# Abstract

My graduation project comprises developing a mobile application for searching and visualizing of natural, bio, vegan, eco, and allergen free products. Based on personal preferences such as skin type, allergens, and personal search history, the app users can search for their desired products.

I developed this project in 7 sprints. First, a mobile application using Flutter, the open-source Google’s UI toolkit, that manage different screen interfaces for the app. And second, I created a back-end web server with NodeJS in order to manage the client search requests and others different functionalities.

Working in this project started from interviewing natural and allergen free real consumers, product manufacturers, product retailers, and consultants to gather information about their right needs in the eco-friendly environment and to optimize their user-experience, then the development and implementation of the requested solution and finished with ambition hopes to optimize this project for better user-experience of the app users and further features.

I analysed, designed, and developed this project within L’Atelier, the Marketing, training and merchandising division of Kilani Group.

© 2021  
Sadok Laouissi   
ALL RIGHTS RESERVED

# Dedication

Sadok Laouissi

# Acknowledgements

I would like to express my deep gratitude to our corporate supervisor Mrs. Ridha Leila and my faculty supervisor Mr. Ben Ahmed Zied, for their patient guidance, enthusiastic encouragement and useful critiques of this work.

I would like also to extend my thanks to the product consultant of L’Atelier Mrs. Dhahri Olfa for her help in providing the professional information about the skin type, ingredients and allergies and also helping in the interview process.

Finally, I wish to thank ARGANIA IT Support Division for their help during this project.

# Table of Contents

[Abstract i](#_Toc82313691)

[Dedication iii](#_Toc82313692)

[Acknowledgements iv](#_Toc82313693)

[Table of Contents v](#_Toc82313694)

[List of Figures viii](#_Toc82313695)

[List of Tables xi](#_Toc82313696)

[General Introduction 12](#_Toc82313697)

[Chapter I: Situation analysis of the project 15](#_Toc82313698)

[1. Introduction 15](#_Toc82313699)

[2. Presentation of the host organization 15](#_Toc82313700)

[3. Presentation of the organization 15](#_Toc82313701)

[4. Study of the existing 21](#_Toc82313702)

[5. Conclusion 24](#_Toc82313703)

[Chapter II: Specifications and methodology 26](#_Toc82313704)

[1. Introduction 26](#_Toc82313705)

[2. Modelling Language 26](#_Toc82313706)

[3. Software Requirements Specifications 27](#_Toc82313707)

[4. Global Use Case 29](#_Toc82313708)

[5. Software Development Methodology 30](#_Toc82313709)

[6. Project Management 36](#_Toc82313710)

[Chapter III: State of the art 43](#_Toc82313711)

[1. Introduction 43](#_Toc82313712)

[2. Conclusion 43](#_Toc82313713)

[Chapter IV: Sprint Zero - Work Environment 45](#_Toc82313714)

[1. Introduction 45](#_Toc82313715)

[2. Work Environment 45](#_Toc82313716)

[3. Application Architecture 52](#_Toc82313717)

[4. Conclusion 54](#_Toc82313718)

[Chapter V: Sprint 1 – Category Management 56](#_Toc82313719)

[1. Introduction 56](#_Toc82313720)

[2. Sprint Backlog 56](#_Toc82313721)

[3. Analysis 57](#_Toc82313722)

[4. Conclusion 67](#_Toc82313723)

[Chapter VI: Sprint 2 – Allergens Management 69](#_Toc82313724)

[1. Introduction 69](#_Toc82313725)

[2. Sprint Backlog 69](#_Toc82313726)

[3. Analysis 70](#_Toc82313727)

[4. Conclusion 80](#_Toc82313728)

[Chapter VII: Sprint 3 – Ingredients Management 82](#_Toc82313729)

[1. Introduction 82](#_Toc82313730)

[2. Sprint Backlog 82](#_Toc82313731)

[3. Analysis 83](#_Toc82313732)

[4. Conclusion 91](#_Toc82313733)

[Conclusion & Perspectives 94](#_Toc82313734)

[Reflection 95](#_Toc82313735)

[Table of Acronyms and Abbreviations 96](#_Toc82313736)

[References 97](#_Toc82313737)

# List of Figures

[Figure 1: Kilani Groupe Logo 16](#_Toc82313738)

[Figure 2: Kilani Groupe Organizational chart 18](#_Toc82313739)

[Figure 3: Kilani Groupe Missions & Activities 19](#_Toc82313740)

[Figure 4: L'Atelier’s Activities 21](#_Toc82313741)

[Figure 5: Yuka Logo 22](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313742)

[Figure 6: INCI Beauty Logo 22](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313743)

[Figure 7: PharmaPocket Logo 23](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313744)

[Figure 8: Think Dirty App Logo 23](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313745)

[Figure 9: UML Diagrams Overview [5] 27](#_Toc82313746)

[Figure 10: Global Use Case 29](#_Toc82313747)

[Figure 11: Agile Methodology workflow 31](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313748)

[Figure 12: Scrum Methodology Overview 31](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313749)

[Figure 13: Scrum Values 32](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313750)

[Figure 14 : Gantt Diagram Part 1 40](#_Toc82313751)

[Figure 15 : Gantt Diagram Part 2 41](#_Toc82313752)

[Figure 16: Android Studio Logo 46](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313753)

[Figure 17: VS Code Logo 46](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313754)

[Figure 18: Postman Logo 47](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313755)

[Figure 19: WampServer Logo 47](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313756)

[Figure 20: Visual Paradigm Online Logo 47](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313757)

[Figure 21: Adobe XD Logo 48](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313758)

[Figure 22: Adobe Photoshop Logo 48](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313759)

[Figure 23: GitHub Desktop Logo 48](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313760)

[Figure 24: TeamGantt Online Logo 49](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313761)

[Figure 25: Dart Logo 49](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313762)

[Figure 26: TypeScript Logo 49](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313763)

[Figure 27: MySQL Logo 50](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313764)

[Figure 28: Angular Logo 50](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313765)

[Figure 29: Node JS Logo 51](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313766)

[Figure 30: Flutter Logo 51](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313767)

[Figure 31: JSON Logo 52](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313768)

[Figure 32 : Application's Physical Architecture 53](#_Toc82313769)

[Figure 33 : Logical Architecture 54](#_Toc82313770)

[Figure 34: Categories Management Use Case Refinement 57](#_Toc82313771)

[Figure 35: Add Category Sequence Diagram 59](#_Toc82313772)

[Figure 36: Screen Capture Add Category Interface 60](#_Toc82313773)

[Figure 37: Edit Category Sequence Diagram 61](#_Toc82313774)

[Figure 38: Screen Capture Edit Category Interface 62](#_Toc82313775)

[Figure 39: Delete Category Sequence Diagram 63](#_Toc82313776)

[Figure 40: Screen Capture Delete Category 64](#_Toc82313777)

[Figure 41: Screen Capture List Categories Interface 65](#_Toc82313778)

[Figure 42: Home Screen Implementation 66](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313779)

[Figure 43: Home Screen Design 66](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313780)

[Figure 44: Allergens Management Use Case Refinement 70](#_Toc82313781)

[Figure 45: Screen Capture Add Allergen Interface 71](#_Toc82313782)

[Figure 46: Screen Capture Edit Allergen 72](#_Toc82313783)

[Figure 47: Screen Capture Delete Allergen 74](#_Toc82313784)

[Figure 48: Sequence diagram of the use case "Associate Ingredient/Allergen" 76](#_Toc82313785)

[Figure 49: Screen Capture Associate Ingredient/Allergen Interface 77](#_Toc82313786)

[Figure 50: Sequence Diagram Edit Personal Allergen Scenario 2 79](#_Toc82313787)

[Figure 51: Personal Allergen Implementation 80](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313788)

[Figure 52: Design Adobe XD: Personal Allergens 80](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313789)

[Figure 53 : Ingredients Management Use Case Refinement 83](#_Toc82313790)

[Figure 54: Screen Capture Ingredients Interface 85](#_Toc82313791)

[Figure 55: Sequence diagram Add Ingredient 87](#_Toc82313792)

[Figure 56: Screen Capture Add Ingredient 87](#_Toc82313793)

[Figure 57: Screen Capture Edit Ingredient 88](#_Toc82313794)

[Figure 58: Screen Capture Delete Ingredient 89](#_Toc82313795)

[Figure 59: Ingredients Screen Realisation 91](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313796)

[Figure 60: Design Adobe XD Ingredients Screen 91](file:///E:\GitHub\nurture_cosmetic\report\PFE-Finale.docx#_Toc82313797)

# List of Tables

[Table 1: The Scrum Team 33](#_Toc82313798)

[Table 2: Product Backlog 39](#_Toc82313799)

[Table 3: Sprints Planning 39](#_Toc82313800)

[Table 4: Hardware Environment Characteristics 45](#_Toc82313801)

[Table 5: Tasks Planning - Sprint 1 57](#_Toc82313802)

[Table 6: Textual description of the use case “Add Category” 58](#_Toc82313803)

[Table 7: Textual description of the use case “Edit Category” 61](#_Toc82313804)

[Table 8: Textual description of the use case “Delete Category” 63](#_Toc82313805)

[Table 9: Textual description of the use case “List Categories” 65](#_Toc82313806)

[Table 10: Textual description of the use case “List Categories (User)” 66](#_Toc82313807)

[Table 11: Tasks Planning - Sprint 2 69](#_Toc82313808)

[Table 12: Textual description of the use case “Add Allergen” 71](#_Toc82313809)

[Table 13: Textual description of the use case “Edit Allergen” 72](#_Toc82313810)

[Table 14:Textual description of the use case “Delete Allergen” 73](#_Toc82313811)

[Table 15: Textual Description Associate Ingredient/allergen 75](#_Toc82313812)

[Table 16 : Textual description Edit Personal Allergen 78](#_Toc82313813)

[Table 17: Tasks Planning Sprint 3 - Ingredients Management 83](#_Toc82313814)

[Table 18: Textual description of the use case “View All Ingredients” 84](#_Toc82313815)

[Table 19: Textual description of the use case “Add Ingredient” 86](#_Toc82313816)

[Table 20: Textual description of the use case “Edit Ingredient” 88](#_Toc82313817)

[Table 21: Textual description of the use case “Delete Ingredient” 89](#_Toc82313818)

[Table 22: Textual description of the use case “View Product's Ingredients” 90](#_Toc82313819)

# General Introduction

Nowadays, many people are thinking about environmental issues and the environmental condition of the Earth. People understand that their irresponsibility hurts the natural environment. Our planet suffers from many problems, which result from excessive human activity.

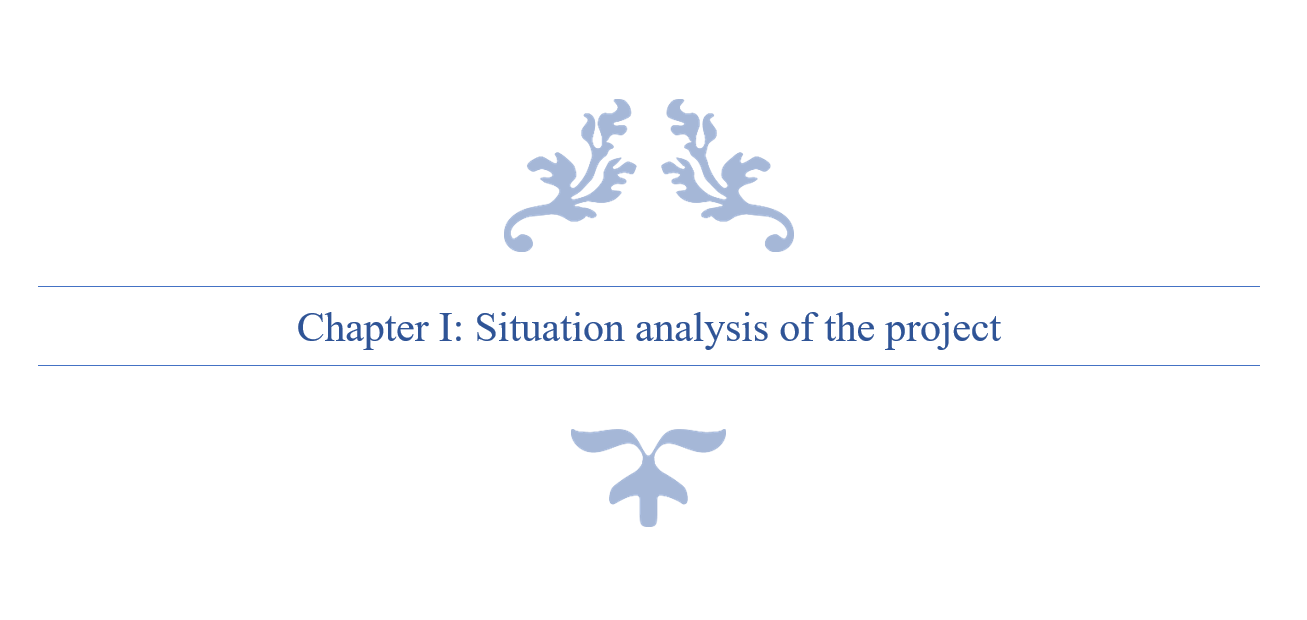
The entire planet is enduring contamination, worldwide warming, deforestation, and imperilling natural life species. These issues are exceptionally pertinent and require quick and comprehensive arrangements. The arrangement for these problems is the alter within the demeanour of humankind towards nature and the natural assets that are being used unrestrictedly. Individuals got to esteem the environment and nature for their survival. Put, individuals ought to “go green” to spare the Soil.

More and more people are “Going Green” in recent years. The primal inspirations for this development have been the diminished presentation to chemicals additionally natural, bio, and allergen-free items are better for the environment. Hence, this new concept has developed the way buyers see the items they use and purchase. Products producers are driving toward a supportability aim and getting to be “Green Manufacturer”. Product Buyers needs more transparency about the ingredients used in making each product and are more willing to advise approximately this rising concept. Thus, taking part in this global movement comes with a part of struggles, the information about allergies, ingredients, allergen... A few customers brag about the product’s quality and harmful ingredients, and others brag about the diversity of products.

The overarching goal of this project is to set up an innovative solution for searching and visualizing of natural, bio, vegan, eco, and allergen free products based on personal preferences such as skin type, allergies, and personal search history.

During the realization of this project, I used a method of fairly efficient development, resulting from the Agile method, namely the SCRUM. I will try through this report to highlight the steps taken, in which I have used the advantages of said method, in particular the plan of productivity and efficiency.

I composed this report of six major chapters. First, I am going to present the situation analysis of the project. Then, the second is about specifications and method. The third chapter is the state-of-the art. The fourth chapter describes the work environment, which is Sprint Zero. And the last two chapters are the sprints implemented to develop this application.



# Chapter I: Situation analysis of the project

## Introduction

In this chapter, I am going to present the host organization first. Then I am going to expose the subject of the project and the work environment. And finally, I will conduct a solution for this problematic.

## Presentation of the host organization

### Introduction

This is a report of the graduation project within the engineer degree majoring in Web & Mobile Software Development at the Private Higher School of Engineering and Technology (ESPRIT). I carried this project out within Kilani Group, the Marketing, training and merchandising department L’Atelier.

