TUNISIAN REPUBLIC
*****
MINISTRY OF HIGHER EDUCATION, SCIENTIFIC RESEARCH AND TECHNOLOGY
*****
GENERAL DIRECTORATE OF TECHNOLOGICAL STUDIES
*****
HIGHER INSTITUTE OF TECHNOLOGICAL STUDIES OF SILIANA
*****
IT TECHNOLOGY DEPARTMENT

End of Study Project Report

Developed with a view to obtaining the diploma of

Applied License in Computer Technologies

Course

Information System Development

Subject

Development and Implementation of a Web Service and Innovative Software Applications for Customer Loyalty in the Field of Catering and Retail

Directed by: **Ons Khdima Allah**

ISET supervisor: **Ms. Sana Zakraoui**

Company Supervisor: **Mr Slim Zouaghia**

Host organization: **PROXICLIC**

Academic year: 2023/2024

Dedications

To those who supported us in times of need,

To those who believed in our potential,

To those who encouraged us to become who we are today,

To those who shared our joys and sorrows,

To those who wished us well,

To our fathers, mothers, sisters, brothers, friends, and family,

To all those who await our success,

We dedicate, with pride and love, the fruit of our success. May God bless you all.

Ons Khadim Allah

Acknowledgements

We would like to express our gratitude to all those who helped us to accomplish our task and who gave us all the attention we needed to prepare this work.

In particular, we would like to thank our two supervisors, who we consider ourselves fortunate to have had the privilege of working with them, and who have accepted with great

and guide us in our work.

Ms. Sana Zakraoui, our pedagogical supervisor, who was very patient with us and helped us a great deal with her knowledge, skills and experience.

helped us a great deal with her knowledge, her enriching experience and, above all

and, her availability to ensure that this project was carried out to the best of his ability.

We would also like to thank Mr Slim Zouaghia, our professional supervisor and

for the interest he showed in our subject, and his teaching philosophy was a great help to us.

philosophy was very helpful.

We would also like to thank the host organization PROXI CLIC for giving us the opportunity to be part of their team.

We learned a lot from them and their principles.

Finally, we'd like to thank our teachers, who were a source of inspiration and motivation.

and the members of the jury who also accepted to evaluate our work.

Table of Contents

Chapitre1 : Project Framework

Introduction

In this chapter, we will present the framework of our project by briefly describing

The host company as well as their various functionalities. Next, we will develop a

Study of the existing situation and our proposed solution. The last part of this chapter sets out the methodology adopted and the modeling language to meet the needs, which will give us a big vision before diving into the details.

I. Presentation of the Host Organization

BI DATA, a French company, provides an extensive array of IT solutions for point-of-sale management, particularly in the catering sector. Their reliable and comprehensive POS systems simplify operations for businesses. They offer turnkey packages tailored to various budgets and establishment types, along with a diverse selection of touchscreen cash registers and accessories.

Their cash register software includes features such as data export, customer account management, employee scheduling, and integrated stock management.

ProxiClic is a group of developers affiliated with the company BI DATA in Tunisia, established in March 2013, specializing in the creation of mobile, desktop, and web solutions. Setting itself apart through its bespoke development service, ProxiClic tailors solutions to meet the unique requirements of each client, ensuring excellence and effectiveness throughout the entire project lifecycle, from planning to execution. Moreover, it provides ongoing technical support to resolve post-implementation issues.

Notably, ProxiClic stands out in the development of applications for the restaurant sector, covering a wide range of features, from online ordering to reservation management and loyalty systems.

II. Areas of activity

The business area of our project revolves around the management of loyalty programs in the restaurant or retail sectors.

1. Customer Loyalty Program Notion

structured program aimed at rewarding and retaining customers based on their purchases, interactions, or other behaviors.

2. Points System Notion

A mechanism where customers earn points for their purchases or interactions, which can be redeemed for rewards or discounts.

3. Reward Management Notion

The process of creating, managing, and distributing rewards to customers based on their loyalty program status.

4. Data Analytics Notion

The use of data and analytics to track customer behavior, identify trends, and optimize loyalty programs for better results.

5. Integration Notion

The ability of the loyalty software to integrate with other systems, such as e-commerce platforms, CRM systems, or POS systems, to collect and analyze customer data.

6. Program Administration Notion:

The tools and features provided by the loyalty software to manage the loyalty program, such as setting up rules and tracking program performance.

7. Point-of-Sale Systems and Self-Service Kiosks Notion :

The use of tactile point-of-sale systems and self-service ordering kiosks to facilitate the implementation and management of loyalty programs. These systems allow customers to sign-up, log in, and manage their loyalty points independently, providing a seamless and integrated experience within the retail or restaurant environment.

III. Study of the market

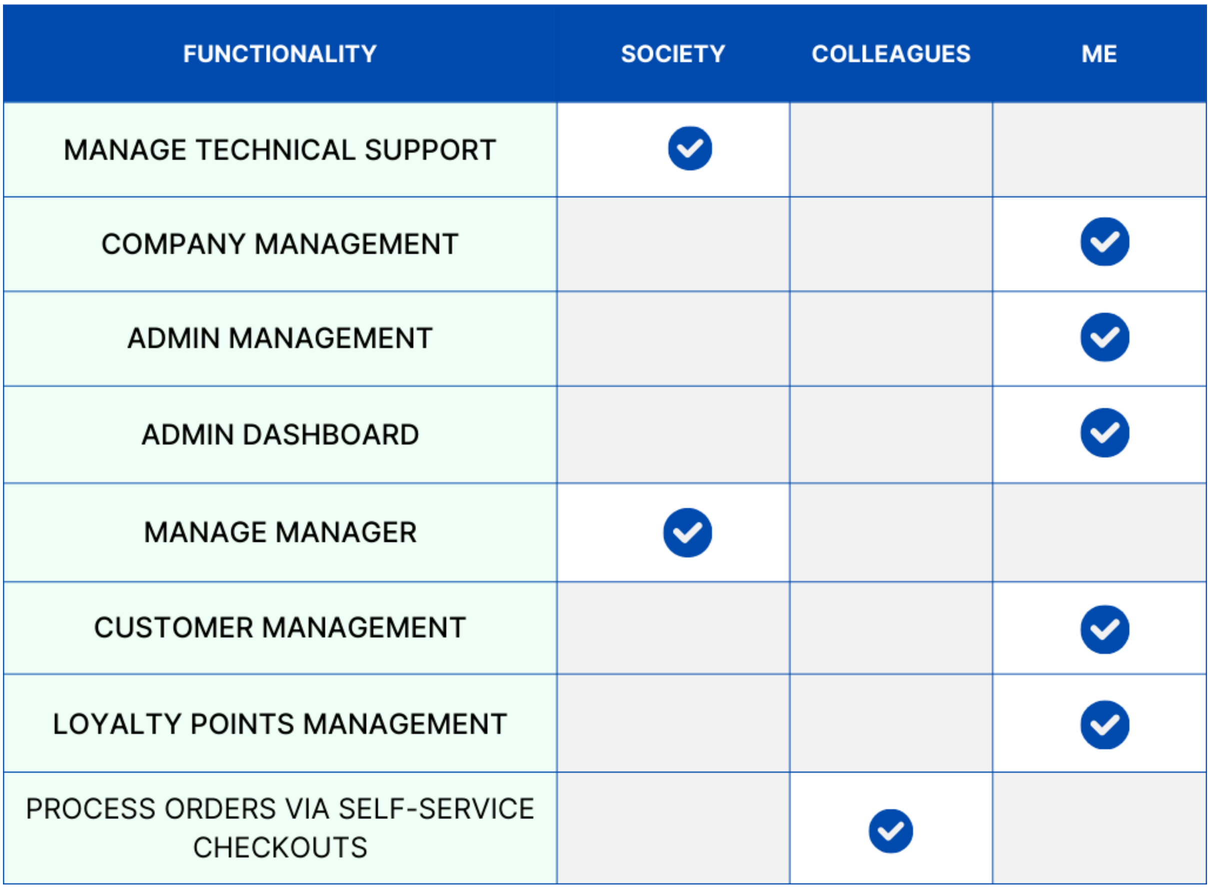
As its name suggests, the study of the market is the analytical phase of a project aimed at understanding market supply and demand to implement the company's sales or commercial strategy. Market research is an essential stage in the development of any IT project.

Currently, customer loyalty in the foodservice sector is primarily handled locally, typically confined to individual establishments. This fragmented method poses notable challenges, such as restricted benefit portability for customers and restaurateurs facing difficulties in attracting and retaining a diverse clientele. Furthermore, the lack of a centralized platform inhibits thorough analysis of customer preferences and behaviors, which is essential for crafting successful loyalty strategies.

IV. Solution proposed

The objective of this initiative is to create an inventive web service that surpasses the constraints of local loyalty systems. This project is to create a comprehensive loyalty points management application tailored for businesses with multiple branches, such as restaurant chains. The proposed solution includes both a web application for administrators and a point-of-sale (POS) system for customer interactions. The functionalities are designed to centralize loyalty program management, facilitate seamless customer interactions, and provide detailed analytics and reporting.

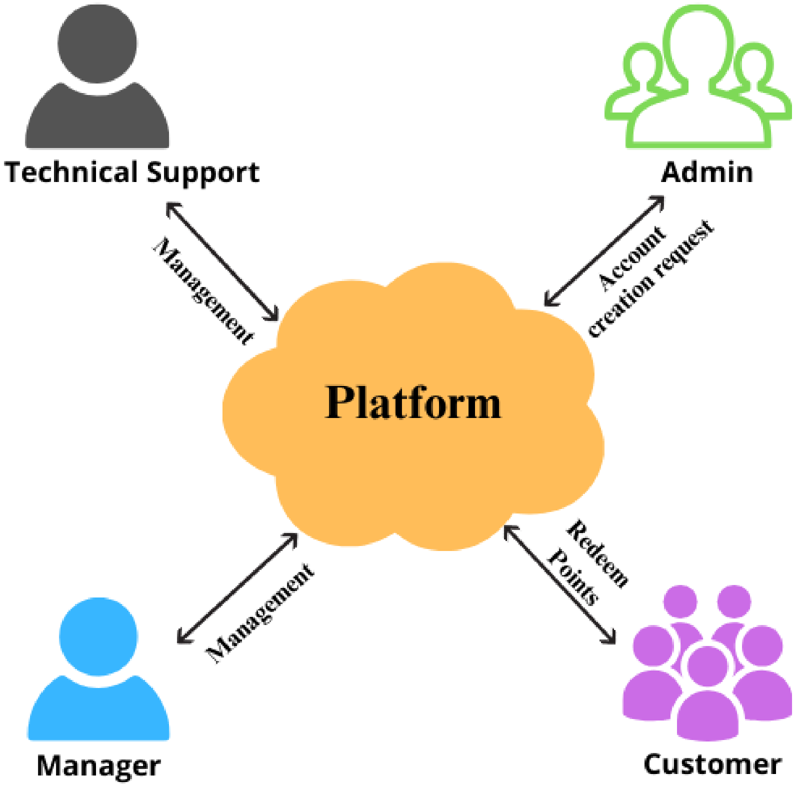
The company has initiated the project, and this schedule clearly indicates the tasks completed before and those I am to complete. It also outlines the responsibilities that my colleagues have already fulfilled.



FUNCTIONALITY	SOCIETY	COLLEAGUES	ME
MANAGE TECHNICAL SUPPORT	✓		
COMPANY MANAGEMENT			✓
ADMIN MANAGEMENT			✓
ADMIN DASHBOARD			✓
MANAGE MANAGER	✓		
CUSTOMER MANAGEMENT			✓
LOYALTY POINTS MANAGEMENT			✓
PROCESS ORDERS VIA SELF-SERVICE CHECKOUTS		✓	

The Application Will Include:

- Account and Company Management
- Admin Dashboard
- Customer Management
- Loyalty Points Management



V. Project Methodologies Adopted

1. Development methodology and modeling language

1.1. Why use a methodology

Generally speaking, a development methodology provides structure to project creation, outlining the lifecycle of the development process to facilitate workflow and consistently meet project deadlines. The advantages of a methodology include:

Improved efficiency: Utilizing the appropriate development methodology enhances efficiency by identifying and prioritizing project objectives from the outset. This ensures that time isn't wasted on irrelevant details and that the entire team is aligned with the same objectives.

Reduced risk: Enhanced project visibility and predictability decrease risk. With clearer insight into team performance and the status of ongoing work, contingency planning becomes more manageable for project managers.

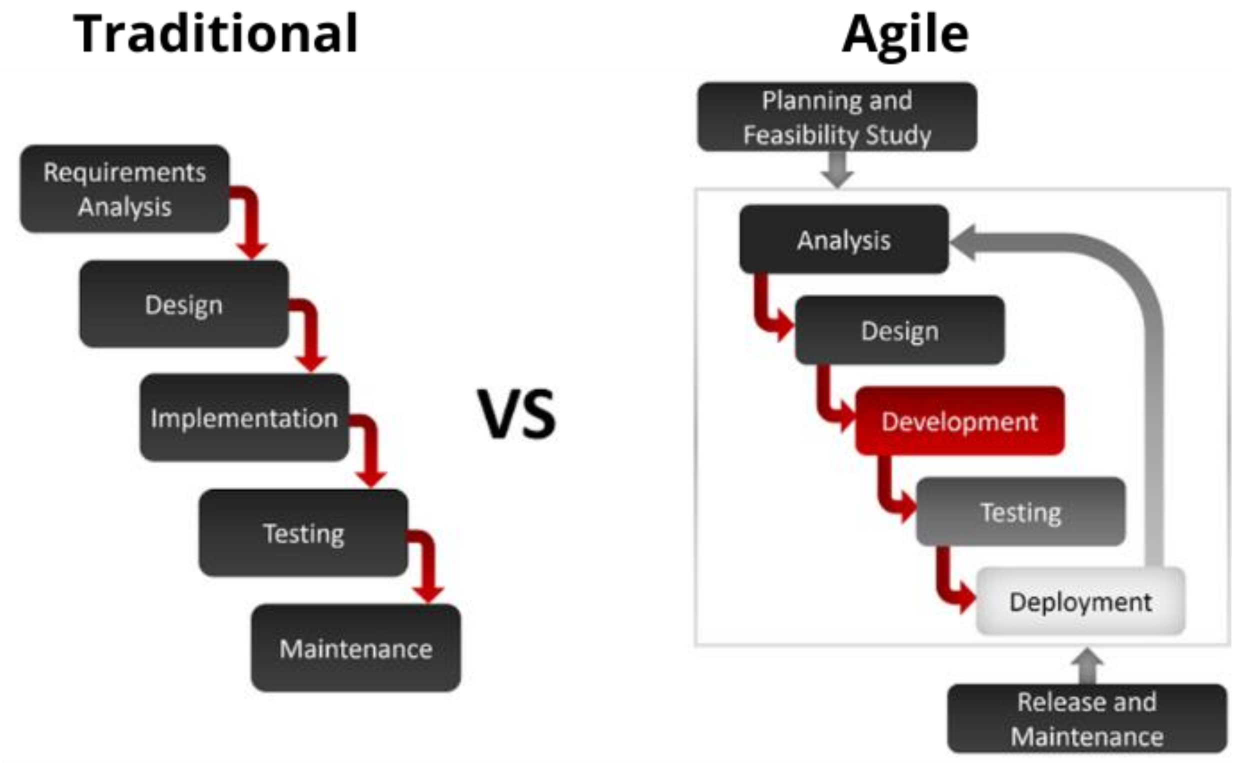
Improved product quality: By applying development methodologies appropriately, higher-quality products can be achieved. The purpose of any approach is to provide guidance for the workflow and prevent sloppy work. Various methodologies accomplish this through different means. The key is to understand which methodology to use for our project and how to integrate it correctly to maximize its effectiveness.

1.2. Choice of methodology

There are several development methodologies, each with its own advantages and disadvantages. Choosing the right method depends on project requirements, team size, objectives, available resources, and schedule.

a. Comparative Study of Traditional and Agile Methods

Agile methods differ from traditional methods in that they prioritize feedback and learning, promoting flexibility and collaboration. Instead of a set process, they allow room for a constantly revised and updated plan of action based on outcomes, customer feedback, and latest results.[1](The Differences between Traditional and Agile Project Management | AgileConnection)



Our choice: Given the advantages of the iterative and incremental approaches, we are adapting to the agile approach.

In addition, this approach allows us to prepare for changes and evolutions without trying to anticipate too much at the outset.

Having decided to adopt Agile development management, we still need to choose the most suitable method for our project.

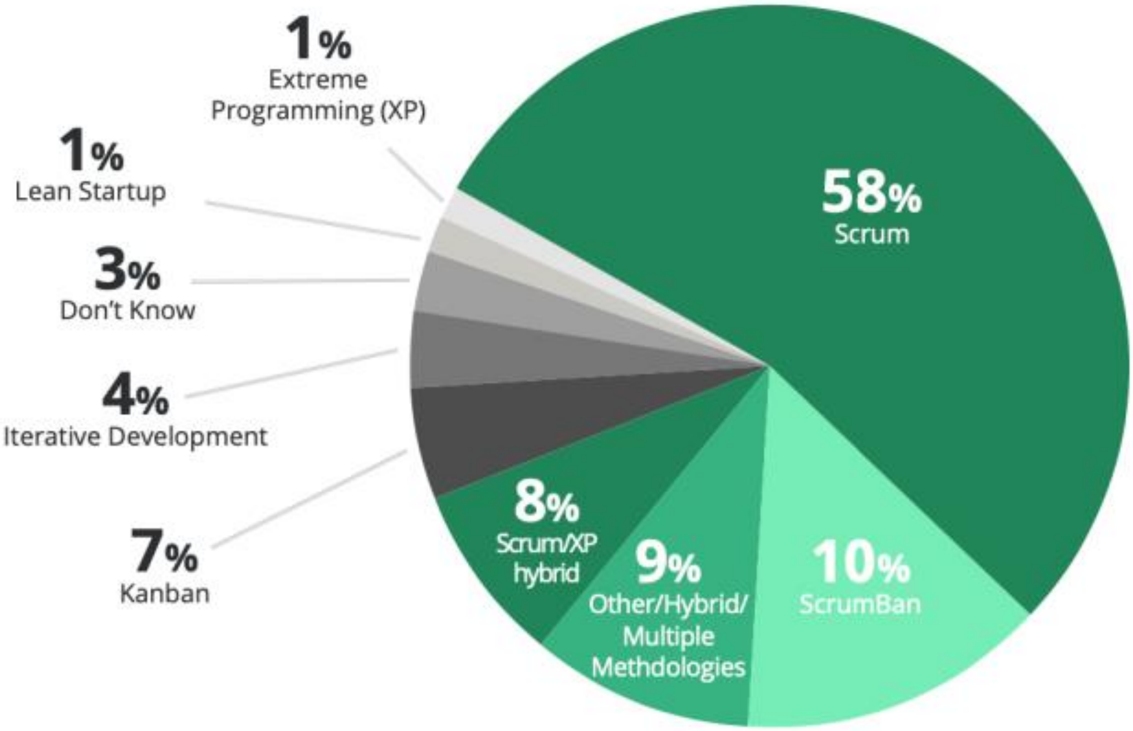
b. Comparison between Agile Methods

(3- The Agile Process: Scrum, Kanban and XP (sealights.io)**)**

The agile methods available are numerous and can be a source of confusion. To help you choose, we've compared the three agile methods: XP, Scrum, and Kanban.

Method	Description
XP	XP (Extreme Programming) is a highly disciplined management method focusing on continually improving the quality and speed of software delivery. It involves close collaboration between the development team and customers, with continuous planning, testing, and feedback to deliver valuable software quickly.
Scrum	Scrum is a highly iterative agile framework operating in sprints of 2-4 weeks. It defines features and objectives before each sprint and aims to reduce risk while delivering value to customers as rapidly as possible.
Kanban	Kanban is an agile-based methodology that operates as a large to-do list, prioritizing work items. It emphasizes visualizing tasks and their statuses as cards on a board, which is visible to all project staff, helping manage work according to priority effectively.

Figure X highlights the numerical results of the 2020 "State of Agile" report. providing statistics on the use of agile methods, as well as a visualization of trends and adoption of these approaches in project management.



We chose to use the Scrum framework for our project, as it is the most popular agile method and offers a number of interesting features. It also complies with the requirements we have defined, including:.

- Fast feedback
- Quicker innovation
- Continuous improvement
- Rapid adaptation to change
- Delighted customers
- Reduced time from idea to deliver

1.3. Introduction to Scrum **(5:** What Is Scrum: A Guide to the Most Popular Agile Framework (scrumalliance.org)**)**

Project management is necessary to achieve the expected objectives, and this fact requires the involvement of the customer, which is what Scrum applies.

Scrum is a framework in which people can address complex adaptive problems while productively and creatively delivering products of the highest possible value.

