

# General Introduction

Leaving home for college or a new adventure is a big step towards independence whether you're a Tunisian student far from family or just living on your own for the first time, it's an exciting but challenging time, especially when it comes to cooking being away from home can leave you feeling a bit lost, especially if you're not sure how to cook.

That's where "Taste & Share" comes in! we understand the struggles of being alone and away from the comfort of home, which is why we've gathered simple and healthy recipes to make cooking easier for you as a student, you're often juggling classes, assignments, and social activities, leaving little time and energy for cooking elaborate meals we get it you need meals that are quick, affordable, and nutritious, so you can stay focused and energized throughout the day , our platform is designed with your needs in mind, offering a variety of easy-to-follow recipes that can be prepared with minimal ingredients and time plus, by cooking at home instead of constantly ordering takeout or dining out, you can save money in the long run while also enjoying healthier meals.

We know that living away from home can sometimes leave you feeling homesick and longing for familiar flavors, that's why our platform offers a diverse array of recipes to suit every palate and culinary skill level. From simple one-pot wonders to comforting dishes that remind you of home-cooked meals, we've got something for everyone.

And with our friendly community of fellow students and food enthusiasts, you can connect with others who are going through the same experience as you. You can find, save, and share recipes with friends easily, making the cooking experience even more enjoyable. So whether you're craving a taste of home or ready to try something new, "Taste & Share" is here to make your cooking journey fun, delicious, and budget-friendly, wherever you are.

In our project, we implemented Scrum, an agile framework for managing work. Scrum breaks down tasks into short sprints, typically lasting one to four weeks, fostering collaboration and adaptability. With daily stand-up meetings, sprint planning, reviews, and retrospectives, Scrum promotes transparency and continuous improvement, enabling teams to deliver value more efficiently.

# CHAPTER 1

## Requirements Specification

### 1.1 Introduction

In this part of our chapter, we'll dig into the details of what our project needs. This means figuring out what it should do (functional requirements), how it should perform (non-functional requirements), and who will use it (actors). Then, we'll create a simple diagram (the global use case diagram) to show how everything fits together. This diagram will help us understand how different parts of our project interact. Finally, we'll talk about our backlog product along with our work environment (Project Management Methodology/Software environment), which is a list of tasks and features we need to prioritize. By doing all this, we'll make sure we're on track to meet our project's goals.

### 1.2 Requirements Identification

The requirements specification is crucial for building a product. It outlines how the system should work and any limitations on its design. By establishing these guidelines upfront, we ensure that the development team creates a product that meets the customers' needs. Functional requirements describe what the system should do, while non-functional requirements focus on how it should perform. Together, these requirements provide a clear roadmap for product development, ensuring that the end result aligns with customer expectations and functions effectively.

#### 1.2.1 Functional requirements

Functional requirements, often referred to as the functional specification, serve as a comprehensive set of guidelines detailing the primary objectives of the system and how it serves its users. These requirements highlight the core functionalities that the development team needs to implement throughout the development process. By clearly defining what the system must accomplish, functional requirements enable the team to monitor their progress effectively and ensure that they're on track to meet the project's goals.

- Authenticate
- Consult Recipes
- Give Opinions
- Contact Other Users

- Manage Recipes
- Manage Users
- Manage Comments
- Manage Categories

### 1.2.2 Non-Functional requirements

Non-functional requirements, in contrast to functional ones, delineate the various attributes and characteristics of the developed system while it executes its use cases. They primarily focus on aspects such as security, portability, usability, and more, essentially setting the standards for how well the functional requirements should be met. These requirements are critical as they ensure that the system operates within predefined parameters and meets user expectations regarding performance and quality. Although the system may technically still function without adhering strictly to non-functional requirements, it would likely fall short in meeting the overall needs and desires of both the owner and the end-user. Thus, it is imperative to prioritize and fulfill these requirements to deliver a successful and satisfactory product.

- Security and Safety :
  - The app must encrypt user's password and ensure the Security of his account with MFA.
  - The app should ensure user's Safety by blocking sensitive contents and ban suspiscious users.
- Easy to use :
  - The app should provide usability for everybody regardless his knowledge in technology.
  - Provide a tutorial for who use the app for the first time.
- Maintainable
  - The app should runs smoothly.
  - The app should remain reliable and functions effectively.
- Availability
  - The app should be accessible to everyone.
  - The app should be available 24/7 to users.

### 1.2.3 Actors Identification

An actor embodies a user's role when interacting with the system. They represent different user personas and their specific interactions with the system's functionalities, guiding its behavior. Identifying these actors helps tailor the system to meet diverse user needs effectively.

<b>Actors</b>	<b>Roles</b>
Administrator	<ul style="list-style-type: none"><li>• Authenticate</li><li>• Manage Recipes</li><li>• Manage Users</li><li>• Manage Comments</li><li>• Manage Categories</li></ul>
User	<ul style="list-style-type: none"><li>• Authenticate</li><li>• Consult Recipes</li><li>• Give Opinions</li><li>• Contact Other Users</li></ul>

Table 1: Detailed description of the actors

### 1.3 Global use case diagram

The following figure showcases the global use case diagram of our project.

