République Tunisienne
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université de Sousse

Institut Supérieur des Sciences
Appliquées et de Technologie
de Sousse

DEPARTEMENT INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ETUDES

En vue de l'obtention du:
Diplôme National d'Ingénieur Informatique
Option: Génie Logiciel-Architecture Logicielle

Conception et développement d'une Lightning application de gestion des utilisateurs

Elaboré par :

Aslene OUAZE

Soutenu le ...../**.....**/**.....** **devant le jury :**

Président :	**Nom & Prénom Président**	**ISSAT de Sousse**
Examinateur :	**Nom & Prénom Examinateur**	**ISSAT de Sousse**
Encadrant :	**Nom & Prénom Encadrant**	**ISSAT de Sousse**
Encadrant Industriel :	**Mme. Rim Ben Abdesslem** **M.Arous Mandhour**	**TECHLEAD**

Année Universitaire : 2022 / 2023

Code Sujet: FI-

Résumé

Ce travail a été développé dans le cadre d'un projet de stage de fin d'études qui a été réalisé au sein de la Société TECHLEAD. Ce projet consiste à concevoir et développer une lightning application dynamique pour le CRM Salesforce pour permettre aux administrateurs et aux directeurs de communautés de gérer leur communauté et ses utilisateurs.
Notre application donne également la possibilité de consulter l'historique de connexion des utilisateurs et un tableau de bord synthétique visualisant les KPI ainsi que la mise à disposition d'un Chatbot intelligent à l'administrateur.

Mots clés

Application Lightning, Gestionnaire de communauté, Chatbot, LWC, JS, CSS, Apex, Aura, SLDS, SOQL, SOSL, Architecture MVC

Summary

This work was developed as part of an end-of-study internship project which was achieved within the TECHLEAD company. This project involves designing and developing a dynamic lightning application for the Salesforce CRM to enable administrators and community managers to manage their community and its users.
Our application also gives the possibility of consulting users' connection history and a synthetic dashboard visualizing the KPIs as well as providing a smart Chatbot to the administrator.

Keywords
Lightning application, Community Management, Chatbot, LWC, JS, CSS, Apex, Aura, SLDS, SOQL, SOSL, MVC Architecture

Contents

List of Figures

List of Tables

General Introduction

In today's digital world, where businesses are increasingly relying on cloud-based services and software,such as the Salesforce Platform, efficient management of user accounts has become crucial. This area demonstrates multiple important investments thanks to the increasing number of interested developers in this platform who offer an infinity of useful applications.

These applications are accessible everywhere and through multiple devices as long as that device is connected to Salesforce, such availability is provided thanks to the robust web and mobile infrastructure of the Salesforce platform.

That's why we wanted to create a lightning solution for individuals and enterprises who want to manage and monitor their user's activity within Salesforce and we took community users as a base point. Our application will provide multiple information and statistics about each user as well as enable modifications to such information, it will also provide useful KPI charts and a smart chatbot solution for administrators and community managers.

Our end-of-study project entitled "Salesforce Community Management Lightning Application" concludes our summer training as a computer engineer.

The project was carried out over six months, within the company TECHLEAD. This report summarizes the stages of realization of this project. Its purpose is to situate the context of the project, to describe the resulting application, the methods, and tools used as well as the results obtained.

This report follows the following organization:

The first chapter is entitled "General Framework of the Project", which is an introductory

chapter presenting the host company, the problem, the solution proposed, and the objectives of the project, a study of the existing and the process of development of our application.

The second chapter, ”Specification of needs”, is used to identify the actors of our application and then to specify the functional and non-functional needs. functionalities to which our application must respond, making it possible to identify its main features.

The third chapter, ”Conception”, serves to describe the conceptual diagrams and the architecture applied to our proposed solution.

The fourth and final chapter, ”Realization”, illustrates the realization of our project through the presentation of the environment and the development tools as well as the visualization of the results of our work through the main application interfaces.

Finally, we end the report with a general conclusion in which we recapitulate the work carried out and we present the prospects.

Chapter 1

General framework of the project

Introduction

We begin this chapter with a general presentation of the organization of the reception. We will then detail the context and the problem of our project and the proposed solution, which will attempt to resolve the inconveniences that already exist. We will finish this chapter by describing the schedule of our internship through the Gantt chart as well as the development process.

1.1 *Presentation of the host organization*

TechLead is a Tunisian IT engineering company whose mission is to design and implement Salesforce solutions for companies to improve their productivity, profitability, and market adaptability.

The company supports its clients throughout the life cycle of their projects, from consulting to the complete implementation of the solution and up to the transfer of skills. The company's young, dynamic, and versatile team assists clients in all stages of implementing their Salesforce solution to better interact with customers, partners, and prospects. The company offers many services such as:

- **Accompaniment:** The company's Salesforce advisors help clients implement and develop your Salesforce solution. They can intervene in the audit and analysis of needs, the design, the integration of data, and the configuration of clients' projects quickly and efficiently.
- **Support:** The company's experienced developers can provide assistance and maintenance to clients' projects in the various administration or development needs with good availability and responsiveness.
- **Salesforce Training:** The company provides training tailored to client's needs and helps them use and leverage the capabilities of Salesforce.

Figure 1. *Host organization TECHLEAD*

1.2 *Project presentation*

In this part, we put our work in its general context. first, we present the context. Second, we present the problem and the reasons that led us to suggest this topic. Third, we present the solution proposed to solve the problem. Finally, we will describe the objectives of this project.