The three roles present in Scrum are:

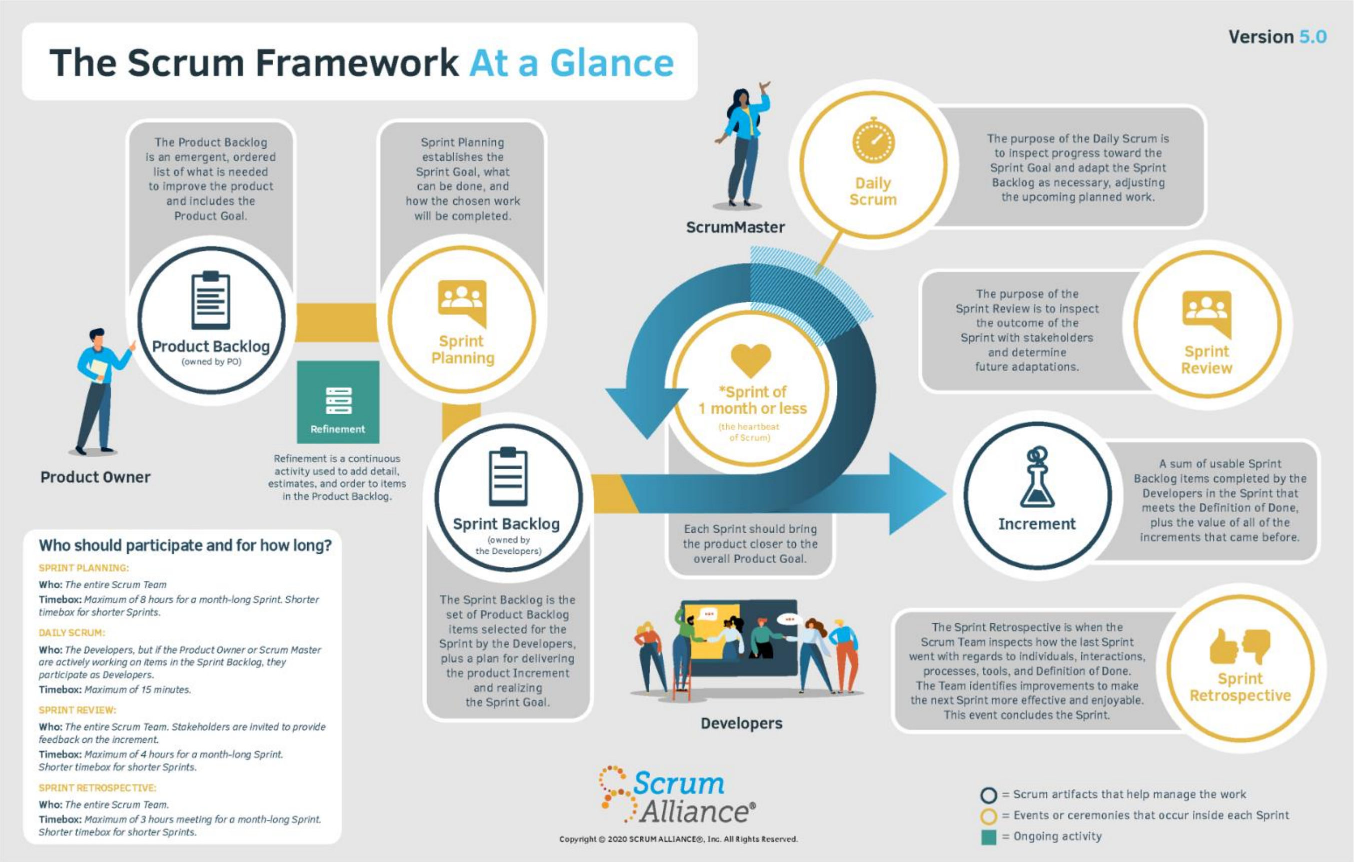
- Scrum Master.
- Product Owner.
- Development Team.

In short, Scrum requires a Scrum Master to foster an environment where:

- A product owner orders the work for a complex problem into a product backlog.
- The Scrum Team turns a selection of this work into a valuable increment during a sprint.
- The Scrum Team and its partners inspect the results and adjust for the next sprint.
- Repeat.

Scrum is based on iterations of 1 to 4 weeks called sprints, whose purpose is to design, build, and test features to achieve a deliverable called an increment. The product backlog contains the prioritized list of requirements. The sprint review is the day for product presentation. Lastly, the sprint retrospective is a meeting for the entire Scrum team to enter into a continuous improvement dynamic and avoid repeating past mistakes.

Figure 5 illustrates the Scrum process, providing a visual representation of the different stages and activities involved in this agile project management methodology.



The Product Backlog is an emergent, ordered list of what is needed to improve the product and includes the Product Goal.

Sprint Planning establishes the Sprint Goal, what can be done, and how the chosen work will be completed.

Refinement is a continuous activity used to add detail, estimates, and order to items in the Product Backlog.

The Sprint Backlog is the set of Product Backlog items selected for the Sprint by the Developers, plus a plan for delivering the product Increment and realizing the Sprint Goal.

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

The purpose of the Sprint Review is to inspect the outcome of the Sprint with stakeholders and determine future adaptations.

Each Sprint should bring the product closer to the overall Product Goal.

A sum of usable Sprint Backlog items completed by the Developers in the Sprint that meets the Definition of Done, plus the value of all of the increments that came before.

The Sprint Retrospective is when the Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and Definition of Done. The Team identifies improvements to make the next Sprint more effective and enjoyable. This event concludes the Sprint.

Who should participate and for how long?

SPRINT PLANNING:

Who: *The entire Scrum Team*

Timebox: *Maximum of 8 hours for a month-long Sprint. Shorter timebox for shorter Sprints.*

DAILY SCRUM:

Who: *The Developers, but if the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.*

Timebox: *Maximum of 15 minutes.*

SPRINT REVIEW:

Who: *The entire Scrum Team. Stakeholders are invited to provide feedback on the increment.*

Timebox: *Maximum of 4 hours for a month-long Sprint. Shorter timebox for shorter Sprints.*

SPRINT RETROSPECTIVE:

Who: *The entire Scrum Team.*

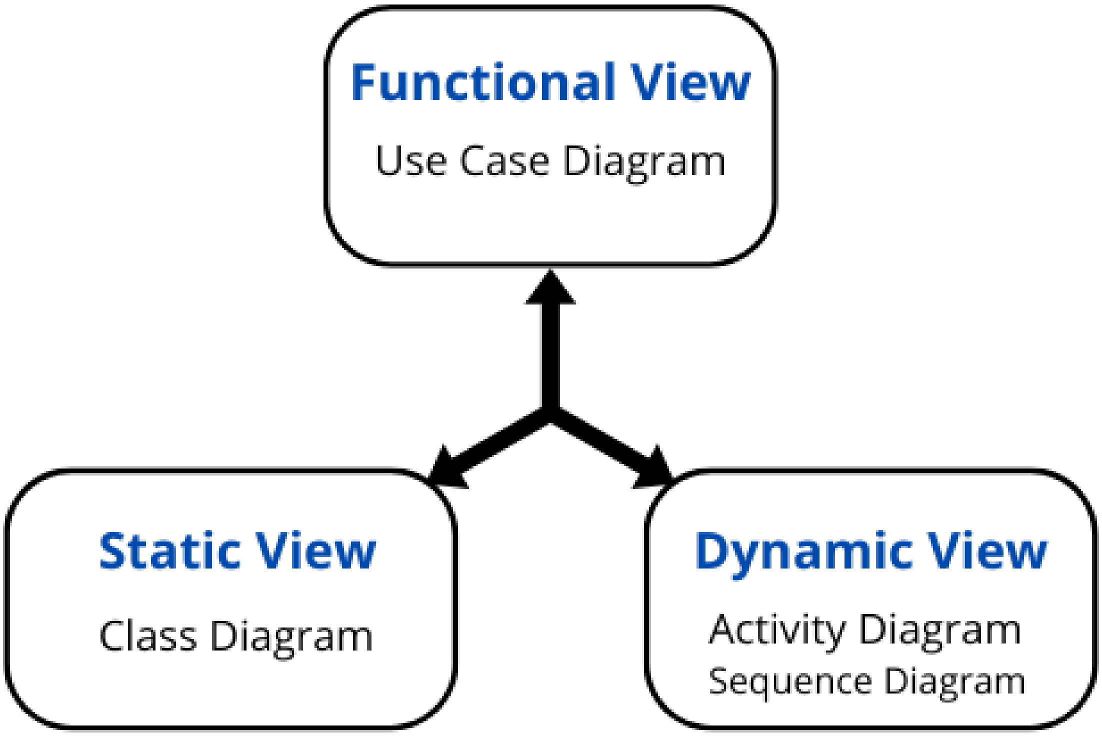
Timebox: *Maximum of 3 hours meeting for a month-long Sprint. Shorter timebox for shorter Sprints.*

○ = Scrum artifacts that help manage the work
○ = Events or ceremonies that occur inside each Sprint
🟩 = Ongoing activity

(Scrum Framework At a Glance: A visual guide to how scrum works.)

1.4. Modeling Language

The aim of modeling is to achieve a viable solution for the system, but this solution cannot be reached in a single iteration. Several steps are necessary to gradually refine the level of detail of the system to be realized. We have chosen to use UML as the unified modeling language because of its ability to cover the various views of the project. Each view requires the use of specific UML diagrams. In our case, we particularly focus on the diagrams presented in Figure 6.



Conclusion

In this chapter, we have introduced our host organization, ProxiClic . Then we set out the general context of the project and presented the choice of development methods and modeling language. In the next chapter, we present the definition of the functionalities to be developed and the planning of their implementation.

CHAPTER 2 : Project Planning

Introduction:

The planning and specification phase is crucial in the information systems development cycle. It involves identifying and documenting the system's functional and non-functional requirements, ensuring that stakeholder needs are clearly understood and met. This chapter will cover the creation of a Product Backlog, the roles within a Scrum team, and the identification of system actors and their interactions. Additionally, we will discuss the development environment and architectural choices, providing a structured approach to system development.

I. Specification of needs

Specifying requirements in an IT project is a crucial step that allows for precisely determining the capabilities and needs of the system to be designed or updated. This step is vital for the project's success and must be carried out carefully to avoid costly errors in terms of time and resources.

1. Functional requirements

Functional requirements define all the services offered to system users. In the case of our system, functional requirements can be summarized as follows:
Account and Company Management: This functionality focuses on managing user accounts and company details. It includes features such as secure user access, admin account creation, company creation and linking, account activation, pending account management, and account status control.

Admin Dashboard: The Admin Dashboard provides a comprehensive overview of the loyalty program's performance metrics. It enables admins to monitor revenue, sales trends, customer behavior, and other key performance indicators through visualized data.

Customer Management: This functionality allows for the management of customer interactions within the system. It enables customers to sign up using self-service checkouts and interactive terminals, allowing them to join and participate in the loyalty program. Restaurant managers are empowered to add new customers, access customer details, update information, and manage accounts.

Loyalty Points Management: This functionality involves handling loyalty points within the system. It includes features such as crediting points to customer accounts after purchases, debiting points upon redemption, and viewing transaction histories.

2. Non-Functional Requirements

Several constraints must be adhered to in the application, including:

- Usability

- Our application must have a simple and intuitive interface that allows users to easily navigate and operate it.

- Performance

- The application must perform efficiently, meeting all user requirements in an optimal manner and within a reasonable response time.

- Integration with Third-Party Systems

The system should have the capability to connect and interact with external systems such as tactile point-of-sale systems,self-service ordering kiosks, and analytics platforms through APIs. This ensures that data and functionalities from these third-party services can be seamlessly incorporated into the system, providing a comprehensive and efficient user experience.

II. Project management with Scrum

The Scrum framework consists of a team with well-defined roles, a ceremonial promoting collaboration and some artifacts.

1. SCRUM team and roles

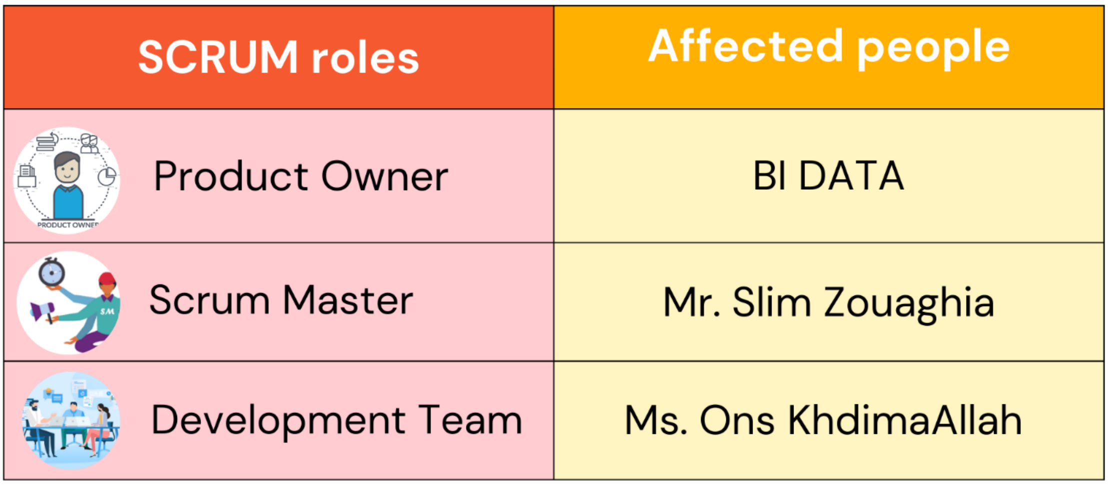
The Scrum Team is made up of a Product Owner, a Development Team and a Scrum Master. Scrum teams are self-organized and self-directed. These teams choose the best ways and procedures to get their work done instead of being commissioned by outsiders.

The different roles of Scrum are:

The Product Owner: The product manager is responsible for the vision of the product requested by the customer. He is the intermediary translating the client's needs to the SCRUM team and vice versa.

Scrum Master: It is above all a member of the project team. It is not intended to lead it, but to guide it in the application of the Scrum methodological framework and help it to move forward autonomously and constantly seeking to improve.

The development team: The team is responsible for developing the software product ordered by the Product owner.



SCRUM roles	Affected people
Product Owner	BI DATA
Scrum Master	Mr. Slim Zouaghia
Development Team	Ms. Ons KhdimaAllah

Tab 4 : Presentation of the Scrum team

2. Backlog of the product (Product Backlog)

The backlog is an important artifact in Scrum. It encompasses all the system's features. These features are prioritized using the following three categories:

- H: High
- M: Medium
- B: Low

Each feature's complexity is defined according to the Fibonacci sequence (1, 2, 3, 5, 8, 13…).

Functionality	Number	User story	priority	complexity	sprint
	1	As a user, I want secure access using email address or telephone number and password so that my account information is protected.	H	3	1
	2	As a new admin, I want to create an account by providing my details and completing a CAPTCHA verification	H	3	
	3	As a new admin, I want to create an account by providing my details so that I can manage my company's loyalty program.	H	3	2
	4	As technical support, I want to create companies and link them to admin accounts so that the admins can manage their specific company's loyalty programs.	H	5	
	5	As an admin, I want my account to be activated after my company is created so that I can start using the system.	H	2	
	6	As technical support, I want to view all pending admin accounts associated with a company, allowing me to choose which accounts to activate or delete.	H	4	

	7	As technical support, I want to copy the details of a specific company	H	3	
	8	As technical support, I want to remove a company from the system	H	4	
	9	As technical support, I want to choose to see the list of either activated accounts or pending accounts to manage them accordingly.	M	3	
	10	As technical support, I want to select if the account should be an admin or not.	H	3	
	11	As technical support, I want to deactivate accounts to prevent further access when necessary.	H	4	
	12	As an admin, I want to access a dashboard with loyalty program performance metrics	H	5	3
	13	As an admin, I want to filter and view the dashboard data based on specific criteria	H	5	
	14	As an admin, I want to view the total revenue (CA TTC) for the selected date range so that I can monitor overall performance.	H	5	
	15	As an admin, I want to view the net revenue (CA HT) for the	H	5	

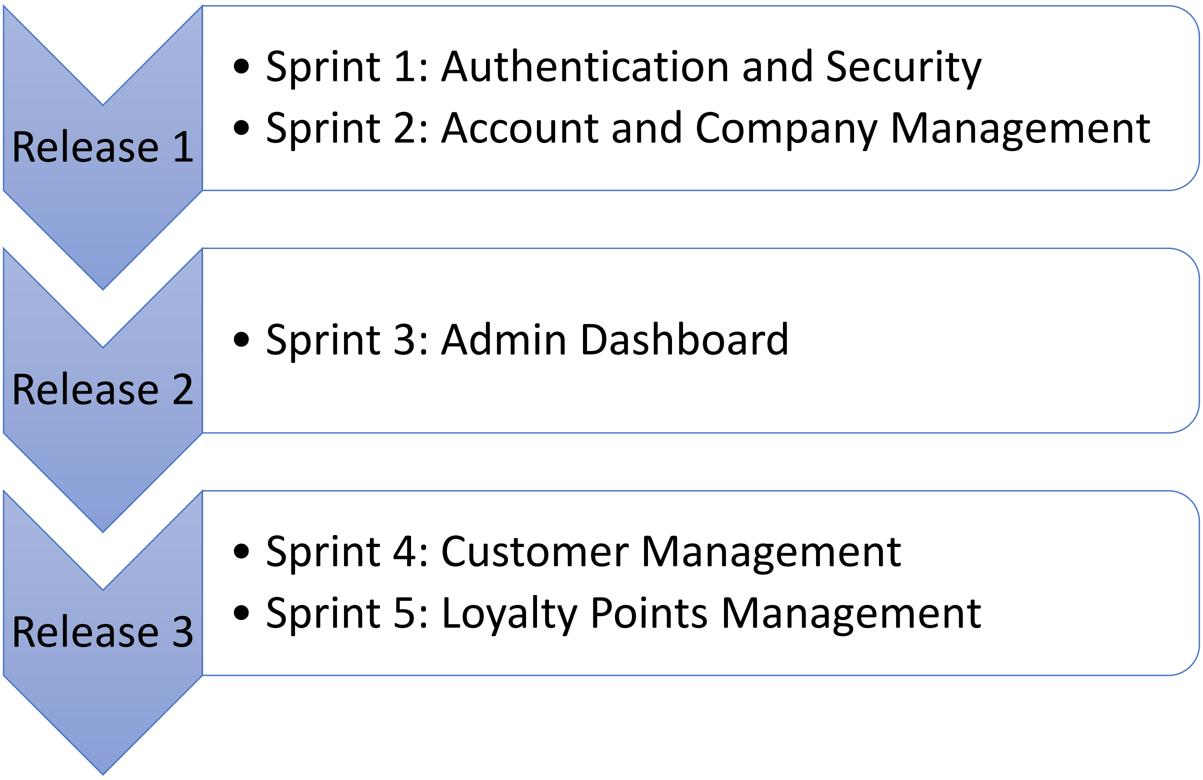
		selected date range to understand the company's financial health.			
	16	As an admin, I want to see the average cart value including taxes (Panier moyen TTC) to gauge customer spending behavior.	H	3	
	17	As an admin, I want to see the average cart value excluding taxes (Panier moyen HT) for financial analysis.	H	3	
	18	As an admin, I want to see the revenue from payments received (CA encaissement) to track cash flow.	H	3	
	19	As an admin, I want to see the number of order cancellations (Annulation) to identify potential issues in order processing.	H	3	
	20	As an admin, I want to view the trend of orders over time (Commande) to monitor sales trends, with the ability to filter the data by date and hour.	H	5	
	21	As an admin, I want to view the breakdown of payment methods (Règlements) to understand customer preferences.	H	3	
	22	As an admin, I want to see the distribution of sales by service	H	5	

		provider (Prestataire) to assess performance.			
	23	As an admin, I want to see the sales by type of sale including taxes (Mode de vente TTC) to analyze sales channels.	H	5	
	24	As an admin, I want to see the sales by type of sale excluding taxes (Mode de vente HT) for detailed financial analysis.	H	5	
	25	As an admin, I want to see the distribution of sales by article or family to understand product performance.	H	5	
	26	As a customer, I want to sign up using self-service checkouts and interactive terminals so that I can easily join and participate in the loyalty program.	H	8	4
	27	As a manager, I want to add new customers to the system so that they can start earning and redeeming loyalty points.	H	3	
	28	As a manager, I want to view customer details and loyalty points so that I can provide personalized service and support.	H	3	
	29	As a manager, I want to update customer information	H	3	
	30	As a manager, I want to credit points to customer accounts	H	3	5

	31	As a manager, I want to debit points from customer accounts	H	3	
	32	As a manager, I can view customer transaction histories.	H	5	
Totale=125					

3. Release planning

In the table below, we illustrate the planning of the different releases of the project. Each of these releases has one or two sprints. Each sprint is set for two or three weeks.



Release 1

- Sprint 1: Authentication and Security
- Sprint 2: Account and Company Management

Release 2

- Sprint 3: Admin Dashboard

Release 3

- Sprint 4: Customer Management
- Sprint 5: Loyalty Points Management

III. General use case diagram

1. Identification of actors

An actor is an external entity (a person, hardware or software) that interacts with the

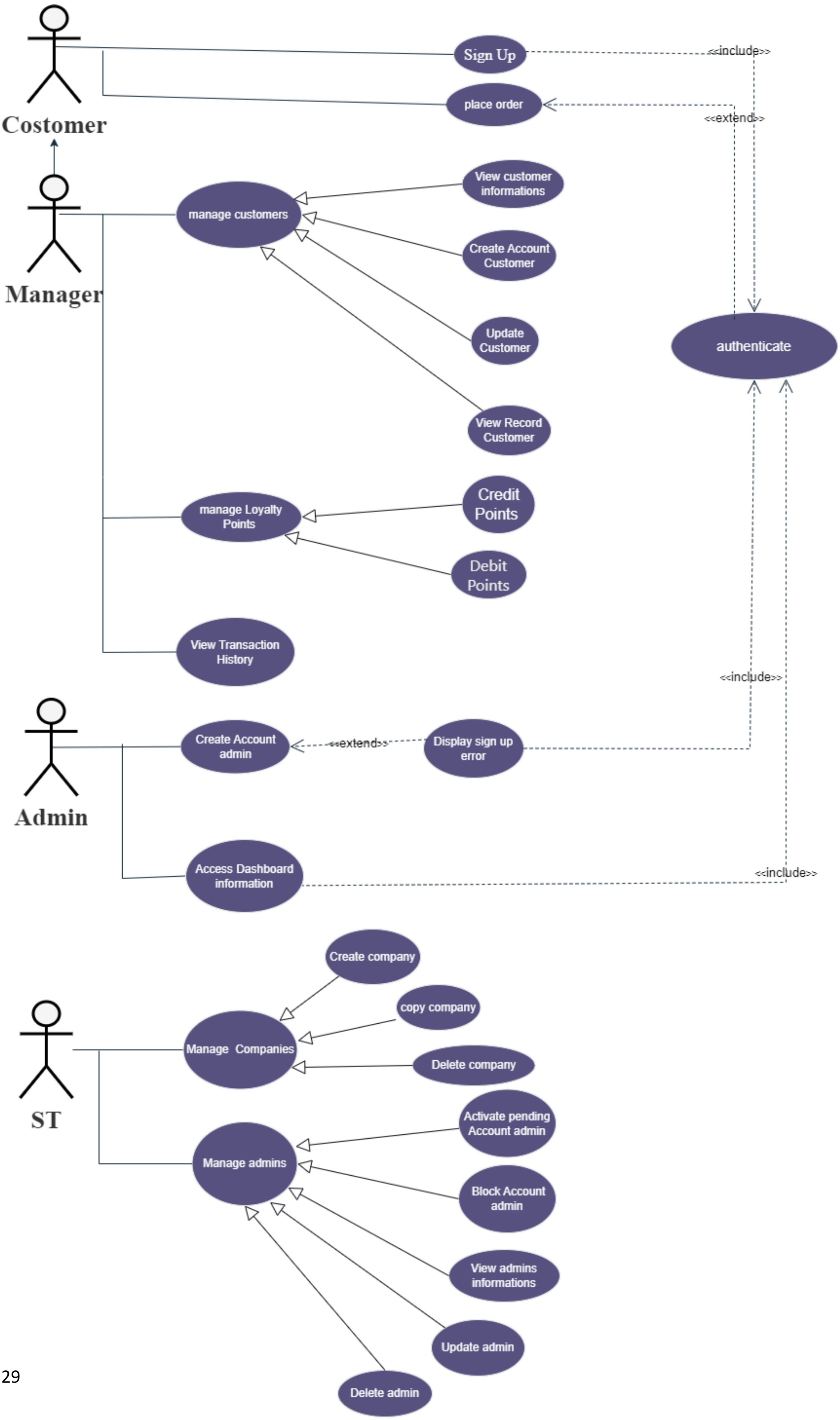
in order to achieve added value. In the case of our project, the actors interacting with

the system are :

- Admin: Admin: Oversees a group of restaurants or any group of businesses, managing administrative tasks and ensuring smooth operations across multiple establishments. They utilize the dashboard to monitor and track the progress of their work efficiently.
- Technical Support :Provides technical assistance within a big data environment, resolving system issues and ensuring optimal performance.
- Manager: Manages a specific restaurant within the administrative group, overseeing day-to-day operations to enhance customer satisfaction.
- Customer: Individuals who sign up for the loyalty program using self-service checkouts and interactive terminals, enabling them to participate in various program benefits.