## Presentation of the organization

A leading actor in the fields of health, beauty, well-being and hygiene in Tunisia and abroad, Kilani Group is driven by the passion of the profession, the ethics as well as the satisfaction of customers and partners.

Since its creation, Kilani Group has expanded its activities around the world of pharmacy and drugstore including the manufacturing of pharmaceuticals, the import and distribution of healthcare, cosmetics & hygiene brands and medical and paramedical devices. Within a spirit of synergy, a range of service activities have been developed for partners operating in the fields of health & beauty.

The Group is thus well positioned in all sectors: industry, wholesale, direct distribution, retail and services.



Figure 1: Kilani Groupe Logo

### Subsidiaries of Kilani Groupe

Kilani Group started its activities in the distribution sector in 1986. Today, the Group is operating in the import, wholesale and direct distribution. Through dedicated sales teams specialized by sector and a logistic network extended all over Tunisia, the group is specialized in pharmaceuticals, paramedical products, surgical equipment and accessories, reagents and laboratory robots, cosmetics, dermo-cosmetics as well as hygiene products.

* ARGANIA
* KIPROPHA
* PROCHIDIA

Leading actor in Tunisia in the pharmaceutical industry, Kilani Group is a drug manufacturer since 1996 with production sites in Tunisia and abroad. The group innovates by constantly investing in research and cutting-edge technology in order to be at the service of health and participate in the development of the pharmaceutical sector.

* TERIAK

Kilani Group is a leader in retail sales in the beauty sector through a network of self-service outlets throughout Tunisia under its own brand. The Group then expanded into the retail of natural beauty and fashion through franchised brands.

* FATALES
* L’OCCITANE EN PROVENCE
* NINE WEST

KILANI Group also offers a range of complementary services to the medical industry, the distribution and the retail channels. One company is specialized in medical detailing, the management of proprietary and partner brands of drugs and the second company offers consulting in Retail marketing, shops layout; Trade Marketing Activation and Digital Marketing for Retail, entities and brands.

* L'Atelier
* Medicis

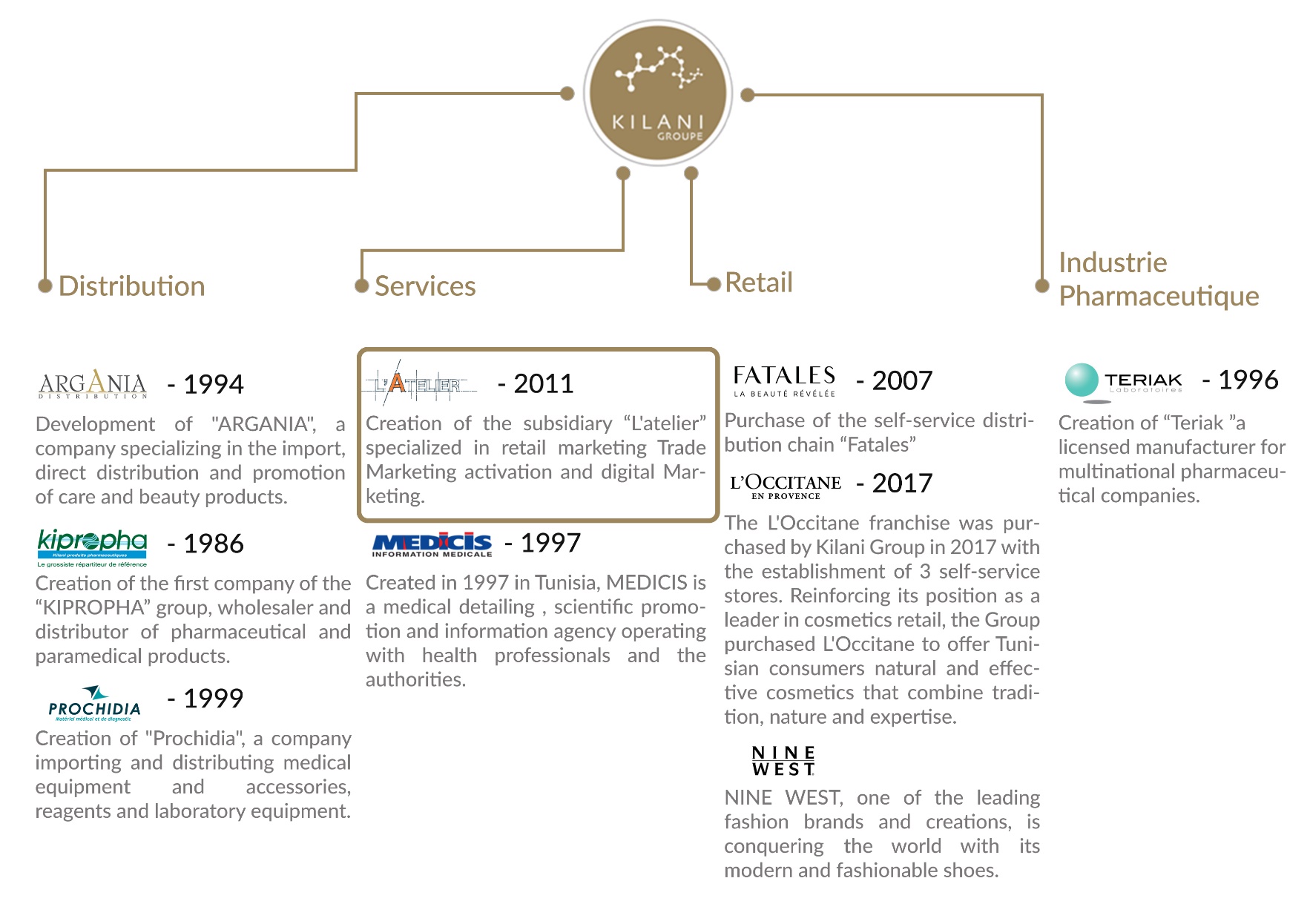


Figure 2: Kilani Groupe Organizational chart

### Missions of Kilani Groupe

Providing well-being and better health to everyone through the development of our brands and those of our partners in drugs, medical devices, care, beauty, hygiene and well-being as well as through our support services for healthcare professionals.

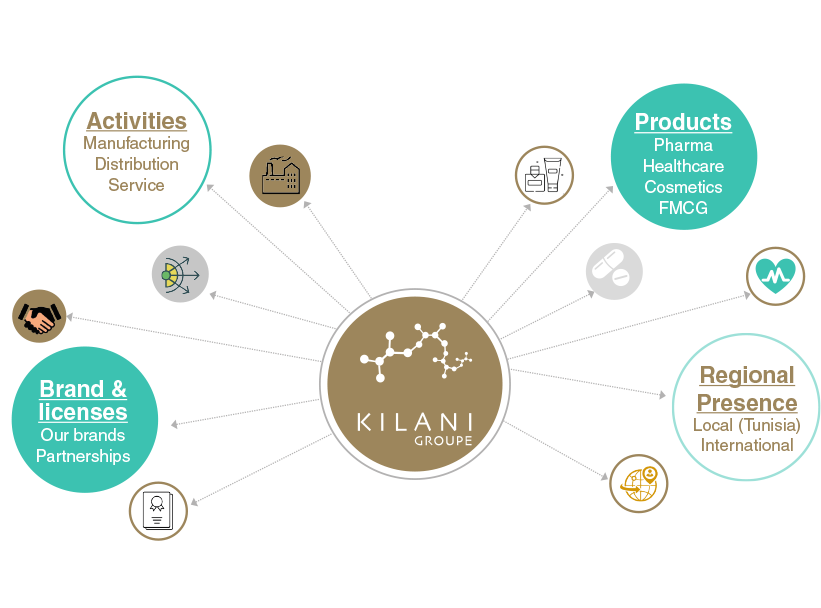


Figure 3: Kilani Groupe Missions & Activities

### Core values of Kilani Groupe

We are driven by a passion, the passion of life. A better life for everyone. Kilani Group has successfully passed on this passion to its subsidiaries and employees who endeavour every day to satisfy all their partners and customers by offering or proposing the best brands, products and services in the fields of health, beauty and well-being.



We adhere to the highest ethical standards. This ethics is that of well performed work and a product that meets the highest quality and safety standards. Ethics that is also based on respect and a sense of responsibility towards our partners, our employees and our customers.

As our main ambition is to satisfy our partners and customers, we do our utmost to innovate through cutting-edge processes to bring out the best in all our fields of activity.



Excellence is our leitmotiv and is embodied through our shared vision with our employees and partners to deliver high quality products and reliable services that meet the requirements of our customers.

### L’Atelier’s Presentation

L'Atelier, a subsidiary of Kilani Group, is specialized in the marketing and communication of brands and outlets.

Composed of a **team of experts**, it supports brands and outlets at all the stages of their projects **from ideation, to launching and marketing activation, in store, out of store as well as digitally.**



Figure 4: L'Atelier’s Activities

L'Atelier's areas of expertise include Marketing Concept Ideation, Outlet Layout and Optimization, Trade Marketing Activation for brands, Marketing and Digital Communication for brands and entities, and Customer Experience Training.

## Study of the existing

### Problematic

After several interviews with customers, producers and retailers in nutriment and cosmetic fields which use, sell, buy or produce natural or bio products, we concluded that currently, there is a strong diversification of organic consumers. Thus, distributors must put in place marketing strategies in order to remain competitive in their market.

The multiple food crises have encouraged the development of the organic market. Indeed, in recent years, demand has grown steadily. Since the early 1990s, the behavior of these consumers has changed dramatically. Combined with a feeling of uncertainty about the quality of the products they buy, they are now turning to certified products, including organic products.

### Presentation of the existing

Based on running tests on the coming concurrent apps, we realized few problems for each one of them.

#### Yuka

Yuka decrypts the labels of your products food and cosmetics and their impact on health.

* Doesn’t have offline mode
* Not suitable for people with special needs such as allergies or diabetes

Figure 5: Yuka Logo

#### INCI Beauty

INCI Beauty is an app that allows you to search and scan a product and consult its ingredients.

Figure 6: INCI Beauty Logo

* Fetching data errors
* Unreliable sources and result not based on scientific background
* Unidentified products
* Complicated to use

#### PharmaPocket

The app allows you to scan the barcodes of cosmetic products and decrypt their composition using simple pictograms.

Figure 7: PharmaPocket Logo

* Last update 2017
* App doesn’t start

#### Think Dirty

Think Dirty app allows us to learn ingredients in our beauty, personal care and household products.

Figure 8: Think Dirty App Logo

* Limited Library
* Exploit the lucrative potential of its proposed service by “Verified Brand”

### Project Solution

Considering this problematic, I propose to develop a mobile application to resolve all of these challenges.

The mission of this project is to provide a responsive mobile app that allows its users to search and filter natural, bio, vegan and allergen-free products with a friendly user interface to inform the users about various information. Then, create a simple and easy form to identify the skin type of the user with a setting page to select its allergies. And last is to implements an algorithm that assist the user in further researches based on his selected preferences.

## Conclusion

My graduation project scope is to develop a mobile application to assist products consumers in their research for the perfect product based on their preferences.



# Chapter II: Specifications and methodology

## Introduction

In this chapter, we’re going to tackle in the first section the modelling language UML, software requirements specifications in the second part, and in the last part, the project management.

## Modelling Language

### Introduction

UML is a standard language for writing software blueprints [4]. UML may be used visualize, construct, and document the artifacts of software. UML is appropriate for modelling systems ranging from enterprise information systems to distributed Web-based applications.

The vocabulary of the UML encompasses three basic terms: **Things**, **Relationships** and **Diagrams**. Things are the abstractions that are firstly build in a model; relationships tie these things together and diagrams group collections of things.

### Unified Modelling Language

A modelling language is a way of expressing building a model, which has been produced during the developing process.

Modelling language defines a collection of model elements. UML, the Unified Modelling language, is the most popular -that will be either by hand or drawing by the tool- needs a number of things that a modelling tool can provide; such as: reporting, integrating with other process model, synchronizations of models and code. Today there are several UML tools on the market that describe the semantics, notations and constructs of UML.

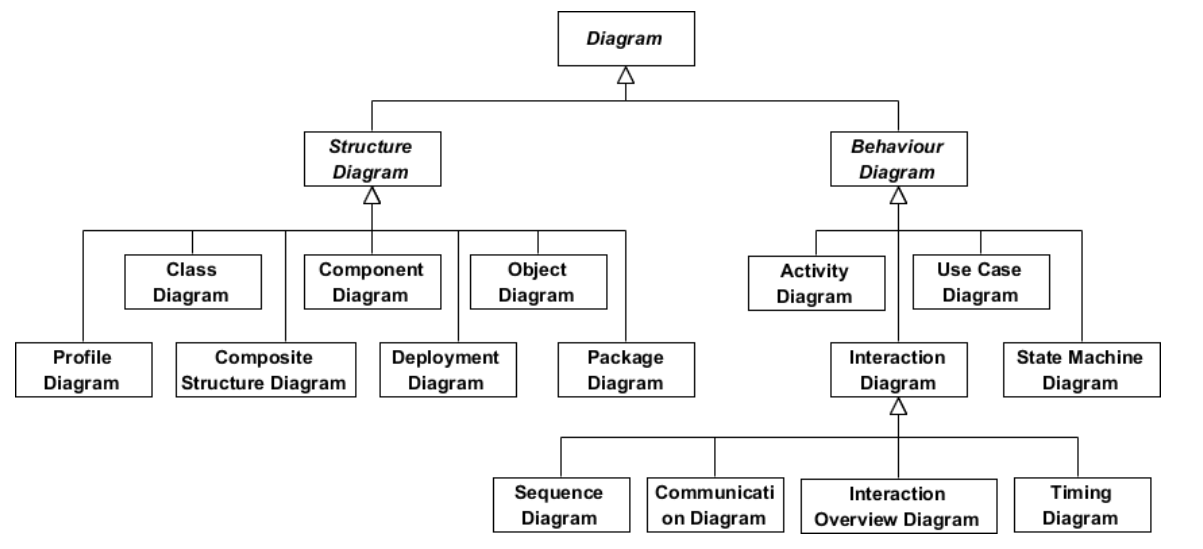


Figure 9: UML Diagrams Overview [5]

## Software Requirements Specifications



### Non-Functional Requirements

Non-functional requirements (NFR’s) cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviours.[6]

In other words, NFS define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

* Reliability and synchronization
* Extensibility
* Friendly user experience

### Functional Requirements

The projects functional requirements features are shown as below:

* + Users’ management
  + Allergens Management
  + Complaints Management
  + Products Management
  + Ingredients Management
  + Search History Management
  + Product Suggestion
  + Favourite Management

## Global Use Case

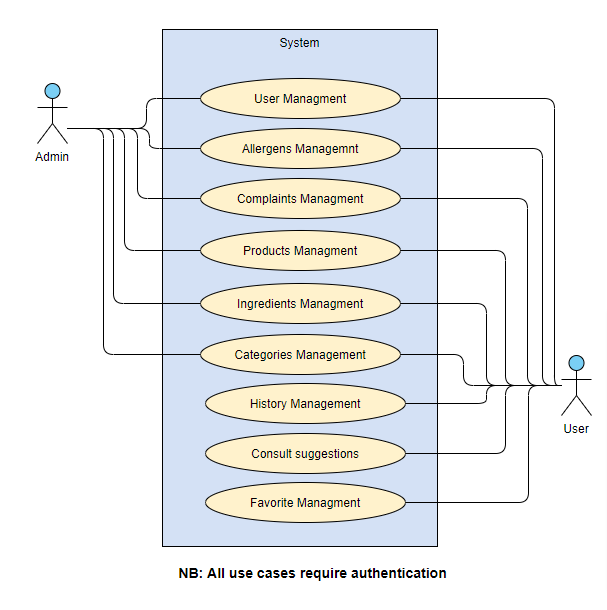


Figure 10: Global Use Case

## Software Development Methodology

**Introduction**

A software development methodology or system development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system.[7]

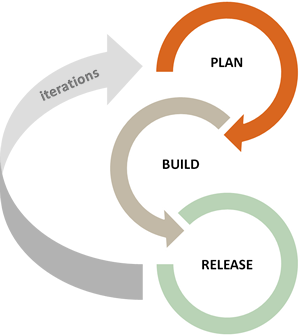
There are the following methodologies:

* Agile Software Development
* Crystal Methods
* Dynamic Systems Development Model (DSDM)
* Extreme Programming (XP)
* Feature Driven Development (FDD)
* Joint Application Development (JAD)
* Lean Development (LD)
* Rapid Application Development (RAD)
* Rational Unified Process (RUP)
* Scrum
* Spiral
* Systems Development Life Cycle (SDLC)
* Waterfall



### Agile Methodology

Agile software development is a conceptual framework for undertaking software engineering projects. There are several agile software development methodologies e.g., Crystal Methods, Dynamic Systems Development Model (DSDM), and Scrum.[8]

We can view agile methods such as Extreme Programming (XP) and Scrum as a reaction to plan-based or traditional methods, which emphasize a "rationalized, engineering-based approach, incorporating extensive planning, codified processes, and rigorous reuse.

