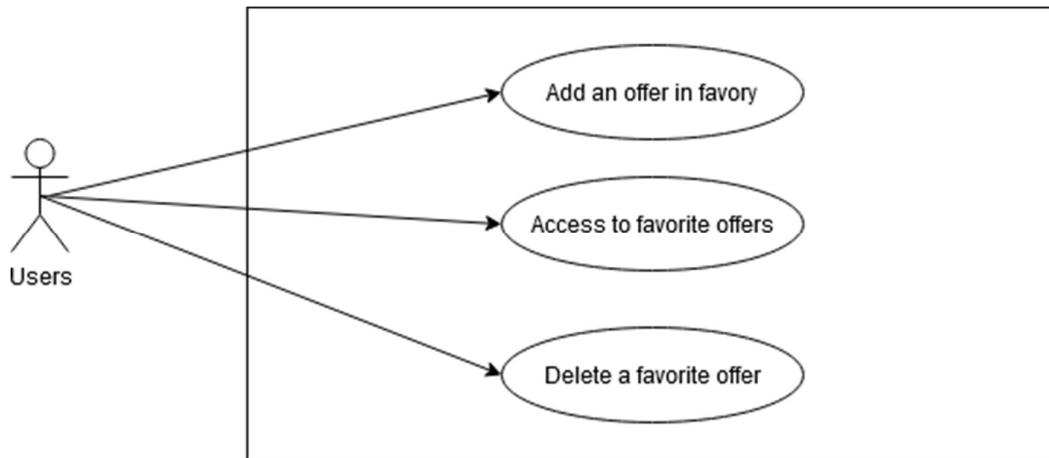
## 8.6 Extension Points

None.

## 9. Favorite System

The screenshot shows the 'FAVORITE OFFERS' section of the Tool Share application. At the top, there is a header bar with the 'TOOL SHARE' logo, a 'Post an offer' button, and navigation links for 'My reservations', 'My offers', 'Offers', 'Favori', 'History', and a user profile icon. Below the header, the title 'FAVORITE OFFERS' is centered. The main content area displays four favorite offers, each with a product name, location, price, category, a 'See offer' button, and a red trash can icon for deletion.

Product name	Location	Price	Category	Action
xxxxxxxx xxxxxxxx	xxxxxxxx	xx € /per day	xxxxxx	<a href="#">See offer</a>
xxxxxxxx xxxxxxxx	xxxxxxxx	xx € /per day	xxxxxx	<a href="#">See offer</a>
xxxxxxxx xxxxxxxx	xxxxxxxx	xx € /per day	xxxxxx	<a href="#">See offer</a>
xxxxxxxx xxxxxxxx	xxxxxxxx	xx € /per day	xxxxxx	<a href="#">See offer</a>



### 9.1 Brief Description

In this use case there is only one type of actors: common users (that we'll name only users).

This use case allows the users to have access to their favorite offers that they found previously. This use case includes adding, accessing and deleting favorite offers.

## 9.2 Flow of Events

### 9.2.1 Basic Flow

This use case starts when a user wants to access the offers that interest him and which he puts in favorite. These offers can possibly interest them or users have already rented the tool concerned by the offer.

The user has access to a panel that allows him to see all the list of his favorite offers.

#### 9.2.1.1 Add an offer in favorite

When a user looks at the offers, he has the possibility to add an offer in favorite. This includes adding the offer to the favorite items panel.

1. The user requests the system to add the wanted offer in favorite.
2. The system adds the offer in favorite.
3. The user has access to all information concerning the offer in the favorite offers panel.

#### 9.2.1.2 Access to favorite offers

1. The user requests the system to see all his favorite offers by going to the pane provided for this functionality.
2. The system retrieves and displays the concerned information.
3. The user has access to all information concerning each offer he puts in favorite (type of tool, owner information, location, price).

#### 9.2.1.3 Delete a favorite offer

1. To delete one of the favorite offers, the user has to access concerning information first. This includes any of the information specified in the **Access to favorite offers** sub-flow.

2. The user requests the system to delete the wanted offer.
3. The system prompts the actor to confirm the deletion thanks to a dialog box.
4. If the user confirms, the system deletes the favorite offer's information from the system.

### 9.2.3 Alternative Flows

#### 9.2.3.1 Delete Cancelled

If in the **Delete a favorite offers** sub-flow, the user decides not to delete the concerning offer, the delete is cancelled and the **Basic Flow** is re-started at the beginning.

## 9.3 Special Requirements

None.

## 9.4 Pre-Conditions

The user must be logged onto the system before this use case begins.

## 9.5 Post-Conditions

If the use case was successful, the information about the user's favorite offer is deleted from the system. Otherwise, the status of the system remains unchanged.