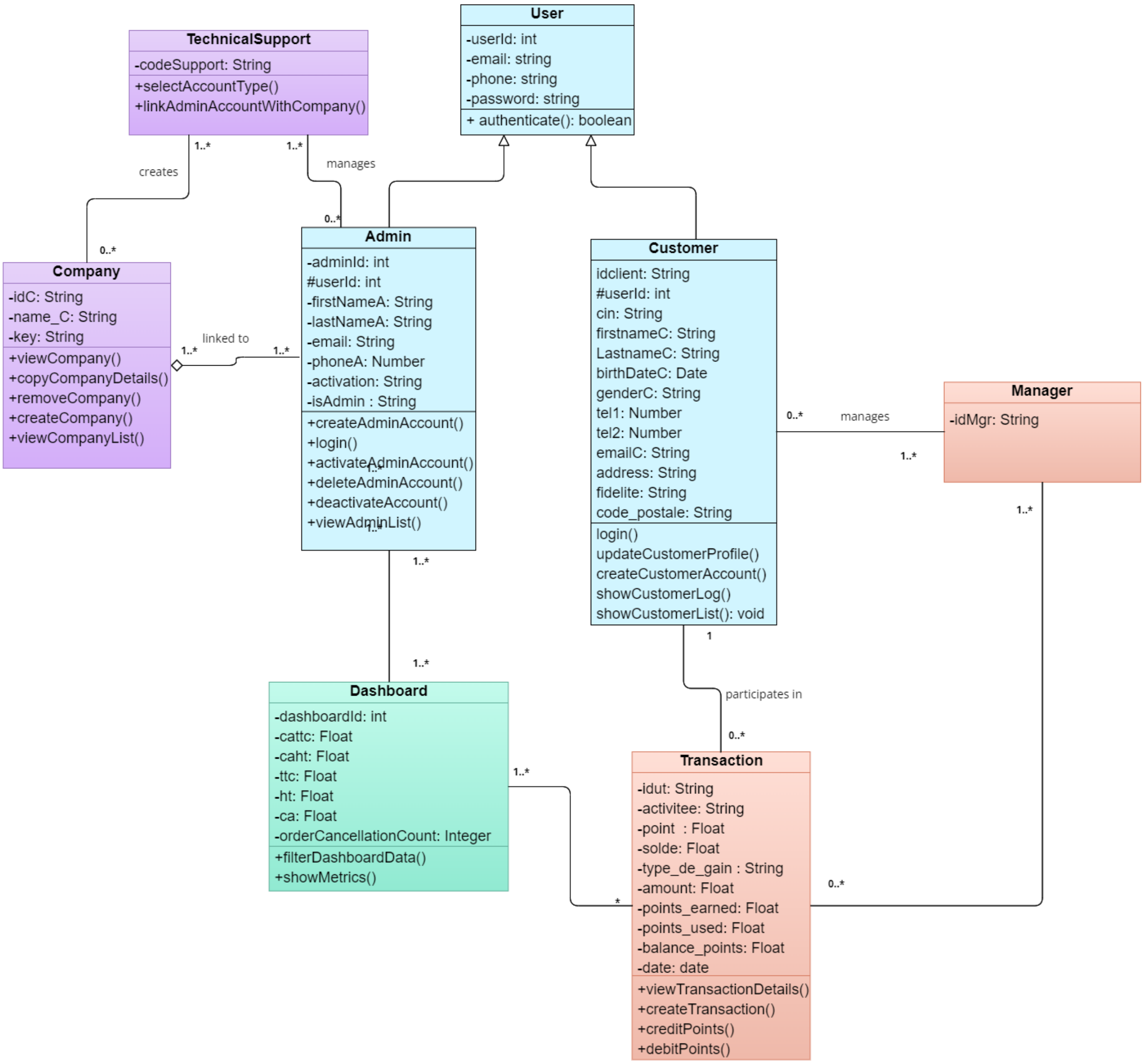
2. The use case diagram

The use case diagram is a UML diagram that provides an overview of the functional behavior of a given system. This diagram helps developers and IT professionals gain a general understanding of the functionalities an information system offers and how actors interact with it.


Costomer
Sign Up
<<include>>
place order
<<extend>>
Manager
manage customers
View customer informations
Create Account Customer
Update Customer
View Record Customer
authenticate
manage Loyalty Points
Credit Points
Debit Points
View Transaction History
<<include>>
Admin
Create Account admin
<<extend>>
Display sign up error
Access Dashboard information
<<include>>
ST
Create company
copy company
Manage Companies
Delete company
Activate pending Account admin
Manage admins
Block Account admin
View admins informations
Update admin
Delete admin

IV. General class diagram

The class diagram represents the classes involved in the system. It is a static representation of the elements that make up a system and their relationships. The class is an abstract concept which makes it possible to represent all the entities of a system. Figure 3.4 illustrates the general class diagram of our project :



V. Development environment and technical choices

The software environment enabled us to give concrete form to our concept and to produce a functional that meets the needs for which it was developed.

The choice of development tools was imposed by the host company.

1. Integrated Development Environments (IDEs)

WinDev and WebDev are both integrated development environments (IDEs) developed by the French company PC SOFT, featuring their own programming language called WLanguage. However, they are designed for different types of application development and **allow the creation of both front-end and back-end web solutions.**

PC SOFT's commitment to excellence is evident in its robust support for various database management systems, ensuring seamless integration with popular platforms like SQL Server, Oracle, MySQL, and PostgreSQL. Additionally, PC SOFT emphasizes the importance of web services, facilitating efficient communication and data exchange between applications.

1.1 WinDev

WinDev, a flagship product of PC SOFT, stands as a versatile software engineering Primarily designed for developing desktop applications, Supports the creation of Windows, Linux, and macOS applications.. With its intuitive visual development environment and drag-and-drop features, WinDev accelerates the creation of applications . Key features include cross-platform compatibility, integrated database management(such as SQL Server, Oracle, MySQL, and PostgreSQL), and a rich component library, enabling rapid development across diverse platforms.

(https://www.clubic.com/telecharger-fiche432943-windev.html

)

Here are **Some Benefits of Using WinDev :**

- Cross-Platform Compatibility:WinDev allows you to build applications that run on various platforms, including Windows, Linux, and mobile devices. This saves development time and effort by enabling code reuse across different platforms.
- Integrated Database Management: WinDev includes built-in database management tools, allowing easy connection to databases such as SQL Server, MySQL, and PostgreSQL. This simplifies data handling and integration within applications.

- Rich Component Library:WinDev offers a wide range of pre-built components (widgets, controls, and UI elements) that can be customized and reused. This speeds up development and ensures consistent user interfaces.
- Multi-Language Support: WinDev provides features for creating multilingual applications, enabling dynamic language switching at runtime. This is crucial for applications that need to be localized for different regions.

1.2. WebDev:

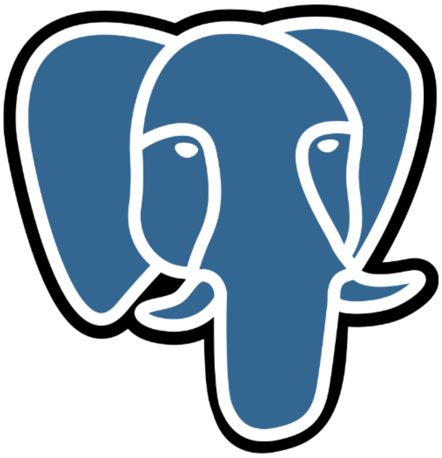
WebDev, another offering from PC SOFT, is specifically crafted for the development of dynamic and interactive web applications. WebDev places a strong emphasis on search engine optimization (SEO) tools, enabling developers to optimize their web applications for better visibility and ranking.

Here are **Some Benefits of Using WebDev :**

- Web Technologies Integration:WebDev supports HTML, CSS, JavaScript, and other web technologies, making it easier to develop dynamic and interactive web applications.
- Database Connectivity:Like WinDev, WebDev offers robust support for database integration, allowing efficient data management with databases like SQL Server, MySQL, Oracle, and PostgreSQL.
- SEO Tools: WebDev includes built-in tools to optimize web applications for search engines, enhancing their visibility and ranking on search engine results pages.
- Web Server Management:WebDev provides features for deploying and managing web applications on web servers, simplifying the deployment process and ongoing maintenance.
- Web Services:Both WinDev and WebDev emphasize the importance of web services, facilitating efficient communication and data exchange between applications. This enhances interoperability and integration capabilities.

2. Database Management

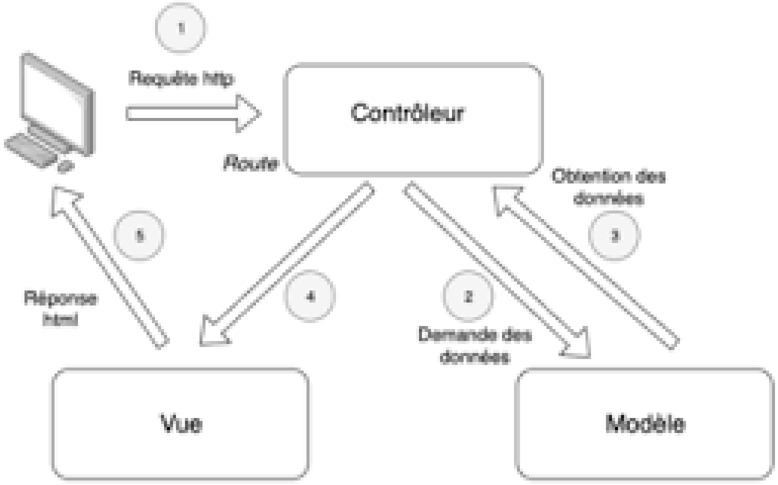
PostgreSQL (PostgreSQL: The world's most advanced open source database) is an advanced, enterprise-class, and open-source relational database system. It supports both SQL (relational) and JSON (non-relational) querying. With more than 35 years of active development, PostgreSQL has earned a strong reputation for its proven architecture, reliability, data integrity, robust feature set, and extensibility.



VI. The MVC Design Pattern

The Model-View-Controller (MVC) design pattern provides a structured approach to developing web applications by separating the application's concerns into three distinct components: Model, View, and Controller. This separation of concerns promotes code clarity, maintainability, and testability, making it an ideal choice for complex applications like your loyalty program system.

Using the MVC pattern aligns seamlessly with Windev and Webdev development environments, as both can handle aspects of the View and Controller. These tools can work together harmoniously with the MVC pattern to ensure a well-structured and efficient development process.



Conclusion

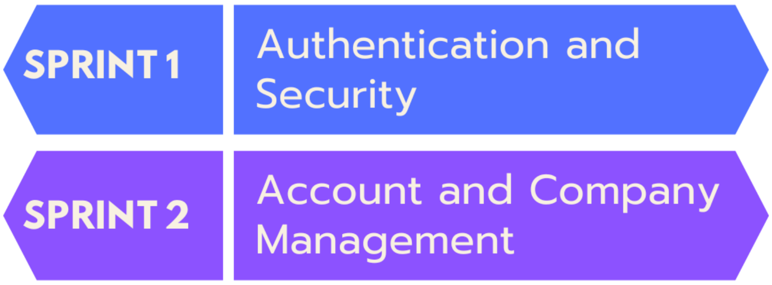
In this chapter, we have prepared our work plan. We have captured the functional and non-functional requirements of our subject, the tasks of the actors, the requirements modeling, and the release planning. In the following chapter, we will present sprint 1.

Chapter 3 : RELEASE 1

Introduction

In this chapter, we delve into the first release of our project, which encompasses the first and second sprints. We will describe the organization of these sprints, the backlog items, and the detailed steps taken during the analysis, design, and implementation phases. This chapter will provide a comprehensive view of the development process through use case diagrams, sequence diagrams, and screenshots of the implemented features. Our goal is to illustrate the progress made and set the stage for subsequent releases.

I. Sprint organization



II. Sprint 1 Backlog

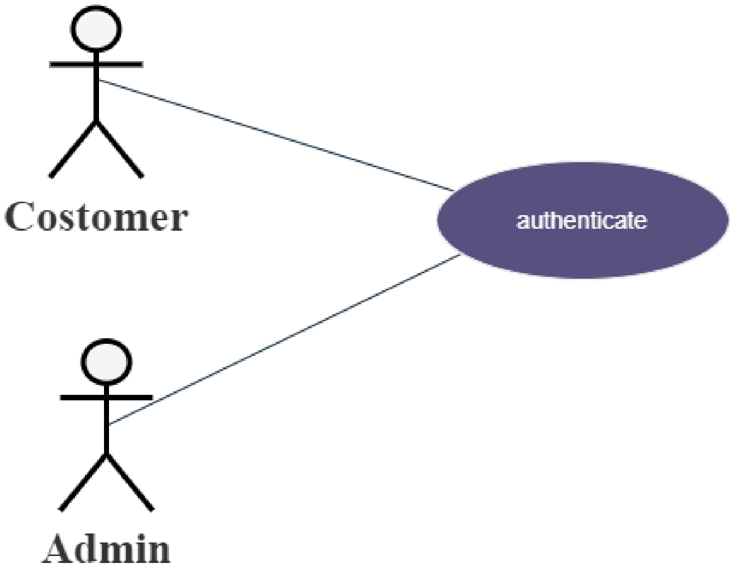
The table below illustrates the product backlog for this sprint:

Number	User story	The tasks	Estimated Time
1	As a user, I want secure access using username and password so that my account information is protected.	Analysis of requirements	
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
2	As a new admin, I want to create an account by providing my details and completing a CAPTCHA verification	Analysis of requirements	
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	

III. Analysis

1. Global Use Case Diagram for Sprint 1

After presenting the sprint backlog, we move on to presenting the use case diagram for the first sprint.



2. Descriptions of Sprint 1

The description tables that enhance the readability of the use cases are below:

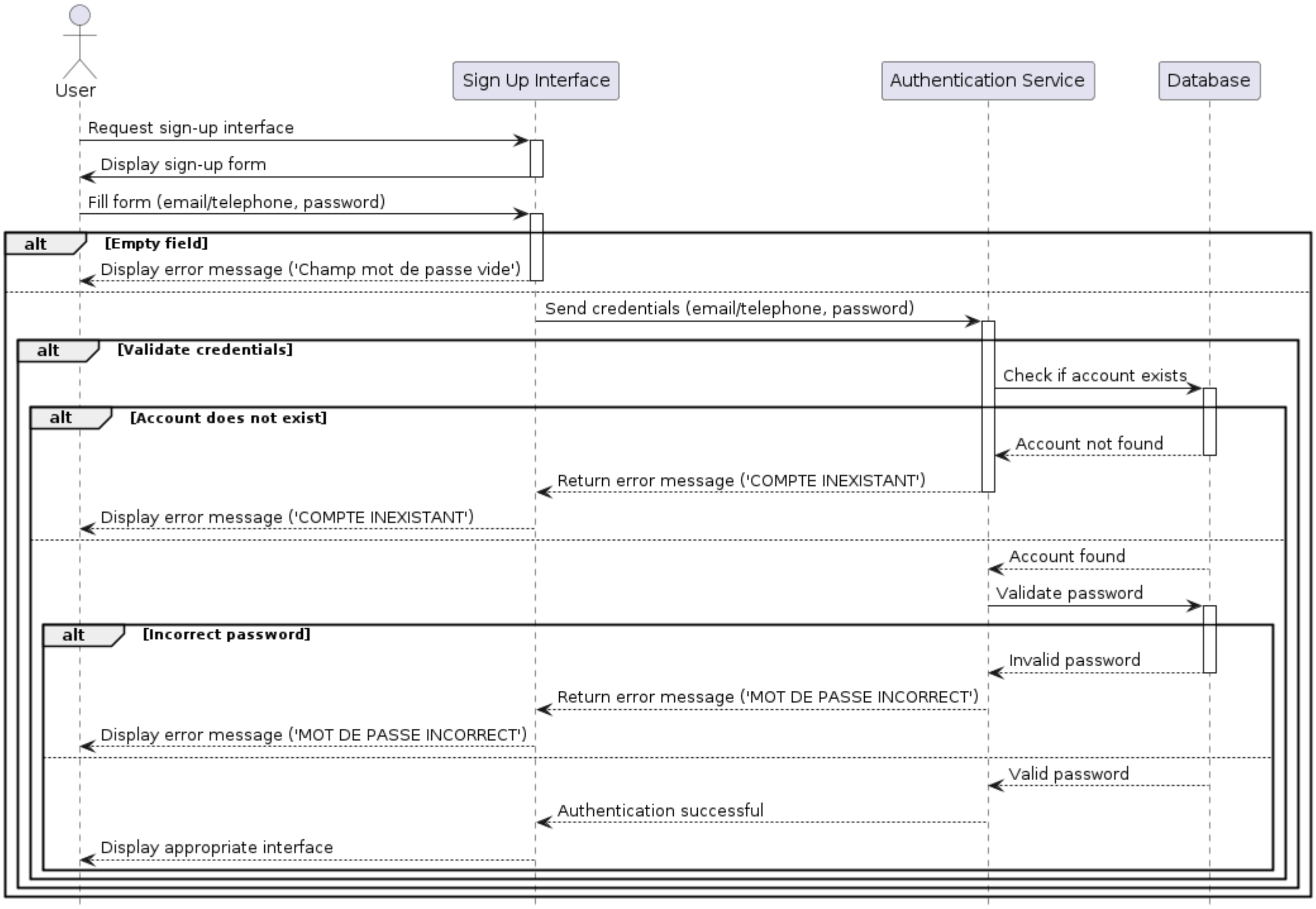
Description of the "Sign Up" use case

Use Case: Sign Up	Sign Up
Actor:	user
Pre-condition:	- The user has a registered account. - The user has a valid email address or telephone number and password.
Post-condition:	The user is authenticated
Main Scenario:	The user enters their email and password. The user confirms by clicking on the 'connection' button. The system displays the appropriate interface for each user.
Alternative Scenario:	The system displays an error message : ‘COMPTE INEXISTANT’ , Champ mot de passe vide’

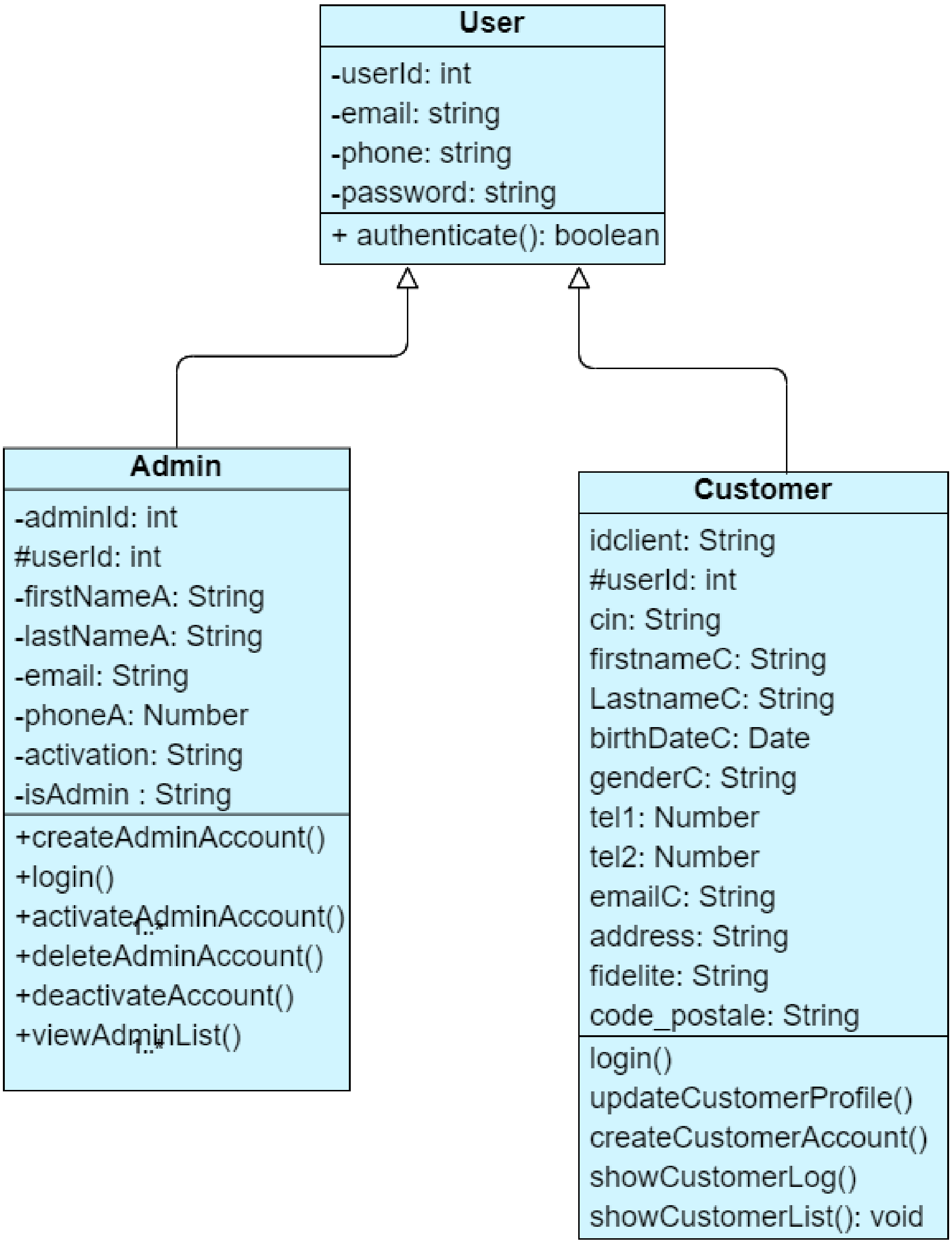
	‘MOT DE PASSE INCORRECT’