In contrast, agile methods address the challenge of an unpredictable world, emphasizing the value competent people and their relationships bring to software development. To clarify the effectiveness of agile methods, we reviewed the agile development literature and conducted a systematic study of what we know empirically about its benefits and limitations.

Figure 11: Agile Methodology workflow



### Scrum Methodology

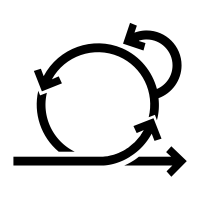
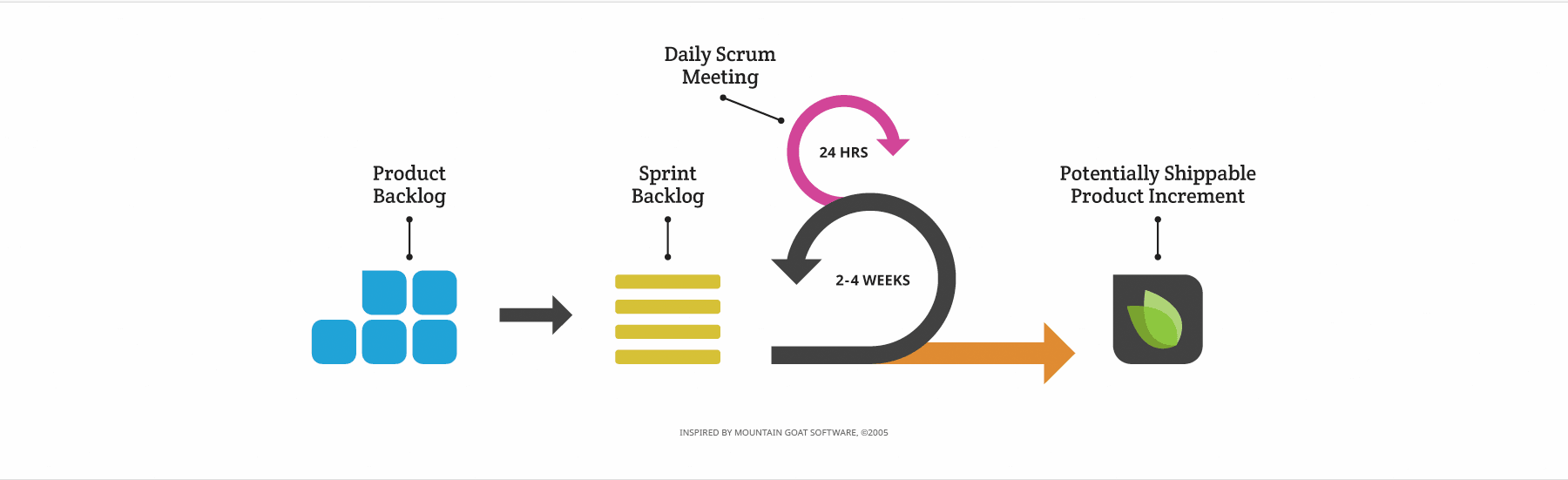
When Jeff Sutherland created the scrum process in 1993, he borrowed the term "scrum" from an analogy put forth in a 1986 study by Takeuchi and Nonaka, published in the Harvard Business Review. In that study, Takeuchi and Nonaka compare high-performing, cross-functional teams to the scrum formation used by Rugby teams. Scrum is the leading agile development methodology, used by Fortune 500 companies around the world.[9]

Figure 12: Scrum Methodology Overview

The Scrum Alliance exists to transform the way we tackle complex projects, bringing the Scrum framework and agile principles beyond software development to the broader world of work.[10]

Figure 13: Scrum Values

#### The Scrum Team

|  |
| --- |
| Product Owner: **Mrs. Ridha Leila** |
| Product owner is the champion for his product. He is focused on understanding business and market requirements, then prioritizing the work to be done by the engineering team accordingly.  Product owner is not a project manager. Product owner is not managing the status of the program. He focusses on ensuring the development team delivers the most value to the business. |
| Scrum Master: **Mr. Ben Ahmed Zied** |
| An effective scrum master deeply understands the work being done by the team and can help the team optimize their delivery flow. As the facilitator-in-chief, they schedule the needed resources (both human and logistical) for sprint planning, stand-up, sprint review, and the sprint retrospective.  Scrum masters also look to resolve impediments and distractions for the development team, insulating them from external disruptions whenever possible. |
| Development Team: **Sadok Laouissi** |
| Development team is the champion for sustainable development practices. The most effective development team is tight-knit and co-located. Team members have differing skill sets, and cross-train each other so no one person becomes a bottleneck in the delivery of work. All members of the team help one another to ensure a successful sprint completion. |

Table 1: The Scrum Team

#### Scrum Events

Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum.[11]

* Sprint Planning:

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Sprint Planning answers the following:

Topic One: What can be done this Sprint?

Topic Two: How will the chosen work get done?

* Daily Scrum Meeting

The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours.

During the meeting, the Development Team members explain:

What did I do yesterday that help the Development Team meet the Sprint Goal?

What will I do today to help the Development Team meet the Sprint Goal?

Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

* Sprint Review

A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed. Based on that and any changes to the Product Backlog during the Sprint, attendees collaborate on the next things that could be done to optimize value.

* Sprint retrospective

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning.

The purpose of the Sprint Retrospective is to:

Inspect how the last Sprint went with regards to people, relationships, process, and tools.

Identify and order the major items that went well and potential improvements.

Create a plan for implementing improvements to the way the Scrum Team does its work.

#### Scrum Artifacts

* Product Backlog

The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

* Sprint Backlog

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a “Done” Increment. The Sprint Backlog makes visible all the work that the Development Team identifies as necessary to meet the Sprint Goal.

## Project Management

### Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | As a(n) | I want to | So that I can | Priority |
|  | Administrator | Authenticate | access the platform | 17 |
|  | Administrator | Add categories | Add its products | 1 |
|  | Administrator | Edit categories | Maintain the app design & functionalities | 2 |
|  | Administrator | Delete Categories | Restrict categories shown | 3 |
|  | Administrator | View Categories | Chose which to update/delete | 4 |
|  | Administrator | View Complaints | Update complaint state | 39 |
|  | Administrator | Update complaint state | (user) Visualize complaint progress | 40 |
|  | Administrator | View Users | View users’ global data | 18 |
|  | Administrator | View Allergens | Visualize allergens data | 6 |
|  | Administrator | Add Allergens | Add allergens to products | 7 |
|  | Administrator | Associate allergens to ingredients | (user) Visualize allergens in ingredients | 8 |
|  | Administrator | Edit Allergens | Change allergens data | 9 |
|  | Administrator | Delete Allergens | Disable allergen | 10 |
|  | Administrator | View Products | Visualize Products data | 24 |
|  | Administrator | Delete Products | Delete unwanted Products | 25 |
|  | Administrator | Enable/disable Product | Un/Restrict for users | 26 |
|  | Administrator | Add Product | (user) Gets latest products | 27 |
|  | Administrator | Associate ingredients to products | (user) Gets product’s ingredients | 28 |
|  | Administrator | Add ingredients | Associate ingredients to products | 12 |
|  | Administrator | View ingredients | View ingredients’ data | 13 |
|  | Administrator | Edit ingredients | Edit ingredients’ data | 14 |
|  | Administrator | Delete ingredients | Delete unused ingredients | 15 |
|  | User | Authenticate | Access the mobile App | 19 |
|  | User | Sign up | Create an account | 20 |
|  | User | Validate account | Enable account | 21 |
|  | User | Take diagnosis | Associate to a skin type | 41 |
|  | User | Add personal allergens | Check product’s compatibility | 11 |
|  | User | Update profile information | Edit user’s profile | 22 |
|  | User | Update password | Secure account | 23 |
|  | User | Add search history | View account history | 36 |
|  | User | Delete search history | Remove unwanted search | 37 |
|  | User | View search history | Track account activity | 38 |
|  | User | Add complaint | Adapt user experience | 43 |
|  | User | View complaints | Visualize complaint progress | 44 |
|  | User | Add to favourite | Save preferred products | 34 |
|  | User | Delete from favourite | Remove unwanted products | 35 |
|  | User | View all categories | Get products based on selected category | 5 |
|  | User | View all products types | Get products based on selected types | 31 |
|  | User | View all products | View products data | 30 |
|  | User | Search all products | Find wanted products | 29 |
|  | User | Filter products | Get products based on selected filters | 32 |
|  | User | View product details | View product’s data | 33 |
|  | User | View product ingredients | Visualize product’s ingredients | 16 |
|  | User | View suggestions | Get similar, popular and newest products | 42 |

Table 2: Product Backlog

### Sprint Planning

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Number | Duration | Start Date | Finish Date |
| Sprint 1 : Categories Management | 15 days | 18/03/2021 | 02/04/2021 |
| Sprint 2 : Allergens Management | 17 days | 05/04/2021 | 22/04/2021 |
| Sprint 3 : Ingredients Management | 21 days | 23/04/2021 | 14/05/2021 |
| Sprint 4 : Users Management | 25 days | 17/05/2021 | 11/06/2021 |
| Sprint 5 : Products Management | 25 days | 14/06/2021 | 09/07/2021 |
| Sprint 6 : Favorite & History Management | 26 days | 12/07/2021 | 09/08/2021 |
| Sprint 7: Complaints & Sugguestions Management | 27 days | 10/08/2021 | 08/09/2021 |

Table 3: Sprints Planning

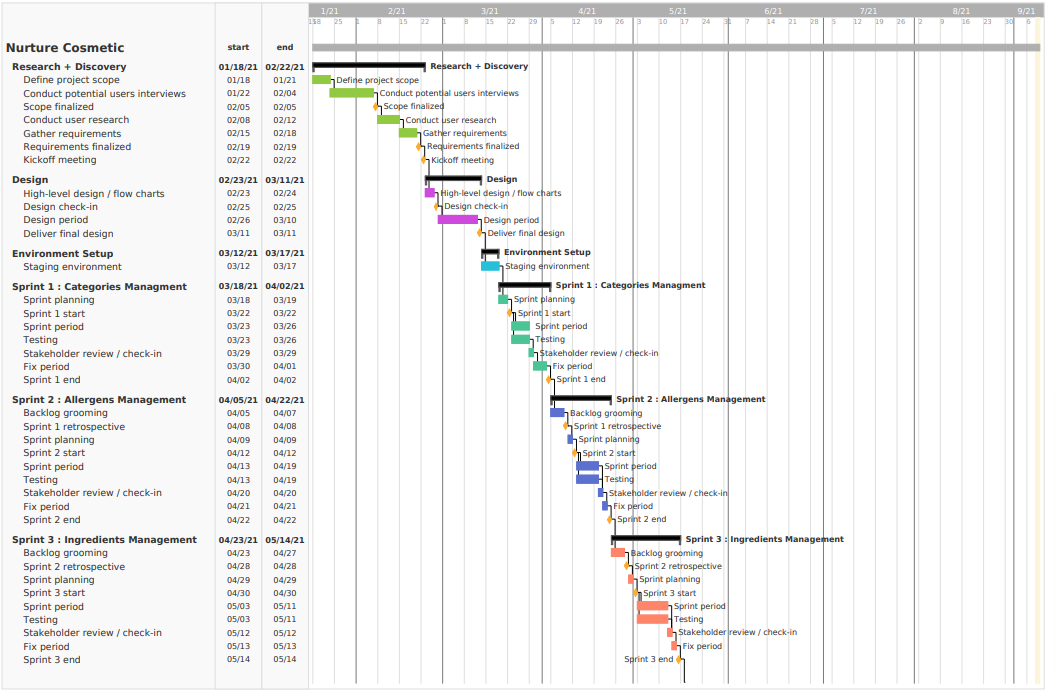


Figure 14 : Gantt Diagram Part 1

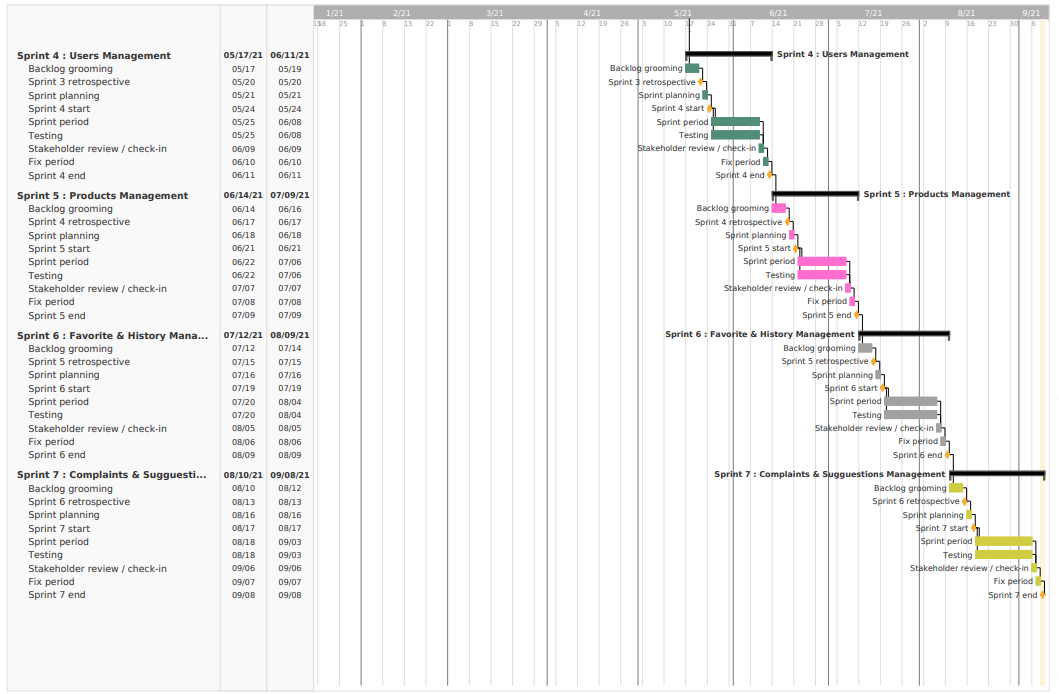


Figure 15 : Gantt Diagram Part 2



# Chapter III: State of the art

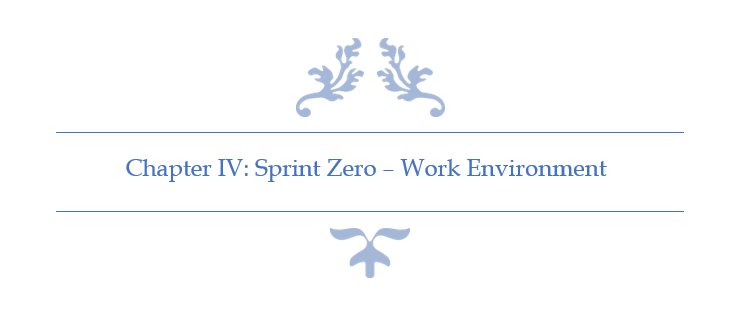
## Introduction

On this part, we are going to explore cloud computing, OpenStack, DevStack, OpenStack Rest API and some implementation features.