1.2.1 Context

Salesforce is a highly customizable advanced CRM, it stores customer data, gives processes to nurture prospective customers, and provides ways to collaborate with other workers. [**1**]
Salesforce comes with a lot of standard functionality, or out-of-the-box products and features

that clients can use to run their business. Here are some common things businesses want to do with Salesforce and the features Salesforce gives that support those activities:[**1**]

You need to:	So we give you:
Sell to prospects and customers	Leads and Opportunities to manage sales
Help customers after the sale	Cases and Communities for customer engagement
Work on the go	The customizable Salesforce mobile app
Collaborate with coworkers, partners, and customers	Slack, Chatter, and Communities to connect your company
Market to your audience	Marketing Cloud to manage your customer journeys

Figure 2. *Salesforce features, Source: [**1**]*

The platform also helps clients move fast. Part of that speed comes from replacing tasks that clients are used to doing by hand with more streamlined processes.[**2**]

The platform's goal is to make big changes with minimal effort and to solve mistakes that impact the buyer using dynamic expandable interfaces using additional extensions.

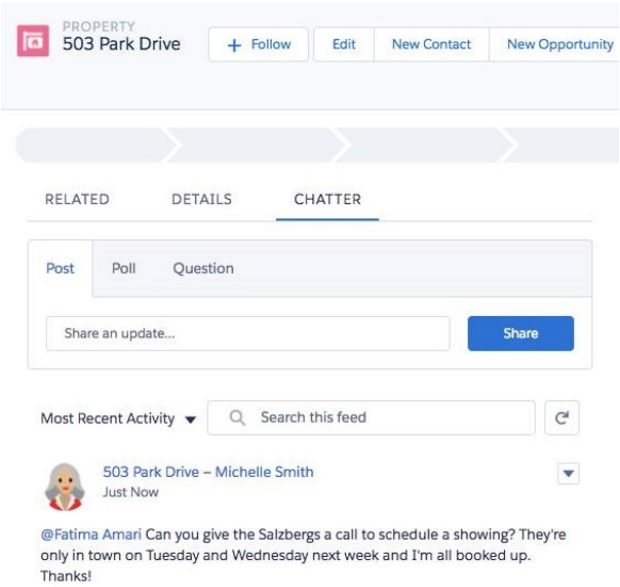

Figure 3. *Chatter extension in Salesforce, Source: [**2**]*

Here are a few use cases for different departments:

For employees who work in...	Customize the platform for...
Finance	• Budget management • Contract management • Pricing
Product	• Warranty management • Preproduction testing • Product ideas and innovation
Supply Chain	• Procurement • Vendor management • Logistics
Ops	• Asset and facilities management • Merger and acquisition enablement • Business agility

Figure 4. *Usecases for Salesforce, Source: [2]*

Salesforce is a cloud company. Everything to offer resides in the trusted, multitenant cloud.[**3**]

The Salesforce platform is the foundation of the services. It's powered by metadata and made up of different parts, like data services, artificial intelligence, and robust APIs for development. [**3**]

All the apps sit on top of the platform. The prebuilt offerings like Sales Cloud and Marketing Cloud, along with apps built using the platform, have consistent, powerful functionality.[**3**]

Everything is integrated. The platform technologies like predictive analytics and the development framework are built into everything to offer and everything to build.[**3**]