IV. Design

1. Sequence Diagrams for Sprint 1


User
Sign Up Interface
Authentication Service
Database
Request sign-up interface
Display sign-up form
Fill form (email/telephone, password)
alt
[Empty field]
Display error message ('Champ mot de passe vide')
Send credentials (email/telephone, password)
alt
[Validate credentials]
Check if account exists
alt
[Account does not exist]
Account not found
Return error message ('COMPTE INEXISTANT')
Display error message ('COMPTE INEXISTANT')
Account found
Validate password
alt
[Incorrect password]
Invalid password
Return error message ('MOT DE PASSE INCORRECT')
Display error message ('MOT DE PASSE INCORRECT')
Valid password
Authentication successful
Display appropriate interface

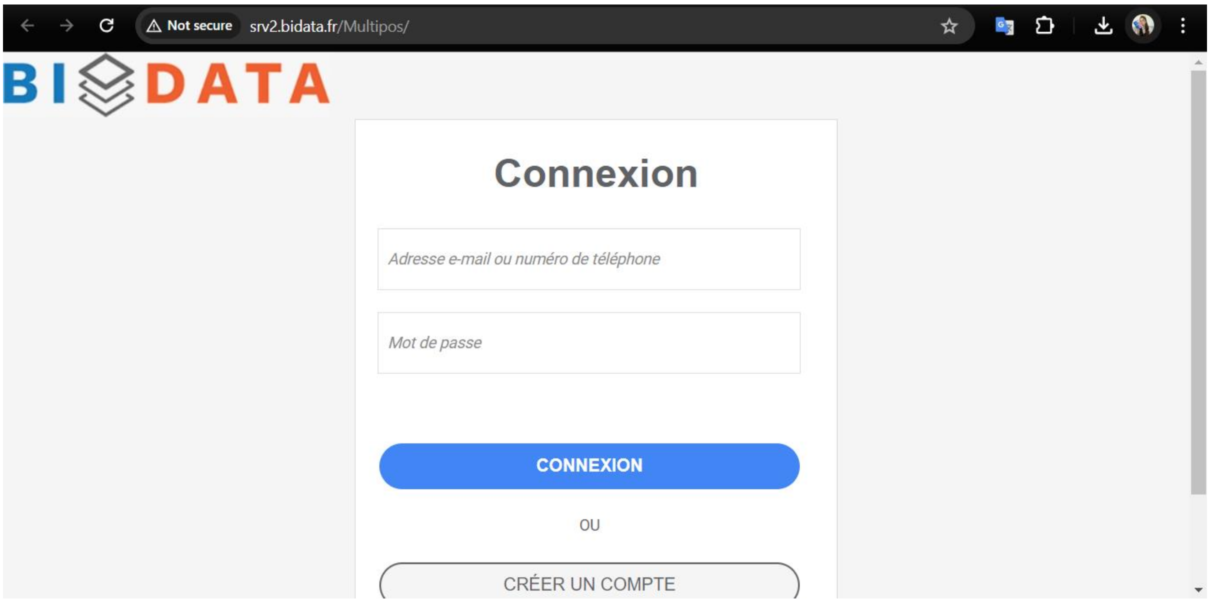
2. Global Class Diagram for Sprint 1



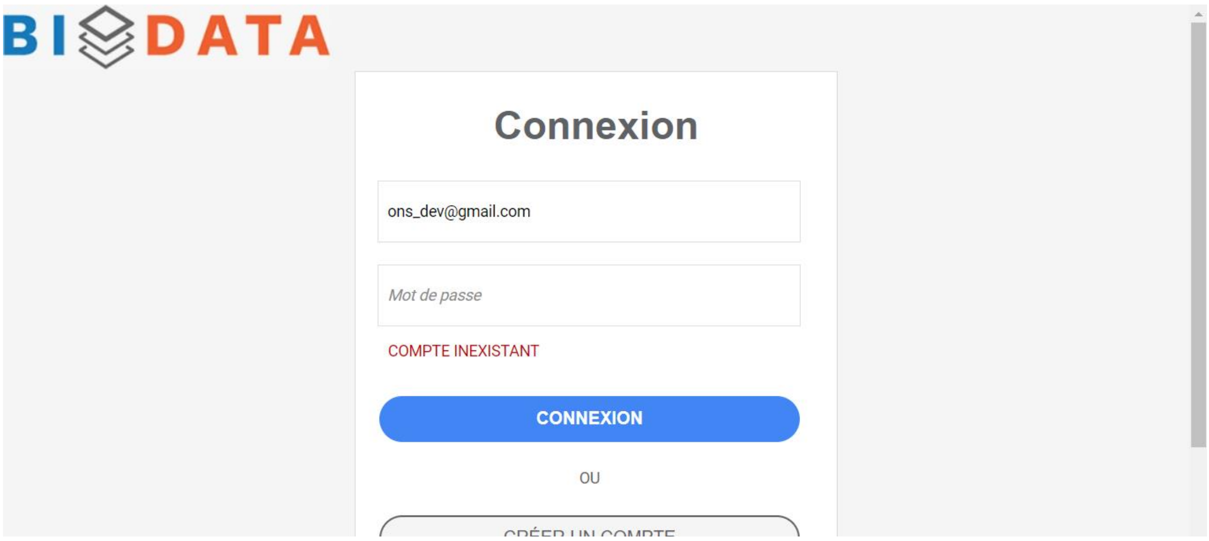
V. Realisation

This section presents the results obtained through screenshots. The screenshots visually illustrate the final outcomes achieved during tests or simulations. Below are the screenshots for each interface related to this sprint:

Authentication Interface for Admin :



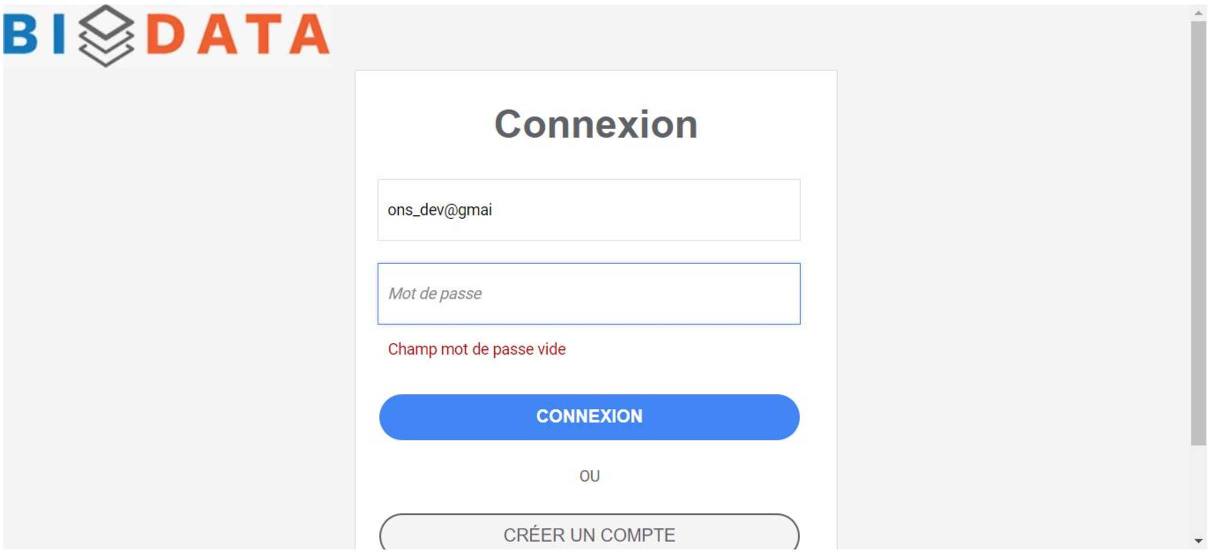
Authentication Interface for Admin: Non-Existent Account



ons_dev@gmail.com

COMPTE INEXISTANT

Authentication Interface for Admin: Empty Password Field



BI DATA

Connexion

ons_dev@gmai

Mot de passe

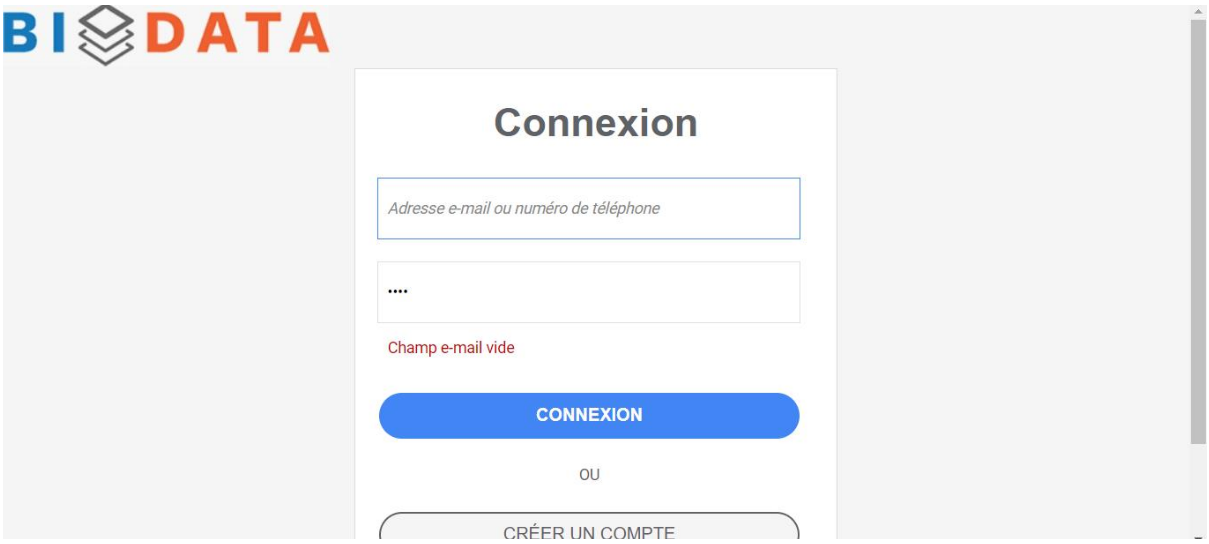
Champ mot de passe vide

CONNEXION

OU

CRÉER UN COMPTE

Authentication Interface for Admin: Empty Email Field



BI DATA

Connexion

Adresse e-mail ou numéro de téléphone

····

Champ e-mail vide

CONNEXION

OU

CRÉER UN COMPTE

Authentication Interface for Customer :

VI. Sprint 2 Backlog

The table below illustrates the product backlog for this sprint:

Number	User story	The tasks	Estimated Time
3	As a new admin, I want to create an account by providing my details so that I can manage my company's loyalty program.	Analysis of requirements	3
		Design	
		Code front-end part	

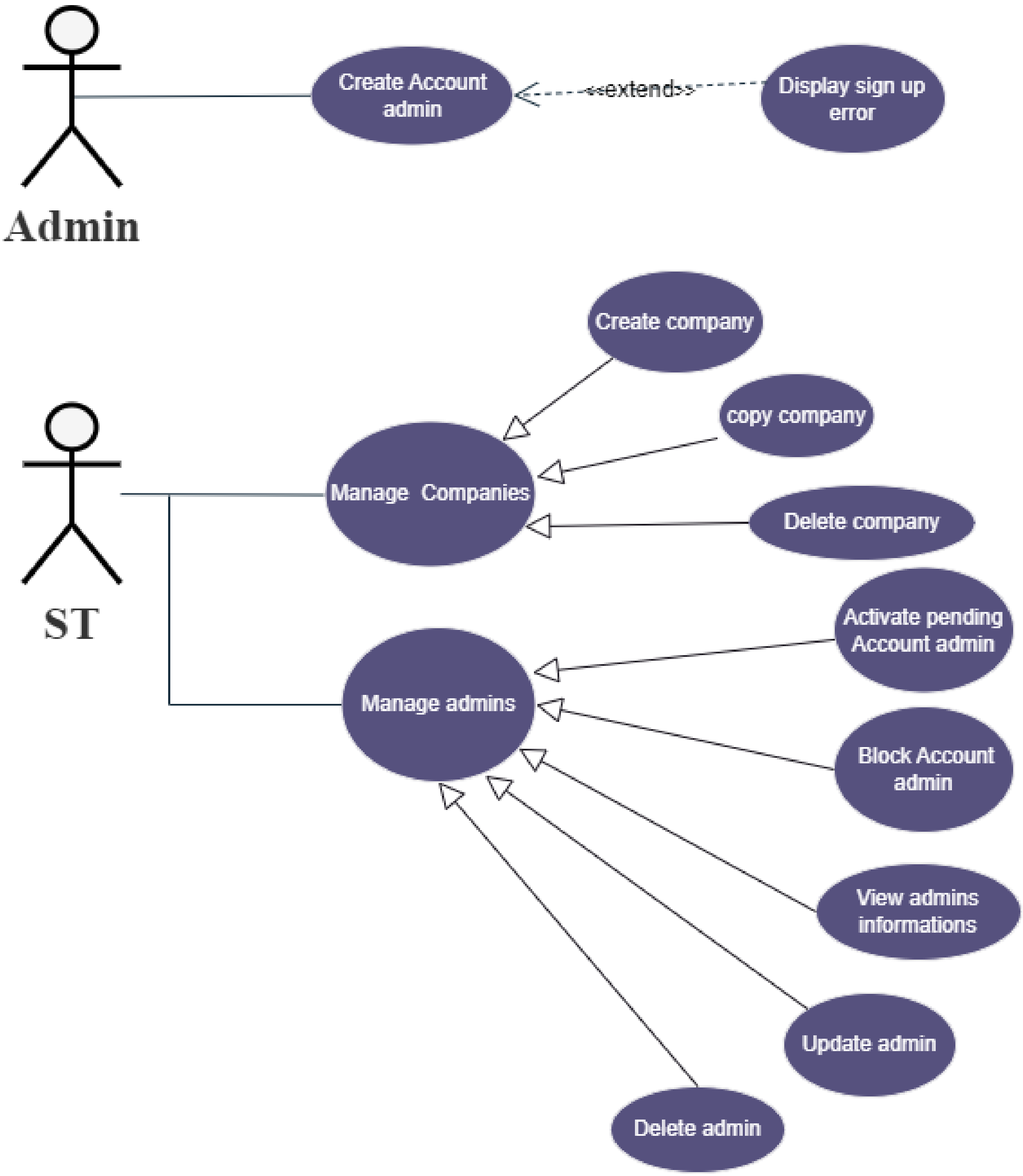
		Code back-end part	
		Integration testing	
4	As technical support, I want to create companies and link them to admin accounts so that the admins can manage their specific company’s loyalty programs.	Analysis of requirements	5
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
5	As an admin, I want my account to be activated after my company is created so that I can start using the system.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
6	As technical support, I want to view all pending admin accounts associated with a company, allowing me to choose which accounts to activate or delete.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
7	As technical support, I want to copy the details of a specific company	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
8	As technical support, I want to remove a company	Analysis of requirements	2
		Design	

		Code front-end part	
		Code back-end part	
		Integration testing	
9	As technical support, I want to choose to see the list of either activated accounts or pending accounts to manage them accordingly.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
10	As technical support, I want to select if the account should be an admin or not.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
11	As technical support, I want to deactivate accounts to prevent further access when necessary.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	

VII. Analysis

1. Global Use Case Diagram for Sprint 2

After presenting the sprint backlog, we move on to presenting the use case diagram for the second sprint.



2. Descriptions of Sprint 2:

Description of the " Create Admin Account" use case

Use Case:	Create Admin Account
Actor:	New Admin
Pre-condition:	None
Post-condition:	Admin account created and activated

Main Scenario:	1. The new admin navigates to the registration page after clicking on the 'Créer un compte' button. 2. The system displays the registration form with a CAPTCHA. 3. The new admin enters their details and completes the CAPTCHA. 4. The system validates the provided information and the CAPTCHA. 5. If the CAPTCHA is valid and all information is correct, the system creates the admin account. 6. The system displays a success message.The user confirms by clicking on the 'connection' button. The system displays the appropriate interface for each user.
Alternative Scenario:	The system displays an error message like : 'Champ nom obligatoire' '1 majuscule' 'Veuillez vérifier votre mot de passe' 'COMPTE EXISTE'

Description of the " Manage Company" use case

Use Case:	Manage Company
Actor:	Technical Support
Pre-condition:	Authenticated technical support account
Post-condition:	- Company added - Company deleted - Company details copied
Main Scenario:	The system displays the page
Add Company	1. The technical support enters the necessary information and confirms by clicking on the 'Ajouter' button. 2. The system automatically creates the ID and key for the company. 3. The system saves the data. 4. The system displays a success message. 5. The system redirects the user to the homepage, showing the new company at the top of the list.

Delete Company	1. The technical support clicks on the ‘supprimer’ button on a selected company. 2. The system deletes the company.
Copy Company Details	1. The user clicks on the ‘Copy’ button on a selected company. 2. The system copies the company details.
Alternative Scenario:	The system displays an error message

Description of the " Manage Admin" use case

Use Case:	Manage Admin
Actor:	Technical Support
Pre-condition:	admin account created
Post-condition:	- Admin account activated - Admin account deactivated - Admin account deleted - Account type modified
Main Scenario:	1. The technical Support clicks on the 'voir' button. 2. The system displays the desired page.
View Active or Pending Accounts	1. The technical support selects ‘compte non activee’ or s ‘compte activee’ and clicks on the ‘Afficher’ button. 2. The system displays the corresponding list of accounts.
Activate Admin Account	1. The technical support selects ‘compte non activee’ and clicks on the ‘Afficher’ button. The Manager clicks on the ‘Voir’ button in front of a customer from the list 2. The system displays the pending admin accounts. 3. The technical support clicks on the ‘Activer’ button in front of a customer from the list . 4. The system records the activation.
Deactivate Admin Account	1. The technical support selects ‘compte activee’ and clicks on the ‘Afficher’ button. 2. The system displays the active admin accounts.

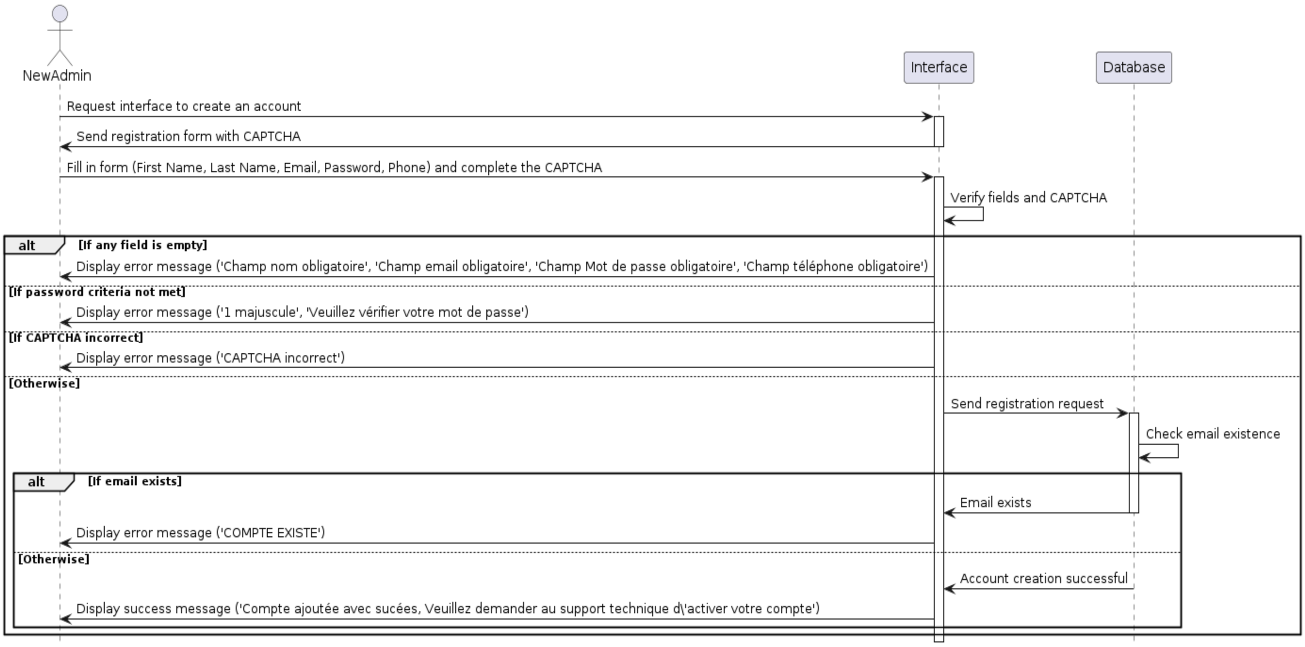
	3. The technical support clicks on the ‘Bloquer’ button in front of a customer from the list 4. The system records the deactivation. 5. The system displays a success message
Modify Account Type	1. The user clicks the account type (admin or not) in the 'admin' table field. 2. The system records the changes. 3. The system displays a success message.
Alternative Scenario:	None

VIII. Design

1. Sequence Diagrams for Sprint 2

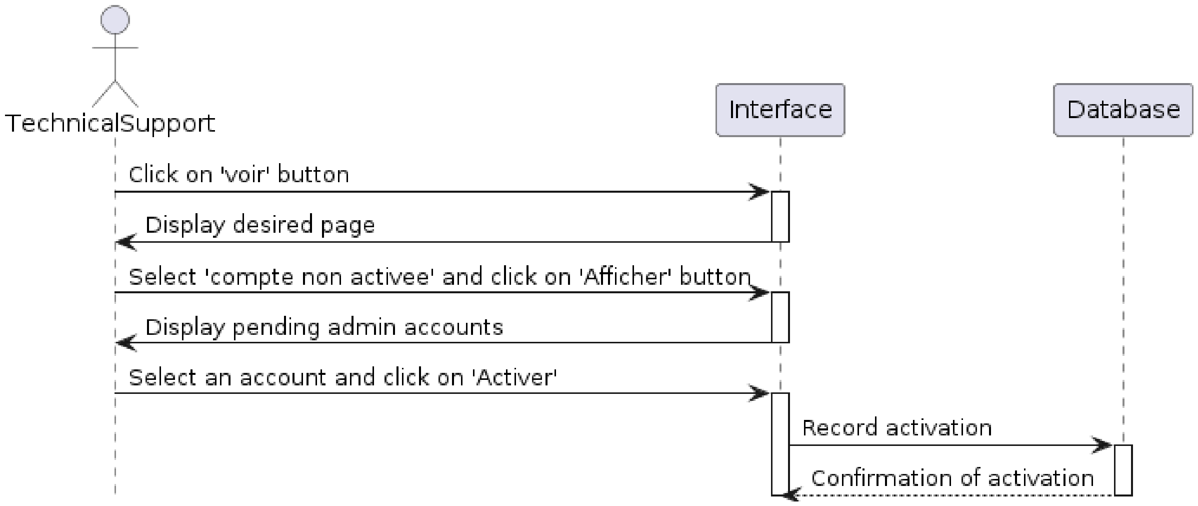
Sequence Diagram for the " Create Admin Account " Use Case