## Conclusion

We discovered Cloud computer on this chapter as well as OpenStack, DevStack and the REST API. The Chosen library used in our situation is Openstack4j. It’s well developed, easy to use and contains various services which will be very helpful to understand OpenStack Components.

On the next chapter, we are going to present the work environment, Hardware and software and the technologies used for the implementation.



# Chapter IV: Sprint Zero - Work Environment

## Introduction

In this chapter I am going to present the different tools and technology used. First, I am going to describe the development environment (Hardware and Software) as well as the implementation features. Then, I am going to present the application’s physical and logical architecture.

## Work Environment

### Hardware Environment

Table 4: Hardware Environment Characteristics

To develop the application, we used as hardware environment 2 laptops and 2 desk- computers which have the following characteristic:

|  |  |  |
| --- | --- | --- |
| Hardware | Laptop | Desktop |
| Mark | HP-Pavilion Gaming 15 | Dell |
| Processor | 2.2 GHz Intel Core i7 | 2.5 GHz Intel Core i7 |
| Storage | 250 To SSD & 1 To HDD | 1 To HDD |
| RAM | 20 Go | 16 Go |
| Operating System | Windows 10 | Windows 10 |

### Software Environment

#### Android Studio v4.2

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development. (DBpedia, s.d.)

Figure 16: Android Studio Logo

#### Visual Studio Code v1.55.2

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 (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). (Microsoft, s.d.)

Figure 17: VS Code Logo

#### Postman v8.3.1

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so we can create better APIs—faster. (Postman, s.d.)

Figure 18: Postman Logo

#### WampServer v3.2.0

WampServer refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language. (Wikipedia, s.d.)

Figure 19: WampServer Logo

#### Visual Paradigm Online

Visual Paradigm is a leading and globally recognized provider for Business and IT Transformation software solutions. It enables organizations to improve business and IT agility and foster innovation through popular open standards. Our award-winning products are trusted by over 320,000 users in companies ranging from small business, consultants, to blue chip organizations, universities and government units across the globe. (Visual Paradigm Online, s.d.)

Figure 20: Visual Paradigm Online Logo

#### Adobe XD

Adobe XD (also known as Adobe Experience Design) is a vector-based user experience design tool for web apps and mobile apps, developed and published by Adobe Inc. It is available for macOS and Windows, although there are versions for iOS and Android to help preview the result of work directly on mobile devices. Adobe XD supports website wireframing and creating click-through prototypes. (Wikipedia, s.d.)

Figure 21: Adobe XD Logo

#### Adobe Photoshop

Photoshop is a computer-assisted editing, processing and drawing software, launched in 1990 on MacOS and then in 1992 on Windows. Published by Adobe, it is mainly used for the processing of digital photographs, but also for the ex-nihilo creation of images. (Wikipe, s.d.)

Figure 22: Adobe Photoshop Logo

### Project Management Tools

#### GitHub Desktop

GitHub Desktop is an application that enables you to interact with GitHub using a GUI instead of the command line or a web browser. GitHub Desktop encourages you and your team to collaborate using best practices with Git and GitHub. You can use GitHub Desktop to complete most Git commands from your desktop with visual confirmation of changes. You can push to, pull from, and clone remote repositories with GitHub Desktop, and use collaborative tools such as attributing commits and creating pull requests. (Github, s.d.)

Figure 23: GitHub Desktop Logo

#### Teamgantt status reportTeamGantt

TeamGantt’s free gantt chart software lets you keep all your tasks, documents, and conversations in a single centralized hub. Storing important information together in your gantt chart tool means you and your team can always find your project's latest files, chats, and updates. (TeamGantt, s.d.)

Figure 24: TeamGantt Online Logo

### Choice of implementation

#### Programming language

##### Dart

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications.

Figure 25: Dart Logo

Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. Dart can compile to either native code or JavaScript. It supports interfaces, mixins, abstract classes, reified generics, and type inference. (Gentile, 2020)

##### TypeScript

TypeScript is a strongly typed programming language which builds on JavaScript giving us better tooling at any scale.

TypeScript may be used to develop JavaScript applications for both client-side and server-side execution (as with Node.js). (Tutorials Point, s.d.)

Figure 26: TypeScript Logo

#### Database

##### MySQL

MySQL is a freely available open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).

Figure 27: MySQL Logo

SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MySQL is an essential part of almost every open-source PHP application. Good examples for PHP & MySQL-based scripts are WordPress, Joomla, Magento and Drupal. (Site Ground, s.d.)

#### Framework

Figure 28: Angular Logo

##### Angular

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks of the Angular framework are Angular components that are organized into NgModules. NgModules collect related code into functional sets; an Angular application is defined by a set of NgModules. An application always has at least a root module that enables bootstrapping, and typically has many more feature modules. (Angular, s.d.)

##### Node JS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. (Wikipedia, s.d.)

Figure 29: Node JS Logo

##### Flutter

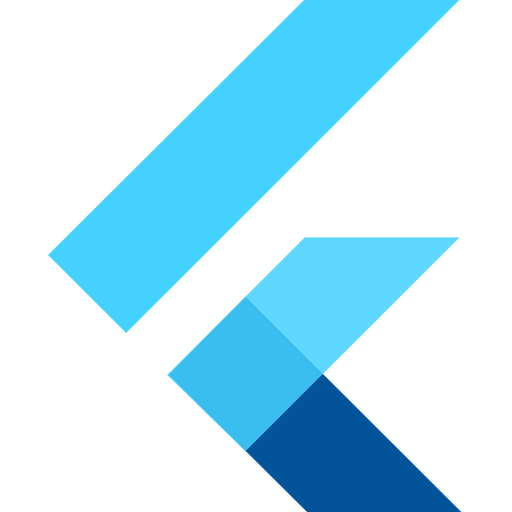
Flutter is Google’s free and open-source UI framework for creating native mobile applications. Released in 2017, Flutter allows developers to build mobile applications for both iOS and Android with a single codebase and programming language. This capability makes building iOS and Android apps simpler and faster. (Perfecto, 2021)

Figure 30: Flutter Logo

The Flutter framework consists of both a software development kit (SDK) and their widget-based UI library. This library consists of various reusable UI elements, such as sliders, buttons, and text inputs. (Perfecto, 2021)

Developers building mobile applications with the Flutter framework will do so using a programming language called Dart. With a syntax like JavaScript, Dart is a typed object programming language that focuses on front-end development. (Perfecto, 2021)

#### REST API vs. SOAP

REST operates through a solitary, consistent interface to access named resources. It’s most commonly used when you’re exposing a public API over the Internet. SOAP, on the other hand, exposes components of application logic as services rather than data. Additionally, it operates through different interfaces. To put it simply, REST accesses data while SOAP performs operations through a more standardized set of messaging patterns. Still, in most cases, either REST or SOAP could be used to achieve the same outcome (and both are infinitely scalable), with some differences in how you’d configure it. (ALTVATER, 2017)

#### Data Format

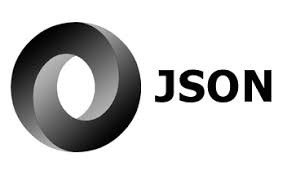
JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. (JSON, s.d.)

Figure 31: JSON Logo

## Application Architecture

### Physical Architecture

My application is based on 3-tiers Architecture as explained in the following figure. The first part is the Website. It’s the project’s operation room. Only Admins have access to the website through his own credentials. It can manage the main parts shown in the mobile app. The second part is the Server which contains the servers web and the application’s back office and contains all the controller. It is also the link between the application’s different parts. The third and the last part is the mobile app which contains products & categories stored in the web server and created through the administration website. The web server communicates with both parts through its own routes.

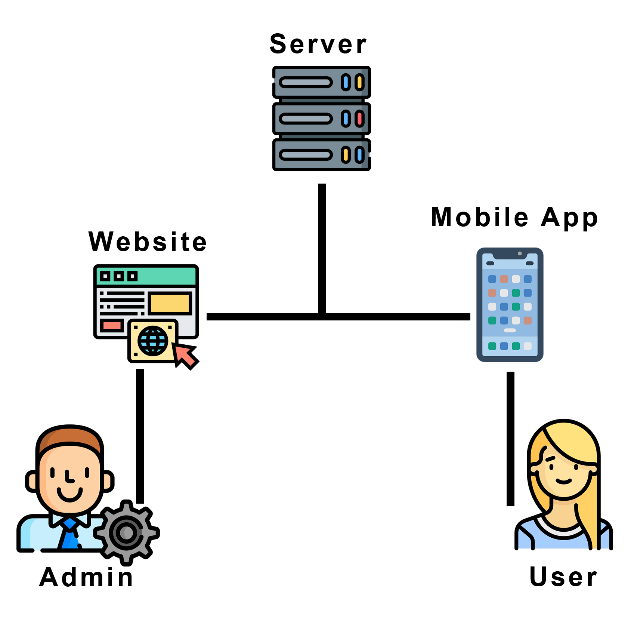


Figure 32 : Application's Physical Architecture

### Logical Architecture

In this project I used MEAN Stack but instead of MongoDB I used MySQL (MySQL, ExpressJS, Angular, NodeJS) for the back-office part and Flutter for the front-office.

First, I used MVVM (Model, View, View Model) for the administration dashboard (Angular Project) which allows us to enable the two-way data binding between the View and the View Model.

Second, I used “BloC” architecture for the mobile app (Flutter) which helps organize data flow. It allows us to keep different layers of our application separate which makes it easy to test and reuse elements in different parts of code.

And last, I used “Single Threaded Event Loop” architecture for the web server (NodeJS) which allows us to handle multiple concurrent clients based on its JavaScript event-based model.

The following figure explains the Pattern used in my situation.

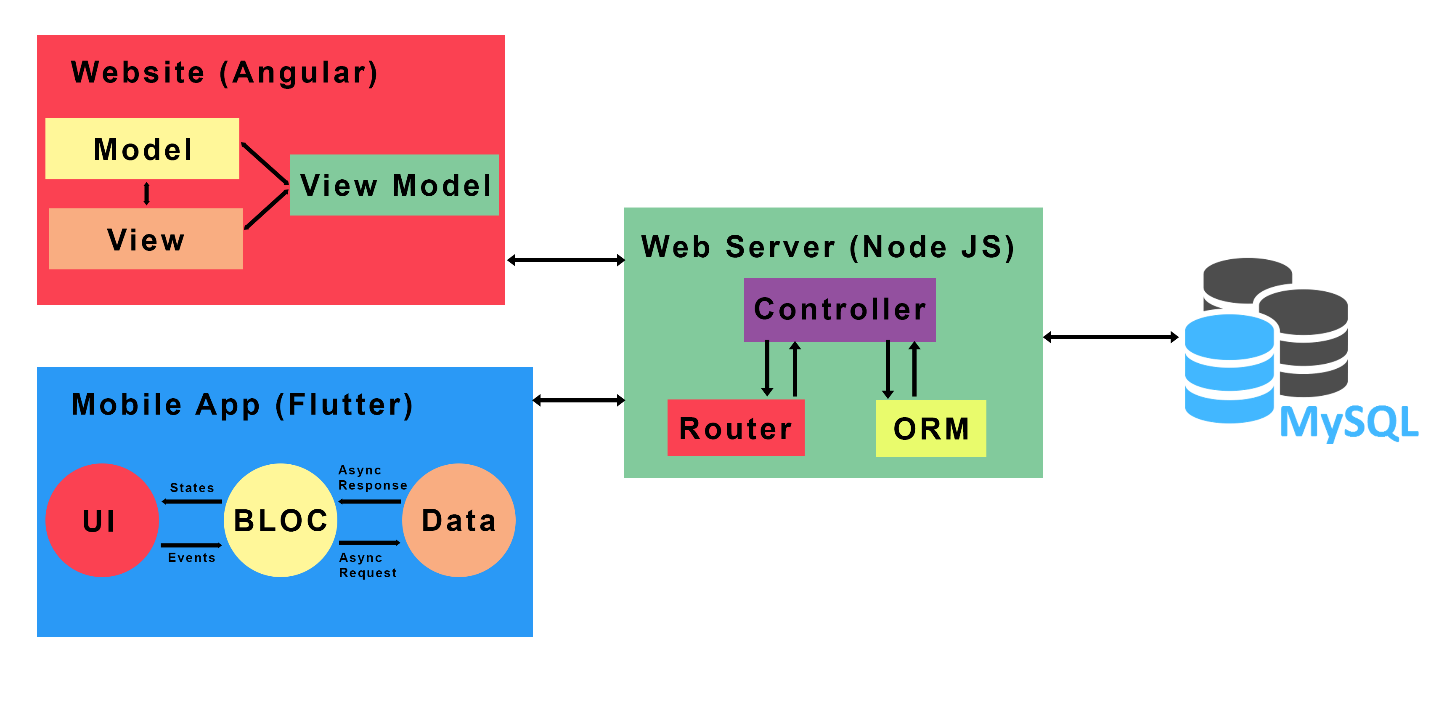


Figure 33 : Logical Architecture

## Conclusion

In this chapter, I described the hardware and software platforms on which I built our applications. Then, I presented the different technologies used in the realization. Also, I presented the application’s physical and logical architecture.