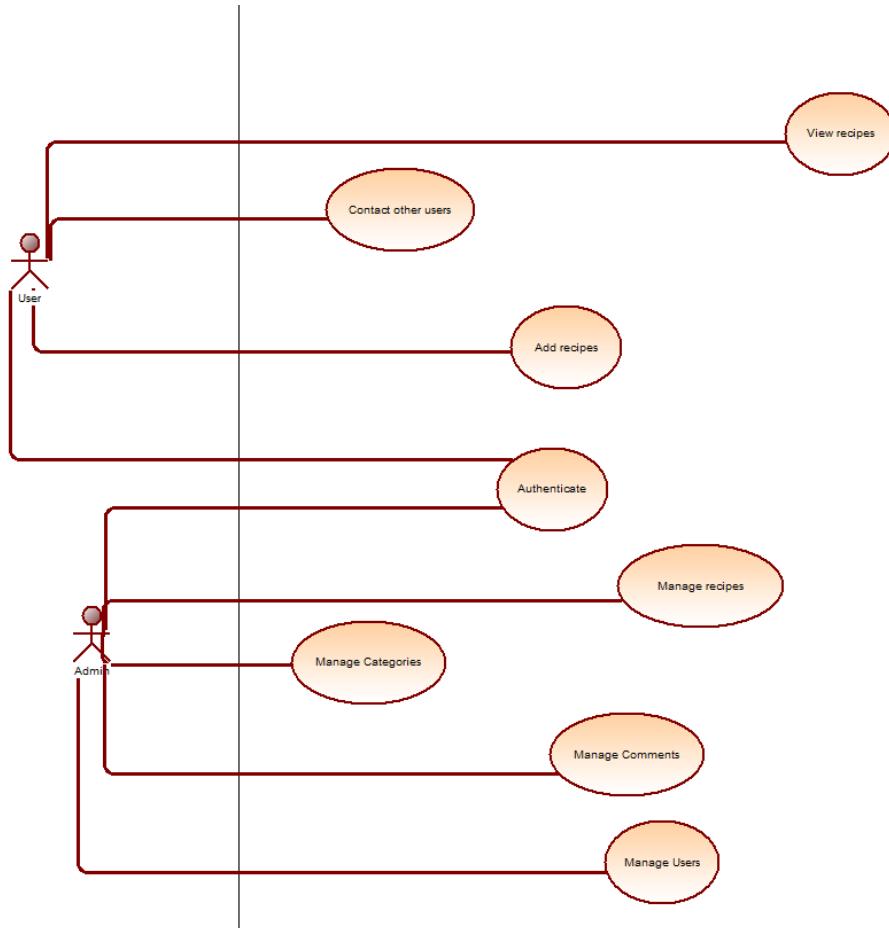


Figure 1: global use case diagram

## 1.4 Product backlog

The backlog product is a prioritized list of tasks essential for developing the project, structured according to the levels necessary to meet system requirements. This hierarchical organization ensures that the most critical tasks are placed at the top level, enabling the development team to identify and focus on delivering them first. By following this approach, the team can streamline their efforts, ensuring that the project progresses in alignment with the system's overarching goals and priorities.

User Story	Priority	Sprint	Estimation
As a user, I can authenticate	1	0	Medium
As a User, I can add recipes	1	0	Medium
As an admin, I can manage categories	1	0	Medium
As an admin, I can authenticate	1	0	Medium
As a user, I can contact other users	2	1	High
As a User I can view recipes	2	1	Medium
As an admin, I can manage users	2	1	High

Table 2: Product backlog

## 1.5 Work environment

In pursuit of our project goals, we've embraced the Agile methodology, specifically Scrum, along with employing various software tools. This approach allows us to adapt quickly to changing requirements, collaborate effectively as a team, and deliver high-quality results efficiently. By leveraging Agile principles and utilizing the right software, we're confident in our ability to achieve our objectives with flexibility and innovation.

### 1.5.1 Project Management Methodology

Scrum is a form of agile project management. You can think of it more like a framework than as a project management methodology in itself. With Scrum, work is split into short cycles known as “sprints”, which usually last about 1-2 weeks. Work is taken from the backlog for each sprint iteration, Small teams are led by a Scrum Master (who is not the same as the project manager) for the duration of the sprint, after which they review their performance in a “sprint retrospective” and make any necessary changes before starting the next sprint.