The following figure shows the sequence diagram illustrating the sign-up process:

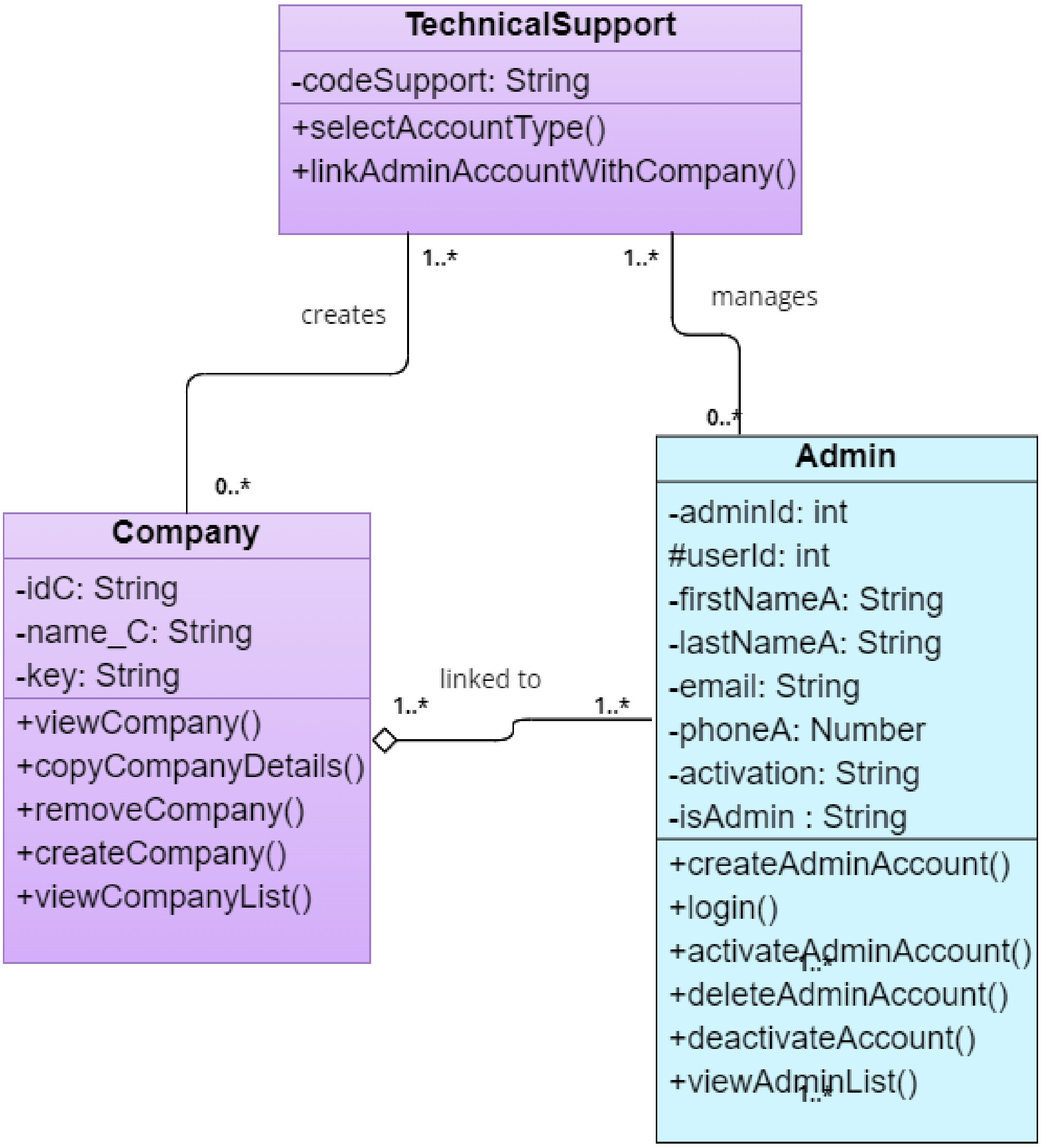

NewAdmin
Interface
Database
Request interface to create an account
Send registration form with CAPTCHA
Fill in form (First Name, Last Name, Email, Password, Phone) and complete the CAPTCHA
Verify fields and CAPTCHA
alt
[If any field is empty]
Display error message ('Champ nom obligatoire', 'Champ email obligatoire', 'Champ Mot de passe obligatoire', 'Champ téléphone obligatoire')
[If password criteria not met]
Display error message ('1 majuscule', 'Veuillez vérifier votre mot de passe')
[If CAPTCHA incorrect]
Display error message ('CAPTCHA incorrect')
[Otherwise]
Send registration request
Check email existence
alt
[If email exists]
Email exists
Display error message ('COMPTE EXISTE')
[Otherwise]
Account creation successful
Display success message ('Compte ajoutée avec sucées, Veuillez demander au support technique d\'activer votre compte')

Sequence Diagram for the " Activate Admin Account " Use Case

The following figure shows the sequence diagram illustrating the sign-up process:


TechnicalSupport
Interface
Database
Click on 'voir' button
Display desired page
Select 'compte non activee' and click on 'Afficher' button
Display pending admin accounts
Select an account and click on 'Activer'
Record activation
Confirmation of activation

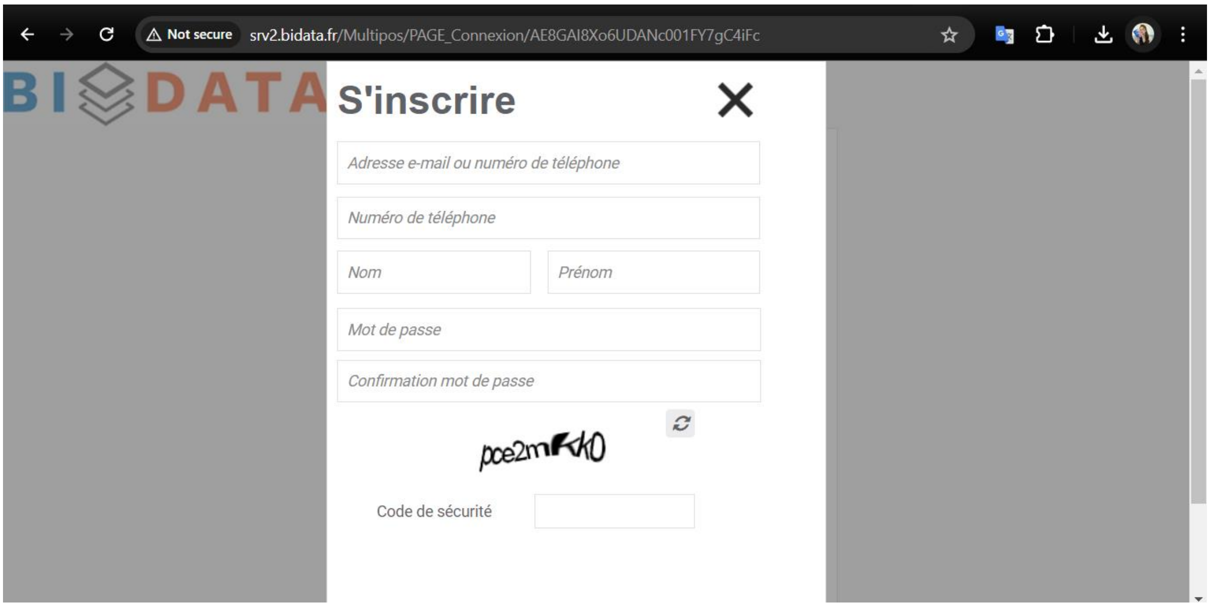
2. Global Class Diagram for Sprint 2



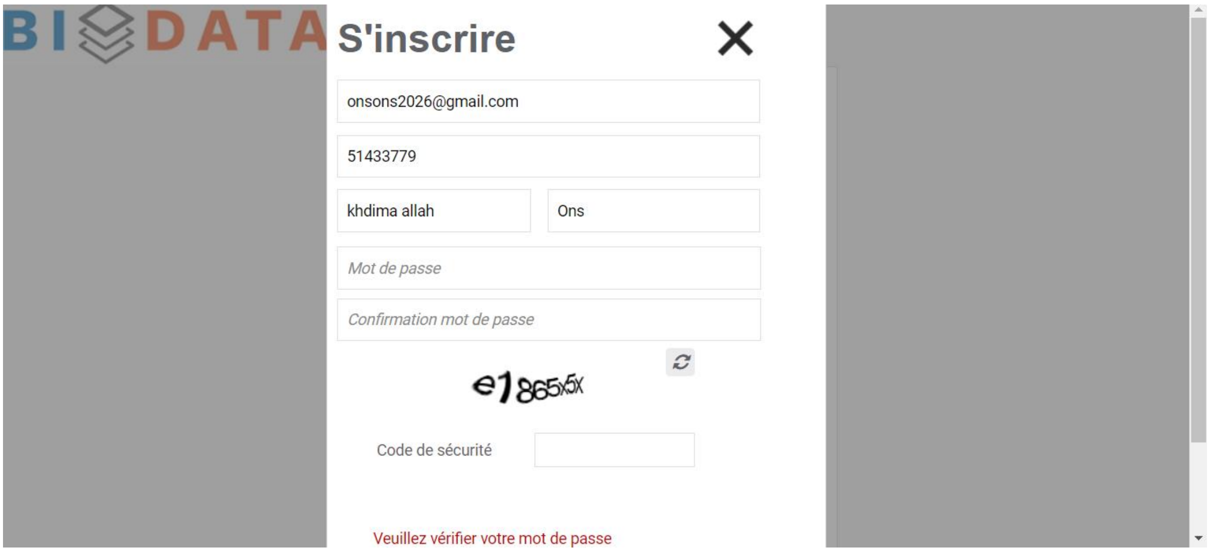
IX. Realisation

This section presents the results obtained through screenshots. The screenshots visually illustrate the final outcomes achieved during tests or simulations. Below are the screenshots for each interface related to this sprint:

Create Admin Account Interface:


Not secure
srv2.bidata.fr/Multipos/PAGE_Connexion/AE8GAI8Xo6UDANc001FY7gC4iFc
BI DATA
S'inscrire
Adresse e-mail ou numéro de téléphone
Numéro de téléphone
Nom
Prénom
Mot de passe
Confirmation mot de passe
pce2mKkO
Code de sécurité

Create Admin Account Interface: Please check your password



onsons2026@gmail.com

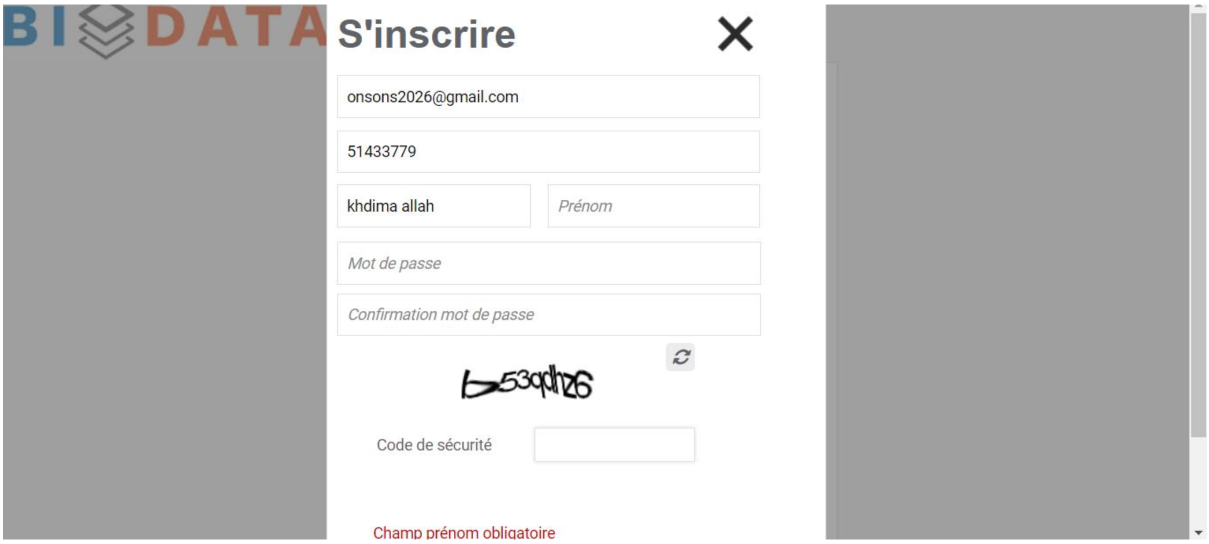
51433779

khdima allah

Ons

Veuillez vérifier votre mot de passe

Create Admin Account Interface: First name field is required



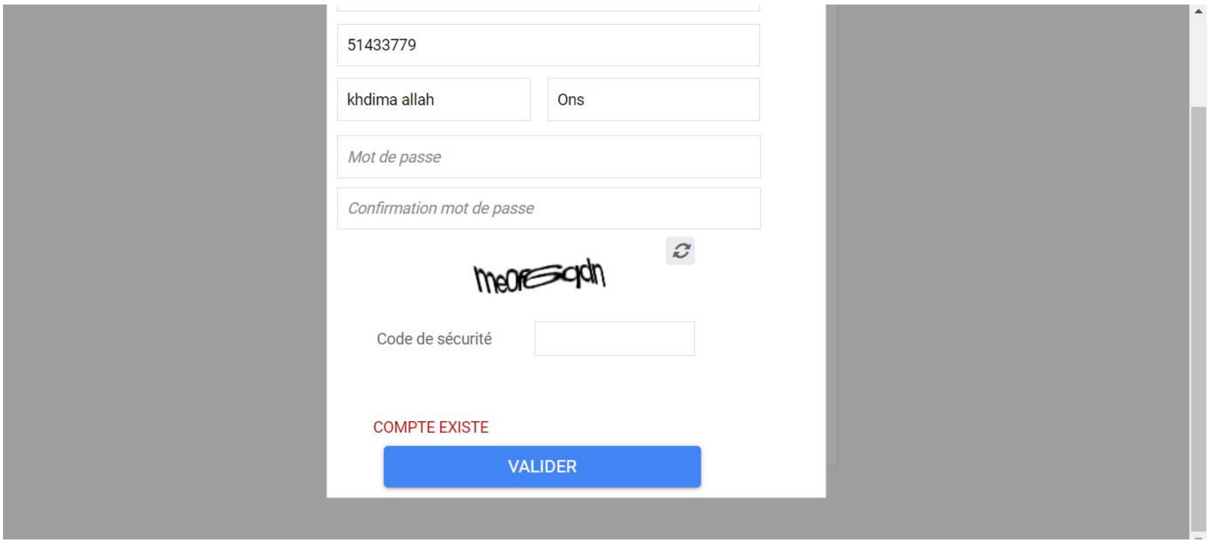
onsons2026@gmail.com

51433779

khdima allah

Champ prénom obligatoire

Create Admin Account Interface: Account already exists



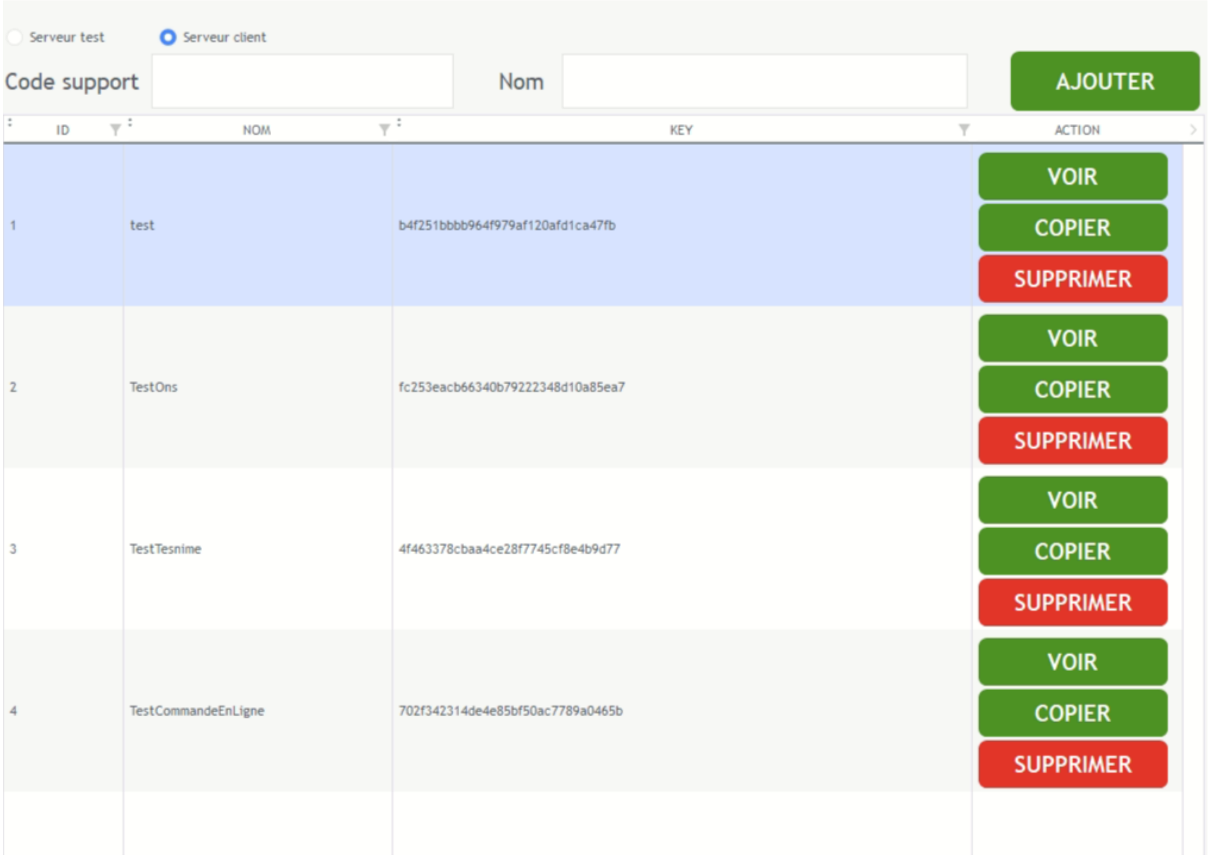
51433779

khdima allah

Ons

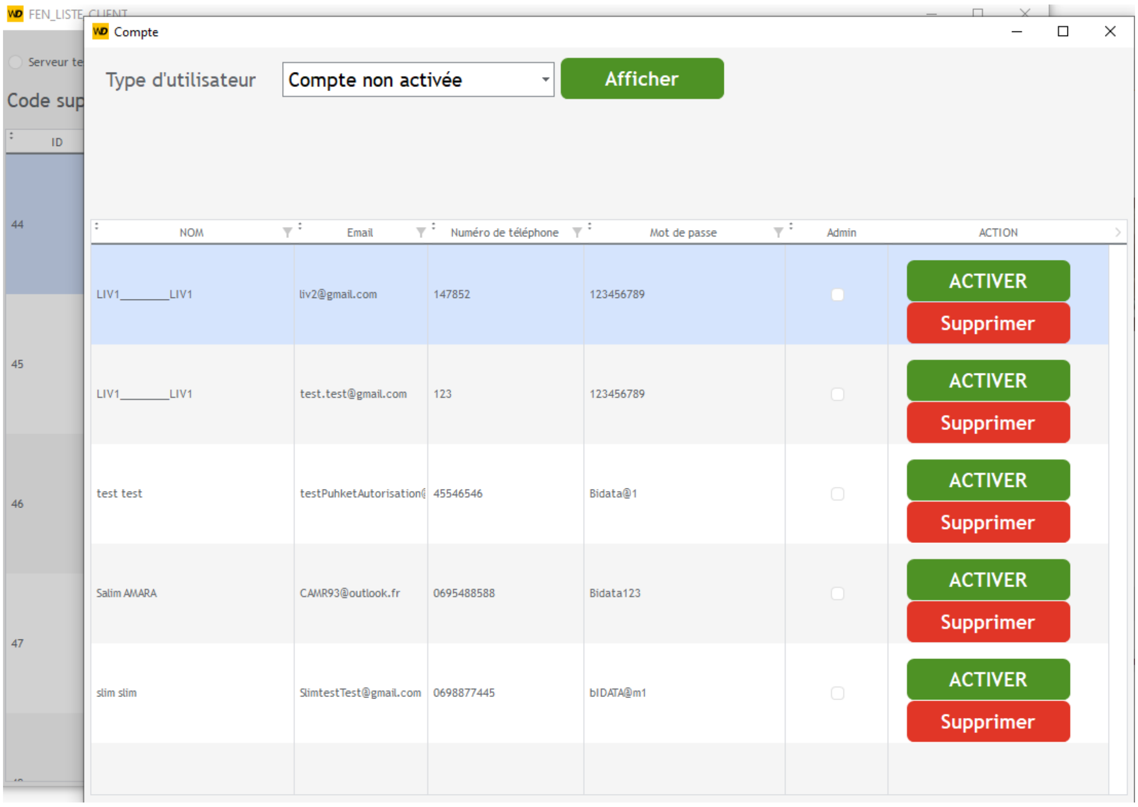
COMPTE EXISTE

Manage Company Interface:



ID	NOM	KEY	ACTION
1	test	b4f251bbbb964f979af120afd1ca47fb	VOIR COPIER SUPPRIMER
2	TestOns	fc253eacb66340b79222348d10a85ea7	VOIR COPIER SUPPRIMER
3	TestTesnime	4f463378cbaa4ce28f7745cf8e4b9d77	VOIR COPIER SUPPRIMER
4	TestCommandeEnLigne	702f342314de4e85bf50ac7789a0465b	VOIR COPIER SUPPRIMER

