

Acknowledgements

First and foremost, I would like to express my sincere gratitude to all people without whom this work would not see the light. Special thanks go to my company supervisors Mr. Hamza Bejaoui and Mr. Skander Amor who offered me the opportunity to work on this project and discover new technologies. Their guidance, continuous feedback and involvement pushed me towards the fulfilment of the project's objectives. I would also like to thank WanToTrip's team for their help during the duration of this capstone project. This work would not have been possible without the support of my academic supervisor Mrs. Hanen Yakoubi. Her dedication, recommendations and continuous follow-up on my work contributed to the improvement of my project and analytical skills. Last but not least, I would like to express my gratitude towards my family, friends and Mariem Beldi alumna of the Higher Institute for Technological Studies of Zaghouan for supporting me during this challenging journey. Thank you!

Contents

Acknowledgements.....	1
Contents.....	2
List of Figures	6
List of Tables	8
General introduction.....	1
Chapter 1 : Training context	2
Introduction	2
1.Presentation of the company	2
1.1.Sector of activities.....	3
1.2.Company hierarchy.....	3
2.Problematic and proposed solution	3
2.1 Analysis of the current state	3
2.1.1 TutterflyCRM.....	4
2.1.2 Dolphin Dynamics	4
2.1.3 Kapture.....	4
2.2 Criticizing the current state.....	5
2.3 Proposed solution.....	5
3.Choice of the methodology	6
3.1 Methodology adopted: Agile scrum.....	6
3.1.1 Scrum methodology.....	6
Conclusion.....	7
Chapter 2 : Project Planning	8
Introduction	8
1.Requirement's specification	8
1.1. Functional requirements specification	8
1.2. Non-functional requirements specification.....	9
2.Actor identification	10
2.1 Product backlog	10
3.Global use case	11
4.Management of the project with Scrum	12
4.1. The scrum team	12

4.2. Sprint planning.....	13
5.General class diagram	13
6.The physical architecture.....	15
7.The application architecture.....	16
7.1. Node JS.....	16
7.2. Vue JS application	17
7.2.1 Composition of Vue Js.....	17
8.The software architecture	17
9.The technological options.....	18
10.The software environment	20
11.The hardware environment.....	21
Conclusion.....	21
Chapter 3 : Study and execution of Sprint 1	22
Introduction	22
1.Sprint Backlog	22
2.Analysis	22
2.1 Refine Use case diagrams	23
2.1.1 Use case « Password update ».....	23
2.1.2 Use case « Password reset ».....	24
2.1.3 Use case « Authentication »	24
3.Conception.....	25
3.1Sequence diagrams.....	25
3.1.1 Sequence diagram “Password update”	25
3.1.2 Sequence diagram “Reset password”.....	26
3.1.3 Sequence diagram “Authentication”	26
4.Realization and testing.....	27
4.1Testing execution and results	27
4.1.1 Testing Authentication.....	27
4.1.2 Testing Reset Password	28
4.1.3 Testing password update	30
Conclusion.....	31
Chapter 4 : Study and execution of Sprint 2	32
Introduction	32

1.Sprint Backlog	32
2.Analysis	32
2.1Use case diagrams.....	33
2.1.1 Use case « Activity management ».....	33
2.1.2 Use case « Role management ».....	36
2.1.3 Use case « User management ».....	38
4.Conception.....	40
4.1 Sequence diagrams	40
4.1.1 Sequence diagram “Add activity”	40
4.1.2 Sequence diagram “Modify activity”	41
4.1.3 Sequence diagram “Delete activity”	42
4.1.4 Sequence diagram “Add role”.....	42
4.1.5 Sequence diagram “Modify role”	43
4.1.6 Sequence diagram “Delete role”	43
4.1.7 Sequence diagram “Add user”	44
4.1.8 Sequence diagram “Modify user”	44
4.1.9 Sequence diagram “Delete user”	45
5.Realization and testing.....	45
5.1Testing execution and results	45
5.1.1 Testing Activity management	45
5.1.2 Testing Role management	47
5.1.3 Testing User management	49
Conclusion.....	50
Chapter 5 : Study and execution of Sprint 3	51
Introduction	51
1.Sprint Backlog	51
2.Analysis	51
2.1Use case diagrams.....	52
2.1.1 Use case « Contact management ».....	52
2.1.2 Use case « Deals management »	55
2.1.3 Use case « Profile management ».....	57
4.Conception.....	59
4.1 Sequence diagrams	59

4.1.1 Sequence diagram “Add Contact”	59
4.1.2 Sequence diagram “Modify Contact”	60
4.1.3 Sequence diagram “Delete Contact”	61
4.1.4 Sequence diagram “Add deal”	61
4.1.5 Sequence diagram “Modify deal”	62
4.1.6 Sequence diagram “Delete deal”	62
4.1.7 Sequence diagram “Update information”	63
4.1.8 Sequence diagram “Change password”	63
5.Realization and testing.....	64
5.1Testing execution and results	64
5.1.1 Testing Contact management.....	64
5.1.2 Testing Deals management.....	65
5.1.3 Testing Profile management.....	67
5.1.4 Testing View dashboard.....	68
5.1.5 Testing Calendar	68
Conclusion.....	69
Chapter 6 : Study and execution of Sprint 4	70
Introduction	70
1.Sprint backlog	70
2.Realization and testing.....	71
2.1 Testing execution and results	71
2.1.1 Testing the authentication interfaces.....	71
2.1.2 Testing the booking process interfaces	73
2.1.3 Testing the trips and events interface	76
2.1.4 Testing the detail trip interface	78
Conclusion.....	80
General conclusion and perspectives	81
Bibliography	82