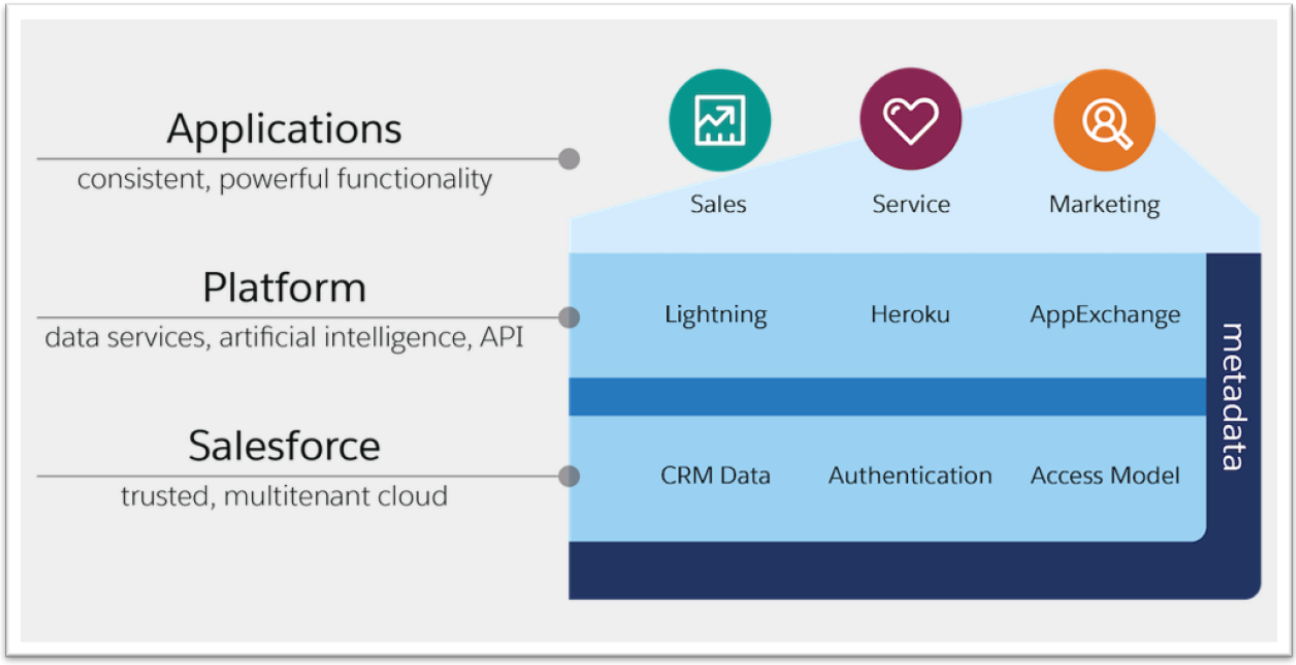

Figure 5. *Salesforce's architecture, Source: [**3**]*

1.2.2 Problem

In Salesforce, each user is identified by a unique username and profile. Along with other settings, the profile determines what tasks a user can perform, what data they can view, and how they can use the data.

As a Salesforce admin, you manage users in your organization. In addition to creating and assigning users, user management includes managing permissions and licenses, delegating users, and more

Users are managed in the community through a Salesforce interface and over several stages which makes it difficult and time-consuming.

1.2.3 Proposed solution

The main objective of our work is to design and develop a powerful tool to facilitate the task of managing the users of the community.

This application is intended to offer any organization a simple and effective means to manage the

”administration” part, then the connection history part, and finally provide a synthetic dashboard visualizing the KPIs as well as a smart Chatbot solution for the administrators and community managers.

1.2.4 Objectives

Ensure user satisfaction by ensuring:

Functional objectives:

1. **Choice of the community:** Select the community to which we will manage the users
2. **Manage users:** The system must allow users to be managed with the functionalities of activation, deactivation, modification, and consultation of the list of users via a data table. Creation of different filters that allow us to facilitate the navigation of the list of users.
3. **Manage connection history:** Create a chart that shows the number of user connections per day, week, year, or according to a well-determined date. This allows the administrator to modify the user's license
4. **Offer a synthetic dashboard**
5. **Offer a smart Chatbot solution**

Non-functional objectives:

1. **Security:** Access to information is only possible after verification of privileges and access rights, for example Authentication, Redirections.
2. **Ergonomics and user-friendliness:** The application will provide a user-friendly and easy-to-use interface that does not require any prerequisites, so it can be used by all types of users (even non-computer specialists).
3. **Extensibility and maintainability:** The architecture of the application will allow the evolution and maintenance (addition or deletion or update) at the level of its various modules in a flexible manner.
4. **Performance:** The application must be efficient, i.e. the system must react within a period that does not exceed 5 seconds, whatever the action of the application.
5. **Availability:** The application will be available on 24/24, and 7/7 except during the maintenance period.

Technical objectives:

1. Organization of the application according to the MVC architecture: a software architecture model that separates the representation of information from the user's interaction with it.

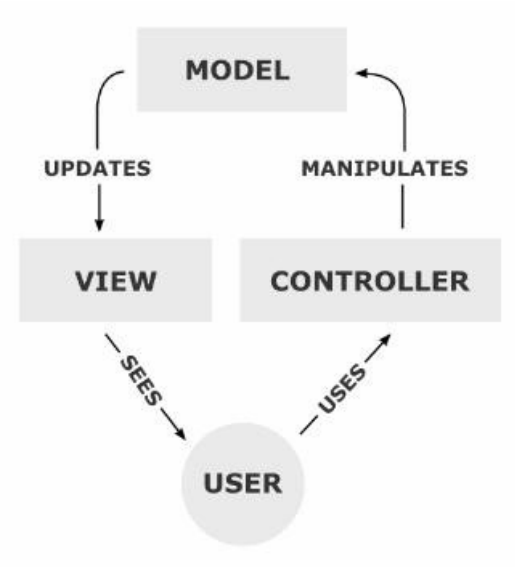

Figure 6. *MVC Architecture*

2. Using the Framework "LWC".
3. Using the programming language "Apex".
4. Using the library "SLDS".
5. Using the Salesforce Object Query Language "SOQL/SOSL".

1.3 *Study of the existing*

In this part, we analyze and criticize the existing applications currently, through the following table propose a solution that solves their drawbacks.

Application	**Pros**	**Cons**
HubSpot	- User-Friendly Interface - Ability to create custom roles and assign specific permissions to users based on their roles. - Ability tracks user activity, allowing businesses to monitor user behavior.	- Interface can be complex, particularly for businesses with a large number of users. - Pricing structure is based on the number of contacts in a business's database, which can make it more expensive for businesses with larger teams. - There may be a learning curve when it comes to managing users and configuring access control.
ZenDesk	- Provides user analytics, allowing businesses to monitor user behavior. - Allows businesses to set up custom notifications for user actions, such as ticket creation or update. - Provides tools for user collaboration, such as shared views and comments.	- May not offer as much granularity as some businesses require. - Interface may not offer as much customization as some businesses require. - Pricing structure is based on the number of agents, which can make it more expensive for businesses with larger teams.
Conga	- Offers customizable workflows. - Offers Centralized user management system. - Provides tools for user collaboration, such as document sharing and commenting	- Limited third-party integrations - Pricing structure can be more expensive for businesses with larger teams. - User management interface can be complex.