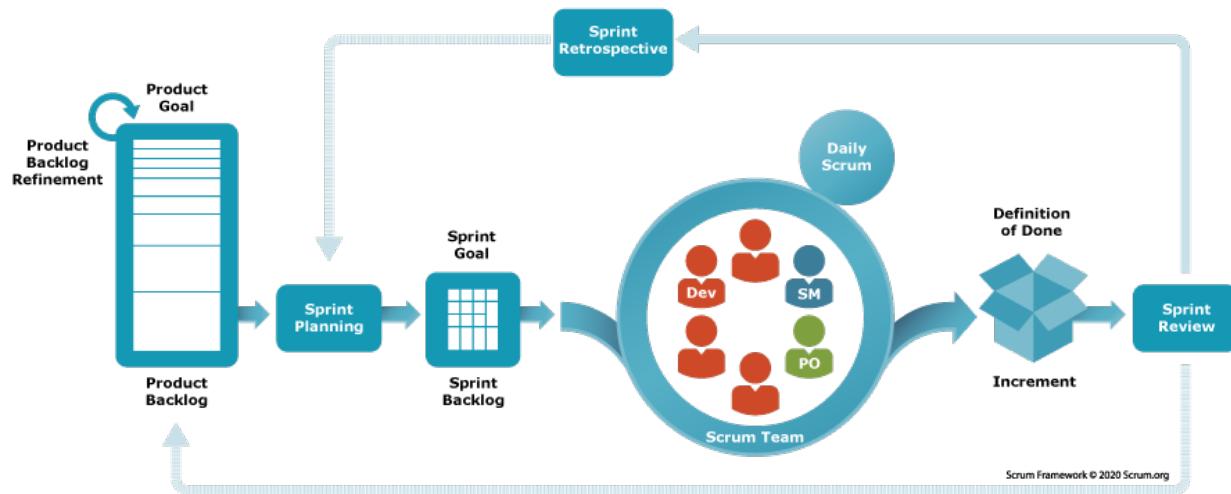


Figure 2: An iteration according to the Scrum method

### 1.5.2 Software environment



Figure 3: Visual Studio Code

**Visual Studio Code :** Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET)[1]



Figure 4: PHP

**PHP :** (recursive acronym for PHP: Hypertext Preprocessor ) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML[2]



Figure 5: Angular

**Angular :** is a TypeScript-based open-source web application framework primarily maintained by Google and a community of developers. It is used for building dynamic single-page web applications (SPAs) and offers a comprehensive solution that includes features such as data binding, dependency injection, routing, and much more. Angular provides a structured framework that follows the Model-View-Controller (MVC) or Model-View-ViewModel (MVVM) architectural patterns, facilitating the development of scalable and maintainable web applications.[3]



Figure 6: MySQL

**MySQL :** is a relational database management system. The database structure is organized into physical files optimized for speed. The logical data model, with objects such as data tables, views, rows, and columns, offers a flexible programming environment.[4]



Figure 7: XAMPP

**XAMPP :** is an open-source software package that provides a local web server environment for testing and development. It helps you test web applications locally before deployment, ensuring they function correctly on a live server.[5]



Figure 8: PowerAMC

**PowerAMC :** PowerAMC supports UML object modeling and data modeling. It is interesting for customers to use the same tool to define objects, database schema, O/R mapping, and to generate the database schema, Java classes and JDO persistence descriptor with O/R mapping definition.[6]



Figure 9: StarUML

**StarUML :** StarUML is a software tool used for creating diagrams to visualize and design software systems. It supports various types of diagrams like class diagrams, sequence diagrams, and more, making it useful for software developers and designers.[7]



Figure 10: Visual Paradigm online

**Visual Paradigm online :** Visual Paradigm Online, an online diagramming software that is perfect for students, teachers and business professionals to reliably create and share all kinds of popular diagrams and charts such as flowcharts, UML, infographic, BPMN, mind maps, Customer Journey Map, organization charts, AWS diagram, Azure diagram, ArchiMate, PERT, SWOT, Value Chain, etc.[8]

## 1.6 Conclusion

In our first chapter, we began by thoroughly detailing the requirements specification process, followed by identifying the actors involved in our system. We then simplified these findings into a concise diagram for easy understanding. Transitioning to our methodology, we highlighted our utilization of the agile Scrum approach for effective project management. Lastly, we rounded off by outlining the software tools integral to our project's development and organization.

# CHAPTER 2

## Sprint 0

### 2.1 Introduction

our primary focus is on setting up the essential groundwork that will pave the way for the development journey ahead. We will be laying the foundation for user authentication, recipe management, admin authentication, and category management functionalities

### 2.2 Identification of Sprint 0 Backlog

The following table contains the backlog elements that are realised during the sprint 0 :

User Story	Priority	Sprint	Estimation
As a User, I can authenticate	1	0	Medium
As a User, I can add recipes	1	0	Medium
As an admin, I can authenticate	1	0	Medium
As an admin, I can manage categories	1	0	Medium

Table 3: Product backlog Sprint 0

### 2.3 Refinement of sprint 0

In this section, we examine various use-case scenarios for the initial sprint.

#### Refinement of the user story "authenticate"

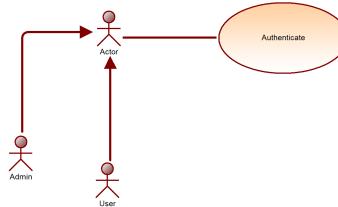


Figure 11: use case diagram “authenticate”

## CHAPTER 2. Sprint 0

---

Use Case Scenario	Authenticate
Actors	Administrator / User
Pre-Conditions	The User must have an account
Post-Conditions	Authenticate
Describe Main Scenario	<ul style="list-style-type: none"><li>• The user types the email and password</li><li>• The user clicks on the login button</li><li>• The system verifies the email and password If they are correct then :</li><li>• The system redirects the user to the home page</li><li>• If not then :</li><li>• The system displays an alert to indicate the error</li></ul>
Scenarios	Give Opinions, View recipes

Table 4: Detailed description of the actors

## CHAPTER 2. Sprint 0

---

### Refinement of the user story "Add Recipes"



Figure 12: use case diagram “Add Recipes”

Use Case Scenario	Add Recipes
Actors	User
Pre-Conditions	The user must be authenticated
Post-Conditions	Recipe added
Describe Main Scenario	<ul style="list-style-type: none"><li>• The user clicks on the "+"(Add) button</li><li>• The system displays the form interface</li><li>• The user fills out the form and clicks on "Add"</li></ul>

Table 5: Detailed description of the Add recipe

### Refinement of the user story "Manage Categories"

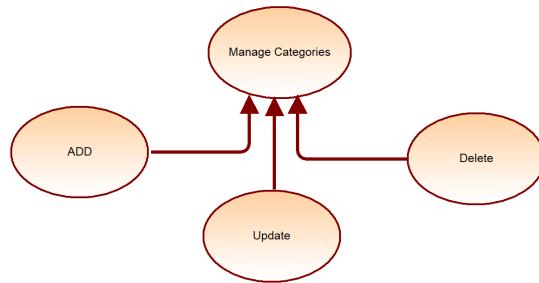


Figure 13: use case diagram “Manage Categories”

Use Case Scenario	Manage Categories
Actors	Admin
Pre-Conditions	<ul style="list-style-type: none"> <li>The administrator must be authenticated</li> <li>The system is operational</li> </ul>
Post-Conditions	Categories Managed
Describe Main Scenario	<ul style="list-style-type: none"> <li>The system displays the interface</li> <li>The admin chooses the operation</li> <li>The system displays the interface according to the choice of the admin</li> </ul>

Table 6: Detailed description of Manage Categories

## 2.4 Conception of Sprint 0

The following figure represents the class diagram of Authenticate.