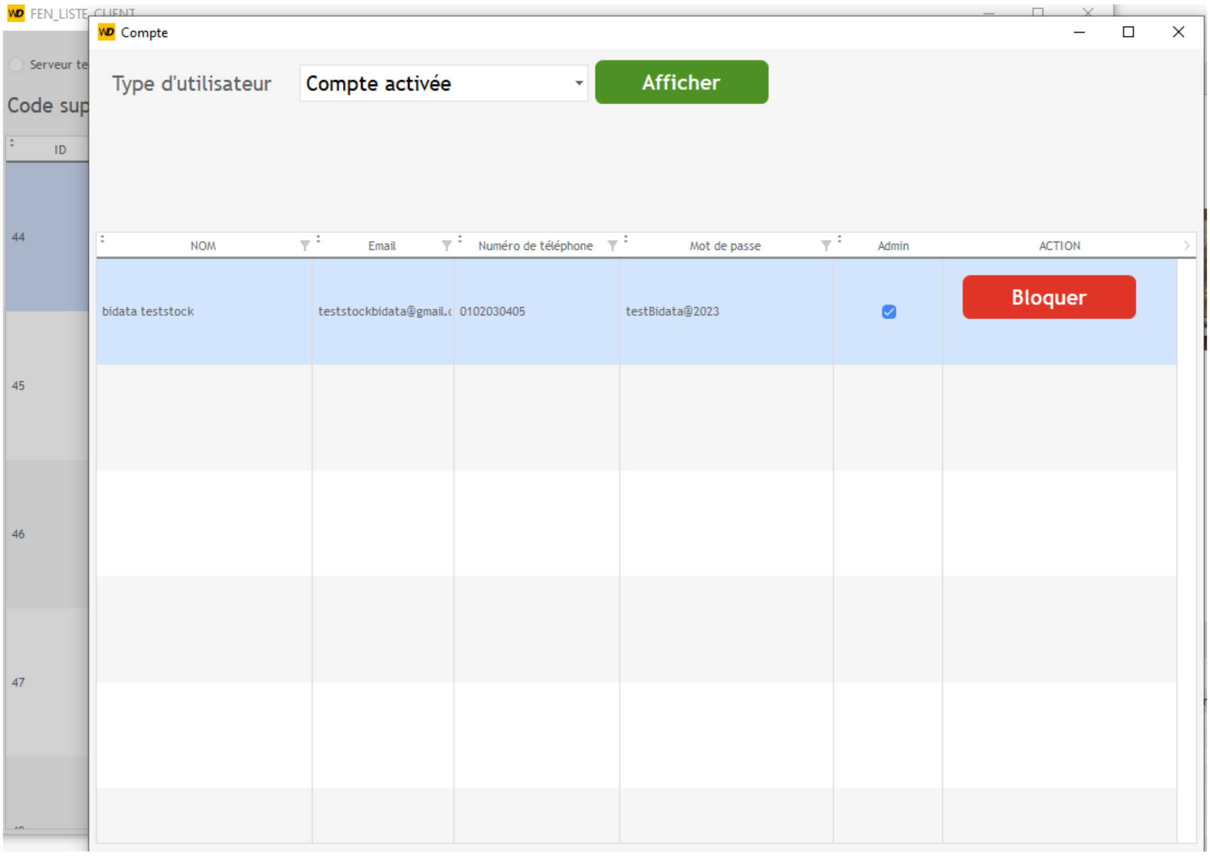
Deactivate Admin Account Interface:



Type d'utilisateur: Compte non activée

NOM	Email	Numéro de téléphone	Mot de passe	Admin	ACTION
LIV1________LIV1	liv2@gmail.com	147852	123456789	□	ACTIVER Supprimer
LIV1________LIV1	test.test@gmail.com	123	123456789	□	ACTIVER Supprimer
test test	testPuhketAutorisation(	45546546	Bidata@1	□	ACTIVER Supprimer
Salim AMARA	CAMR93@outlook.fr	0695488588	Bidata123	□	ACTIVER Supprimer
slim slim	SlimtestTest@gmail.com	0698877445	bIDATA@m1	□	ACTIVER Supprimer

Activate Admin Account Interface:



Type d'utilisateur Compte activée

NOM	Email	Numéro de téléphone	Mot de passe	Admin	ACTION
bidata teststock	teststockbidata@gmail.c	0102030405	testBidata@2023	☑	Bloquer

Conclusion

In this chapter, we have explored the first release, focusing on the user stories and tasks for the first and second sprints. Detailed analysis and design elements, including global use case diagrams and sequence diagrams, have been presented. The chapter concludes with the realization phase, highlighting the implementation outcomes through screenshots. Moving forward, the next chapter will continue with the second release, focusing on the tasks and objectives of the subsequent sprint.

Chapter 4 : RELEASE 2

Introduction

This chapter focuses on the second release of our project. We will outline the organization and tasks of the third sprint, detailing the user stories and requirements addressed.

I. Sprint organization

II. Sprint 3 Backlog

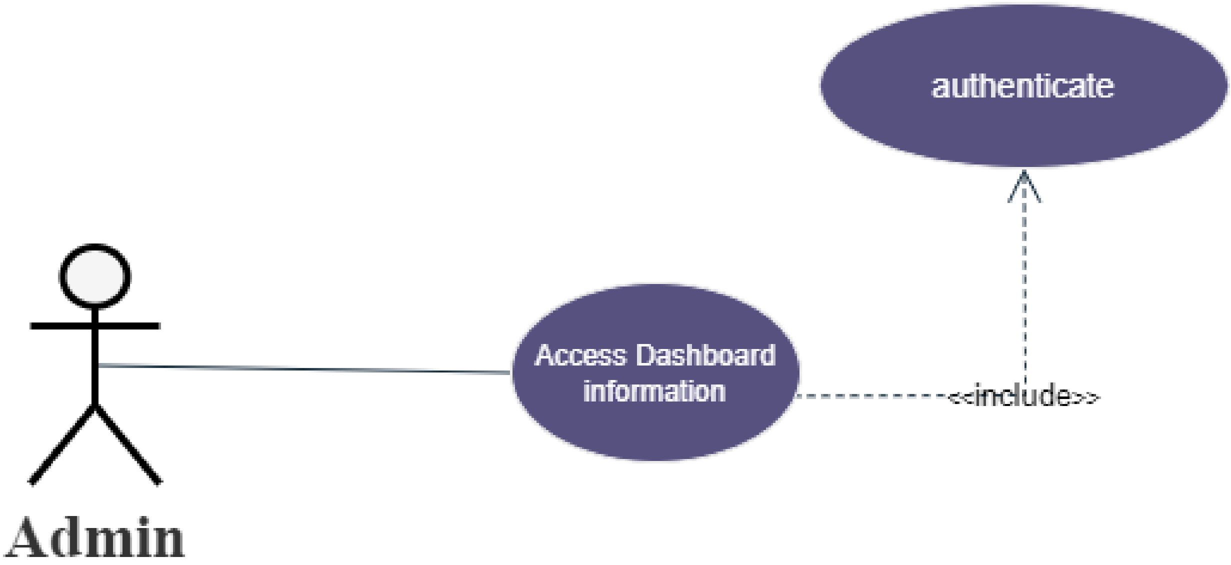
Number	User story	The tasks	Estimated Time
12	As an admin, I want to access a dashboard with loyalty program performance metrics	Analysis of requirements	3
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
13	As an admin, I want to filter and view the dashboard data based on specific criteria	Analysis of requirements	5
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
14	As an admin, I want to view the total revenue (CA TTC) for the selected date range so that I can monitor overall performance.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	

15	As an admin, I want to view the net revenue (CA HT) for the selected date range to understand the company's financial health.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
16	As an admin, I want to see the average cart value including taxes (Panier moyen TTC) to gauge customer spending behavior.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
17	As an admin, I want to see the average cart value excluding taxes (Panier moyen HT) for financial analysis.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
18	As an admin, I want to see the revenue from payments received (CA encaissement) to track cash flow.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
19	As an admin, I want to see the number of order cancellations (Annulation) to identify potential issues in order processing.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	

20	As an admin, I want to view the trend of orders over time (Commande) to monitor sales trends, with the ability to filter the data by date and hour.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
21	As an admin, I want to view the breakdown of payment methods (Règlements) to understand customer preferences.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
22	As an admin, I want to see the distribution of sales by service provider (Prestataire) to assess performance.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
23	As an admin, I want to see the sales by type of sale including taxes (Mode de vente TTC) to analyze sales channels.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
24	As an admin, I want to see the sales by type of sale excluding taxes (Mode de vente HT) for detailed financial analysis.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
25	As an admin, I want to see the distribution of sales by article or family to understand product performance.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	

III. Analysis

1. Global Use Case Diagram for Sprint 3



2. Descriptions of Sprint 3

Description of the " Access Dashboard information" use case

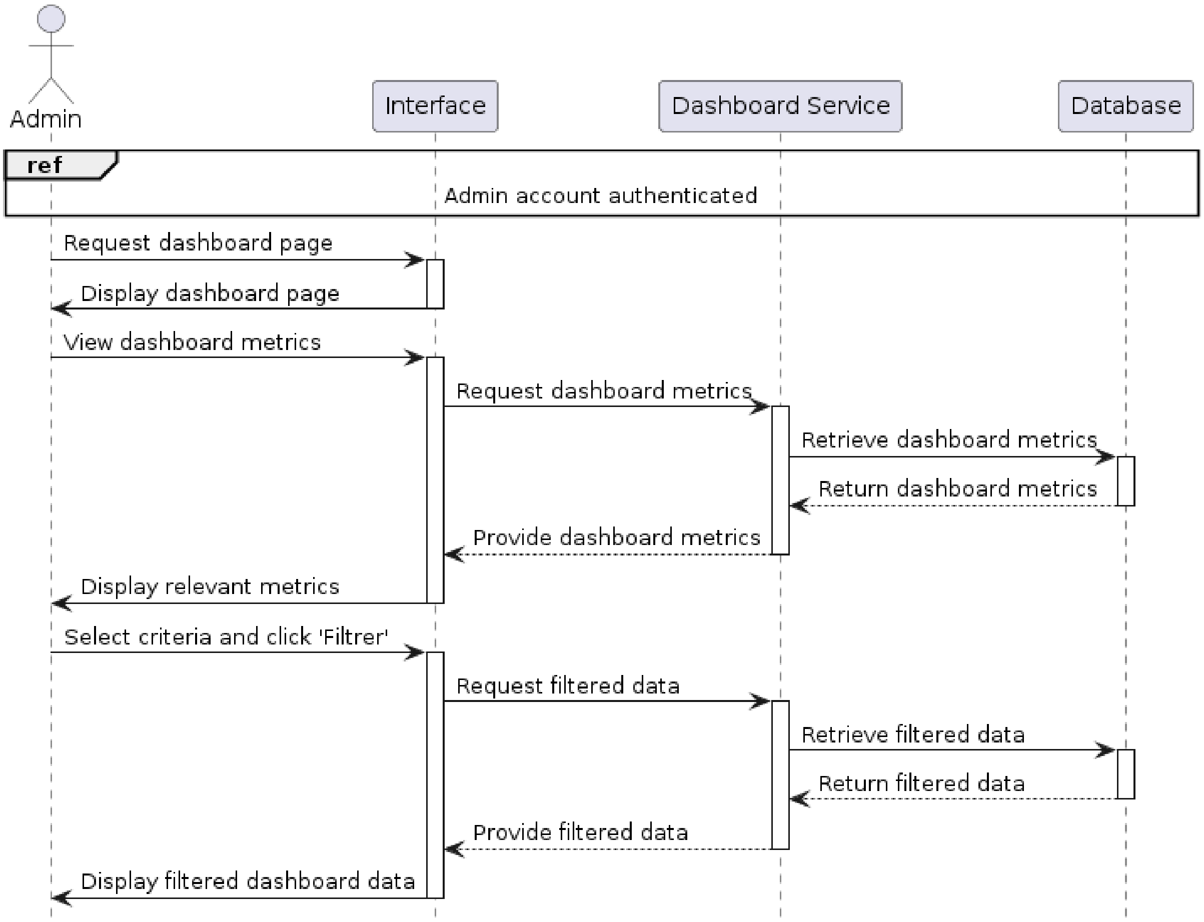
Use Case:	Access Dashboard information
Actor:	Admin
Pre-condition:	Admin account created and authenticated
Post-condition:	- Dashboard data viewed - Data filtered based on specific criteria
Main Scenario:	The system displays the dashboard page.
View Dashboard Metrics	1. The admin views the dashboard with loyalty program performance metrics. 2. The system displays the relevant metrics.
Filter Dashboard Data	1. The admin selects specific criteria and clicks on the 'Filtrer' to filter the data. 2. The system filters and displays the dashboard data based on the selected criteria.
Alternative Scenario:	None

IV. Design

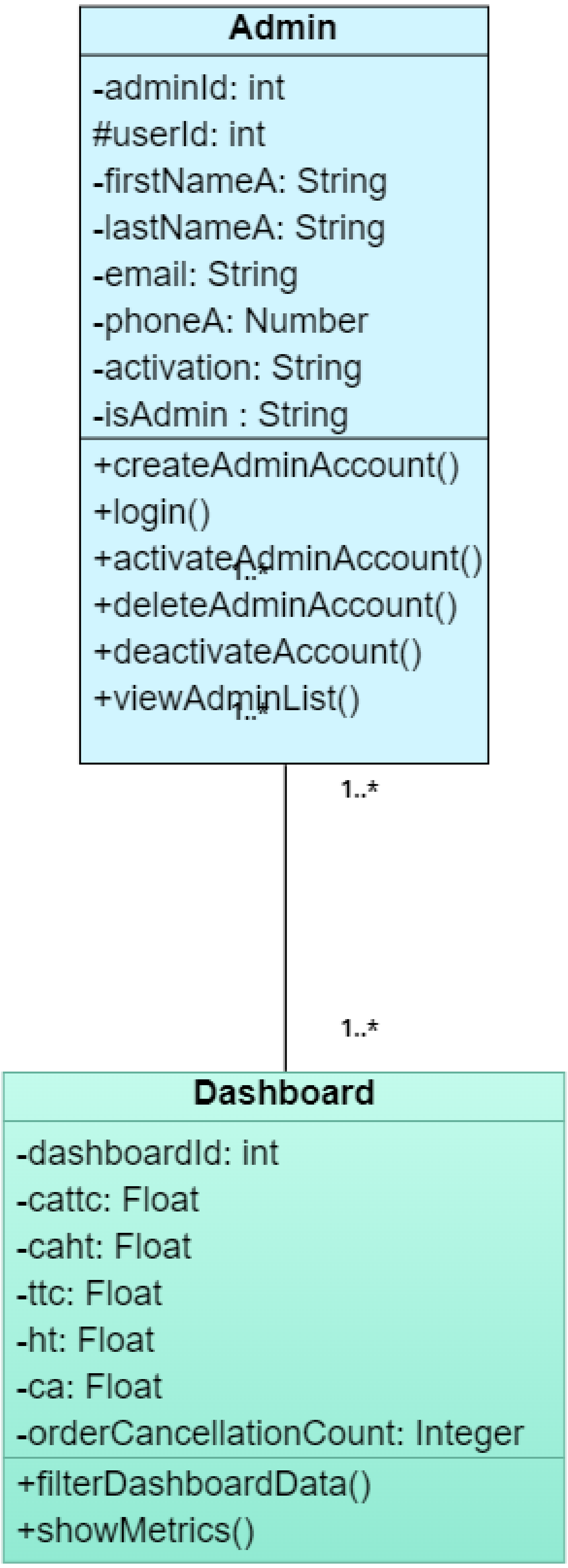
1. Sequence Diagrams for Sprint 3

Sequence Diagram for the " Access Dashboard information " Use Case

The following figure shows the sequence diagram illustrating the sign-up process:



2. Global Class Diagram for Sprint 3



V. Realisation

This section presents the results obtained through screenshots. The screenshots visually illustrate the final outcomes achieved during tests or simulations. Below are the screenshots for each interface related to this sprint:

Dashboard Admin Interfaces

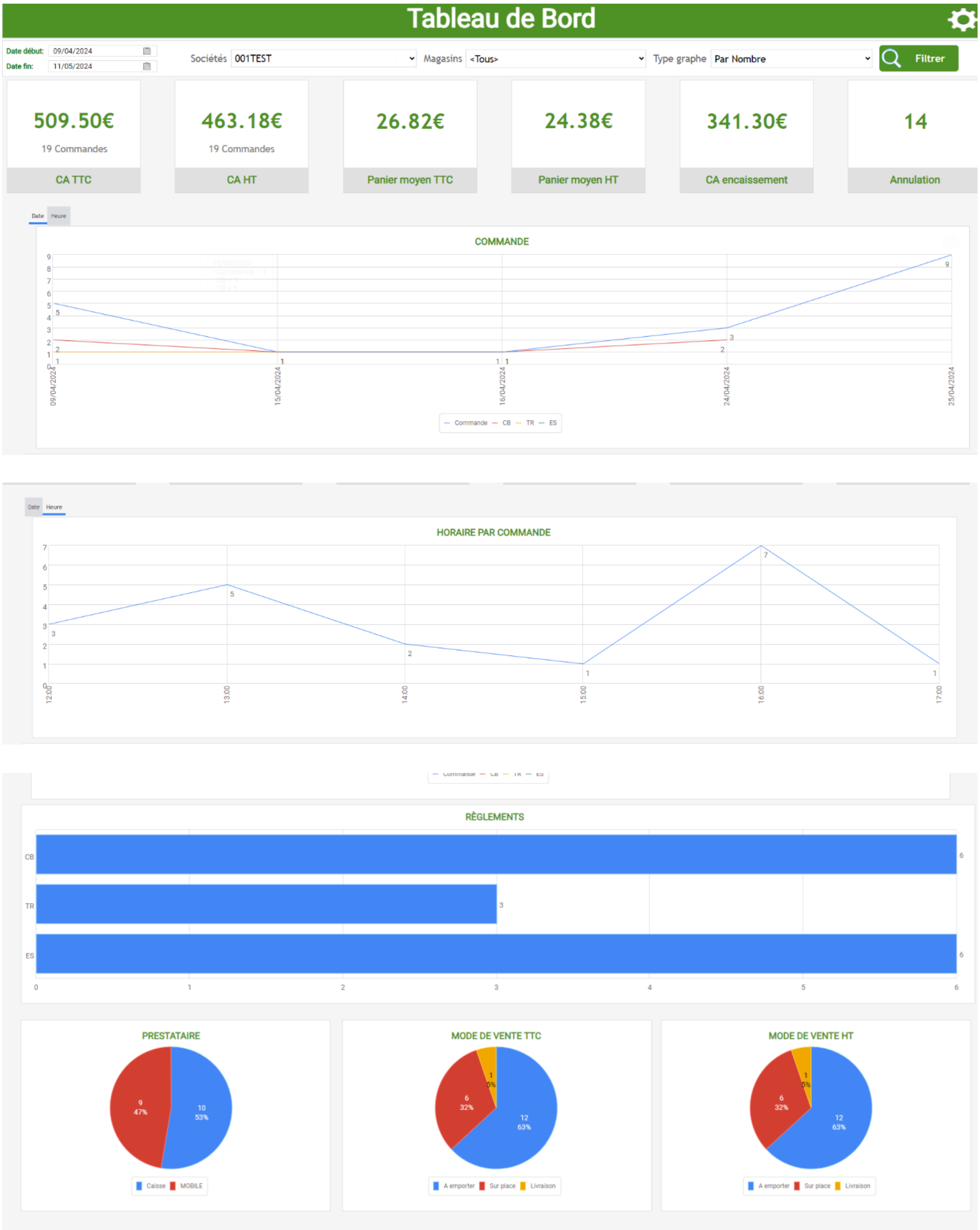
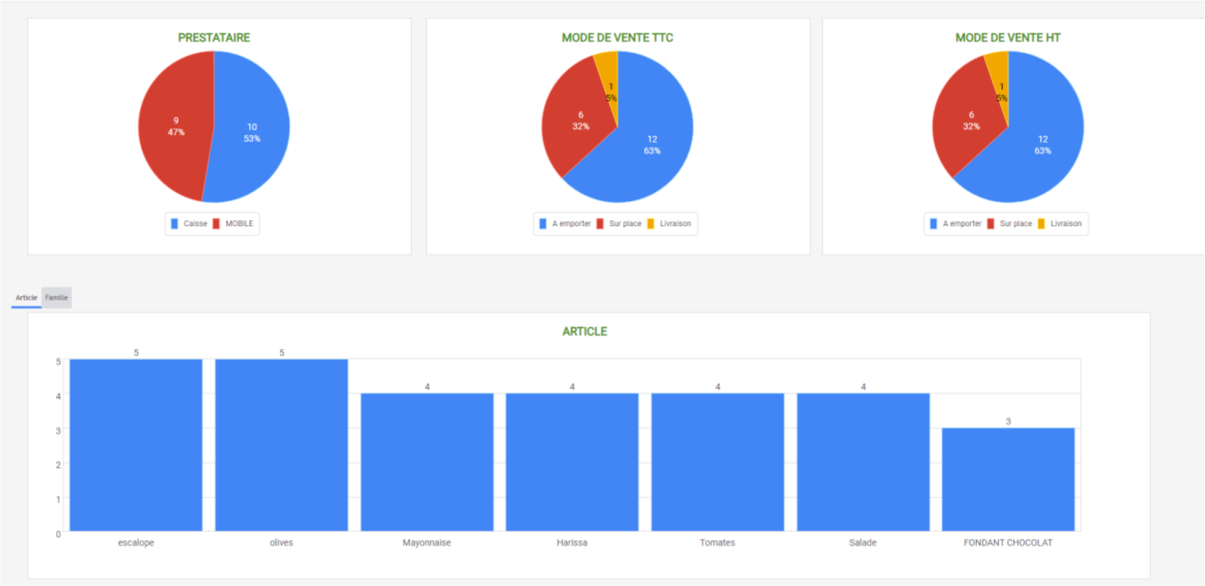


Tableau de Bord

Date début: 09/04/2024
Date fin: 11/05/2024
Sociétés: 001TEST
Magasins: <Tous>
Type graphe: Par Nombre

CA TTC	CA HT	Panier moyen TTC	Panier moyen HT	CA encaissement	Annulation
509.50€ 19 Commandes	463.18€ 19 Commandes	26.82€	24.38€	341.30€	14



Conclusion

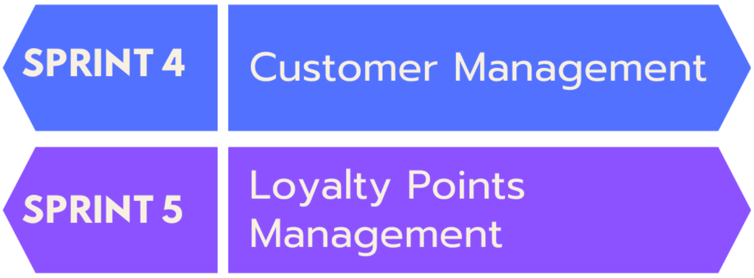
This chapter has covered the organization and execution of the second release. We have elaborated on the user stories and tasks assigned for the third sprint, accompanied by relevant analysis, design, and implementation details. The chapter also includes screenshots showcasing the progress and completion of the sprint. In the next chapter, we will move on to the third release, addressing the objectives and tasks for the subsequent sprints.