Table 1. *Study of the existing*

1.4 *Development process*

A software development process is a set of related activities followed by a team led to the production of the software within the organization. It consists of a detailed plan describing how to develop, design, test, deploy, and maintain the product. [**4**]

1.4.1 Incremental development

In this context, we adopt the process of incremental development as an approach to the realization of our project. According to this process, the customer's needs are specialized, the software is globally designed, then the realization is done by an increment of functionalities.[**4**] Each increment is considered an executable part of the final system. These increments are successively integrated into the final product and at each stage, the software is tested, operated, and maintained as a whole.[**4**]

Implementing the software by increment makes it possible to take into account the risk analysis to facilitate the detection of errors at the earliest according to customer feedback and to reduce time and cost of production, which helps in the realization of software quality. [**4**]

1.4.2 Provisional schedule of tasks

A Gantt chart is a graphical tool that represents the management of the project over time, which facilitates its implementation.

Indeed, the internship within TECHLEAD will run for a period of 4 months. The following timeline illustrates a provisional schedule set early in development, representing the main stages leading to a functional solution that meets the criteria defined by previously mentioned specifications.

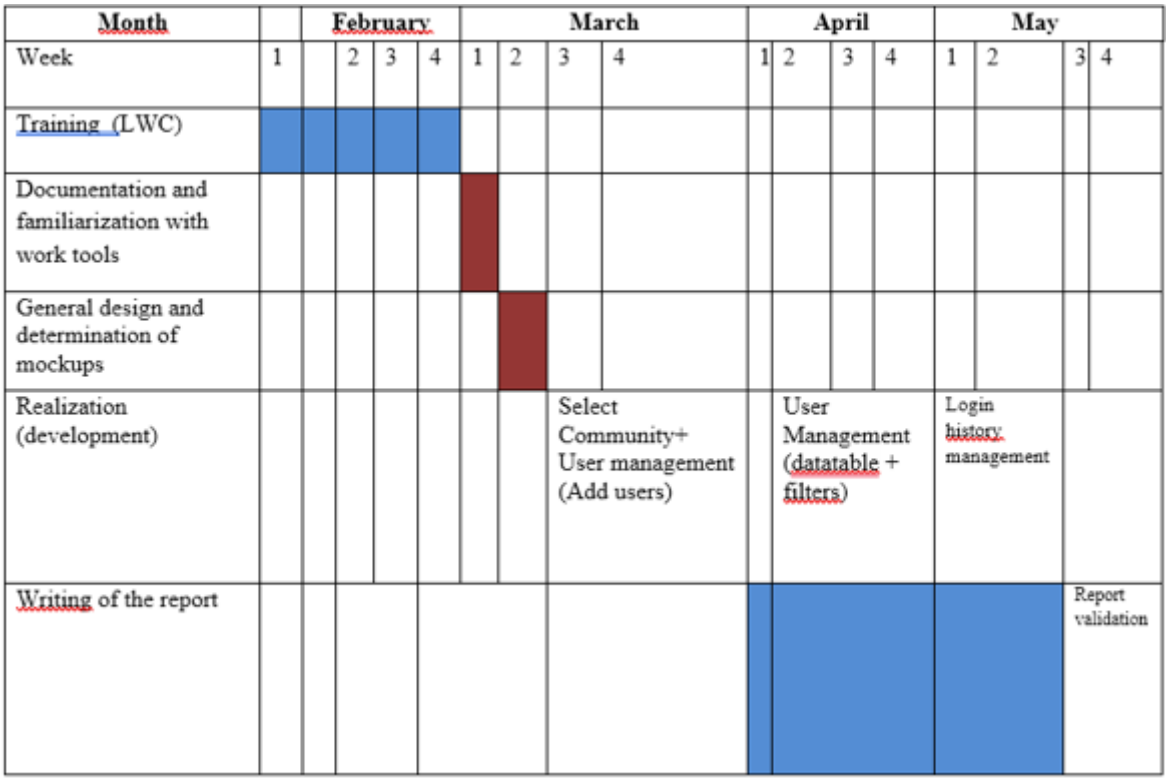

Figure 7. *Gantt diagram*

Conclusion

In this chapter, we have introduced the context of our project by representing the host organization in the first place. Secondly, we have introduced the sales platform and its features. Thirdly, we have cleared up the problem. Then, we described the proposed solution and the objectives to be achieved. After that, we analyzed the existing applications. Lastly, we have depicted the advancement of activities throughout the project according to the adopted development process. In the next chapter, we will specify the functional requirements and the non-functional needs.

Chapter 2

Specification of needs

Introduction

In this chapter, we will identify the actors, then we will specify the functional and non-functional needs that the proposed solution must meet. Finally, we present the use case diagrams explaining our application's main functionalities.

2.1 *Identification of actors*

An actor represents an external entity that interacts directly with the system. It can be either a human person or a system. We distinguish two types of actors, the main actor, and a secondary actor. Indeed, a principal actor obtains an observable result of the system while a secondary actor is asked for additional information.

Main actors

Community Manager

The community manager is the first main user of our application who should be able to :

- Choose the community that he has the right to access and manage.

- Consult users of a specific community inside a well-organized table with pagination options.
- Filter said users by name, user-name, Salesforce account name, status(active or not active), and Salesforce profile.
- Consult and update the details of each user.
- Activate users within a specific community.
- Deactivate users within a specific community.
- Send a ”Welcome to community” email to a specific active user.
- Send a ”Reset password” email to a specific active user.
- Add one or multiple users at once to a specific community.

In case when the community manager has a Salesforce role he will be also able to :

- Add one or multiple users at once to a specific community.