List of Figures

Figure 1: Company hierarchy	3
Figure 2: Scrum methodology	6
Figure 3: Global use case.....	12
Figure 4:Sprint planning.....	13
Figure 5: General class diagram	14
Figure 7: MongoDB database Model	15
Figure 8: Deployment diagram	15
Figure 9: Node Js backend anatomy	16
Figure 10: The software architecture	17
Figure 11: Use case diagram Sprint1.....	23
Figure 12: Sequence diagram "Password update"	26
Figure 13: Sequence diagram "Reset password".....	26
Figure 14: Sequence diagram "Authentication"	27
Figure 15: Authentication error – data not valid	28
Figure 16: Authentication error – email is invalid	28
Figure 17: The interface "Forgot password"	29
Figure 18: Email with the reset link	29
Figure 19: Reset password interface.....	30
Figure 20: Update password interface.....	30
Figure 21: Use case diagram Sprint2.....	33
Figure 22: Use case "Activity management".....	33
Figure 23: Use case description "Role management"	36
Figure 24: Use case "User management"	38
Figure 25: Use case description "Delete user"	40
Figure 26: Sequence diagram "Add activity"	41
Figure 27: Sequence diagram "Modify activity"	41
Figure 28: Sequence diagram "Delete activity"	42
Figure 29: Sequence diagram "Add role"	43
Figure 30: Sequence diagram "Modify role"	43
Figure 31: Sequence diagram "Delete role"	43
Figure 32: Sequence diagram "Add user"	44
Figure 33: Sequence diagram "Modify user"	45
Figure 34: Sequence diagram "Delete user"	45
Figure 35: User interface "Activity management"	46
Figure 36: Test "Add activity"	46
Figure 37: Test add activity successfully	47
Figure 38: User interface "Role management"	47
Figure 39: Test "Add role"	48
Figure 40: Test role added successfully	48
Figure 41: User interface "User management".....	49
Figure 42: Test "Add user"	49

Figure 43: Test user added successfully.....	50
Figure 44: Use case diagram Sprint3.....	52
Figure 45: Use case "Contact management"	52
Figure 46: Use case description "Deal management"	55
Figure 47: Use case "Profile management"	57
Figure 48: Sequence diagram "Add contact"	60
Figure 49: Sequence diagram "Modify contact"	60
Figure 50: Sequence diagram "Delete contact"	61
Figure 51: Sequence diagram "Add deal"	62
Figure 52: Sequence diagram "Modify deal"	62
Figure 53: Sequence diagram "Delete deal"	63
Figure 54: Sequence diagram "Update information"	63
Figure 55: Sequence diagram "Change password"	63
Figure 56: User Interface "Contact management"	64
Figure 57: Test "Add contact"	64
Figure 58: Test add contact successfully.....	65
Figure 59: User interface "Deals management"	65
Figure 60: User interface "Deals won"	66
Figure 61: Test modify deal successfully.....	66
Figure 62: User interface "Profile management"	67
Figure 63: User interface "Edit profile".....	67
Figure 64: User interface "Dashboard"	68
Figure 65: User interface "Calendar"	68
Figure 66: The login interface before the improvements.....	71
Figure 67: The sign-up interface before the improvements.....	72
Figure 68: The login interface after the improvements.....	72
Figure 69: The sign-up interface after the improvements.....	73
Figure 70: The booking interface 1 before the improvements.....	74
Figure 71: The booking interface 2 before the improvements.....	74
Figure 72: The first booking interface after the improvements	75
Figure 73: The booking interfaces after the improvements	75
Figure 74: The previous events interface before the improvements	76
Figure 75: The trips interface before the improvements	76
Figure 76: The upcoming trips and events interface 1 after the improvements.....	77
Figure 77: The upcoming trips and events interface 2 after the improvements.....	78
Figure 78: The trips details before the improvements	79
Figure 79: The trips details after the improvements	79

List of Tables

Table 1: Presentation of the company.....	2
Table 2: TutterflyCRM	4
Table 3: Dolphin Dynamics.....	4
Table 4: Kapture.....	4
Table 5: Product backlog.....	10
Table 6: The scrum team.....	12
Table 7: The hardware environment	21
Table 8: Sprint1 Backlog.....	22
Table 9: Use case description "Password update"	23
Table 10: Use case description "Password reset"	24
Table 11: Use case "Authentication"	24
Table 12: Sprint2 Backlog.....	32
Table 13: Use case description " Add activity"	34
Table 14: Modify activity.....	34
Table 15: Use case description "Delete activity"	35
Table 16: Use case description "Add role"	36
Table 17: Use case description "Modify role"	37
Table 18: Use case description "Delete role"	37
Table 19: Use case description "Add user"	38
Table 20: Use case description "Modify user"	39
Table 21: Sprint3 Backlog.....	51
Table 22: Use case description " Add contact"	53
Table 23: Modify activity.....	53
Table 24: Use case description "Delete activity"	54
Table 25: Use case description "Add deal"	55
Table 26: Use case description "Modify deal"	56
Table 27: Use case description "Delete deal"	56
Table 28: Use case description "Update information"	58
Table 29: Use case description "Password change"	58
Table 30: Sprint 4 Backlog.....	70

General introduction

According to the most recent startup act annual report, between 2019 and 2020 over 416 startups applied for the startup act and 248 received the label. These figures indicate the increase in the number of startups in Tunisia and the growth of the startup ecosystem. However, startups today face two main problems, one is the lack of organization and the other is the budget, especially for early-stage startups.

The lack of organization weakens the collaborative effort required to have efficient employees, leading to a total collapse of startup productivity. Therefore, having a customer relationship management system can help improve team efficiency, improve information organization, and improve customer service.

As part of my end of study internship, the objective is to develop a web application to manage the relationship with the client of the startup WanToTrip. This report describes the different steps taken to create the functionalities and complete the web application.

This report will be divided into five chapters. The first one entitled “Training context” which is specifically dedicated to present the hosting organization, the project managed and the methodology adopted. The second chapter entitled “Project planning” is mainly devoted to present the requirements specification, actor identification, the overall conception of the project and the technologies used to develop this project. The last three chapters represent Sprint 1, 2, and 3 respectively, and specifically present the Sprint implementation. Each sprint will contain 4 main parts; sprint backlog, the analysis in which we will show the different use case diagrams, the conception to identify the sequence diagrams and the realization part. Finally, we will conclude this report with a general conclusion.

Chapter 1 : Training context

Introduction

This chapter is dedicated to introducing the hosting company and the issues I resolved during the graduation defense. We will first introduce the hosting company, we will analyze the current state, we will propose the solution, and then we will introduce the plan to choose the agile method that will be used in our project.

1. Presentation of the company

Wantotrip is a newborn Tunisian startup legally founded on September 19th, 2019, their office is in Rue de l'Argentine, Tunis. Wantotrip is a platform that aims to create a new way of traveling by offering affordable and unique travel experiences. It allows everyone to travel with a group of likeminded people and share memorable experience with influencers and famous bloggers.