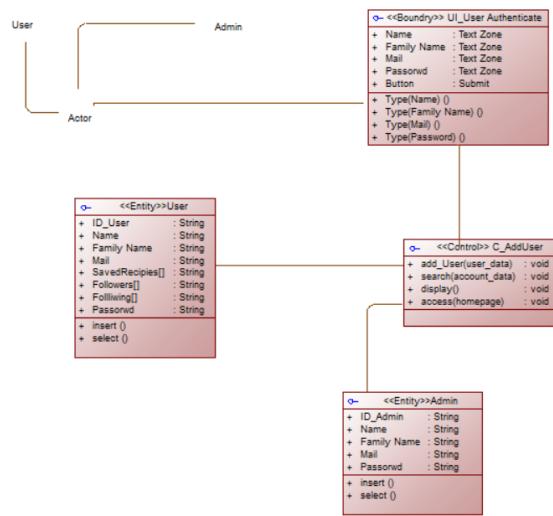


Figure 14: use case diagram “Authenticate”

The following figure represents the sequence diagram of Authenticate.

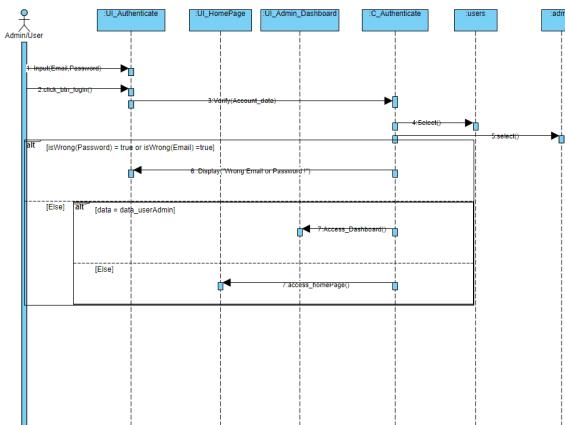


Figure 15: use case diagram “Authenticate”

## CHAPTER 2. Sprint 0

---

The following figure represents the class diagram of Add recipes.

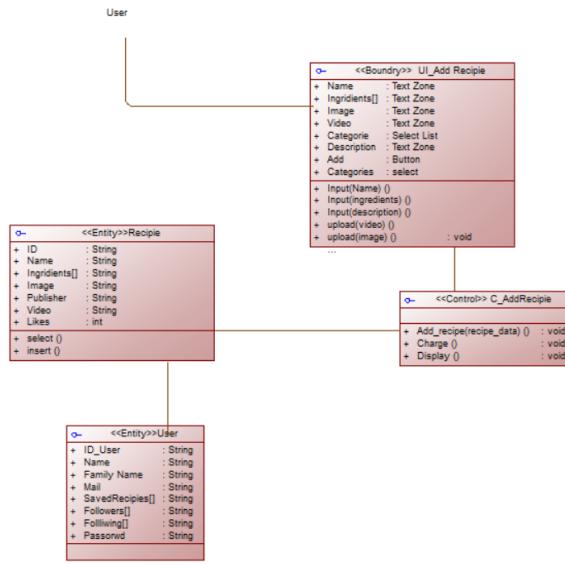


Figure 16: use case diagram “Add Recipes”

The following figure represents the sequence diagram of Add recipes.

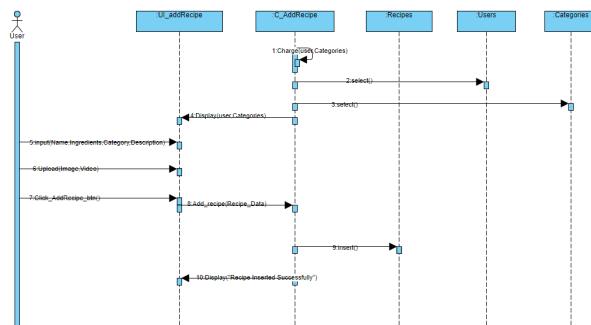


Figure 17: use case diagram “Add Recipes”

## CHAPTER 2. Sprint 0

---

The following figure represents the class diagram of Manage categories.

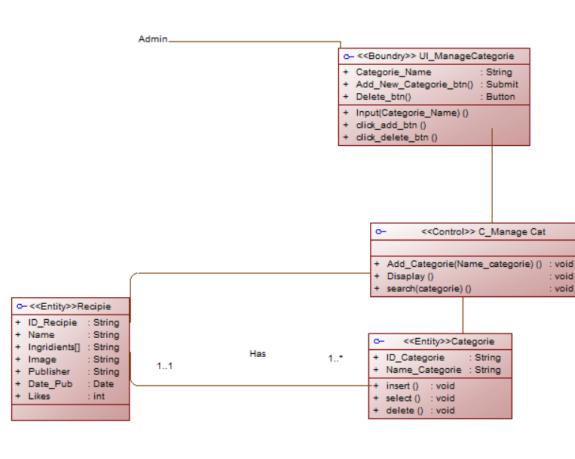


Figure 18: use case diagram “Manege Categories”

The following figure represents the sequence diagram of Manage categories.