Salesforce Administrator

The Salesforce administrator is the second main user of our application who should be able to :

- Perform all the actions, mentioned above, that the community manager can.
- Consult a bar chart that shows the number of logins of each user within the selected community.
- Filter chart results by the period between two specific dates.
- Consult details about each user displayed in the chart.
- Update the Salesforce user license for each user displayed in the chart.
- Consult users failed login attempts to a specific community inside a well-organized table with pagination options.

- Filter said login attempts by name, user-name, status(Invalid password, No community access, etc...) and event date and time.
- Consult detailed information about each login attempt.
- Send a security warning email to the account owner about the login attempt event.
- Access a Chatbot that provides information about the selected community or the Salesforce organization.

Secondary actors

Salesforce System

This actor is the system, previously developed and deployed in a cloud server by the Salesforce organization, interactable through our application, and it's responsible for :

- Sending an automatic "Welcome to community" email upon adding a new member to the community by our application user.
- Sending an automatic "Welcome to community" email upon activating a previously deactivated user by our application user.
- Generating reset password URL upon sending a "Reset password" email to a community member by our application user.
- Tracking users' successful and failed login attempts and saving them to the organization database.

2.2 *Functional Needs*

Functional needs are expressed by the user of the application which makes it possible to identify the functionalities of this application.
In our case, the functional needs are:

- Choice of the community:

  Select the community to which we will manage the users.

- Manage users:

  The system must allow users to be managed with the functionalities of activation, deactivation, modification, and consultation of the list of users via a data table. Creation of different filters that allow us to facilitate the navigation of the list of users.

- Manage connection history:

  Create a chart that shows the number of user connections per day, week, year, or according to a well-determined date. This allows the administrator to modify the user's license.

- Offer a synthetic dashboard:

  Allowing the system administrator / the community manager to visualize the KPIs (Key Performance Indicators) of his organization / community.

2.3 *Non-functional Needs*

Non-functional requirements represent the characteristics of the system. They relate to the constraints to be taken into consideration to set up an adequate solution.
For our application, the non-functional requirements are:

- Security:

  Access to information is only possible after verification of privileges and access rights, for example Authentication, Redirections.

- Ergonomics and user-friendliness:

  The application will provide a user-friendly and easy-to-use interface that does not require any prerequisites, so it can be used by all types of users (even non-computer specialists).

- Extensibility and maintainability:

  The architecture of the application will allow the evolution and maintenance (addition or deletion or update) at the level of its various modules in a flexible manner.

- Performance:

  The application must be efficient, i.e. the system must react within a period that does not exceed 5 seconds, whatever the action of the application.

- Availability:

  The application will be available on 24/24, and 7/7 except during the maintenance period.

2.4 *Use case diagrams*

In this section, we will highlight the system's functionalities to be from the functional needs mentioned above based on the UML(Unified Modeling Language) diagrams which group together all the use cases of the system.