On the next chapter, I am going to start the development of the first sprint.



# Chapter V: Sprint 1 – Category Management

## Introduction

The purpose of this sprint is to discover, design, develop and test all features related to Categories.

## Sprint Backlog

### Sprint Scope

In this sprint, our main goal is to implement category service which allows the user to search for products based on their categories and it also allows the administrator to add, edit and delete categories.

### Tasks Planning

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Day number | | | | | | | | | | |
| User stories# | Tasks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2  &  3  &  4  &  5 | Database design and implementation | 6 | 0 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 0 |
| API implementation | 0 | 6 | 2 | 2 | 0 | 2 | 4 | 2 | 2 | 0 |
| Persistence development | 0 | 0 | 4 | 2 | 5 | 1 | 2 | 0 | 0 | 0 |
| Front end development | 0 | 0 | 0 | 2 | 2 | 3 | 0 | 4 | 2 | 4 |
| Test | 2 | 2 | 2 | 0 | 1 | 2 | 0 | 2 | 2 | 0 |
| Documentation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 |
| User stories# | Tasks | 11 | 12 | 13 | 14 | 15 |
| 6 | Database design and implementation | 4 | 4 | 0 | 2 | 0 |
| API implementation | 2 | 2 | 4 | 2 | 3 |
| Persistence development | 0 | 0 | 2 | 2 | 2 |
| Front end development | 0 | 0 | 0 | 2 | 1 |
| Test | 2 | 2 | 2 | 0 | 2 |
| Documentation | 0 | 0 | 0 | 0 | 0 |

Table 5: Tasks Planning - Sprint 1

## Analysis

### Categories Management Use Case Refinement

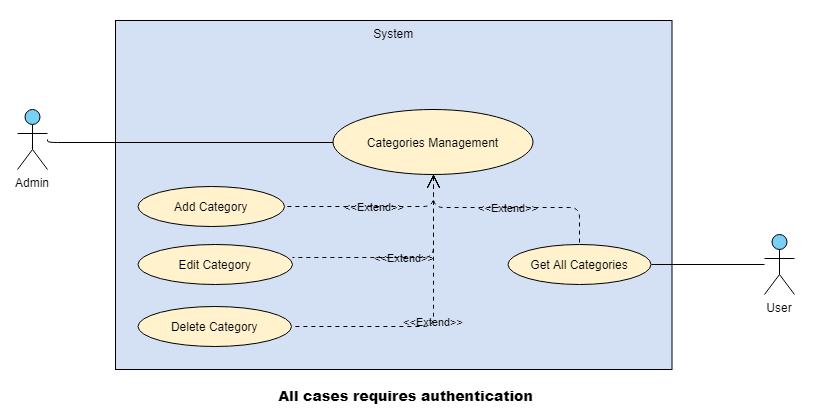


Figure 34: Categories Management Use Case Refinement

#### Use case “Add Category”

##### Textual description of the use case “Add Category”

The below table explain the “success” scenario of adding a new category.

|  |  |
| --- | --- |
| Use Case | Add Category |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | New category inserted. |
| Nominal Scenario | * The admin clicks on “Categories” dropdown on the sidebar. * The admin clicks on “Add Category” button on the sidebar. * The system shows the desired interface. * The admin fills the form and clicks on “Confirm” button. |

Table 6: Textual description of the use case “Add Category”

##### Sequence diagram of the use case “Add Category”

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

The below figure is the sequence diagram of add Category:

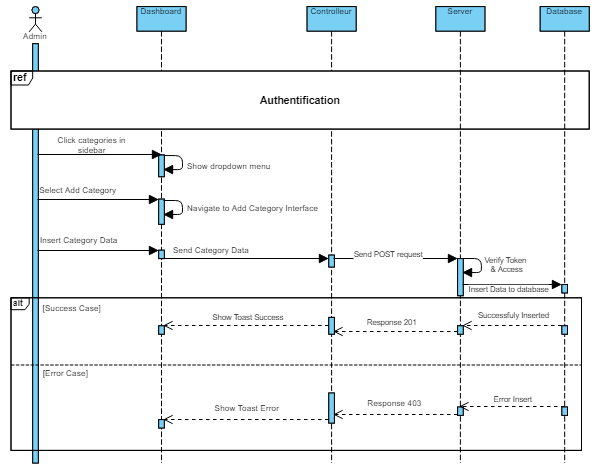


Figure 35: Add Category Sequence Diagram

##### Screen Captures

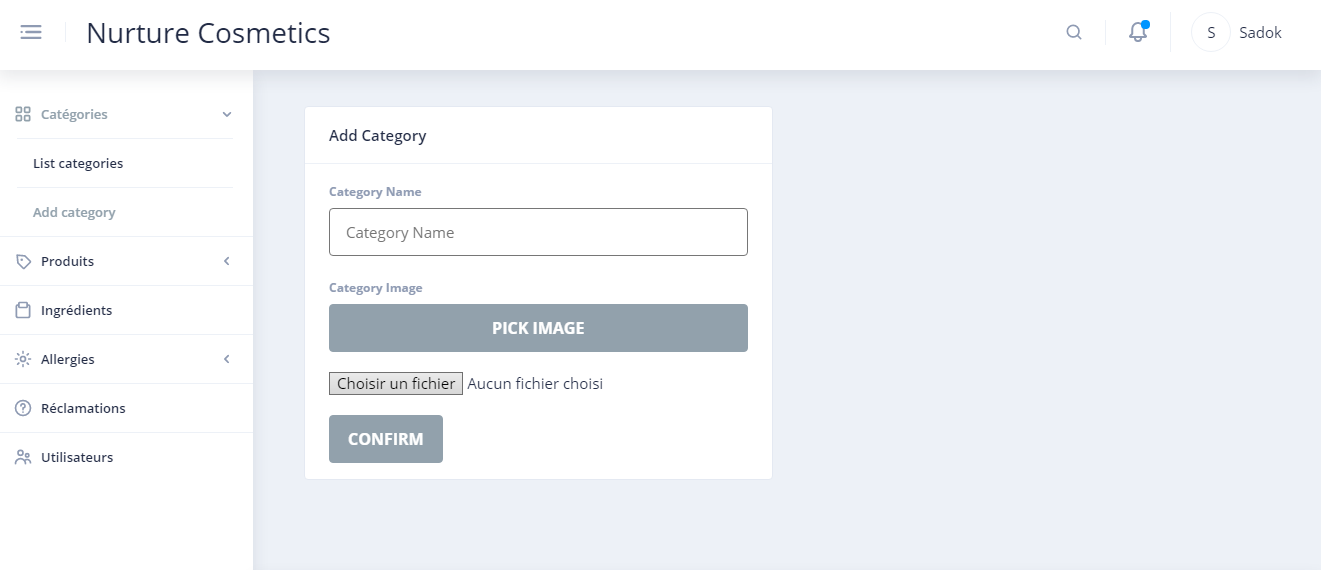


Figure 36: Screen Capture Add Category Interface

#### Use case “Edit Category”

##### Textual description of the use case “Edit Category”

The below table explain the “success” scenario of editing a category.

|  |  |
| --- | --- |
| Use Case | Edit Category |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | Category updated. |
| Nominal Scenario | * The admin clicks on “Categories” dropdown on the sidebar. * The admin clicks on “List Categories” button on the sidebar. * The system shows the desired interface. * The admin clicks on “pencil” button on the desired category. * The admin fills the form and clicks on “Check-mark” button. |

Table 7: Textual description of the use case “Edit Category”

##### Sequence diagram of the use case “Edit Category”

The below figure is the sequence diagram of Edit Category:

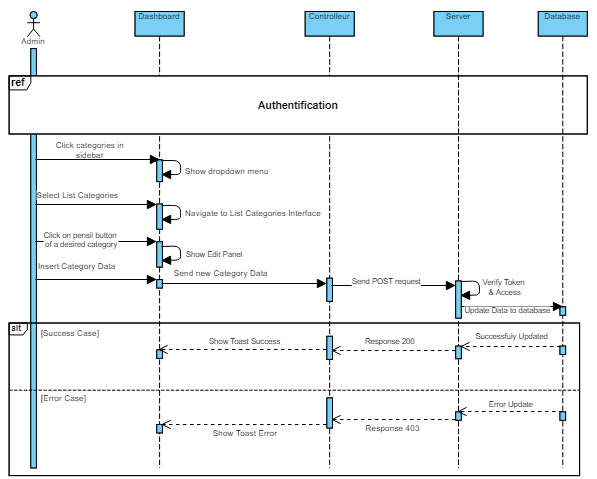


Figure 37: Edit Category Sequence Diagram

##### Screen Captures



Figure 38: Screen Capture Edit Category Interface

#### Use case “Delete Category”

##### Textual description of the use case “Delete Category”

The below table explain the “success” scenario of deleting a category.

|  |  |
| --- | --- |
| Use Case | Delete Category |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | Category deleted. |
| Nominal Scenario | * The admin clicks on “Categories” dropdown on the sidebar. * The admin clicks on “List Categories” button on the sidebar. * The system shows the desired interface. * The admin clicks on “close” icon on the desired category. * The admin confirms delete request. |

Table 8: Textual description of the use case “Delete Category”

##### Sequence diagram of the use case “Delete Category”

The below figure is the sequence diagram of Delete Category:

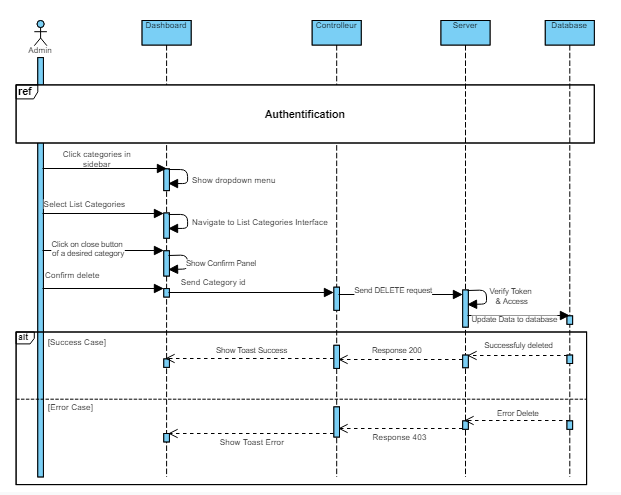


Figure 39: Delete Category Sequence Diagram

##### Screen Captures

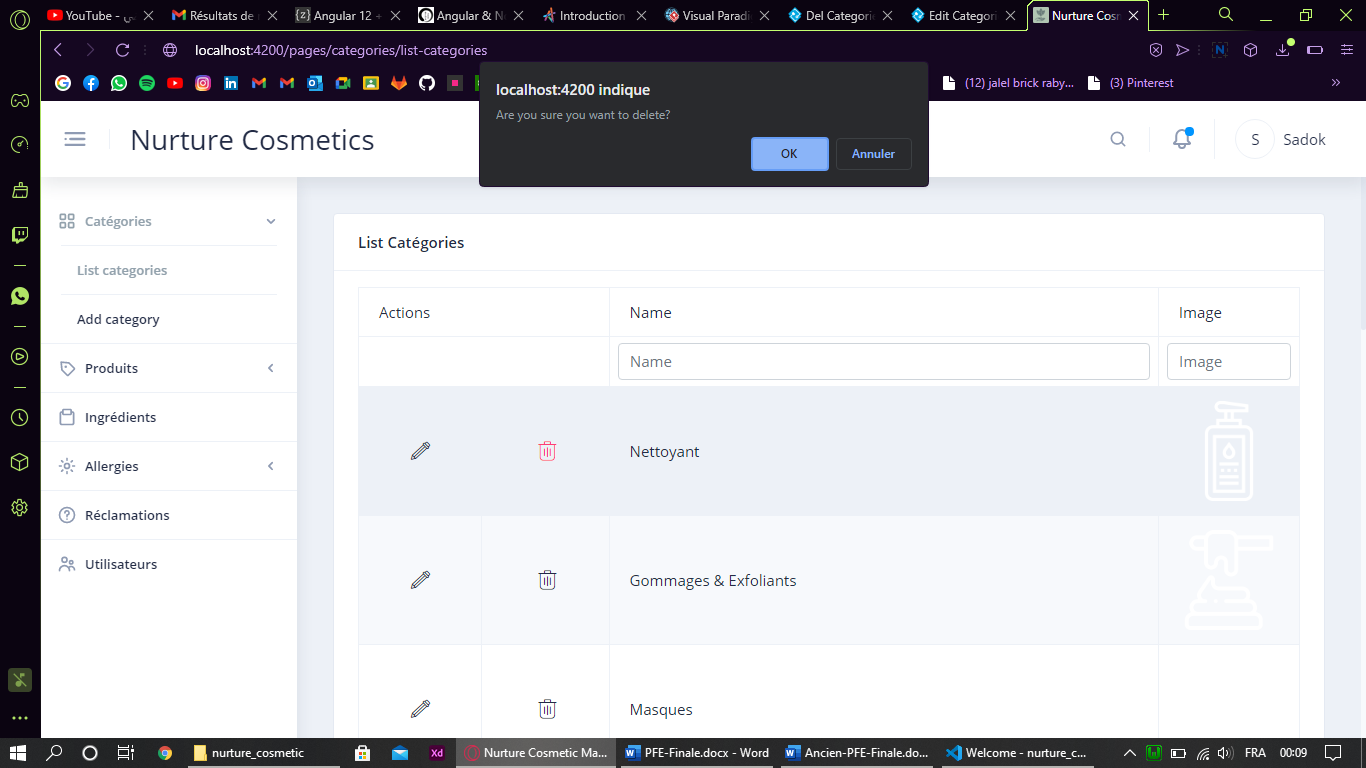


Figure 40: Screen Capture Delete Category

#### Use case “List All Categories (Admin)”

##### Textual description of the use case “List All Categories (Admin)”

The below table explain the “success” scenario of get all categories.

|  |  |
| --- | --- |
| Use Case | List All Categories |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Nominal Scenario | * The admin clicks on “Categories” dropdown on the sidebar. * The admin clicks on “List Categories” button on the sidebar. * The system shows the desired interface. |

Table 9: Textual description of the use case “List Categories”