Table 1: Presentation of the company

• Company name	Wanto trip (Wantotrip)
• Logo	
• Date of creation	19/09/2019
• Legal form	IT activity
• Sector of activity	Community travelling
• Valuation	1.7 Million TND
• Website	http://www.wantotrip.net
• Email	i@wantotrip.net

1.1. Sector of activities

The principal activity for Wantotrip is community travelling, by that we mean Wantotrip organize for bloggers & influencers that we prefer to call them community leaders customized trips according to their main activity on social media with their community of followers. So for instance, if an influencer main activity is yoga and health-wise the planning of the trip gets updated, in other words, we break the digital gap between the community leader and his community to go on amazing trips altogether and share the same activities and passion.

1.2. Company hierarchy

The following picture illustrate the current company hierarchy.

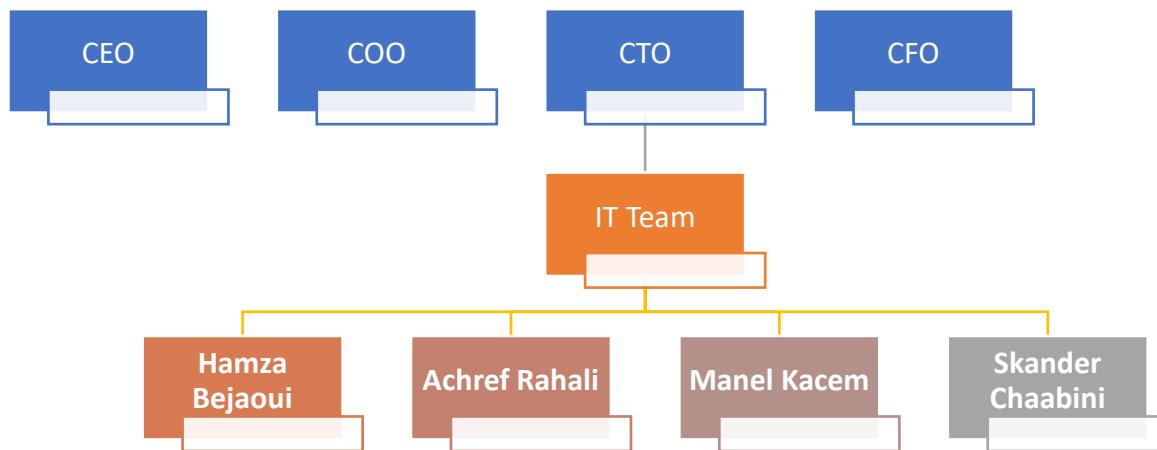


Figure 1: Company hierarchy

2. Problematic and proposed solution

This section is devoted to present both the existing problem and the solution proposed.

2.1 Analysis of the current state

There are plenty of customer relationship management platforms that exist nowadays which will allow us to identify the problems that exist in them and their benefits. In this part, we present some

internationally known platforms then we compare the different services offered by them. We have chosen to present, in what follows, the best known and most used between these applications:

2.1.1 TutterflyCRM

TutterflyCRM is an industry-leading travel sales solution which offers company-wide customer management processes. It also integrates sales processing, management, and oversight modules to provide comprehensive software for running business operations. [1]

Table 2: TutterflyCRM

Pros	Cons
<ul style="list-style-type: none"> ▪ Fully customizable sales, leads, and opportunities reporting. ▪ Completely cloud-based 	<ul style="list-style-type: none"> ▪ Lack of activity managements. ▪ Lack of contact database.

2.1.2 Dolphin Dynamics

Dolphin Dynamics is a cloud-based travel agency management solution that helps search for travel packages, track sales, and business patterns. [2]

Table 3: Dolphin Dynamics

Pros	Cons
<ul style="list-style-type: none"> ▪ Offers various products for both business and leisure travel agencies... ▪ Let you send out branded, personalized invoices and statements. 	<ul style="list-style-type: none"> ▪ The interface is not user friendly. ▪ Does not support desktop nor mobile version.

2.1.3 Kapture

Kapture CRM is a cloud customer support automation platform. Kapture is highly scalable and configurable to meet the needs of growing customer-facing enterprises. [3]

Table 4: Kapture

Pros	Cons
<ul style="list-style-type: none"> ▪ Provides preset email templates for itineraries, bookings, and packages. ▪ Enables you to securely save customer documentation (such as passports) for future bookings 	<ul style="list-style-type: none"> ▪ Expensive. ▪ Lack of activity management and opportunity management.

2.2 Criticizing the current state

After analyzing the current situation, we realized that there are many factors that can improve the relationship with customers. One is to have software that connects leads and customers in one place and helps organize and improve efficiency. The second is to meet user needs and maintain the authenticity of the product through a good user experience. The systems analyzed in the previous section certainly allows the management of customer relationships but does not take into account the specific needs of WanToTrip. In addition, we noticed that most of the researched software lacked an easy-to-use interface and ergonomic design. The existing interfaces are full of information, sometimes unnecessary and disturbing, or they are very simple and do not allow easy and efficient navigation between the application components, leaving customers dissatisfied.

2.3 Proposed solution

Our goal is to create a customer relationship management web application that can meet the needs of WanToTrip. The solution will not only contain the basic features of a CRM as seen in the analysis of the current state. However, it will provide the company with a data base full of the potential community leaders which will allow them to identify their next leaders and generate more revenue. The application will contain the following main features:

- Managing roles.
- Managing contacts (Client, agencies, and community leaders).
- Managing activities.

In addition, we decided to improve the current website to strengthen the relationship with customers and have an effective contact database. To this end, we decided to optimize and improve the existing user interface to better visualize the services provided by WanToTrip, attract the attention of visitors, and keep them on the website for as long as possible.