2.4.1 Global use case diagram

The following figure illustrates the global use case diagram of our application

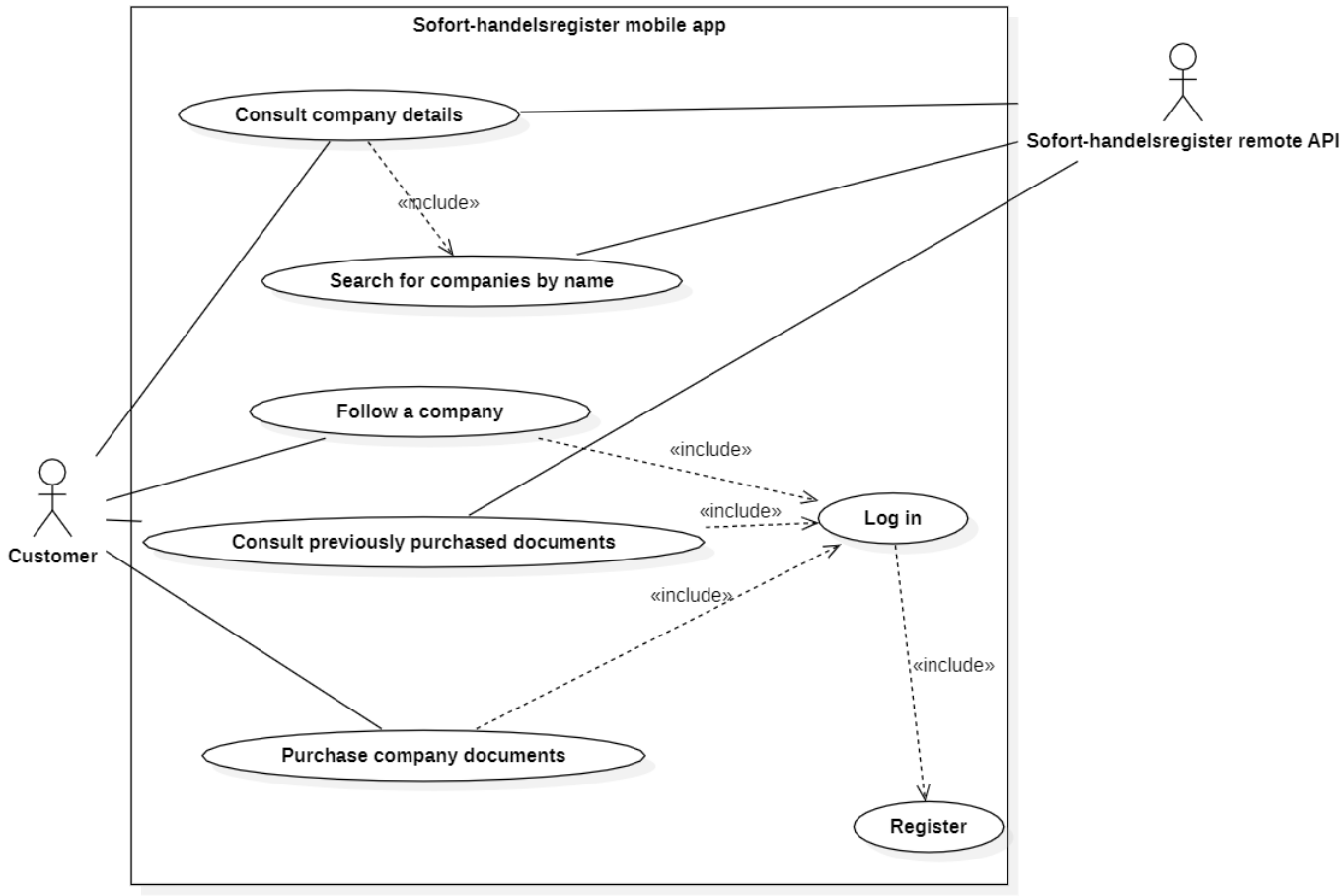

Figure 8. *Global use case diagram*

2.4.2 Use case refinement

In this section, we will detail the main use cases.

2.4.2.1 Register use case refinement

In our application, a customer must register before benefiting from the service of purchasing a document through our application.

The following figure shows the Register use case diagram.

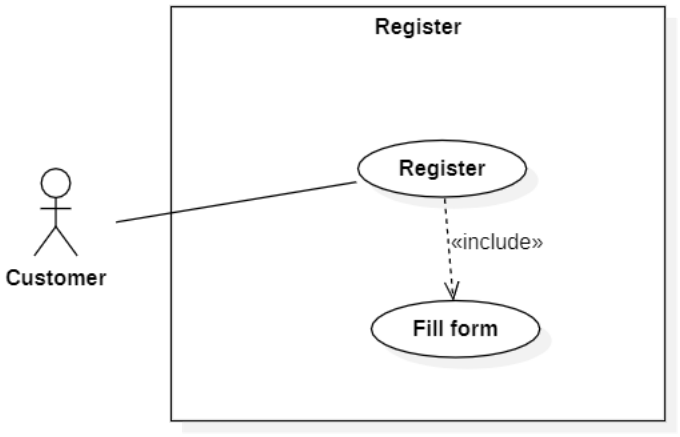

Figure 9. *Register use case diagram*

The following table details the tasks to be performed by the customer to register to our app.

Summary	
Title	Register
Objectif	Allowing the user to register to be able to purchase documents and follow companies through the application
Actors	Customer
Description of sequences	
Pre-condition	User should start the application and go to the registration interface.
Post-condition	The user is registered and his contact details are saved in the database.
Normal scenario	1. The user accesses the registration interface. 2. The user completes the form and complies. 3. Registration is completed.
Alternative scenario	1. Empty fields: The system sends an error message: You must complete all fields 2. The email or password entered is not valid: The system displays an error message describing the validation conditions for these fields. 3. The email is already in use: An alert is sent by the system
Non functional constraints	1. The interface must be ergonomic. 2. Error messages should be understandable and clear.

Table 2. *Register use case*

2.4.2.2 Login use case refinement

In our application, a customer must log in before benefiting from the service of purchasing a document through our application and tracking a specific company through notifications.
The following figure shows the Login use case diagram.

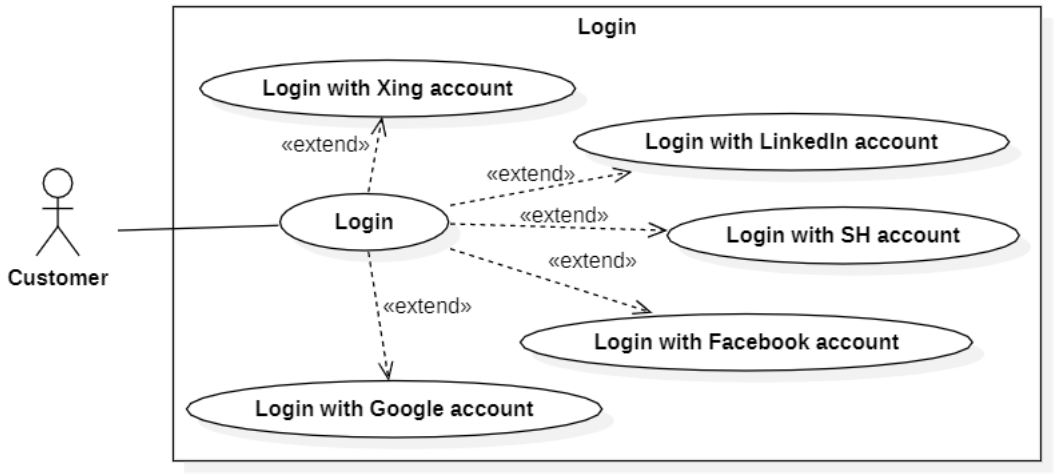

Figure 10. *Login use case diagram*

The following table details the tasks to be performed by the customer to log in to our app.