##### Screen Captures

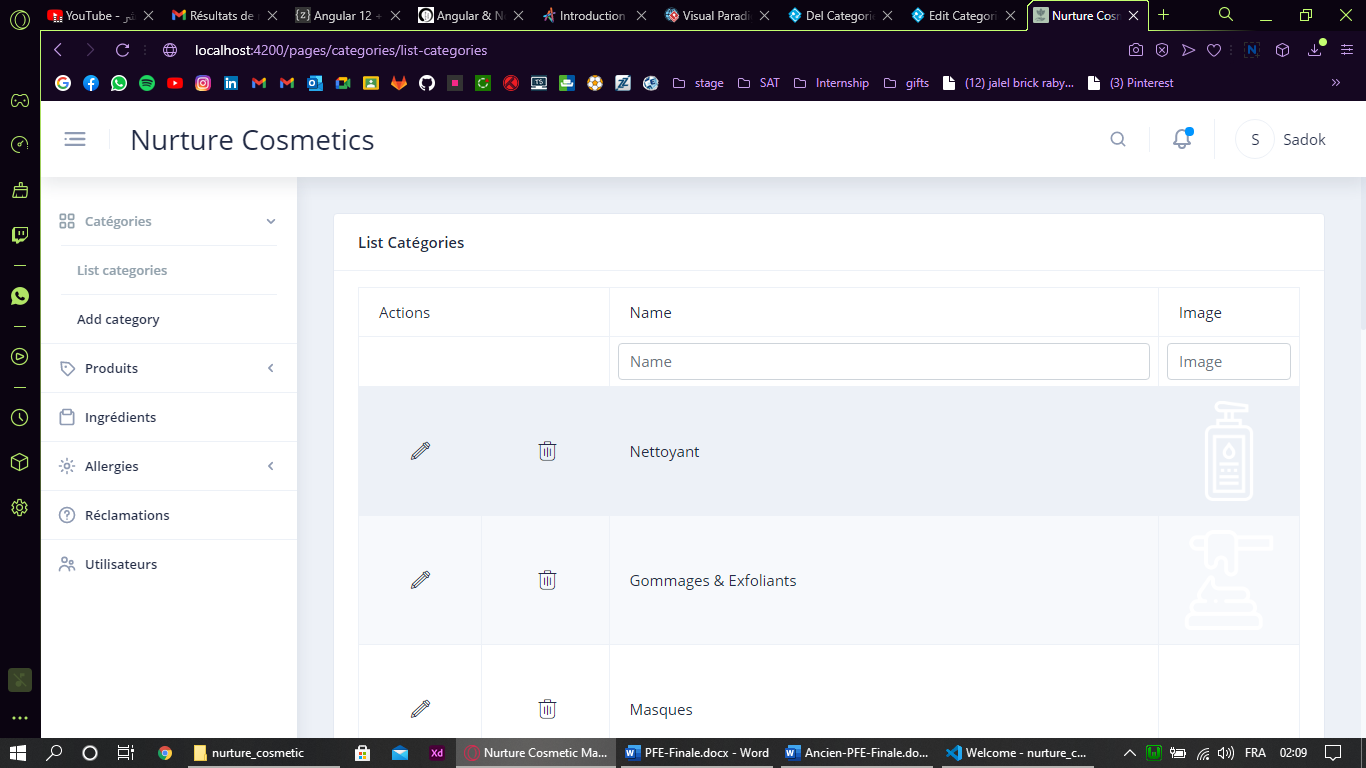


Figure 41: Screen Capture List Categories Interface

#### Use case “List All Categories (User)”

##### Textual description of the use case “List All Categories (User)”

The below table explain the “success” scenario of get all categories.

|  |  |
| --- | --- |
| Use Case | List All Categories |
| Actor | User |
| Pre-condition | The user must be authenticated. |
| Nominal Scenario | * The system shows all categories on home screen. |

Table 10: Textual description of the use case “List Categories (User)”

##### Screen Captures

The pictures below show the design made using Adobe XD and the realisation:



Figure 42: Home Screen Implementation



Figure 43: Home Screen Design

## Conclusion

All categories’ features are now implemented and tested.



# Chapter VI: Sprint 2 – Allergens Management

## Introduction

On this sprint our main objective is to implement all allergens methods such as Create, Update, Delete, View… Allergen’s service is very important for this project because it allows users to select their personal allergens which help them with the search for products.

## Sprint Backlog

### Sprint Scope

On this sprint, our main task is the management of allergens and their association with ingredients.

### Tasks Planning

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Day number | | | | | | | | | | | | | | | | |
| User stories# | Tasks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 6 -- 11 | Database design and implementation | 4 | 4 | 2 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| API implementation | 2 | 2 | 4 | 2 | 2 | 4 | 3 | 1 | 4 | 5 | 2 | 2 | 0 | 0 | 1 | 0 | 0 |
| Persistence development | 0 | 0 | 0 | 4 | 4 | 2 | 2 | 3 | 1 | 3 | 1 | 0 | 0 | 0 | 0 | 4 | 1 |
| Front end development | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 2 | 3 | 4 | 2 | 3 | 4 |
| Test | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 3 | 0 | 0 | 3 | 2 | 1 | 1 | 4 | 1 | 3 |
| Documentation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 3 | 1 | 0 | 0 |

Table 11: Tasks Planning - Sprint 2

## Analysis

### Allergens Management Use Case Refinement



Figure 44: Allergens Management Use Case Refinement

#### Use case “Add Allergen”

##### Textual description

The below table explain the “success” scenario of adding a new allergen.

|  |  |
| --- | --- |
| Use Case | Add Allergen |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | New Allergen inserted. |
| Nominal Scenario | * The admin clicks on “Allergens” dropdown on the sidebar. * The admin clicks on “Add Allergen” button on the sidebar. * The system shows the desired interface. * The admin fills the form and clicks on “Confirm” button. |

Table 12: Textual description of the use case “Add Allergen”

##### Screen Capture

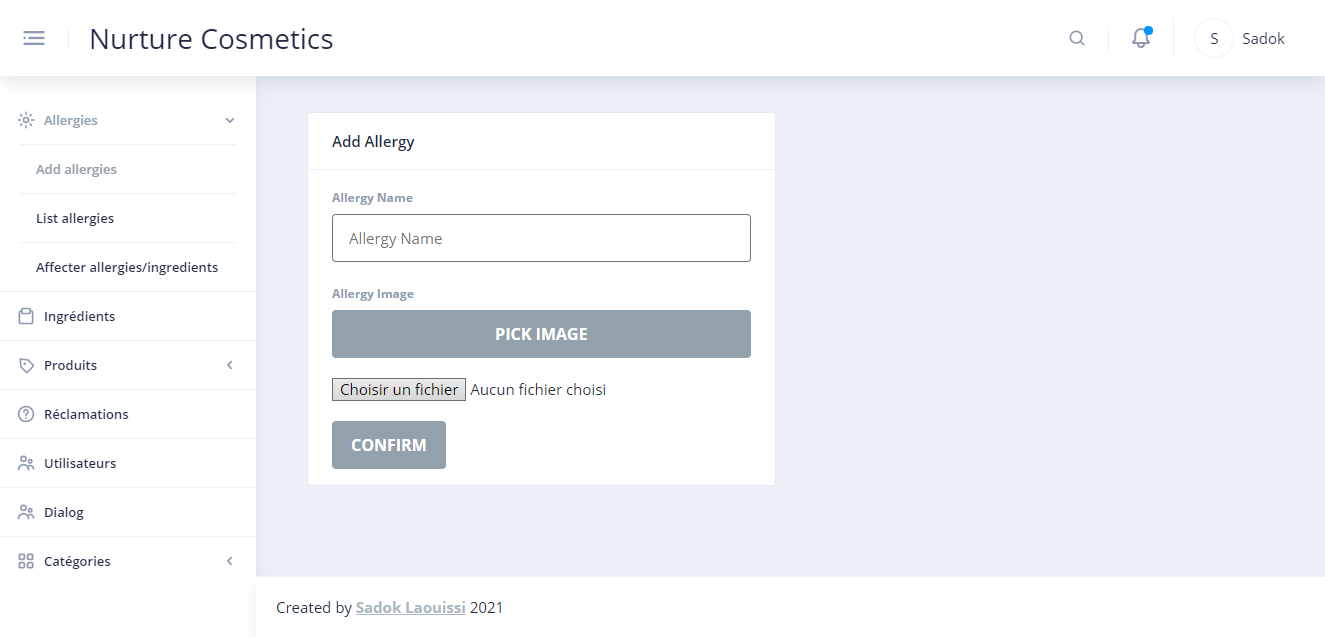


Figure 45: Screen Capture Add Allergen Interface

#### Use case “Edit Allergen”

##### Textual description

The below table explain the “success” scenario of editing an allergen.

|  |  |
| --- | --- |
| Use Case | Edit Allergen |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | Allergen updated. |
| Nominal Scenario | * The admin clicks on “Allergens” dropdown on the sidebar. * The admin clicks on “List Allergens” button on the sidebar. * The system shows the desired interface. * The admin clicks on “pencil” button on the desired allergen. * The admin fills the form and clicks on “Check-mark” button. |

Table 13: Textual description of the use case “Edit Allergen”

##### Screen Capture

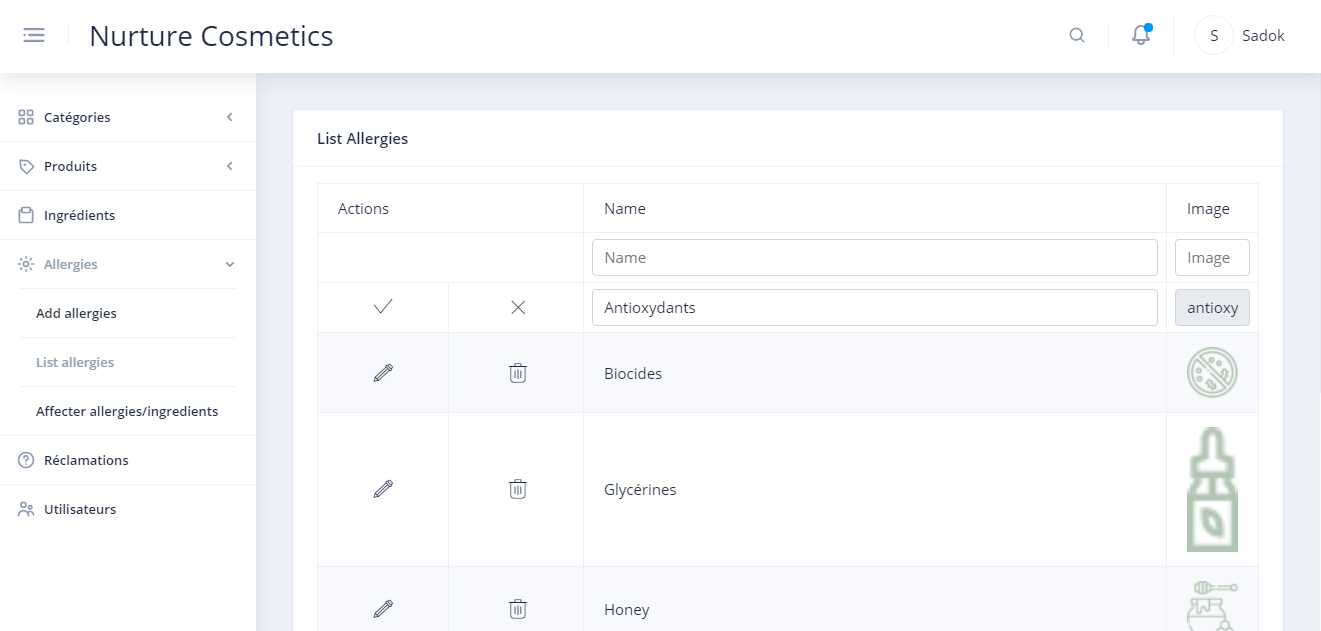


Figure 46: Screen Capture Edit Allergen

#### Use case “Delete Allergen”

##### Textual description

The below table explain the “success” scenario of deleting an allergen.

|  |  |
| --- | --- |
| Use Case | Delete Category |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | Allergen deleted. |
| Nominal Scenario | * The admin clicks on “Allergens” dropdown on the sidebar. * The admin clicks on “List Allergens” button on the sidebar. * The system shows the desired interface. * The admin clicks on “trash” icon on the desired allergen. * The admin confirms delete request. |

Table 14:Textual description of the use case “Delete Allergen”

##### Screen Capture

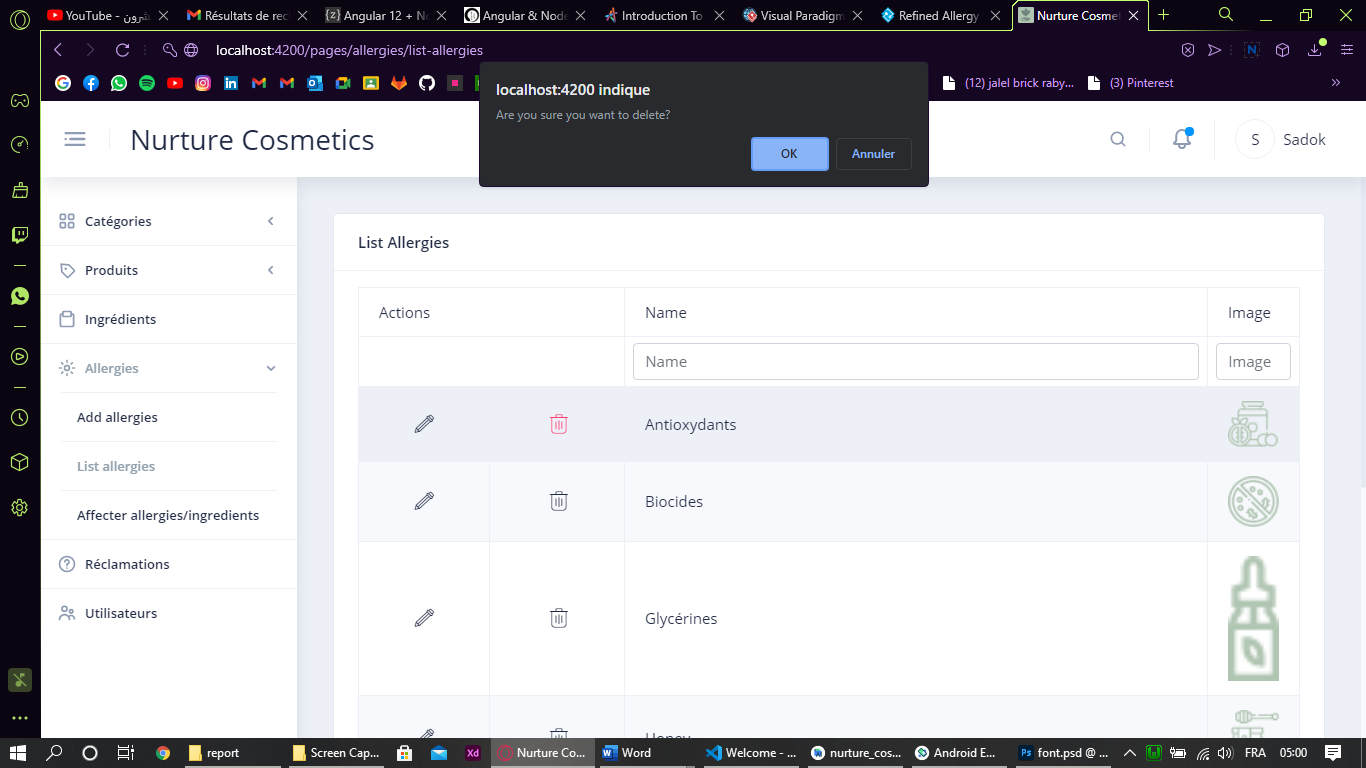


Figure 47: Screen Capture Delete Allergen

#### Use case “Associate Ingredient/Allergen”

##### Textual description

The below table explain the “success” scenario of association an ingredient to an allergen.