3. Choice of the methodology

3.1 Methodology adopted: Agile scrum.

3.1.1 Scrum methodology

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems. Scrum software development proceeds to its completion via a series of iterations called Sprint. Small teams consisting of a 6-10 people (it may vary) divide their work into “mini projects (iterations)” that have duration of about one – four weeks during which a limited number of detailed user stories are done. [4]

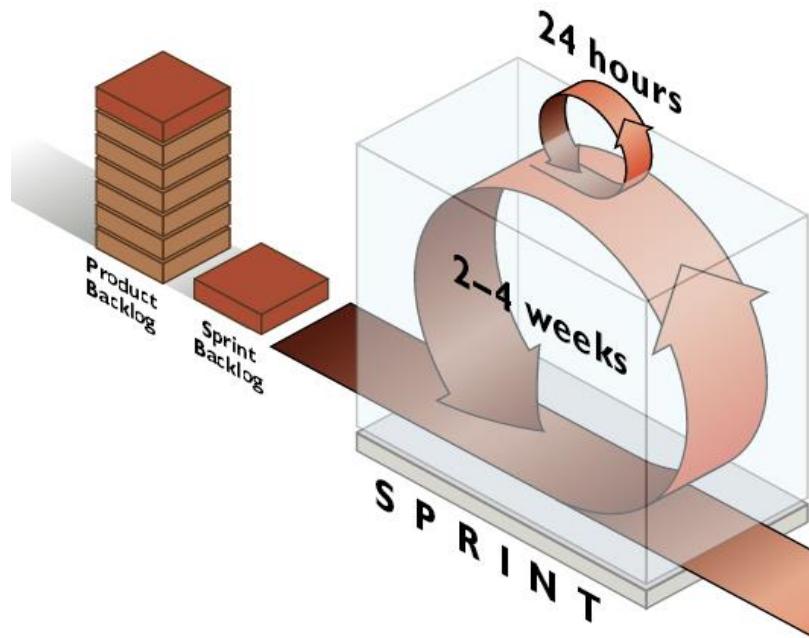


Figure 2: Scrum methodology

The methodology scrum defines three core roles:

- **Scrum master:** The scrum master is the facilitator of the scrum development process. In addition to holding daily meetings with the scrum team, the scrum master makes certain

that scrum rules are being enforced and applied as intended. The scrum master's responsibilities also include coaching and motivating the team, removing impediments to sprints, and ensuring that the team has the best possible conditions to meet its goals and produce deliverable products. [5]

- **Product owner:** The product owner represents stakeholders, which are typically customers. To ensure the scrum team is always delivering value to stakeholders and the business, the product owner determines product expectations, records changes to the product and administers a scrum backlog, a detailed and constantly updated to-do list for the scrum project. The product owner is also responsible for prioritizing goals for each sprint, based on their value to stakeholders, such that the most important and deliverable features are built in each iteration. [5]
- **Scrum team:** The scrum team is a self-organized group of three to nine individuals who have the business, design, analytical and development skills to carry out the actual work, solve problems and produce deliverable products. Members of the scrum team self-administer tasks and are jointly responsible for meeting each sprint's goals. [5]

Conclusion

To sum up, in the first chapter we presented the host organization WanToTrip, then we studied the various existing application in the market to identify the problems that exist, next we presented the solution found that both eliminates the issues found and meets the needs of the organization. Lastly, we studied the methodology of work which we adopted during the development of this project. Nevertheless, the next chapter will be devoted to the analysis and specification of the project needs while describing the product backlog, which will contain all the different features desired in the product.

Chapter 2 : Project Planning

Introduction

Since we will adopt the SCRUM methodology in this project, this chapter will be dedicated to first determine the functional and non-functional requirements specification of the project. In addition, we will identify the actors of our application, the product backlog and plan the sprints. Finally, we will present the application architecture and the technologies used.

1. Requirement's specification

In this section, we will focus on the specification of functional and non-functional project requirements.

1.1. Functional requirements specification

Our system must allow the users to:

- **Authenticate:**
 - Any user already registered in the system can authenticate and access the application using the email and password provided.
- **Password reset:**
 - Any user can request password reset and change it in the case of a forgotten password.
- **Password update:**
 - Any user added by the owner must change their password when trying to authenticate.
- **Profile management:**
 - The user must be able to edit, modify or delete its personal information.
- **Role management:** Only the owner of the application must be able to:
 - Add a role to a specific user and functionalities;
 - Modify role;

- Delete role.
- **User management:** The user of the application must be able to:
 - Add a new user;
 - Modify the role of a specific user;
 - Delete users.
- **Contact management:**
 - The user will be able to consult the details of a specific contact (The deals related to them),
 - The user will be able to add a new contact (Client, Community leader or Agency),
 - The user will be able to modify or delete a contact.
- **Activity management:**
 - The user will be able to add a new activity, modify it or delete it.
- **Deals management:** the user must be able to add new deal, modify it or delete it.
- **View dashboard:** the user must be able to check their activities, deals won or lost, the total of contacts and the total of users in one dashboard.
- **View calendar:** The user must be able to check their activities and deals in a calendar.

1.2. Non-functional requirements specification

- **Usability:** The application must provide a user friendly and ergonomic interfaces to facilitate the use of it by the user even in cases where the user is not a specialist.
- **Security:** The platform must guarantee data integrity, ensure the accuracy of data provide confidentiality. In addition, the system must ensure the availability of data for a proper functionality.
- **Reliability:** The application must function without failure, for a specific range of time and under stated conditions.
- **Reusability:** The application must use existing assets in some form within the software product development process.

- **Availability:** The system's functionalities must be available all time without any interruption or failure.

2. Actor identification

An actor is a role played by anything that interacts with our system. It can be a physical person or even another system. In addition, an actor can directly consult and/or modify the actions of our project by sending or receiving messages likely to carry data. In this project, the role specifies the right of access of each actor to system functionality. These roles vary and depend on the organization WanToTrip. For that, we have chosen two actors for our project. By default, an owner has access to all features of the application. On the other hand, the user has access to the other features depending on the role assigned by the company.

The following are the actors of our system:

- The user: which can be one of WanToTrip employee and has a specific role. For example, it can be a sales representative, a program manager, a designer, or a developer etc.
- The owner: which has access to all features of the user in addition to the role management.