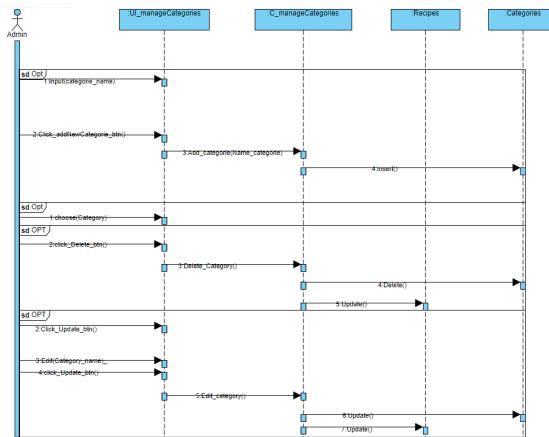
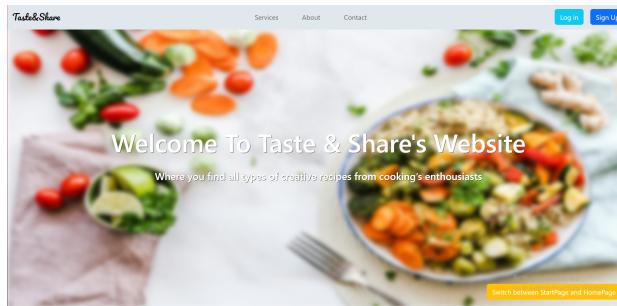


Figure 19: use case diagram “Manege Categories”

## 2.5 Realization of Sprint 0

### Realization of the user story “authenticate”

the following figure displays the login interface used by all users. It includes two fields for entering their credentials. Once authenticated, users can access role-specific dashboards.



**Figure:** Home Page

The user interacts with this interface to sign in or log in. A sign-up form will pop up when the user presses the "Sign Up" button.

A screenshot of the authenticate interface, specifically the login form. The title bar says "Welcome to Taste Share" and has a close button "X". The form contains two input fields: "Email" and "Password", each with its own input box. Below the password field is a checkbox labeled "Show password". At the bottom is a large blue "Login" button.

Figure 20: Authenticate Interface

### Realization of the user story “Add Recipes”

The image exhibits the Add Recipes interface accessible to all users, presenting six fields for inputting Recipe information. Upon completion, users can simply click 'Add Recipe' to publish it.

The screenshot shows the 'Add Recipe' form on a web page. At the top, there is a navigation bar with icons for 'Taste&Share', a house, a search bar, a plus sign, a bell, and a user profile icon. Below the navigation bar, the title 'Add Recipe' is centered. The form consists of six input fields:

- 'Recipe Name'
- 'Ingredients' (with a note: 'Each ingredient must be separated by / or ,')
- 'Image url'
- 'Video url (Optional if there is a description)' (with a note: 'Video url (Optional if there is a description)')
- 'Publisher'  
1
- 'Description (optional if there is a video)'  
Description

At the bottom of the form are two buttons: 'Add Recipe' (blue) and 'Cancel' (red).

Figure 21: Add Recipes Interface

## CHAPTER 2. Sprint 0

---

### Realization of the user story “Manage Categories”

The depicted figure showcases the Manage Categories interface, granting the admin comprehensive control over categories management tasks such as updating, deleting categories

The screenshot shows a web-based application interface titled "Manage Categories". At the top, there is a search bar labeled "Category Name:" with a placeholder and a blue button labeled "Add New Category". Below the search bar is a table with three columns: "Id\_Category", "Name Category", and "Actions". The table contains two rows. Row 1 has an ID of 1 and a category name of "Unlisted", with a note "No Actions". Row 2 has an ID of 2 and a category name of "Dessert", with "Update" and "Delete" buttons. At the bottom left, there is a green button labeled "Cat Commander". The top navigation bar includes icons for user profile, search, and other system functions.

**Figure:** Manage Categories Interface

The admin can manage all categories, including updating categories or deleting them when the admin clicks the "Update" button, another page will open to update categorie.

The screenshot shows a modal window titled "Edit Category". It contains a single input field labeled "Category Name" with a placeholder. Below the input field are two buttons: a green "Update" button and a red "Reset" button.

**Figure:** Update Category Page

The admin can update category name using this interface.

## 2.6 Conclusion

In this sprint, we've focused on three key components: authentication, recipe addition, and category management. Through detailed class diagrams and sequence diagrams, we've outlined the architecture and functionality of each module, ensuring a thorough understanding of their operations. Additionally, we've provided images of each interface, showcasing the user-friendly design and seamless integration of these features.

# CHAPTER 3

## Sprint 1

### 3.1 Introduction

we're shifting our focus towards enhancing user interaction by introducing features that enable users to explore, engage, and manage their interactions within the platform. We'll be implementing the ability to view recipes, contact other users, and manage user accounts

### 3.2 Identification of Sprint 1 Backlog

The following table contains the backlog elements that are realised during the sprint 1 :

User Story	Priority	Sprint	Estimation
As a User I can view recipes	2	1	Medium
As a user, I can contact other users	2	1	High
As an admin, I can manage users	2	1	High

### 3.3 Refinement of sprint 1

In this section, we examine various use-case scenarios for the sprint 1.