|  |  |
| --- | --- |
| Use Case | Associate Ingredient/Allergen |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | Ingredient associated to/removed from allergen. |
| Nominal Scenario | * The admin clicks on “Allergens” dropdown on the sidebar. * The admin clicks on “Associate Ingredient/Allergen” button on the sidebar. * The system shows the desired interface. * The admin clicks on “Select Allergen” dropdown and select the desired allergen. * The system shows its provoking ingredients and all other ingredients. * The admin clicks on plus icon to add other ingredients or trash icon to remove from its current ingredients. |

Table 15: Textual Description Associate Ingredient/allergen

##### Sequence diagram of the use case Associate Ingredient/Allergen

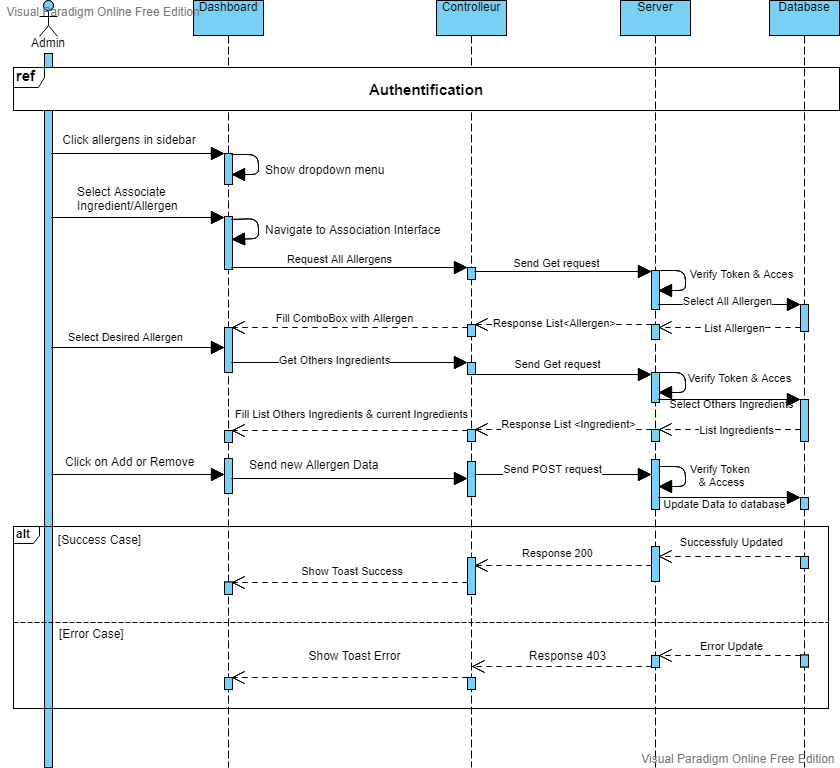


Figure 48: Sequence diagram of the use case "Associate Ingredient/Allergen"

##### Screen Capture

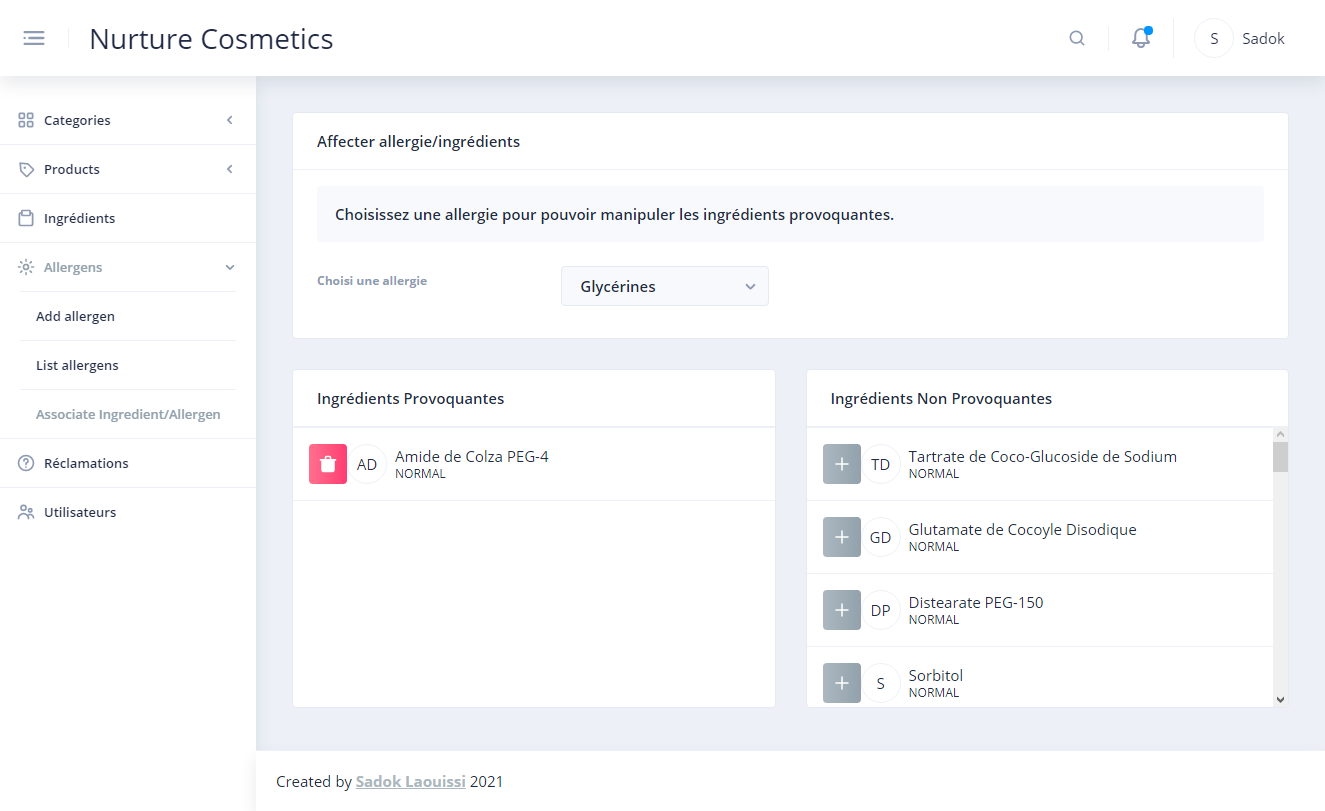


Figure 49: Screen Capture Associate Ingredient/Allergen Interface

#### Use case “Edit Personal Allergen”

##### Textual description

The below table explain the “success” scenario of association an ingredient to an allergen.

|  |  |
| --- | --- |
| Use Case | Edit Personal Allergen |
| Actor | User |
| Pre-condition | The user must be authenticated or new account verified. |
| Post-condition | User’s allergens updated. |
| Nominal Scenario 1  (Case Authenticated) | * The user clicks on “Drawer Icon” button. * The system shows the drawer. * The user clicks on “Parameters” button. * The system shows the Parameters Interface. * The user clicks on “Mon Compte” button. * The system shows the Account Interface. * The user clicks on “Modifier allergens” button. * The system shows Allergen Interface. * The user clicks on “clipboard” icon. * The system shows All Allergen with select box. * The user selects its own allergen. |
| Nominal Scenario 2  (Case Account Verified) | * The system shows All Allergen with select box. * The user selects its own allergen. |

Table 16 : Textual description Edit Personal Allergen

##### Sequence diagram

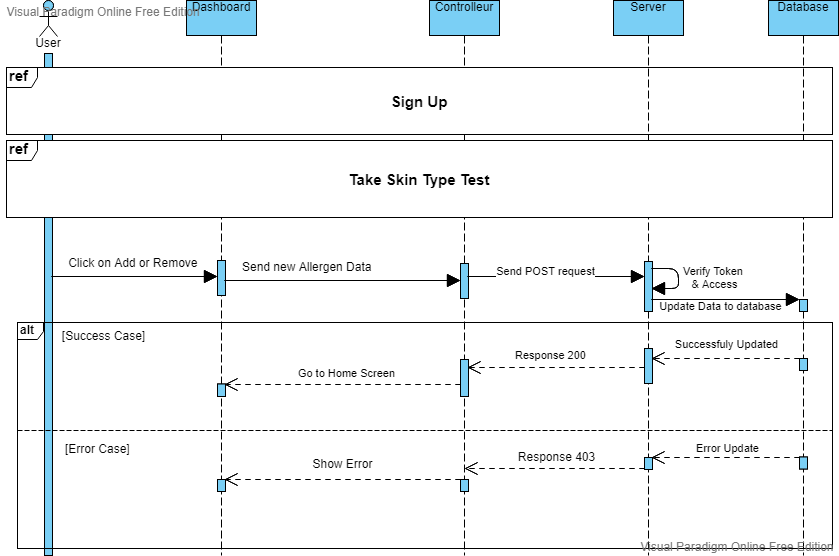


Figure 50: Sequence Diagram Edit Personal Allergen Scenario 2

##### Screen Capture



Figure 51: Personal Allergen Implementation



Figure 52: Design Adobe XD: Personal Allergens

## Conclusion

On this sprint, we made it possible to users to edit their personal allergies so they can see which product is convenient for them based on allergens in product’s ingredients. On the next sprint, we are going to implement ingredients management.



# Chapter VII: Sprint 3 – Ingredients Management

## Introduction

The search by personal allergens is based on ingredients’ allergens so this sprint is important for the rest of the project.

## Sprint Backlog

### Sprint Scope

The main goal of this sprint is to implement the management of the ingredients for the back office and the visualisation for the app users.

### Tasks Planning

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Day number | | | | | | | | | | |
| User stories# | Tasks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 12  &  13  &  14 | Database design and implementation | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 0 | 0 | 0 |
| API implementation | 6 | 6 | 2 | 2 | 0 | 2 | 4 | 2 | 2 | 0 |
| Persistence development | 0 | 0 | 4 | 2 | 5 | 1 | 2 | 0 | 0 | 0 |
| Front end development | 0 | 0 | 0 | 2 | 2 | 3 | 0 | 4 | 2 | 4 |
| Test | 2 | 2 | 2 | 0 | 1 | 2 | 0 | 2 | 2 | 0 |
| Documentation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Day number | | | | | | | | | | | |
| User stories# | Tasks | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 15  &  16 | Database design and implementation | 2 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| API implementation | 0 | 2 | 2 | 2 | 6 | 4 | 2 | 2 | 2 | 0 | 0 |
| Persistence development | 5 | 1 | 4 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 |
| Front end development | 2 | 3 | 0 | 2 | 0 | 0 | 4 | 4 | 2 | 4 | 4 |
| Test | 1 | 2 | 2 | 0 | 2 | 0 | 2 | 2 | 2 | 0 | 0 |
| Documentation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 4 |

Table 17: Tasks Planning Sprint 3 - Ingredients Management

## Analysis

### Ingredients Management Use Case Refinement

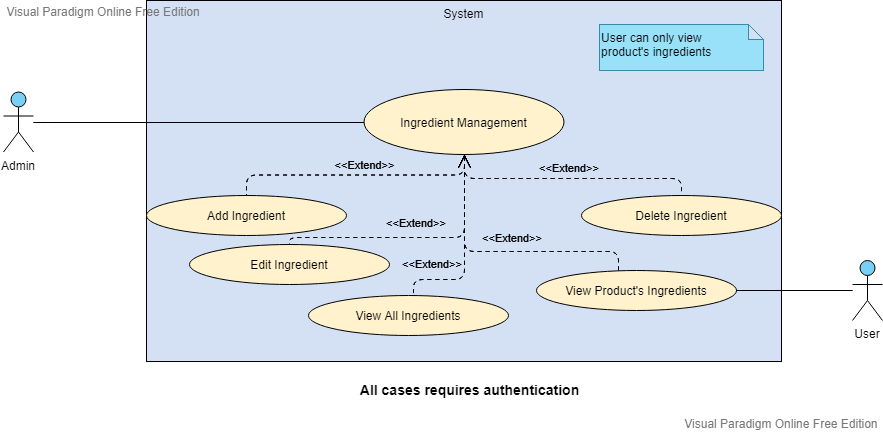


Figure 53 : Ingredients Management Use Case Refinement

#### Use case “View All Ingredients”

##### Textual description

The below table explain the “success” scenario of visualising ingredients.

|  |  |
| --- | --- |
| Use Case | Add Allergen |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Nominal Scenario | * The admin clicks on “Ingredients” on the sidebar. * The system shows the desired interface. |

Table 18: Textual description of the use case “View All Ingredients”

##### Screen Capture

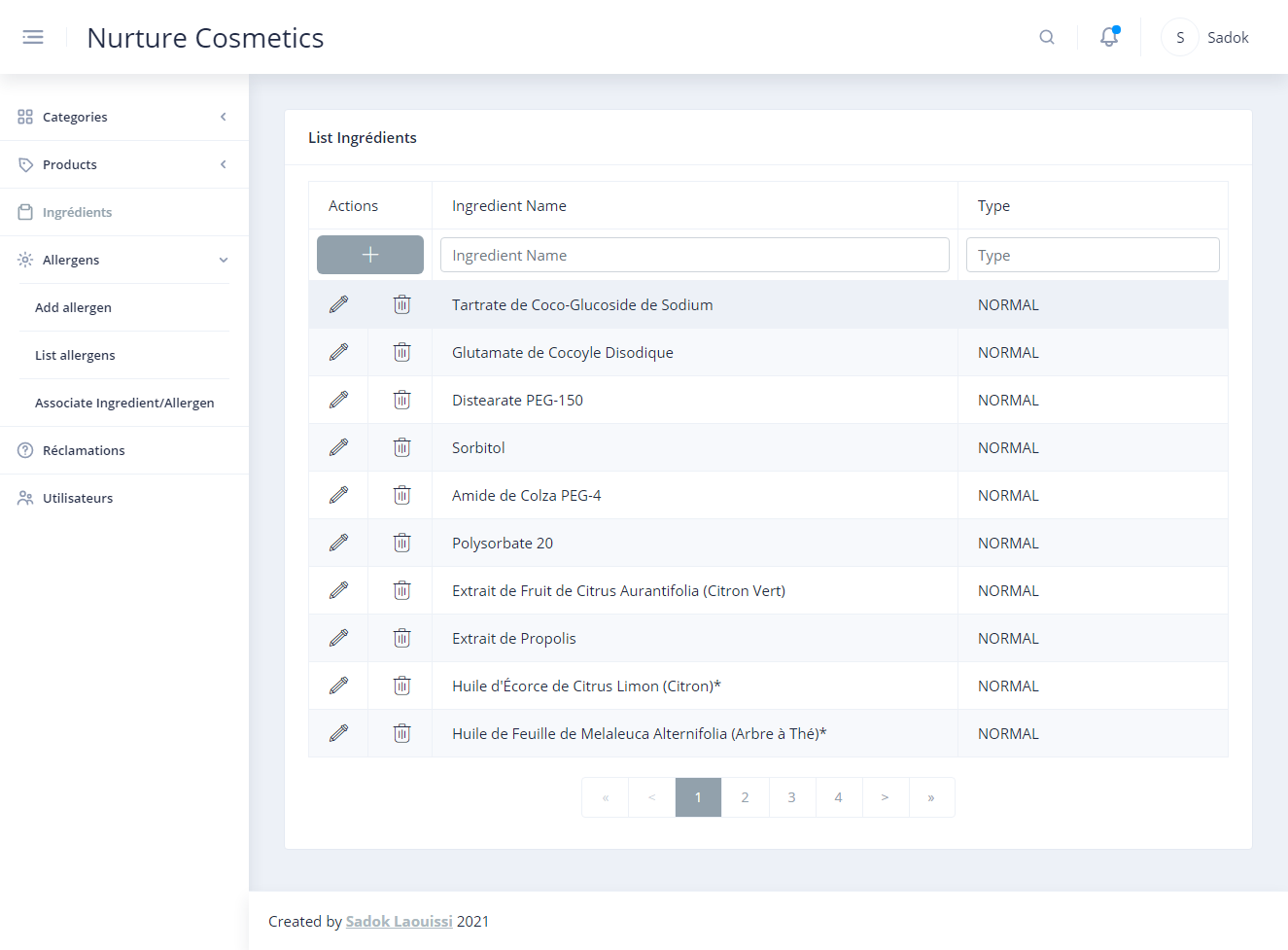


Figure 54: Screen Capture Ingredients Interface

#### Use case “Add Ingredient”

##### Textual description

The below table explain the “success” scenario of adding ingredient.