2.1 Product backlog

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team. Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining activities. Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items. This is an ongoing activity to add details, such as a description, order, and size. Attributes often vary with the domain of work. [6]

Table 5: Product backlog

Actor	User story	Priority
User	As a user I want to be able to register users in the platform to have a profile which will allow them to access the various features.	1

	As a user I want to be able to authenticate using an email and password to access the platform.	1
	As a user , I want to be able, to reset my password in the case of forgetting it.	1
	As a user , I want to be able to manage my profile personal information.	2
	As a user , if I have the right role for it, I want to be able to manage the different users.	2
	As a user , if I have the right role for it, I want to be able to manage the activities.	2
	As a user , if I have the right role for it, I want to be able to manage the contacts.	2
	As a user , if I have the right role for it, I want to be able to manage the deals.	2
	As a user , I want to be able to check activities, deals won or lost, total of contacts and total of users in one dashboard.	2
	As a user , I want to be able to view the calendar.	3
	As a user , I want to improve the user experience to collect the contact data via the public website.	4
Owner	As owner , if I have the right role for it, I want to be able to manage the roles.	2

3. Global use case

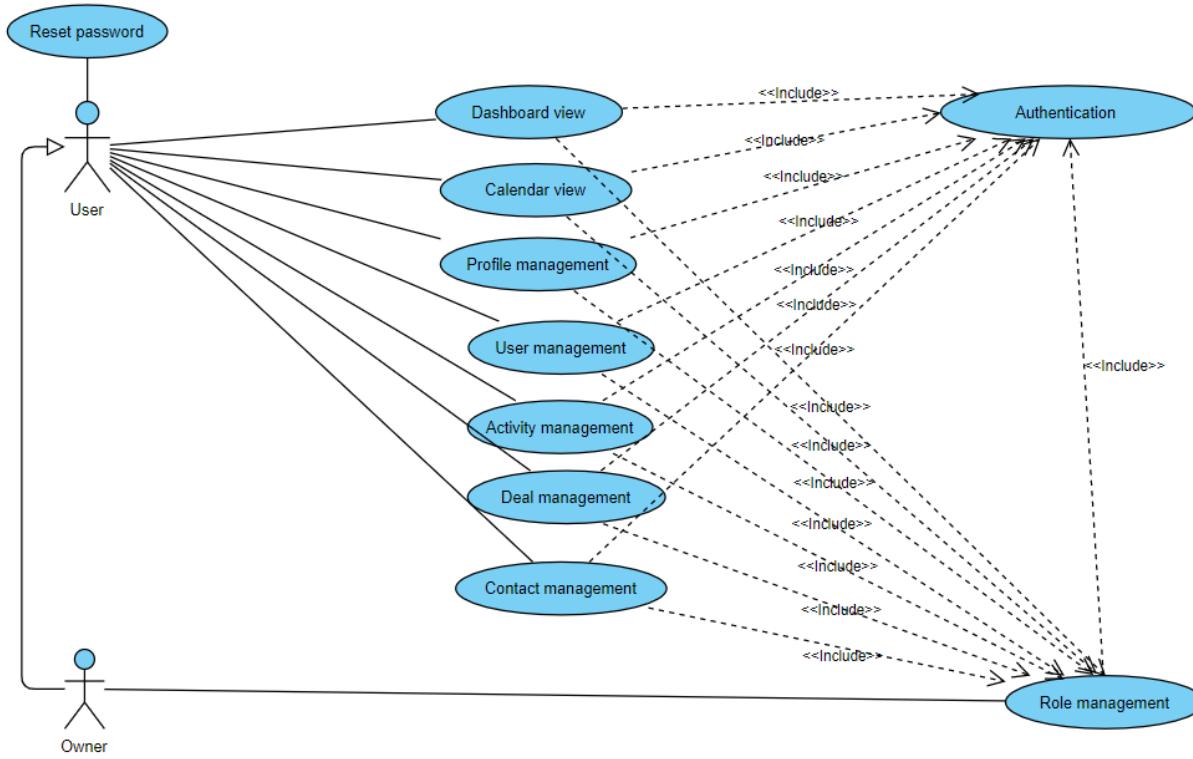


Figure 3: Global use case

4. Management of the project with Scrum

In this section, we will explain how we implemented the scrum methodology in our project starting with the scrum team to planning each sprint.

4.1. The scrum team

Table 6: The scrum team

ROLE	MEMBER NAME
PRODUCT OWNER	Skander Amor
SCRUM MASTER	Hamza Bejaoui
SCRUM TEAM	Manel Kacem

4.2. Sprint planning

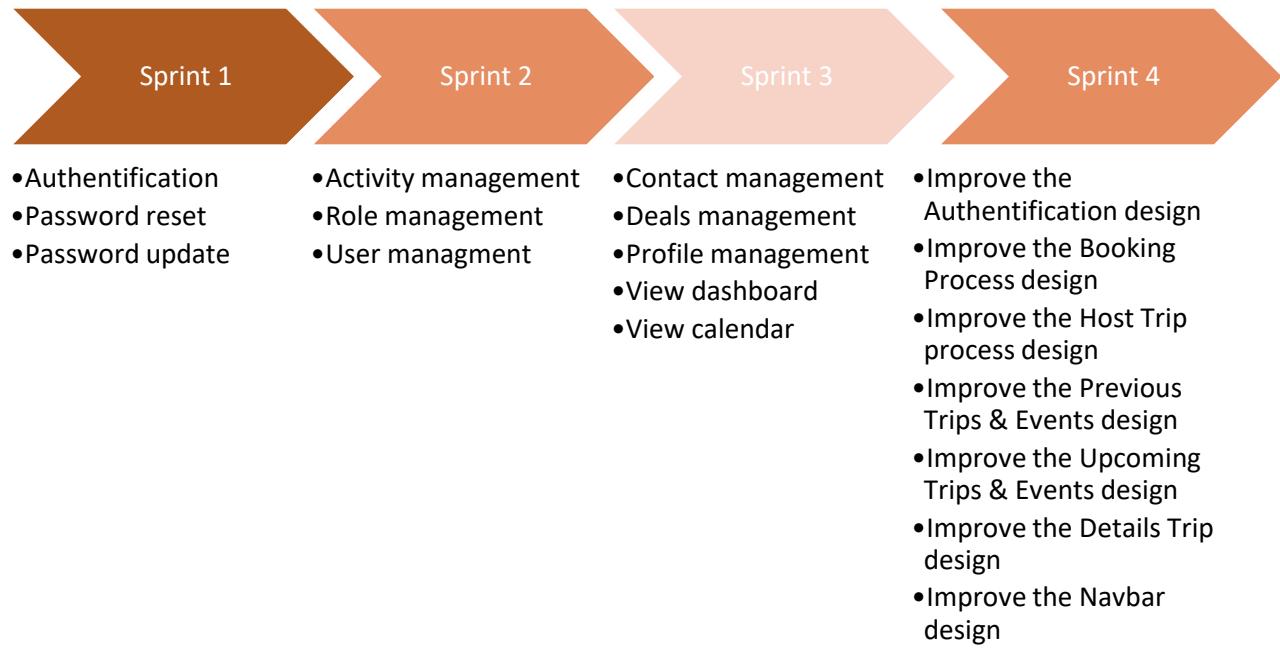


Figure 4:Sprint planning

5. General class diagram

By definition, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. [7] The following diagram is a representation of the general class diagram of our project:

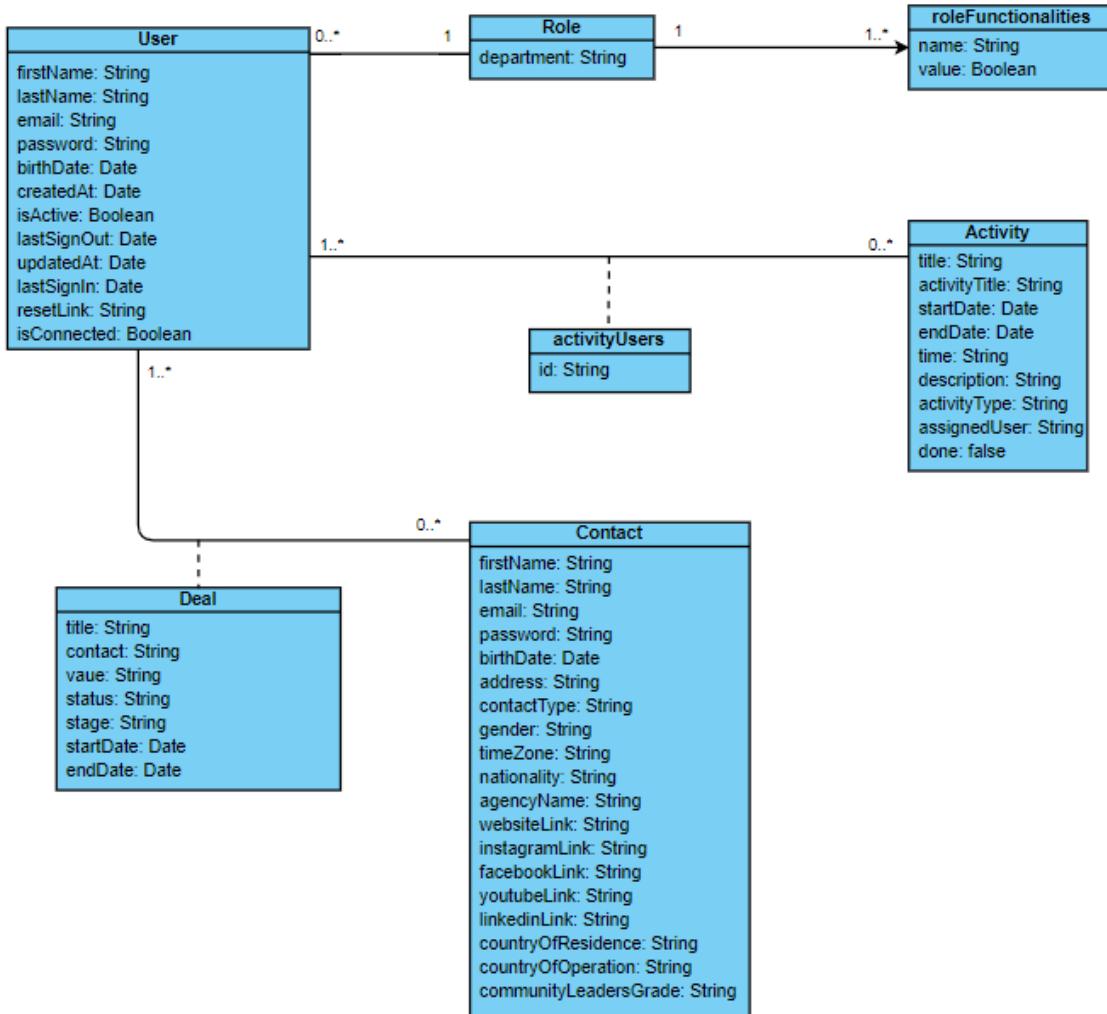


Figure 5: General class diagram

Besides, class diagrams are known for being the main block in object-oriented modeling. However, in our project, the information is stored and retrieved from a NoSQL database, specifically MongoDB and NoSQL databases approach modeling from a different point view. For this reason, we choose to represent the current database model in the following figure below (Figure 6) to better visualize the data.