Summary	
Title	Login
Objectif	Allowing the user to login to be able to purchase documents and follow companies through the application
Actors	Customer
Description of sequences	
Pre-condition	User should start the application and go to the log in interface.
Post-condition	The user is logged in and his session is cached in the application.
Normal scenario	1. The user accesses the login interface. 2. The user completes the form and complies or he clicks on one of the social media buttons and completes the external login process. 3. Login is completed. 4. The login token is cached within the application.
Alternative scenario	1. Empty fields: The system sends an error message: You must complete all fields 2. The email or password entered is not valid: The system displays an error message describing the validation conditions for these fields.
Non functional constraints	1. The interface must be ergonomic. 2. Error messages should be understandable and clear.

Table 3. *Login use case*

2.4.2.3 Company search use case refinement

The following figure illustrates the use case diagram ”Company Search”

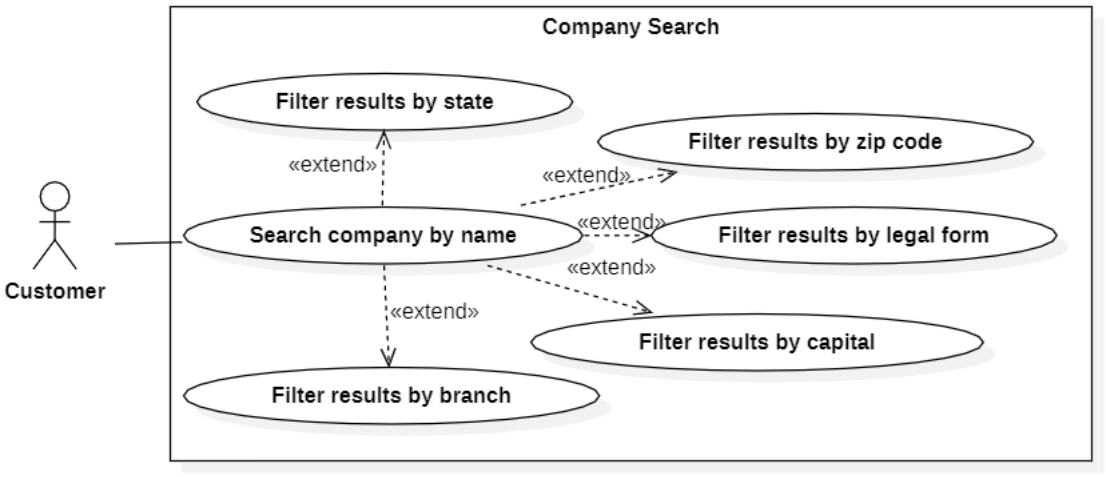

Figure 11. *Company Search use case diagram*

The following table details the tasks to be performed by the customer to search for companies within our app.

Summary	
Title	Company Search
Objectif	Allowing the user to search for specific companies through the application
Actors	Customer
Description of sequences	
Pre-condition	User should start the application and go to the search company interface.
Post-condition	The user chooses a company from the list of results and consults details about it.
Normal scenario	1. The user accesses the company search interface. 2. The user types the company name or the first letters of it. 3. The user may choose a specific state, a legal form, or a branch to filter the results. 4. The user may also type zip code or capital as a filter. 5. The user consults details about companies from the list of results.
Alternative scenario	1. No results: The system sends a message: No results were found with the given parameters 2. No internet: The system sends an error message: No internet connection found.
Non functional constraints	1. The interface must be ergonomic. 2. Error messages should be understandable and clear.

Table 4. *Company Search*

2.4.2.4 Follow Company use case refinement

The following figure illustrates the use case diagram ”Follow Company”

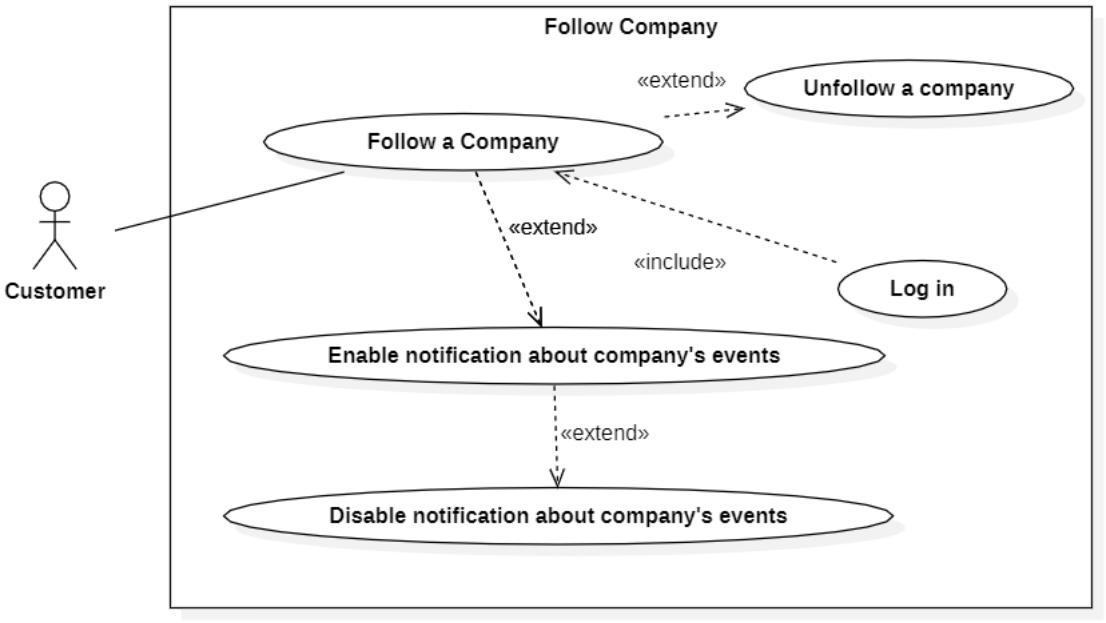

Figure 12. *Follow Company use case diagram*

The following table details the tasks to be performed by the customer to follow a company within our app.