|  |  |
| --- | --- |
| Use Case | Add Ingredient |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | New ingredient added. |
| Nominal Scenario | * The admin clicks on “Ingredients” on the sidebar. * The system shows the desired interface. * The admin clicks on plus button. * The system shows a new field data * The user inserts ingredient name and select the type. * The user confirm insert. * The system shows a toast based on result. |

Table 19: Textual description of the use case “Add Ingredient”

##### Sequence diagram

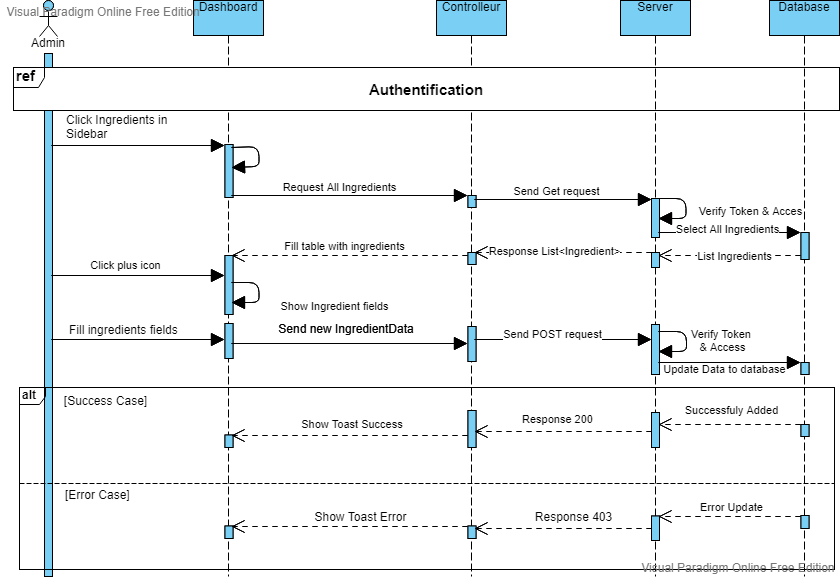


Figure 55: Sequence diagram Add Ingredient

##### Screen Capture

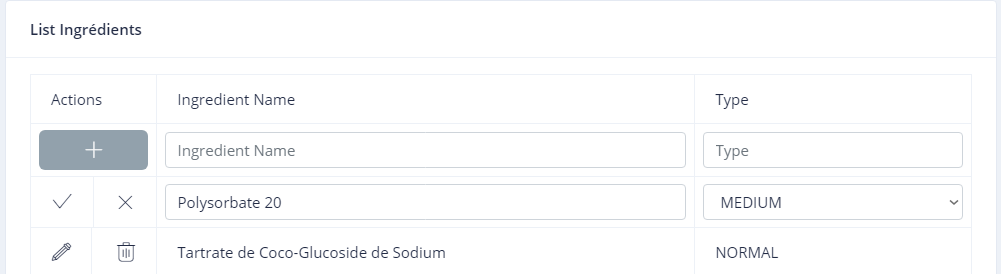


Figure 56: Screen Capture Add Ingredient

#### Use case “Edit Ingredient”

##### Textual description

The below table explain the “success” scenario of editing an ingredient.

|  |  |
| --- | --- |
| Use Case | Edit Ingredient |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | Ingredient edited. |
| Nominal Scenario | * The admin clicks on “Ingredients” on the sidebar. * The system shows the desired interface. * The admin clicks on pencil icon on desired ingredient. * The system shows a current field data * The user edits ingredient name and select the type. * The user confirms editing. * The system shows a toast based on result. |

Table 20: Textual description of the use case “Edit Ingredient”

##### Screen Capture

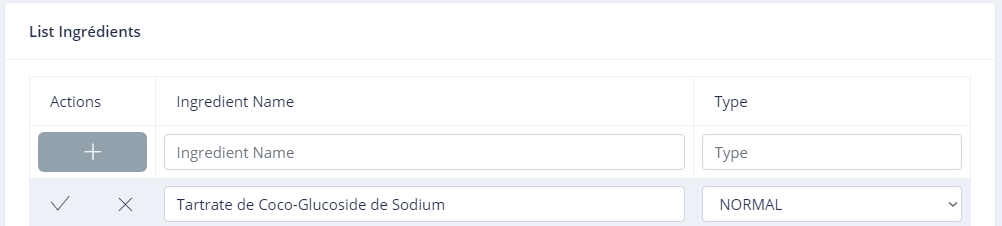


Figure 57: Screen Capture Edit Ingredient

#### Use case “Delete Ingredient”

##### Textual description

The below table explain the “success” scenario of deleting an ingredient.

|  |  |
| --- | --- |
| Use Case | Delete Ingredient |
| Actor | Admin |
| Pre-condition | The administrator must be authenticated. |
| Post-condition | Ingredient deleted. |
| Nominal Scenario | * The admin clicks on “Ingredients” on the sidebar. * The system shows the desired interface. * The admin clicks on trash icon on desired ingredient. * The system shows confirmation popup. * The user confirms delete. * The system shows a toast based on result. |

Table 21: Textual description of the use case “Delete Ingredient”

##### Screen Capture

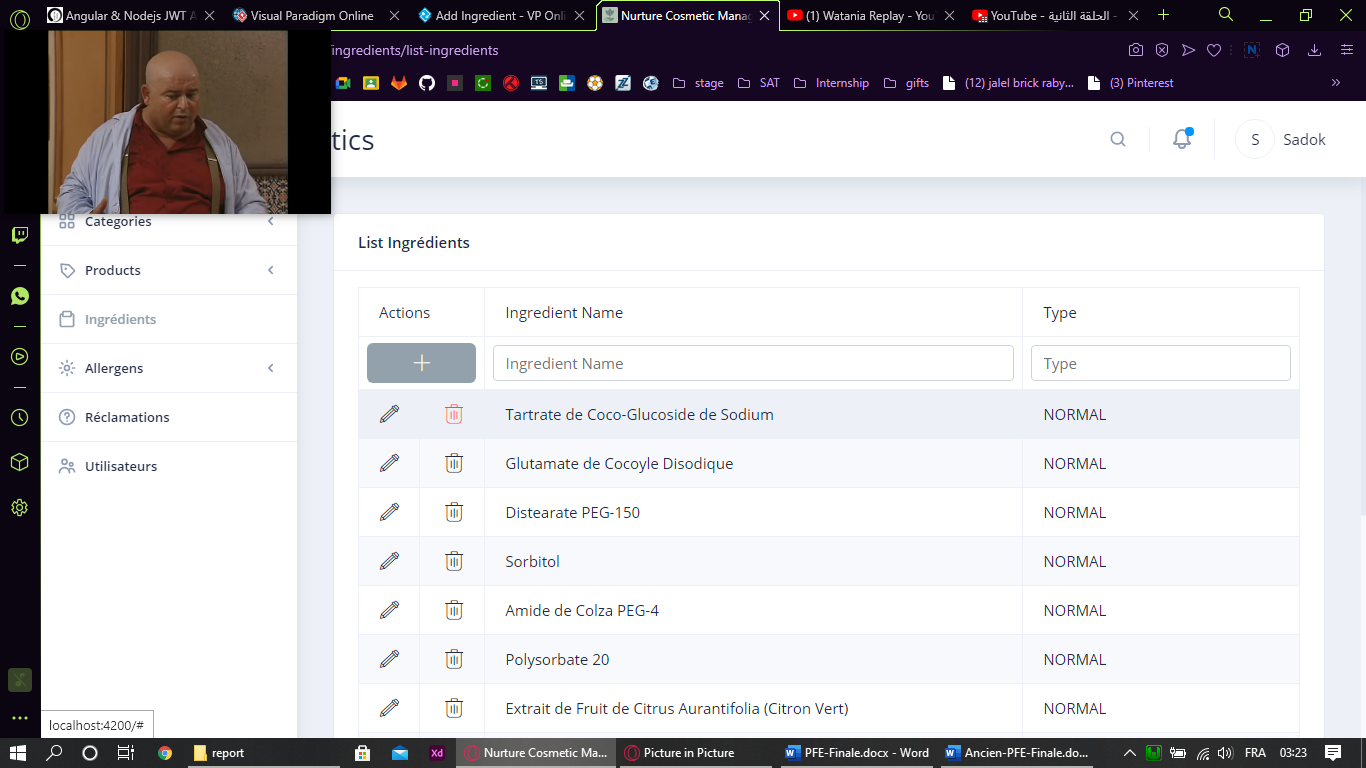


Figure 58: Screen Capture Delete Ingredient

#### Use case “View Product Ingredients”

##### Textual description

The below table explain the “success” scenario of viewing product’s ingredients.

|  |  |
| --- | --- |
| Use Case | View Product Ingredients |
| Actor | User |
| Pre-condition | The user must be authenticated. |
| Nominal Scenario | * The user clicks on desired product. * The system shows product details. * The user swipe left on “Plus Informations”. * The system shows product’s ingredients. |

Table 22: Textual description of the use case “View Product's Ingredients”

##### Screen Captures



Figure 59: Ingredients Screen Realisation



Figure 60: Design Adobe XD Ingredients Screen

## Conclusion

Ingredients management is now done and users will be able to view product’s ingredients. The next sprint we are going to implement user’s services.



# Chapter VIII: Sprint 4 – Users Management

# Conclusion & Perspectives

The misuse of IT resources, the lack of control over computer, and especially the time waste during the balancing between different operating systems could all be resolved using cloud computing tools for provisioning and monitoring. That’s the idea behind our project.

To realize this solution, we used the essential framework Openstack4j and Apache Camel to use their APIs to communicate with DevStack that we had already installed using command line. Then we implemented our MySQL database using the Hibernate framework since we’re using JEE. These is overall the technologies that we used to build our web application.

After succeeding to build the app, we’re able to communicate with Keystone to manage users, roles, courses, and tokens, we were able also to communicate with Glance to manage OS images, and finally Compute to communicate with flavors, virtual machines, and floating IPs.[2]

For better results and improvements, implementing the service Magnum would be a good feature. Magnum is responsible for orchestrating an OS image and run it, in either virtual machines or bare metal in a cluster configuration.[24]

In other words, if we have many virtual machines using the same operating system we could use containers to minimize the use of resources by using the same OS to build VMs on it .

# Reflection

When I started this project, I knew nothing about OpenStack or even the cloud computing technology, it was quite challenging to learn about the process and to find the right framework to work with. But here we are now, documenting our work and planning for new features to improve our work.

At the beginning, it was tough to understand the concept of cloud computing and how to implement it in our work. It was the first time to work with these technologies. We spend a lot of time doing researches, information wasn’t easy at all. The concept of provisioning and monitoring cloud is new in Tunisia, and even worldwide, developing it using java wasn’t an easy choice, since python was already implemented in these kind of application, after all OpenStack is written in python.

For sure, we didn’t regret our choice, we feel very mature and more confident after achieving this project, we know we did something very challenging. We believe after this experience, we’re going to dig more and more in the cloud computing and why not build something very big and innovative for Tunisia.

We can now look back and realize that this experience has helped us both as students and as young professionals.[1]

# Table of Acronyms and Abbreviations

|  |  |
| --- | --- |
| API | Application Programming Interface |
| JEE | **Java** Platform, **Enterprise Edition** |
| JSON | **JavaScript Object Notation** |
| IaaS | **Infrastructure as a service** |
| SaaS | **Software as a service** |
| PaaS | **Platform as a service** |
| URI | **Uniform Resource Identifier** |
| VM | **Virtual Machine** |
| DB | **Database** |
| XML | **Extensible Markup Language** |
| UI | **User Interface** |
| REST | **Representational State Transfer** |

# References

ALTVATER, A., 2017. *SOAP VS REST.* [Online]   
Available at: https://stackify.com/soap-vs-rest/

Angular, n.d. *Introduction to Angular concepts.* [Online]   
Available at: https://angular.io/guide/architecture

Anon., 2005. *Software and Services in Large Enterprises.* [Online]   
Available at: www.forrester.com/Research/Document/Excerpt/0,7211,38659,00.html

DBpedia, n.d. *Android Studio.* [Online]   
Available at: https://dbpedia.org/page/Android\_Studio

Gentile, V., 2020. *Dart (programming language).* [Online]   
Available at: https://www.codemotion.com/magazine/Glossary/dart-programming-language/

Github, n.d. *Getting started with GitHub Desktop.* [Online]   
Available at: https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop

JSON, n.d. *Introducing JSON.* [Online]   
Available at: https://www.json.org/json-en.html

Microsoft, n.d. *Getting Started.* [Online]   
Available at: https://code.visualstudio.com/docs

Perfecto, 2021. *What Is the Flutter Framework?.* [Online]   
Available at: https://www.perfecto.io/blog/what-is-flutter-framework

Postman, n.d. *The Postman API Platform.* [Online]   
Available at: https://www.postman.com/api-platform/

Site Ground, n.d. *What is SQL and MySQL.* [Online]   
Available at: https://www.siteground.com/tutorials/php-mysql/mysql/

TeamGantt, n.d. [Online]   
Available at: https://www.teamgantt.com

Tutorials Point, n.d. *TypeScript - Overview.* [Online]   
Available at: https://www.tutorialspoint.com/typescript/typescript\_overview.htm

Visual Paradigm Online, n.d. *About Us.* [Online]   
Available at: https://online.visual-paradigm.com/fr/about-us/

Wikipedia, n.d. *Adobe XD.* [Online]   
Available at: https://en.wikipedia.org/wiki/Adobe\_XD

Wikipedia, n.d. *Node.js.* [Online]   
Available at: https://en.wikipedia.org/wiki/Node.js

Wikipedia, n.d. *Wamp Server.* [Online]   
Available at: https://en.wikipedia.org/wiki/WampServer

Wikipe, n.d. *Adobe Photoshop.* [Online]   
Available at: https://www.wikipe.wiki/wiki/fr/Photoshop