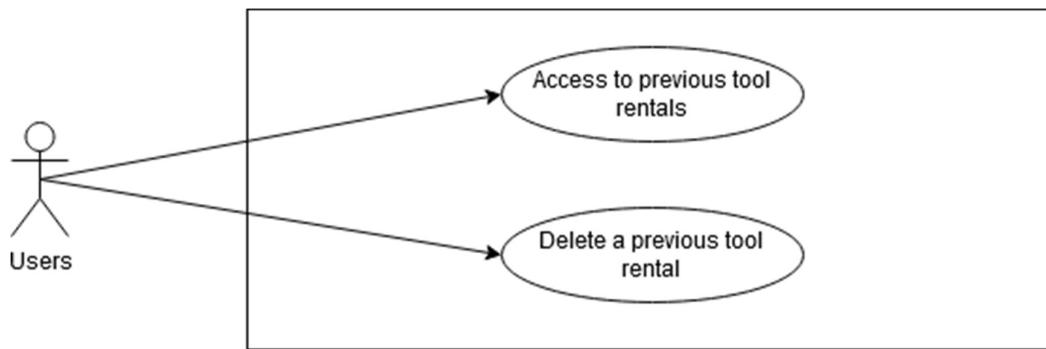
## 9.6 Extension Points

None.

# 10. History System

The screenshot shows the 'History' section of the Tool Share application. At the top, there is a navigation bar with the 'TOOL SHARE' logo, a 'Post an offer' button, and links for 'My reservations', 'My offers', 'Offers', 'Favory', 'History', and a user profile icon. Below the navigation bar, the word 'HISTORY' is centered. The main content area displays a list of four previous tool rentals, each with the following information:

Product name :	xxxxxx xxxx	Price paid:	xx €	Location date :	01/02/2020	Rate	See offerer	Delete
Category :	xxxxxx							
Product name :	xxxxxx xxxx	Price paid:	xx €	Location date :	01/01/2020	Rate	See offerer	Delete
Category :	xxxxxx							
Product name :	xxxxxx xxxx	Price paid:	xx €	Location date :	01/12/2019	Rate	See offerer	Delete
Category :	xxxxxx							
Product name :	xxxxxx xxxx	Price paid:	xx €	Location date :	01/11/2019	Rate	See offerer	Delete
Category :	xxxxxx							



## 10.1 Brief Description

In this use case there is only one type of actors: common users (that we'll name only users).

This use case allows the users to have access to their previous tool rentals. This includes accessing and deleting old rentals.

## 10.2 Flow of Events

### 10.2.1 Basic Flow

This use case starts when an user wants to access his previous tool locations to maybe rent the same tool again, or maybe contact the tool owner to talk about the state of the tool, or to delete a previous rental which is no longer useful to him.

The user has access to a panel that allows him to see all the list of his old tool location.

#### 10.2.1.1 Access to previous tool rentals

1. The user requests the system to see old tool locations by going to the pane provided for this functionality.
2. The system retrieves and displays the concerned information.
3. The user has access to all information concerning each old location (type of tool, owner information, date of rental, location, price).

#### 10.2.1.2 Delete a previous rental

1. To delete previous location information, the user has to access concerning information first. This includes any of the information specified in the **Access to previous tool rentals** sub-flow.
2. The user requests the system to delete the wanted tool rental.
3. The system prompts the actor to confirm the deletion thanks to a dialog box.
3. If the user confirms, the system deletes the rental information from the system.

### 10.2.3 Alternative Flows

#### 10.2.3.1 Delete Cancelled

If in the **Delete a previous rental** sub-flow, the user decides not to delete the concerning rental, the delete is cancelled and the **Basic Flow** is re-started at the beginning.

## 10.3 Special Requirements

None.

## 10.4 Pre-Conditions

The user must be logged onto the system before this use case begins.

## 10.5 Post-Conditions

If the use case was successful, the information about the user's previous tool location is deleted from the system. Otherwise, the status of the system remains unchanged.

## 10.6 Extension Points

None.

# 11. Category

&lt; View Admin &gt;

New Category

TOOL SHARE [Post an offer](#)

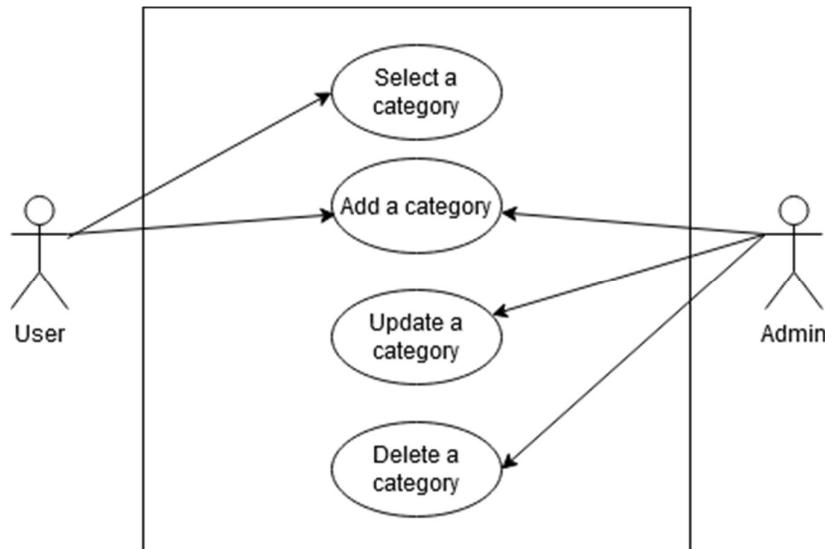
Add a new Category

Category

- Proposed category [Delete](#)
- Another One [Delete](#)
- And a third [Delete](#)

Name

Validate



## 11.1 Brief Description

This use case allows an user to select categories when posting an offer, but he can also propose a category which needs to be validated by the admin. The user can also filter its research by selecting a category of item.