Chapter 5 : RELEASE 3

Introduction

In this chapter, we examine the third release, covering the fourth and fifth sprints. We will present the sprint organization, backlog items, and the detailed analysis, design, and implementation processes. By exploring the global use case diagrams, sequence diagrams, and implementation outcomes, we aim to illustrate the final stages of our project's development.

I. Sprint organization



SPRINT 4 | Customer Management

SPRINT 5 | Loyalty Points Management

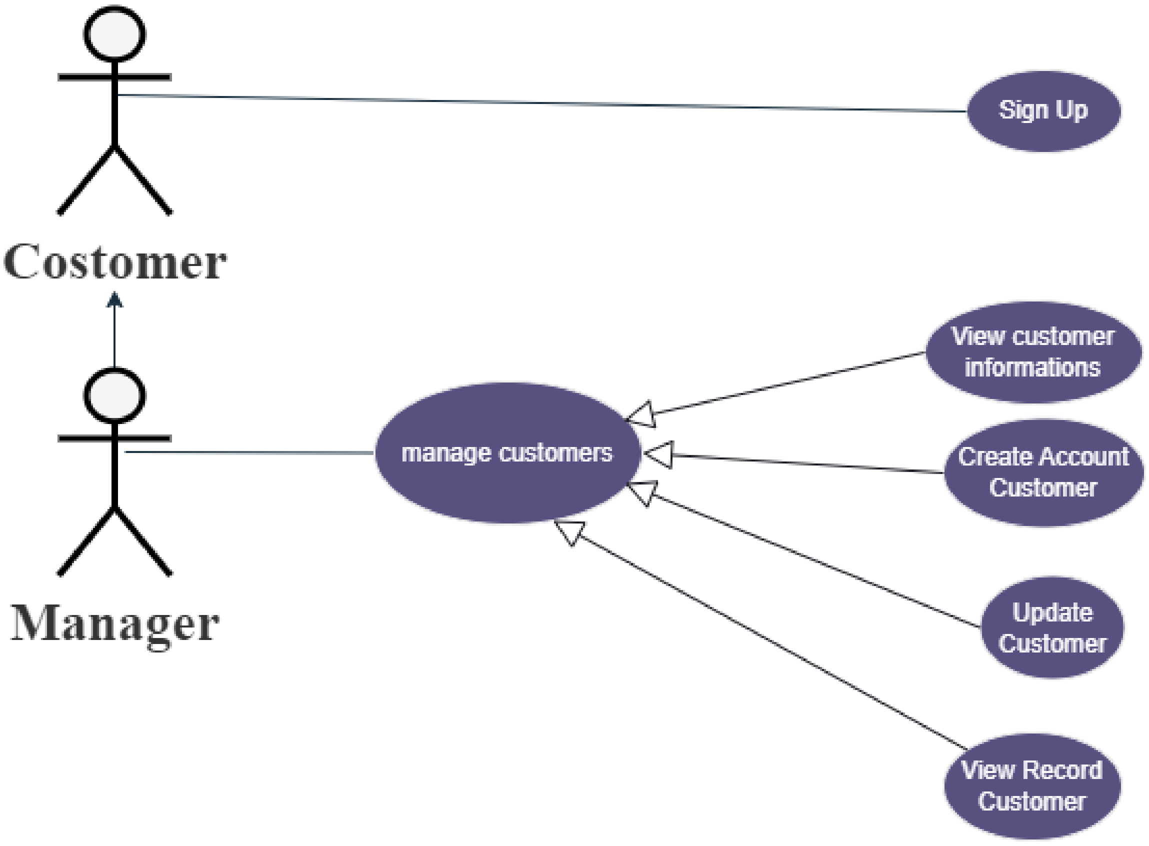
II. Sprint 4 Backlog

Number	User story	The tasks	Estimated Time
26	As a customer, I want to sign up using self-service checkouts and interactive terminals so that I can easily join and participate in the loyalty program.	Analysis of requirements	3
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
27	As a manager, I want to add new customers to the system so that they can start earning and redeeming loyalty points.	Analysis of requirements	5
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
28		Analysis of requirements	2

	As a manager, I want to view customer details and loyalty points so that I can provide personalized service and support.	Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
29	As a manager, I want to update customer information	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	

III. Analysis

1. Global Use Case Diagram for Sprint 4



2. Descriptions of Sprint 4

Description of the " Sign Up" use case

Use Case:	Sign Up
Actor:	Customer
Pre-condition:	Self-service checkouts and interactive terminals are operational.
Post-condition:	-New customer account is created and saved in the system. -Existing customer information is updated if the phone number matches.
Main Scenario:	1. The customer accesses a self-service checkout or interactive terminal. 2.The customer enters the details in the form like: CIN,Nom,Prénom... 3.The customer clicks the ‘Créer’ button. 4.The system checks if the phone number already exists in the database
Alternative Scenario: Phone Number Exists	1. If the phone number exists, the system updates the existing customer information with the new data except for the phone number. 2.The system displays a success message indicating that the customer's information has been updated.
Alternative Scenario: Phone Number Does Not Exist	1. If the phone number does not exist, the system saves the new customer data. 2.The system displays a success message indicating that the new customer account has been created.

Description of the " Manage Customer" use case

Use Case:	Manage Customer
Actor:	Manager
Pre-condition:	Customer accounts exist in the system
Post-condition:	- New customer added

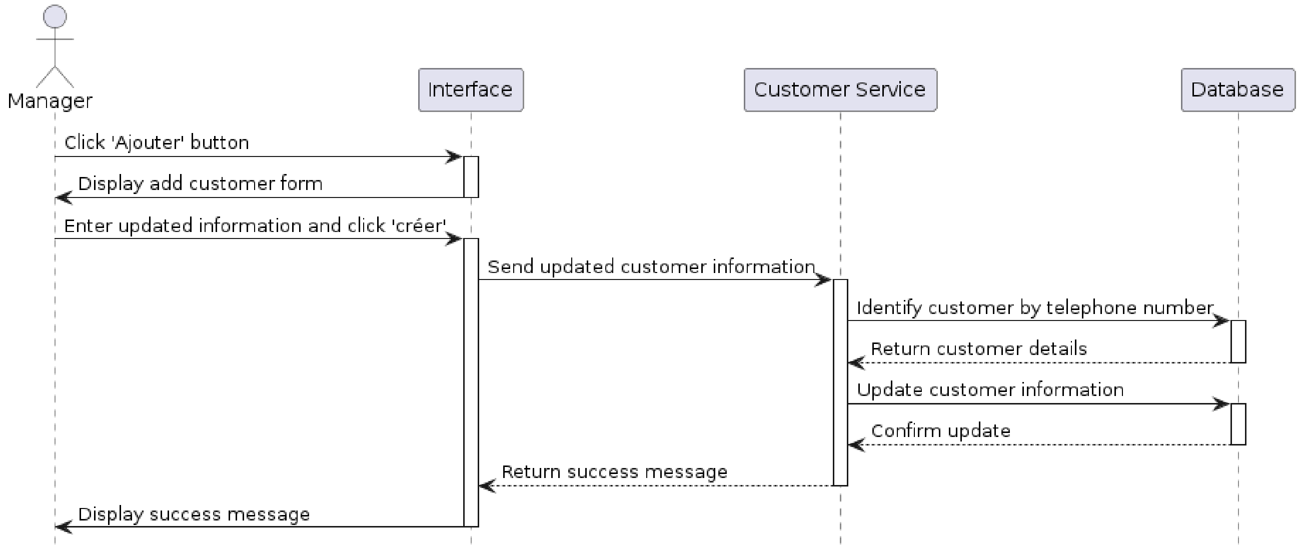
	- Customer details viewed - Customer information updated
Main Scenario:	The system displays the list customer page.
Add New Customer	1. The manager clicks on the ‘Ajouter’ button. 2. The system displays the add customer form. 3. The manager enters the necessary information and confirms by clicking on the ‘créer’ button. 4. The system saves the customer data. 5. The system adds the customer.
View Customer Details	1. The manager clicks on the ‘Voir’ button in front of a customer. 2. The system displays the customer details and loyalty points.
Update Customer Information	1. The manager clicks on the ‘Ajouter’ button. 2. The system displays the add customer form. 3. The manager enters the updated information, ensuring the same telephone number is used, and clicks ‘créer’. 4. The system identifies the existing customer by the telephone number. 5. The system updates the customer's information with the new data, except for the telephone number. 6. The system displays a success message .
Alternative Scenario:	None

IV. Design

1. Sequence Diagrams for Sprint 4

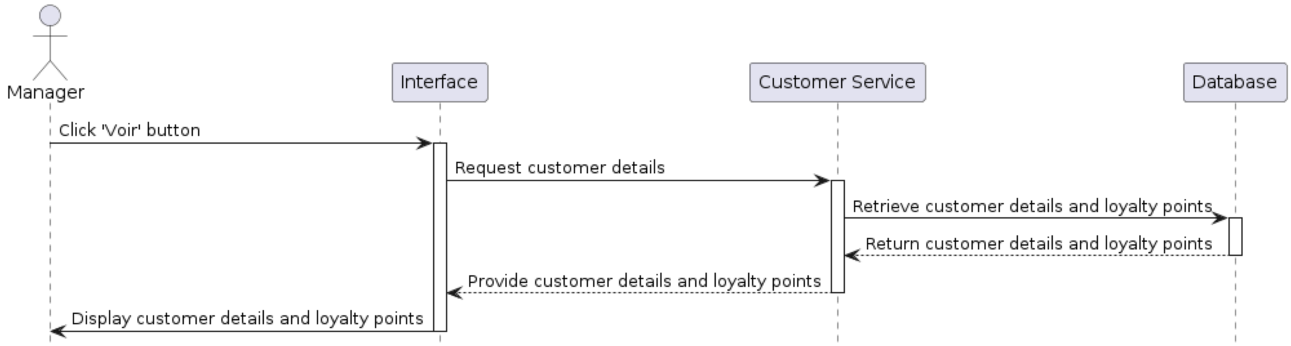
Sequence Diagram for the " Update Customer Information" part of the "Manage Customer" use case

The following figure shows the sequence diagram illustrating the sign-up process:

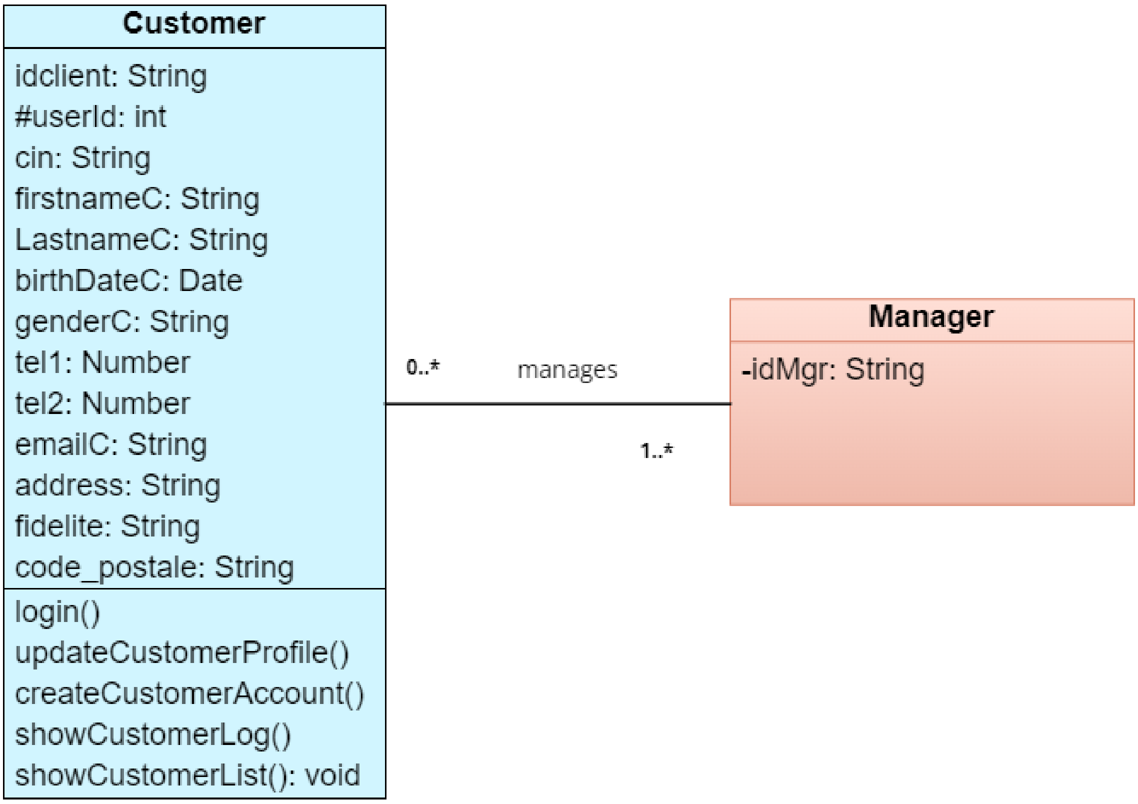

Manager
Interface
Customer Service
Database
Click 'Ajouter' button
Display add customer form
Enter updated information and click 'créer'
Send updated customer information
Identify customer by telephone number
Return customer details
Update customer information
Confirm update
Return success message
Display success message

Sequence Diagram for the " View Customer Details" part of the "Manage Customer" use case

The following figure shows the sequence diagram illustrating the sign-up process:


Manager
Interface
Customer Service
Database
Click 'Voir' button
Request customer details
Retrieve customer details and loyalty points
Return customer details and loyalty points
Provide customer details and loyalty points
Display customer details and loyalty points

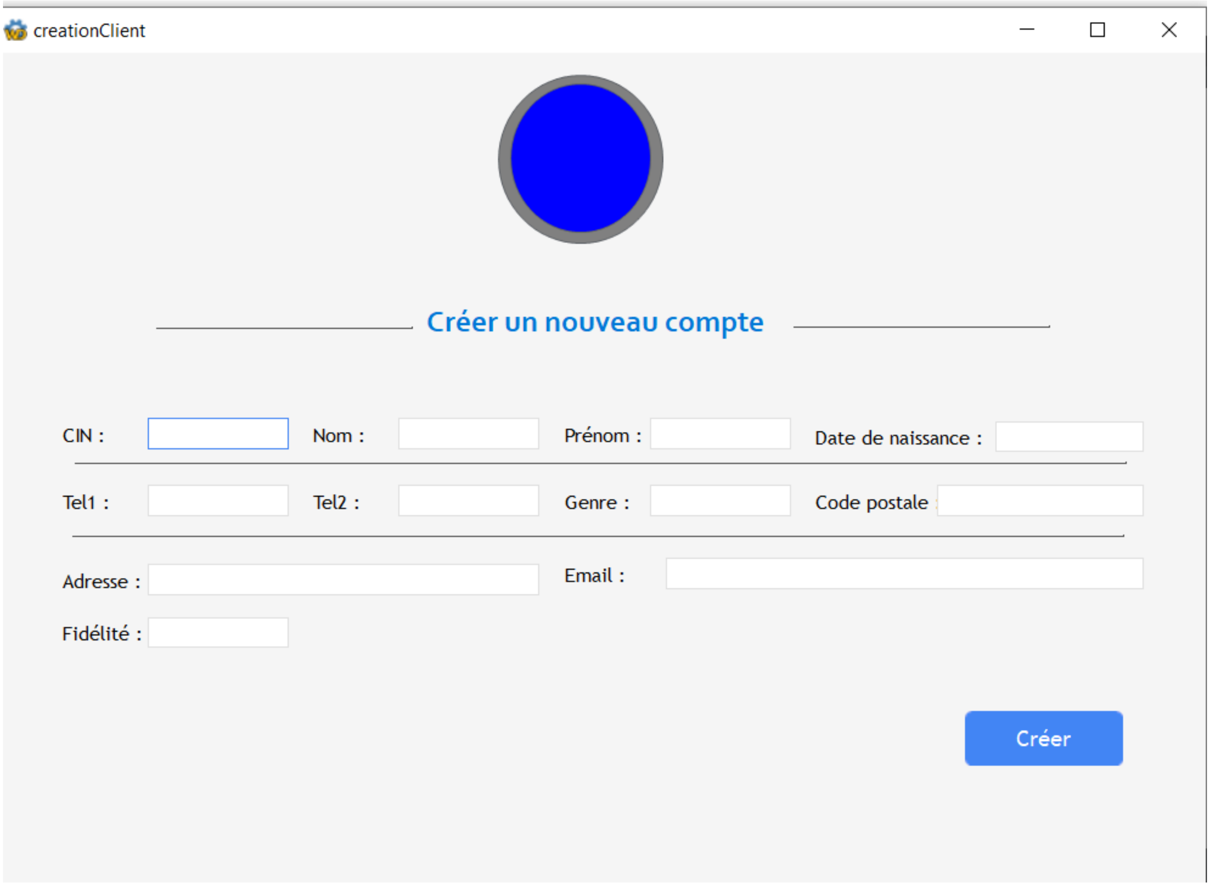
2. Global Class Diagram for Sprint 4



V. Realisation

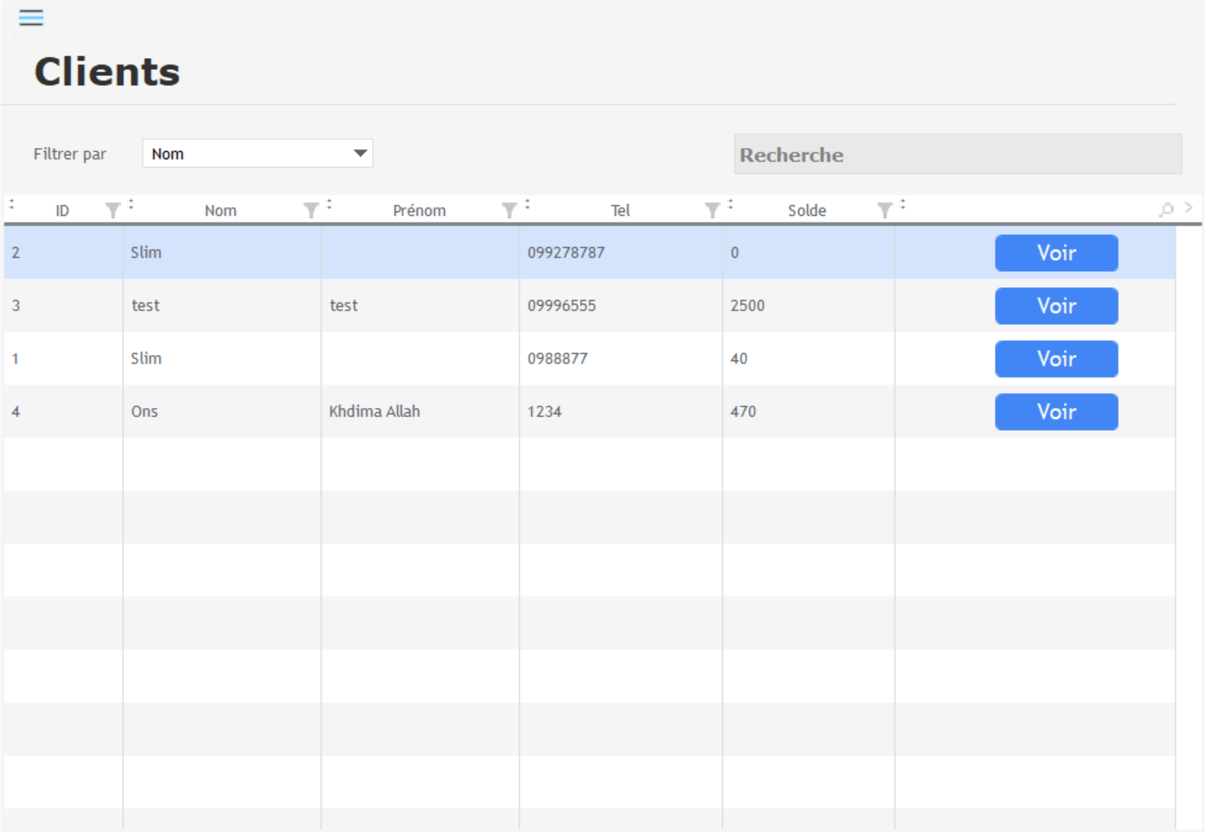
This section presents the results obtained through screenshots. The screenshots visually illustrate the final outcomes achieved during tests or simulations. Below are the screenshots for each interface related to this sprint:

Create or update Customer Account Interface:



CIN :
Nom :
Prénom :
Date de naissance :
Tel1 :
Tel2 :
Genre :
Code postale
Adresse :
Email :
Fidélité :