Summary	
Title	Follow Company
Objectif	Allowing the user to follow specific companies to receive notifications about its events through the application
Actors	Customer
Description of sequences	
Pre-condition	User should start the application, be logged in, search for a specific company and access the consult company details interface.
Post-condition	The tracked company is added to the user's follow list.
Normal scenario	1. The user accesses the consult company interface. 2. The user taps the follow button. 3. The user may choose to unfollow the company or enable event notifications. 4. Upon activating notifications, the user will receive notifications through the application when a notice is published about said company. 5. The user can tap the notification to access the notice.
Alternative scenario	1. Not logged in: The system sends an error message: You must be logged in to follow a company.
Non functional constraints	1. The notification must be as close to real-time as possible

Table 5. *Follow Company use case*

2.4.2.5 Purchase Documents use case refinement

The following figure illustrates the use case diagram ”Purchase Documents”

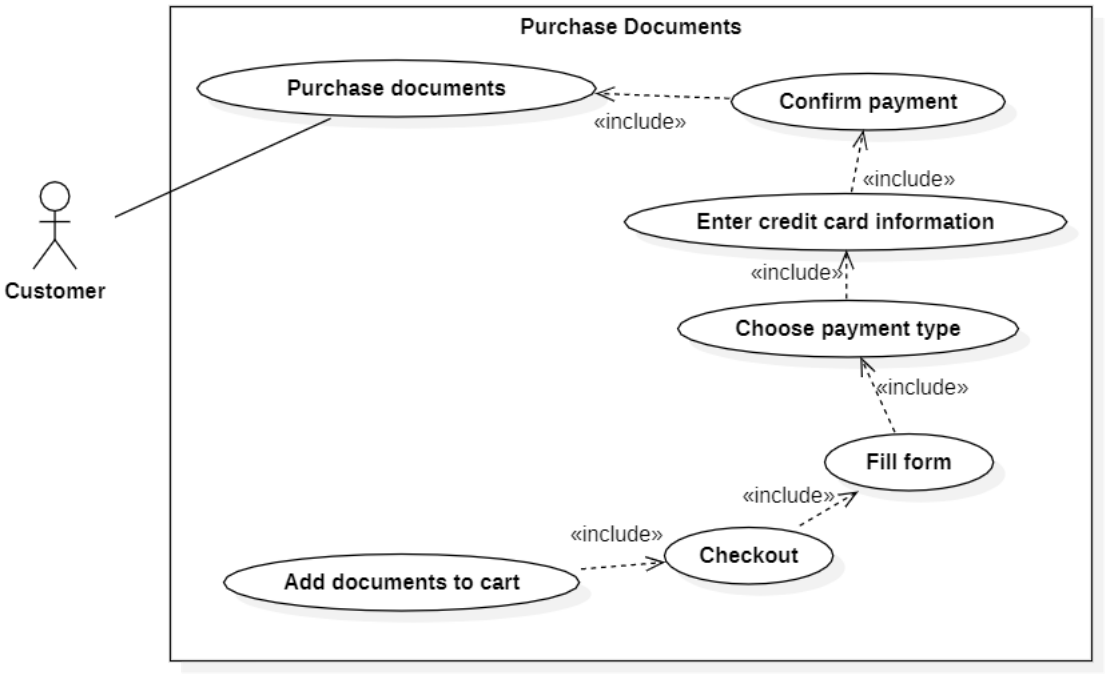

Figure 13. *Purchase Documents use case diagram*

The following table details the tasks to be performed by the customer to purchase documents within our app.

Summary	
Title	Purchase Documents
Objectif	Allowing the user to purchase official documents about the consulted company, if they are available.
Actors	Customer
Description of sequences	
Pre-condition	User should start the application, be logged in, search for a specific company, access the consult company details interface and scroll to the documents section.
Post-condition	The purchased documents are added to their user's purchased documents interface where he can download them on demand.
Normal scenario	1. The user accesses the consult company interface. 2. The user taps the wanted documents. 3. The documents will be added to the shopping cart 4. Upon checkout, the user will be redirected to the payment interface 5. The user fills in the contact information form 6. The user chooses the payment method 7. The user enters credit card details 8. The user confirms the payment 9. Purchase is completed
Alternative scenario	1. Not logged in: The system sends an error message: You must be logged in to purchase documents.
Non functional constraints	1. The interface must be ergonomic. 2. Error messages should be understandable and clear. 3. The payment must be as secure as possible

Table 6. *Purchase Documents use case*

Conclusion

In this chapter, we have described the specification phases of the needs of our developed application to identify the different actors as well as the features and services that our application must provide.

We have detailed these features with use case diagrams. The next chapter will be devoted to the design phase.

Bibliography

[1] Get Started with the Salesforce Platform. [trailhead.salesforce.com].

[2] Discover Use Cases for the Platform. [trailhead.salesforce.com].

[3] Understand the Salesforce Architecture. [trailhead.salesforce.com].

[4] Oumayma LAZREG : Rapport PFE, Conception et développement d'une application mobile intelligente de e _ consulation et d'estimation des maladies. (ISSAT de Sousse), 2019.