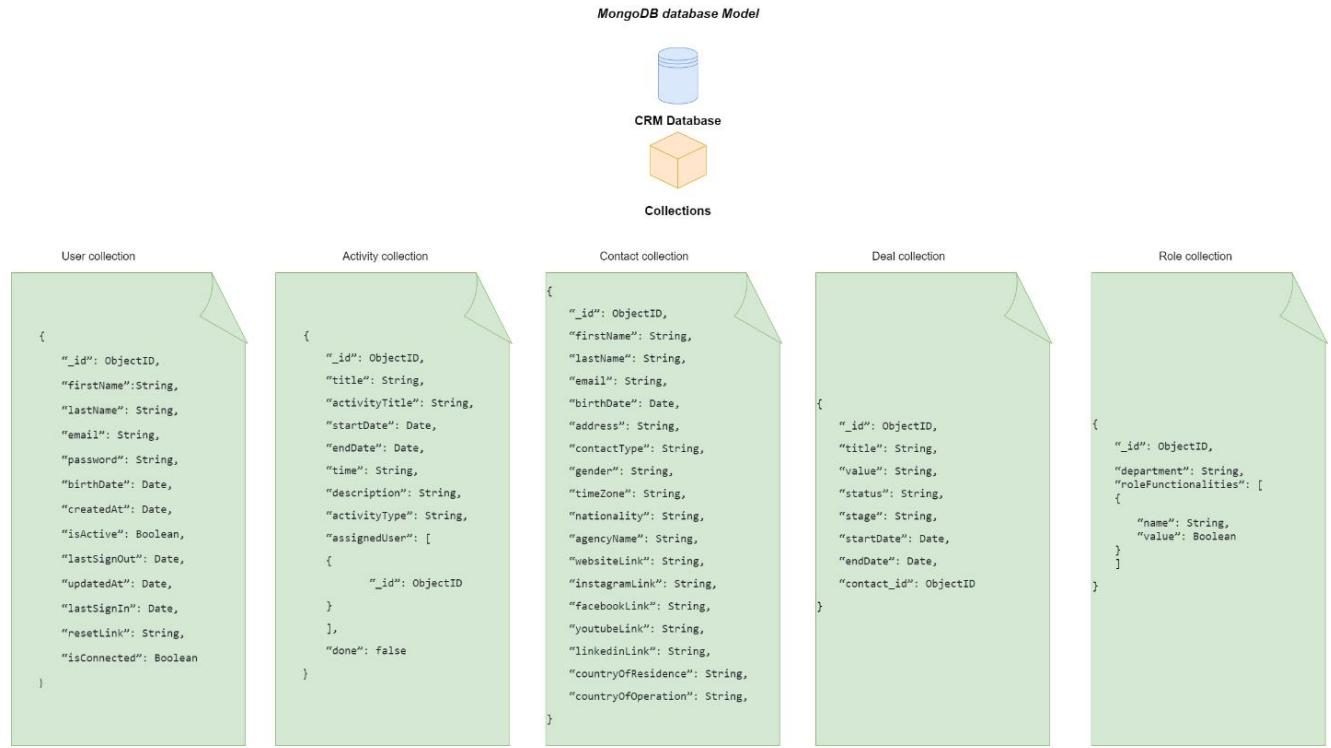


Figure 7: MongoDB database Model

6. The physical architecture

This section is dedicated to showcase the deployment diagram which help to visualize the hardware topology of our system.

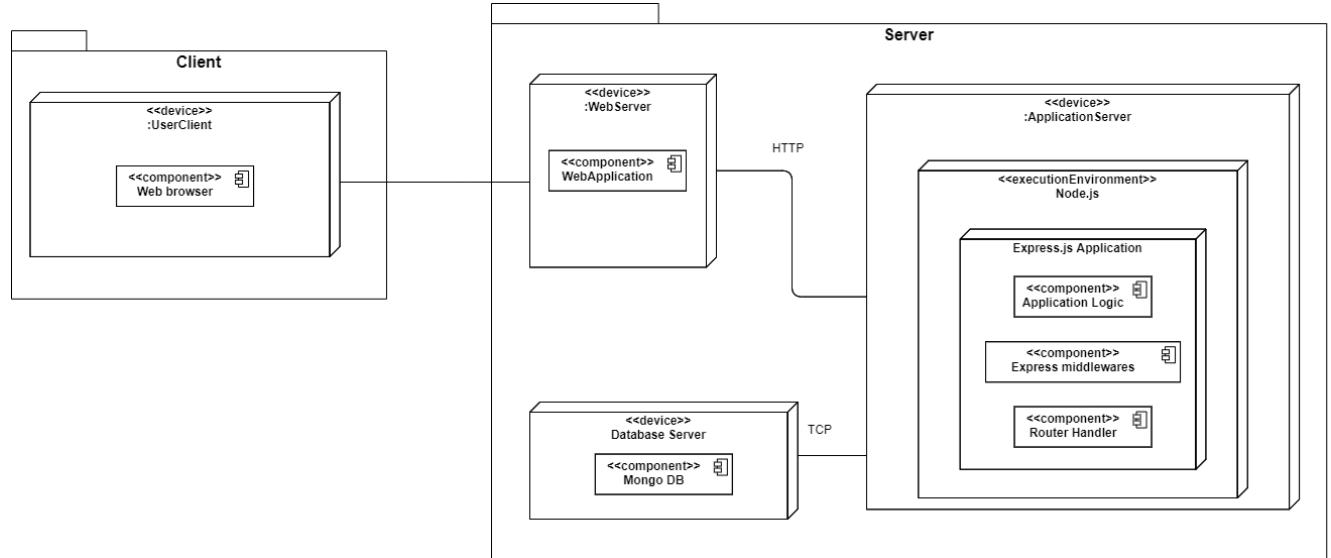


Figure 8: Deployment diagram

7. The application architecture

7.1. Node JS

In our project we used the application programming interfaces (APIs). APIs enables the software to communicate with other pieces of software internal or external consistently, which is a key ingredient in both scalability and reusability. So as to build our API we used the following:

- Express, which vastly simplifies building out common web server tasks under Node.js and is standard fare in building a REST API back end;
- Mongoose, which will connect our back end to a MongoDB database.

Besides, the backend anatomy is as follow:

- **Module:** that makes our code cleaner by encapsulating business logic operations into functions that middleware and routers can call;
- **Middleware:** that will validate prerequisite conditions before Express.js calls the appropriate controller function;
- **Routes:** that use services to process the request before finally sending a response to the requester;
- **Models:** it represents the structure of data, the format, and the constraints with which it is stored. It maintains the data of the application. Essentially, it is the database part of the application.

```
> middleware
> model
> module
> node_modules
< routes\ v1
> public
> secure
JS api.file.router.js
JS api.public.router.js
JS api.secure.router.js
JS apii.router.js
```

Figure 9: Node Js backend anatomy

7.2. Vue JS application

7.2.1 Composition of Vue Js

- **Vue Js components:** Components are one of the most powerful features of Vue.js. They help you extend basic HTML elements to encapsulate reusable code. At a high level, Components are custom elements that Vue.js' compiler would attach specified behavior to. In some cases, they may also appear as a native HTML element extended with the special `is` attribute. [7]
- **Vue Js router:** Vue Js router allow the routing between the different existing components of the web application.
- **Network:** In most case scenarios, a web application communicates with external services such as Rest APIs. This will most likely be done using Axios to fetch data.