#### Refinement of the user story "View recipes "

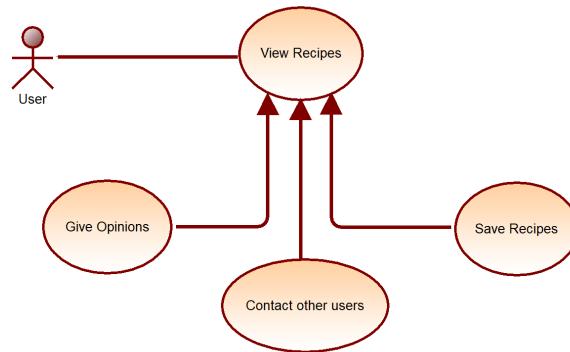


Figure 22: use case diagram “View recipes”

Use Case Scenario	View recipes
Actors	User
Pre-Conditions	The user must be authenticated
Post-Conditions	Recipe accessed
Describe Main Scenario	<ul style="list-style-type: none"> <li>• The user clicks on the recipe</li> <li>• The system displays the recipe interface</li> <li>• The user selects their preferred format for viewing the recipe: either video or text</li> </ul>

Table 7: Detailed description of View Recipes

## CHAPTER 3. Sprint 1

---

### Refinement of the user story "Give Opinions"



Figure 23: use case diagram “Give Opinions”

Use Case Scenario	Give Opinions
Actors	User
Pre-Conditions	<ul style="list-style-type: none"><li>The administrator must be authenticated</li></ul>
Post-Conditions	Opinion Given
Describe Main Scenario	<ul style="list-style-type: none"><li>- The user clicks on the recipe</li><li>- The system displays the recipe interface</li><li>- The user reacts to the recipe with a heart and writes a comment</li></ul>

Table 8: Detailed description of Give Opinions

### Refinement of the user story "Save Recipes"



Figure 24: use case diagram “Save Recipes”

Use Case Scenario	Save Recipes
Actors	User
Pre-Conditions	<ul style="list-style-type: none"><li>• The user must be authenticated</li></ul>
Post-Conditions	Recipe saved
Describe Main Scenario	<ul style="list-style-type: none"><li>• - The user clicks on the save button below any recipe</li><li>• - The system displays a message : "Recipe saved"</li></ul>

Table 9: Detailed description Save Recipes

## CHAPTER 3. Sprint 1

---

### Refinement of the user story "Contact other users"

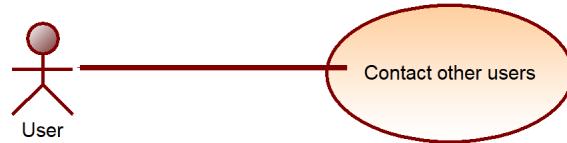


Figure 25: use case diagram “Contact other users”

Use Case Scenario	Contact other users
Actors	User
Pre-Conditions	The user must be authenticated
Post-Conditions	Other users contacted
Describe Main Scenario	<ul style="list-style-type: none"><li>• The user clicks on the recipe</li><li>• The system displays recipe</li><li>• The user clicks on the "Contact The Publisher" button</li></ul>

Table 10: Detailed description of contact other users

### Refinement of the user story "Manage Users"



Figure 26: use case diagram “Manage Users”

Use Case Scenario	Manage Users
Actors	Admin
Pre-Conditions	<ul style="list-style-type: none"><li>• The administrator must be authenticated</li><li>• The system is operational</li></ul>
Post-Conditions	Users Managed
Describe Main Scenario	<ul style="list-style-type: none"><li>• - The administrator chooses whether to add or ban users</li><li>• - The administrator chooses to add or remove moderators</li></ul>

Table 11: Detailed description of Manage Users

### 3.4 Conception of Sprint 1

The following figure represents the class diagram of View recipes.

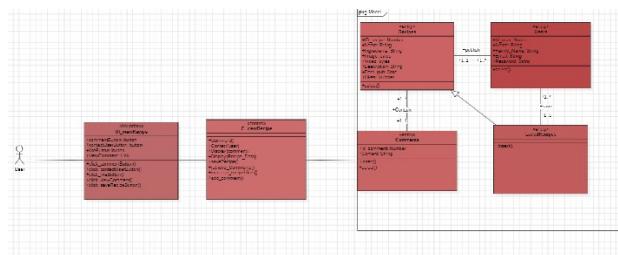


Figure 27: use case diagram “View recipes”

The following figure represents the sequence diagram of View recipes.

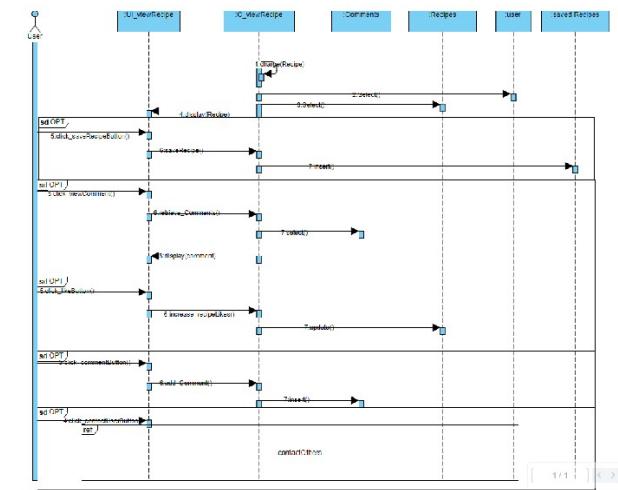


Figure 28: use case diagram “View recipes”

## CHAPTER 3. Sprint 1

---

The following figure represents the class diagram of Contact other Useres.

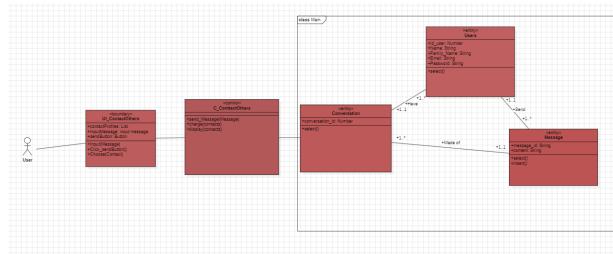


Figure 29: use case diagram “Contact other users”

The following figure represents the sequence diagram of Contact other Useres.