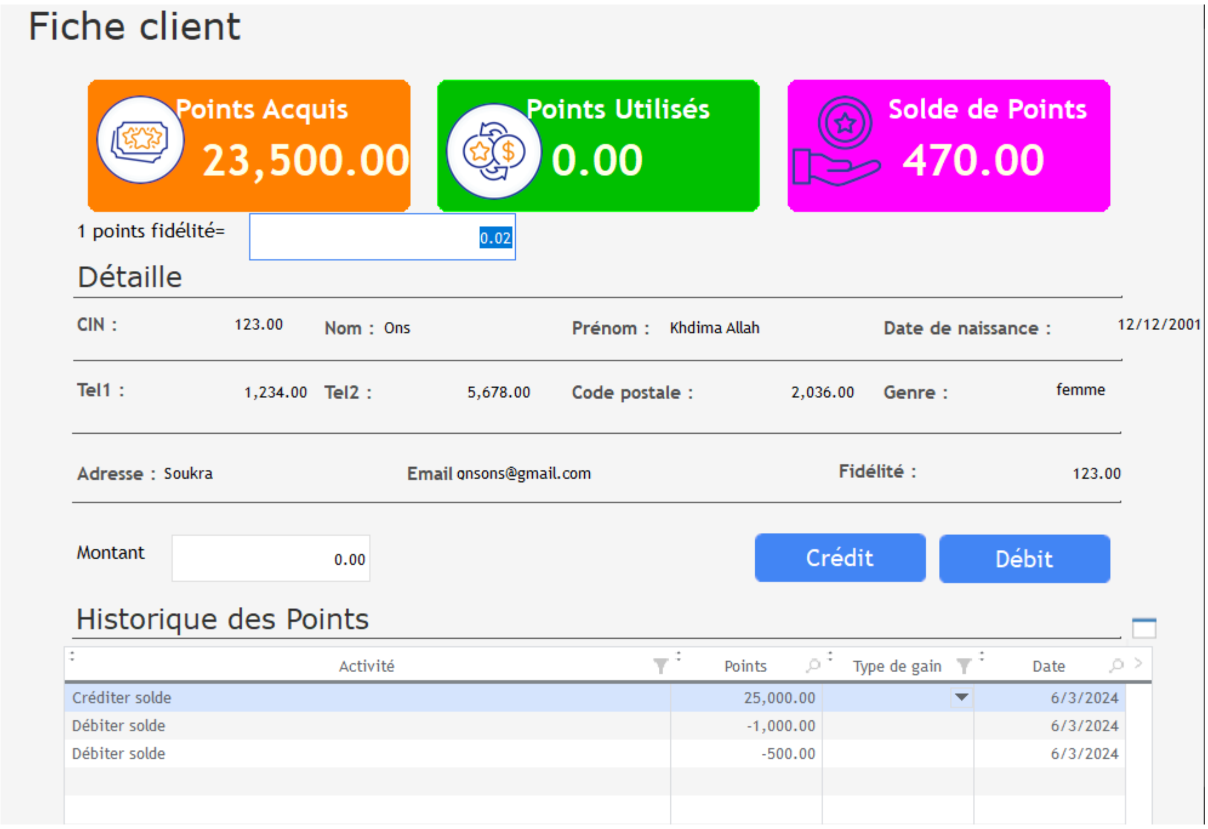
List customer Interface



Filtrer par Nom

ID	Nom	Prénom	Tel	Solde	
2	Slim		099278787	0	Voir
3	test	test	09996555	2500	Voir
1	Slim		0988877	40	Voir
4	Ons	Khdima Allah	1234	470	Voir

Customer Details Interface



Fiche client

Points Acquis	Points Utilisés	Solde de Points
23,500.00	0.00	470.00

1 points fidélité= 0.02

Détaille

CIN :	123.00	Nom :	Ons	Prénom :	Khdima Allah	Date de naissance :	12/12/2001
Tel1 :	1,234.00	Tel2 :	5,678.00	Code postale :	2,036.00	Genre :	femme
Adresse :	Soukra	Email	onsons@gmail.com	Fidélité :	123.00		

Montant 0.00

Historique des Points

Activité	Points	Type de gain	Date
Créditer solde	25,000.00		6/3/2024
Débiter solde	-1,000.00		6/3/2024
Débiter solde	-500.00		6/3/2024

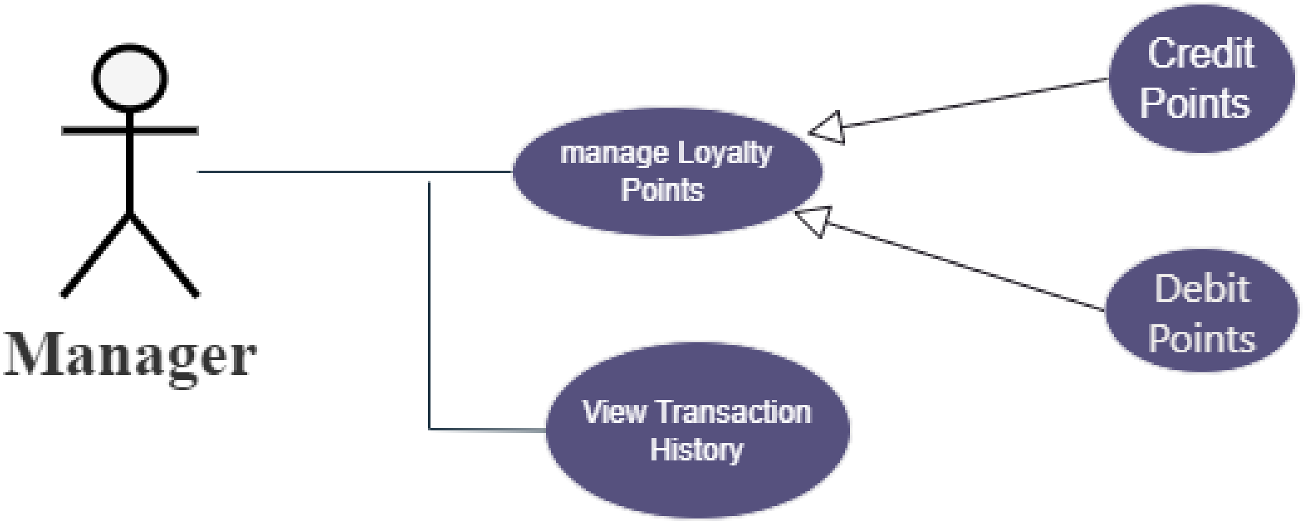
VI. Sprint 5 Backlog

Number	User story	The tasks	Estimated Time
30	As a manager, I want to credit points to customer accounts	Analysis of requirements	3
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	
31	As a manager, I want to debit points from customer accounts	Analysis of requirements	5
		Design	
		Code front-end part	
		Code back-end part	

		Integration testing	
32	As a manager, I can view customer transaction histories.	Analysis of requirements	2
		Design	
		Code front-end part	
		Code back-end part	
		Integration testing	

VII. Analysis

1. Global Use Case Diagram for Sprint 5



2. Descriptions of Sprint 5

Description of the "Manage Loyalty Points" Use Case

Use Case:	Manage Loyalty Points
Actor:	Manager
Pre-condition:	. Customer accounts exist in the system. The manager is on the page displaying customer details.
Post-condition:	Customer accounts are updated with credited or debited loyalty points.
Main Scenario:	1.The manager clicks on the ‘Voir’ button in front of a customer from the list.

	2.The system displays the customer details, including current loyalty points. 3.The manager enters the value of 1 loyalty point in dollars in the provided field.
Credit Points	1.The manager enters the amount and clicks ‘Crédit’. 2.The system calculates the equivalent loyalty points based on the entered value of 1 loyalty point in dollars. 3.The system updates the customer’s loyalty points balance by adding the calculated number of points.
Debit Points	1.The manager enters the amount and clicks ‘Débit’. 2.The system calculates the equivalent loyalty points based on the entered value of 1 loyalty point in dollars. 3. The system updates the customer’s loyalty points balance by subtracting the calculated number of points.
Alternative Scenario:	None

Description of the "View Transaction History" Use Case

Use Case:	Manage Loyalty Points
Actor:	Manager
Pre-condition:	. Customer accounts exist in the system. The manager is on the page displaying customer details.
Post-condition:	The transaction history of the selected customer is displayed to the Manager
Main Scenario:	1,The system displays the client management page. 2.The Manager clicks on the ‘Voir’ button in front of a customer from the list. 3.The system displays the customer details, including current loyalty points and transaction history. 4.The Manager scrolls down to the "Historique des Points" (Points History) section to view the transaction history.

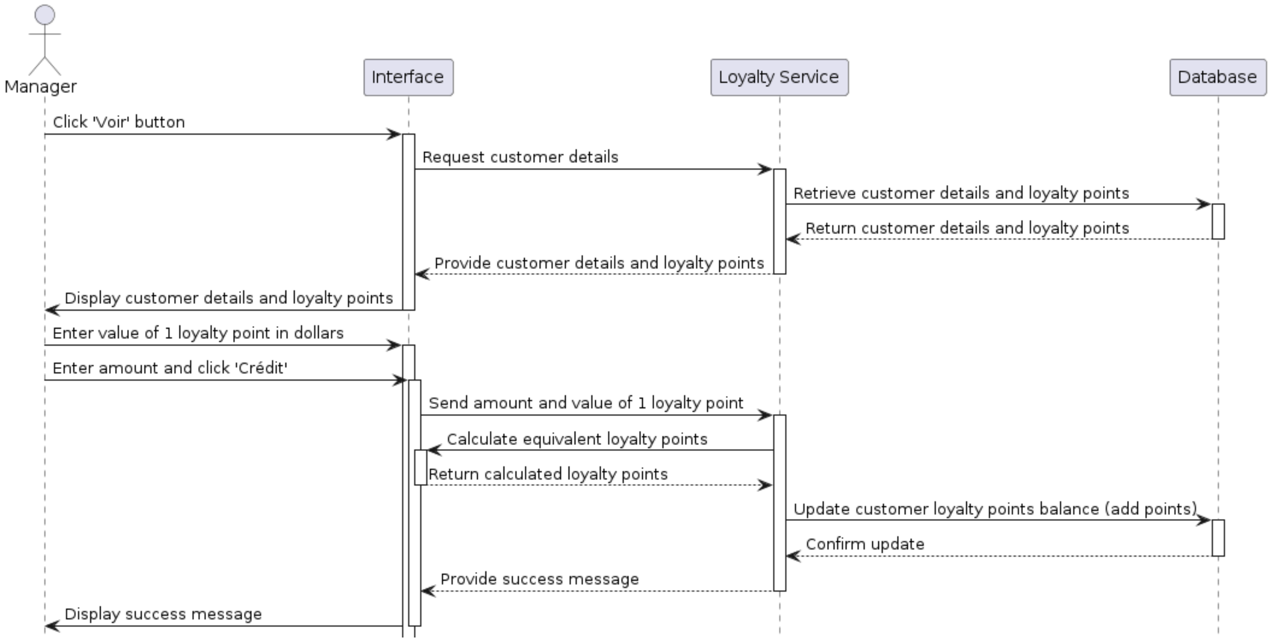
Alternative Scenario:	None

VIII. Design

1. Sequence Diagrams for Sprint 5

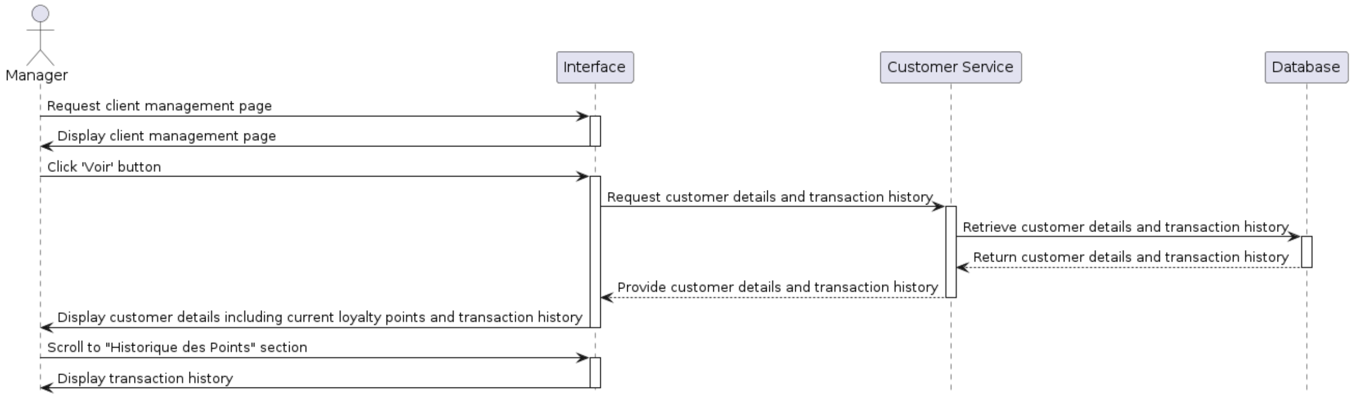
Sequence Diagram for the " Credit Points " part of the " Manage Loyalty Points " use case

The following figure shows the sequence diagram illustrating the sign-up process:

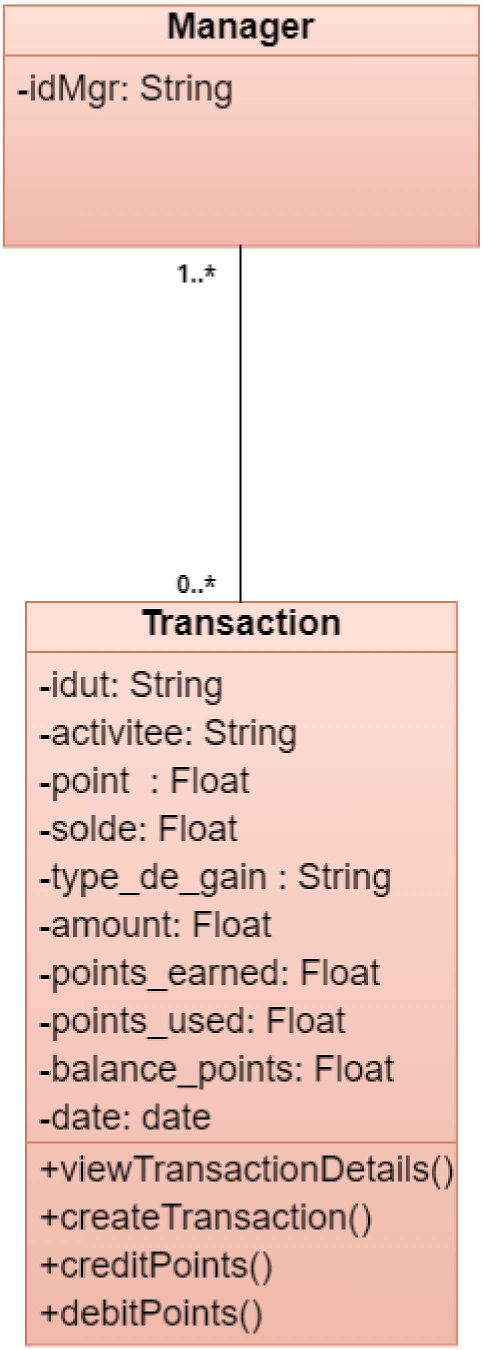

Manager
Interface
Loyalty Service
Database
Click 'Voir' button
Request customer details
Retrieve customer details and loyalty points
Return customer details and loyalty points
Provide customer details and loyalty points
Display customer details and loyalty points
Enter value of 1 loyalty point in dollars
Enter amount and click 'Crédit'
Send amount and value of 1 loyalty point
Calculate equivalent loyalty points
Return calculated loyalty points
Update customer loyalty points balance (add points)
Confirm update
Provide success message
Display success message

Sequence Diagram for the " View Transaction History " use case

The following figure shows the sequence diagram illustrating the sign-up process:


Manager
Interface
Customer Service
Database
Request client management page
Display client management page
Click 'Voir' button
Request customer details and transaction history
Retrieve customer details and transaction history
Return customer details and transaction history
Provide customer details and transaction history
Display customer details including current loyalty points and transaction history
Scroll to "Historique des Points" section
Display transaction history

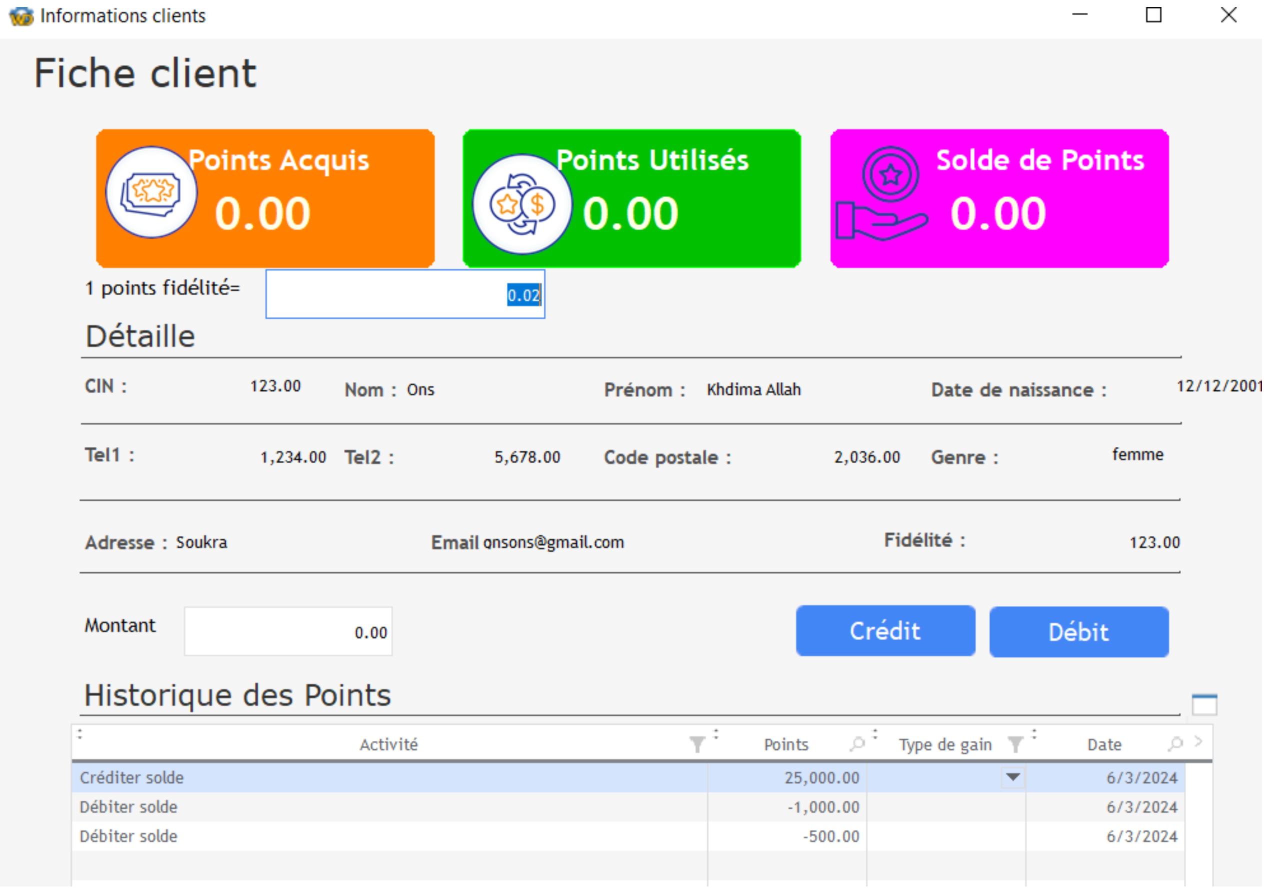
2. Global Class Diagram for Sprint 5



IX. Realisation

This section presents the results obtained through screenshots. The screenshots visually illustrate the final outcomes achieved during tests or simulations. Below are the screenshots for each interface related to this sprint:

Manage Loyalty Points and View Transaction History Interface:



Informations clients

Fiche client

Points Acquis 0.00

Points Utilisés 0.00

Solde de Points 0.00

1 points fidélité= 0.02

Détaille

CIN : 123.00 Nom : Ons Prénom : Khdima Allah Date de naissance : 12/12/2001

Tel1 : 1,234.00 Tel2 : 5,678.00 Code postale : 2,036.00 Genre : femme

Adresse : Soukra Email onsons@gmail.com Fidélité : 123.00

Montant 0.00

Historique des Points

Activité	Points	Type de gain	Date
Créditer solde	25,000.00		6/3/2024
Débiter solde	-1,000.00		6/3/2024
Débiter solde	-500.00		6/3/2024

Conclusion

In this chapter, we have explored the third release, focusing on the user stories and tasks for the fourth and fifth sprints. Detailed analysis and design elements, including global use case diagrams and sequence diagrams, have been presented. The chapter concludes with the realization phase, highlighting the implementation outcomes through screenshots. Moving forward, the general conclusion will provide a summary of the entire project and its achievements.

CONCLUSION GENERALE

Throughout the preparation of our final year project, we have tried to put into practice the knowledge acquired during our university studies with the aim of realizing a very interesting project. Thanks to this project, we have opened ourselves to new technologies and got acquainted with tools that were more or less unknown to us. We discovered professional life within a company.

Following the Scrum methodology allowed us to develop this application step by step, thus touching on all aspects of its creation. Specifying requirements and designing with UML diagrams helped us understand the importance of these tools in establishing a solid foundation for development. The application development phase allowed us to learn new technologies and domains .

Continuous work and team instructions helped us understand and grasp the business side and the project's lexicon. The opportunity to be supervised by experienced managers allowed us to evolve in terms of behavior, diligence, punctuality, and the way we express ourselves .

Additionally, this project gave us the opportunity to work with various programming languages. While we primarily developed in Windev, I also worked on backend development using **Node.js**, demonstrating my ability to develop with popular languages as well. Though the Node.js development was not included in the main system, this experience proves my versatility and my ability to adapt to modern technologies while maintaining efficiency and professional rigor.

BIBLIOGRAPHIE: BIBLIOGRAPHY

خلاصة

ندرج هذا التقرير ضمن مشروع التخرج للحصول على االجازة التطبيقية في تكنولوجيا المعلومات اختصاص تطوير النظم المعلوماتية

يعتمد العمل على إنشاء تطبيق شامل لإدارة نقاط الولاء. هذا التطبيق مصمم خصيصًا للشركات ذات الفروع المتعددة، مثل سلاسل المطاعم، للتفاعل مع العملاء. الهدف الرئيسي هو مركزية إدارة برامج الولاء، وتسهيل تفاعلات (POS) ويتضمن تطبيق ويب للمسؤولين ونظام نقطة بيع العملاء بسلاسة، وتقديم تحليلات وتقارير مفصلة

(POS) الكلمات المفتاحية: سكرم، ويندیف، ويب ديف، بوستجريس كيو إل، يو إم إل، قائمة المنتج المتراكمة، برنامج الولاء، نظام نقطة البيع

Résumé

Ce rapport **résume les travaux réalisés dans** le cadre du projet de fin d'études pour la licence du développement des systèmes informatiques.

Le travail repose sur la création d'une application complète de gestion des points de fidélité. Cette application est conçue pour les entreprises avec plusieurs branches, telles que les chaînes de restaurants, et comprend à la fois une application web pour les administrateurs et un système de point de vente (POS) pour les interactions avec les clients. L'objectif principal est de centraliser la gestion des programmes de fidélité, de faciliter les interactions avec les clients de manière fluide, et de fournir des analyses et des rapports détaillés.

Mots-clés : Scrum, WINDEV, WEBDEV, POSTGRESQL, UML, Backlog Produit, Programme de Fidélité, système de point de vente (POS

Summary

This report summarizes the work done as part of the graduation project for the Bachelor of Computer Systems Development.

The work is based on the creation of a comprehensive loyalty points management application. This application is tailored for businesses with multiple branches, such as restaurant chains, and includes both a web application for administrators and a point-of-sale (POS) system for customer interactions. The primary goal is to centralize loyalty program management, facilitate seamless customer interactions, and provide detailed analytics and reporting

Keywords : Scrum, WINDEV, WEBDEV, POSTGRESQL, UML , Product Backlog , Loyalty Program, a point-of-sale (POS) system