The admin is able to manage categories.

## 11.2 Flow of Events

### 11.2.1 Basic Flow (share by user and admin)

If an actor is an user wishes to select categories while creating his offer, then “**Flow for Users**” is executed.

If the actor is an admin then he can *add, modify, delete* categories. the “**Flow for Admins**” is executed.

### 11.2.2 Flow for Users (user)

This use case starts when the actor wishes to select categories when posting an offer or when he wants to use the search bar.

1. The system requests that the actor selects given categories

If the actor was using the search bar, then the use case of **Offers** is executed.

else the flow moves to the second state.

If the actor wants to “Add a category” which is not in the list, the “**Add a category**” sub flow is executed.

2. The actor validates the categories chosen.
3. The system validates the chosen categories when the actor validates his offer.

### 11.2.2.1 Add a category

This use case starts when the user wishes to add a new category

1. The system requests that the actor enter a name of category
2. The actor validates the value entered.
3. The system enters the new category which needs to be validated by the admin in order to be used.

### 11.2.3 Flow for Admins (admin)

This use case starts when the admin wishes to *add, modify, delete* any category.

1. The system requests that the actor specify which function he would like to use.
  - If the Admin selected “Add a Category”, the “**Add a category**” sub flow is executed.
  - If the Admin selected “Edit a Category”, the “**Edit a category**” sub flow is executed.
  - If the Admin selected “Delete a Category”, the “**Delete a category**” sub flow is executed

#### 1.2.3.1. Add a category

The system requests that the admin enters the category name.

1. Once provided and validated, the system checks if the given name is not already listed.
2. The system adds the new category to the list.

#### 11.2.3.2. Edit a category

The system requests that the admin select the category he would like to update.

1. Once selected and validated, the admin rewrites the name of the category.
2. The system checks if the new entered value is not already existing.
3. When validated, the system updates the selected category

#### 11.2.3.3. Delete a category

The system requests that the admin select the category he would like to delete.

1. Once selected and validated, the system asks for validation (yes or no).
2. If the admin selects yes, the system deletes the selected category

### 11.2.3 Alternative Flows

#### 11.2.3.1 No Category selected

If the user does not select any category while posting his offers, the system displays the error "You need to select at least one category."

#### 11.2.3.2 Category Already Existing

If the actor wants to add an already existing category, then the system displays an error message with "This category already exists."

#### 11.2.3.2 Update Cancelled

If the admin decides not to edit the selected category, the system sends the admin back to the **Basic Admin Flow**.

#### 11.2.3.2 Delete Cancelled

If the admin decides not to delete the selected category, the system sends the admin back to the **Basic Admin Flow**.

## 11.3 Special Requirements

None.

## 11.4 Pre-Conditions

The Actor must be logged onto the system before this use case begins.

To execute the common flow the Actor must be logged with an admin account or a user account.

To execute the user, flow the Actor must be logged with a user account and not admin account.

## 11.5 Post-Conditions

If the use case was successful, the category is now added, deleted and visible into the system. If not, the system state is unchanged.

## 11.6 Extension Points

None.

# Annex

Bryan is the responsible of the use cases

## 1. Work Distribution

Bryan: Priority Offers, Category,

Rémy: Payment, Evaluation System,

Solène: User, History System, Favery System

Valentin: Q&A, Reservation, Offer,

## 2. Facts

In order to split the use cases, we made a chat call on Discord Thursday and we decided who wanted what. The repartition was equal to everybody.

Our group started with first the description of the use case which lasted, then the diagram which lasted both 3days (Friday – Sunday).

We did a meeting in order to decide how we will design the app on Monday. For the mock-ups, we used Balsamiq and it lasted 2 days (Monday-Tuesday)

During the last two days we started the final report, regrouping all the use cases description. Everyone was checking and correcting issues he was seeing in order to have this last version.

## 3. Interpretation

Our Group thinks that there was an efficient way of working together. It first started with a meeting in order to split the use cases. Once it was done, we started on our own to work our descriptions. The main downside of this week was the time we spent on the descriptions of the use cases. It was not clear how it was supposed to be detailed so our group decided to do a model and the apply it to all use cases.

There was not one difficult use case among the others, it was just the time management that was tricky. When the description and the diagrams was finished, a meeting on discord was planned in order to plan how the app will be designed. Directly after that, the mock-ups started to appear, and everything was rolling.

Our group dedicated the last two days to fix any issues that could be seen. The report was edited as soon as one pointed an issue.