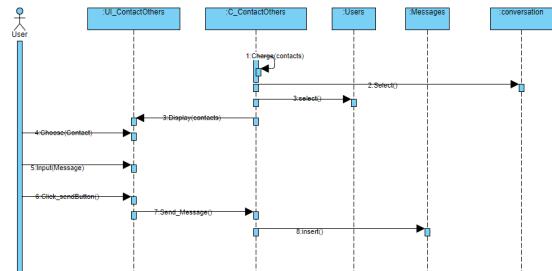


Figure 30: use case diagram “Contact other users”

CHAPTER 3. Sprint 1

The following figure represents the class diagram of Manage Users.

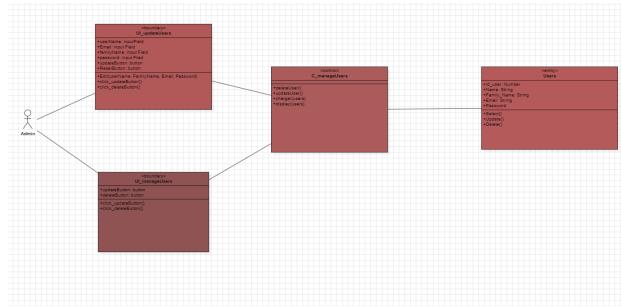


Figure 31: use case diagram “Manage User”

The following figure represents the sequence diagram of Manage Users.

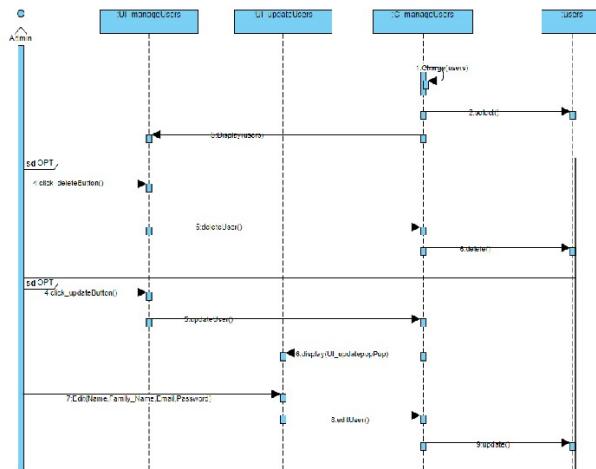
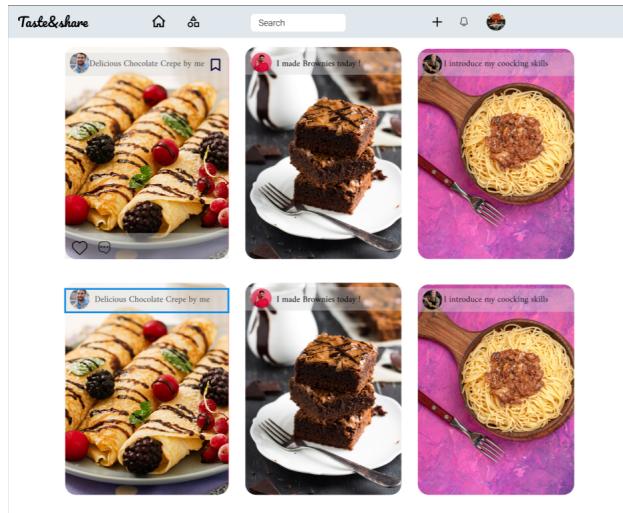


Figure 32: use case diagram “Manage User”

### 3.5 Realization of Sprint 1

#### Realization of the user story “View Recipes”

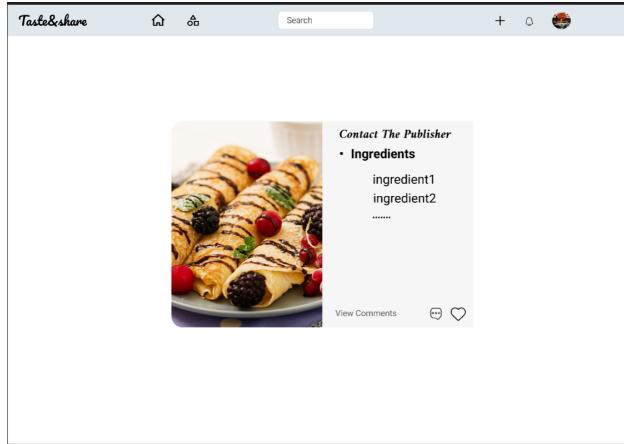
The depicted figure showcases the View Recipes interface, empowering users to seamlessly access recipes, save their favorites, rate dishes, and connect with other users for culinary insights.



**Figure:** View Recipes Page  
The user can view other users' recipes using this interface.

## CHAPTER 3. Sprint 1

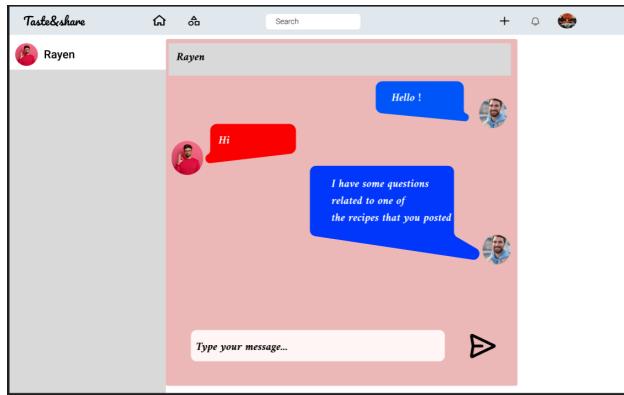
---



**Figure:** View Recipes Page (Detail)

When the user clicks on a recipe, the website will display the recipe details, including any available video.

The user can interact with the recipe by saving it, liking it, commenting on it, viewing ingredients, or contacting the publisher.



**Figure:** Contact Publisher Interface

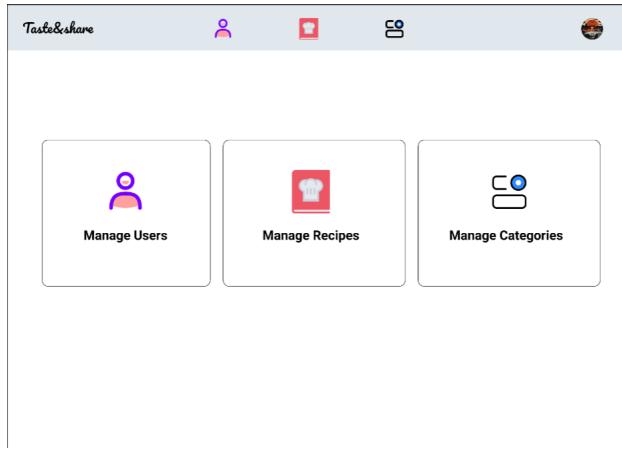
When the user clicks on the "Contact the Publisher" button, a small chat room will be created between the user and the publisher.

## CHAPTER 3. Sprint 1

---

### Admin Dashboard

This interface allows the Admin to Manage all users , recipes , and categories



**Figure:** Admin Dashboard

based on what the Admin wants to manage , he needs to click on buttons provided

## CHAPTER 3. Sprint 1

---

### Realization of the user story “Manage Users”

The depicted figure showcases the Manage Users interface, granting the admin comprehensive control over user management tasks such as updating, deleting, and overseeing user accounts. Additionally, the admin possesses the capability to manage and remove comments efficiently.

ID User	User Name	User Email	Actions	
1	Mr.Dayekh Rayen	Rayen@gmail.com	<button>Update</button>	<button>Delete</button>
2	Mr.Ben Amira Ilyes	Layes.Noxian@gmail.com	<button>Update</button>	<button>Delete</button>

**Figure:** Manage Users Interface

The admin can manage all users, including updating user information or deleting users. When the admin clicks the "Update" button, another page will open to update user information.

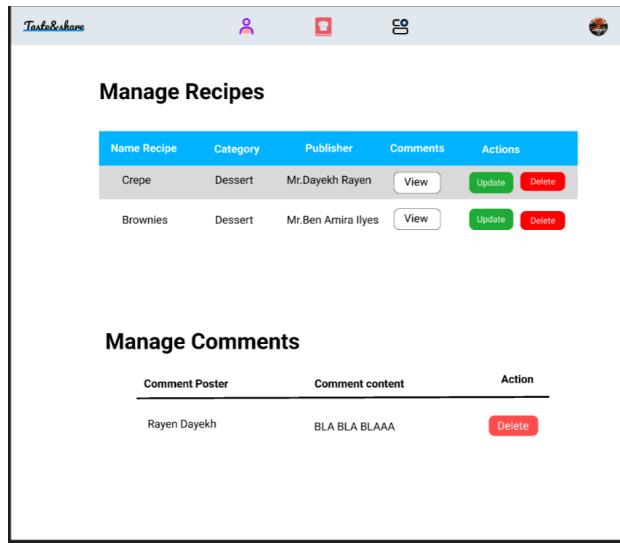
The interface is titled "Update User". It contains four input fields arranged in a 2x2 grid: "User Name" and "Family Name" in the top row, and "Email" and "Password" in the bottom row. Below the input fields are two buttons: a green "Update" button and a red "Reset" button.

**Figure:** Update User Page

The admin can update user information using this interface.

## CHAPTER 3. Sprint 1

---



**Figure:** Manage Recipes Interface

The admin can manage recipes using this interface. Clicking the "View" button will display all comments on that recipe, allowing the admin to delete them. Clicking the "Update" button will open another page to update the recipe name and category.

### 3.6 Conclusion

In the sprint section, we've thoroughly outlined our project's architecture and functionality for user management, recipe management, and recipe viewing. Detailed class diagrams and sequence diagrams provide an insight into the inner workings of each module, ensuring a clear understanding of their operations. Furthermore, we've included images of each interface, illustrating the user-friendly design and functionality of our system. These components collectively demonstrate our dedication to creating a seamless user experience while managing users, recipes, and viewing recipes.

# General Conclusion

Our website offers a rich culinary experience where users can discover, contribute, and connect over a wide array of recipes. With the ability to view, add, follow, and share recipes, our platform provides a dynamic space for food lovers to explore and exchange ideas. Whether you're seeking new recipes to try, sharing your own creations, or engaging with other users, our website is designed to inspire and foster a sense of community around the joy of cooking. Join us today and become part of our vibrant foodie community!

## Bibliography

1. <https://code.visualstudio.com/docs>
2. <https://www.php.net/manual/fr/intro-whatis.php>
3. <https://angular.io/guide/what-is-angular>
4. <https://www.oracle.com/mysql/what-is-mysql/>
5. <https://www.apachefriends.org/fr/about.html>
6. <https://www.powerdesigner.biz/>
7. <https://sourceforge.net/projects/staruml/>
8. <https://appsource.microsoft.com/en-cy/product/office/wa104381637?tab=overview>