8. The software architecture

N-tier architecture is also called multi-tier architecture because the software is engineered to have the processing, data management, and presentation functions physically and logically separated. That means that these different functions are hosted on several machines or clusters, ensuring that services are provided without resources being shared and, as such, these services are delivered at top capacity. The “N” in the name n-tier architecture refers to any number from 1. Not only does your software gain from being able to get services at the best possible rate, but it is also easier to manage. This is because when you work on one section, the changes you make will not affect the other functions. And if there is a problem, you can easily pinpoint where it originates. [8]



Figure 10: The software architecture

N-tier architecture would involve dividing an application into three different tiers. These would be the

- Logic tier: This layer coordinates the application, processes command, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers. ➔ The API of the project.
- The presentation tier: The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand. ➔ The Web application
- The data tier: The information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user. ➔ Mongo Database

9. The technological options



Vue Js is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries. [7]



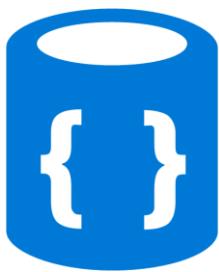
Vuetify is a complete UI framework built on top of Vue.js. The goal of the project is to provide developers with the tools they need to build rich and engaging user experiences. Unlike other frameworks, Vuetify is designed from the ground up to be easy to learn and rewarding to master with hundreds of carefully crafted components from the Material Design specification. Vuetify takes a mobile first approach to design which means

your application just works out of the box, whether it is on a phone, tablet, or desktop computer. [8]



Bulma

Bulma is an Open-source CSS framework developed by Jeremy Thomas. This framework is based on the CSS Flexbox property. It is highly responsive, minimizing the use of media queries for responsive behavior. [11]



NoSQL databases (aka "not only SQL") are non-tabular, and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads. [9]



mongoDB

Mongo DB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and function which is the equivalent of relational database tables. MongoDB is a database which came into light around the mid-2000s. [10]



Node JS is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. In addition, Node.js provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent. [11]

10. The software environment



Visual studio code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. [12]



Git is a free and open-source version control system, originally created by Linus Torvalds in 2005. Version control systems (VCSs) are tools that can track changes to programming code (or other collections of files). VCSs track changes to a folder and its contents in a series of snapshots and maintain metadata. The track record is saved in a database, a repository in Git. [13]



Postman is a tool when trying to dissect RESTful APIs made by others or test ones you have made yourself. It offers a sleek user interface with which to make HTML requests, without the hassle of writing a bunch of code just to test an API's functionality. [14]



Trello is the visual work management tool that empowers teams to ideate, plan, manage, and celebrate their work together in a collaborative, productive, and organized way. [16]



Visual Paradigm (VP-UML) is a UML case tool supporting UML 2, SysML and Business Process Modeling Notation (BPMN) from the Object Management Group (OMG). In addition to modeling support, it provides report generation and code engineering capabilities including code generation. It can reverse engineer diagrams from code,

and provide round-trip engineering for various programming languages. [18]

11. The hardware environment

This web application was developed under the following computer:

Table 7: The hardware environment

Owner	Manel Kacem
Processor	AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx, 2100 Mhz, 4 Core(s), 8 Logical Processor(s)
RAM	16.0 GB
Operating system	Microsoft Windows 10 Home

Conclusion

To sum up, in the second chapter we included everything related to the project management. Starting from the determining the functional and non-functional requirement specification moving to identifying the different actors of the application, to organizing the needs of the application in the product backlog and plan the sprints. Finally, we presented the application architecture, technological options and the environment used.

Chapter 3 : Study and execution of Sprint 1

Introduction

This chapter includes four main parts. The first contains a to-do list for the sprint also known as the sprint backlog, which will display a list of user stories that will be completed during this sprint. The second part is the analysis, we will identify the use case diagram and the description of the related scenarios. The third part will be dedicated to concepts, mainly we will identify sequence diagrams. Finally, we will cover the testing of each feature in this sprint.

1. Sprint Backlog

In this sprint, our goal is to satisfy the following user stories described in the table below.

Table 8: Sprint1 Backlog

Feature name	User Story	Difficulty	Estimation
Authentication	As a user, I want to be able to authenticate using an email and password to access the platform.	Difficult	7
Password reset	As a user, I want to be able to reset my password in the case of forgetting it.	Difficult	7
Password update	As a user, I want to be able to update my password when first set to default.	Difficult	7

2. Analysis

The use-case diagrams model the behavior of a system and help to capture the requirements of the system. These diagrams help identify the interactions between the system and its actors.

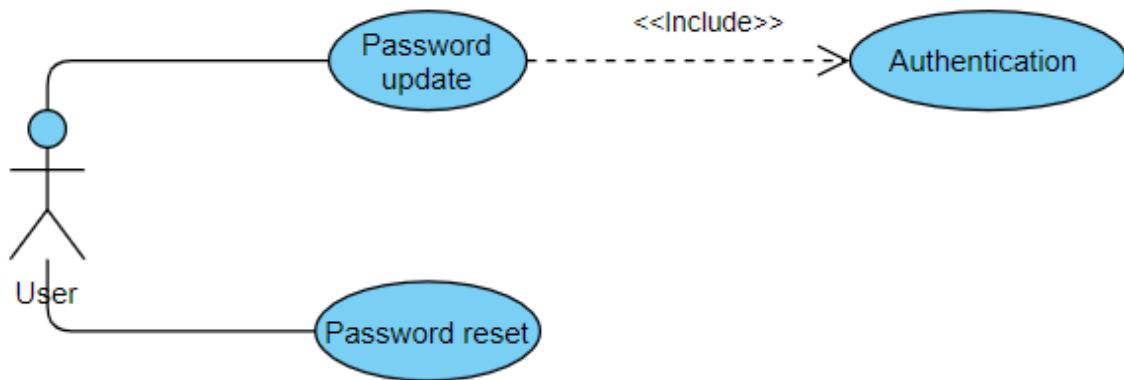


Figure 11: Use case diagram Sprint1

2.1 Refine Use case diagrams

2.1.1 Use case « Password update »

Table 9: Use case description "Password update"

Title	Password update
Actor	User
Goal	User can update their password after being added to the application.
Pre-conditions	User must be registered first.
Nominal scenario	<ol style="list-style-type: none"> 1. The system displays the update password interface; 2. The user enters both the new password and confirm password; 3. The user clicks the save button; 4. The system verifies the information entered; 5. The system redirects user to the login page.
Alternative scenario	2.a New password does not match confirm password; 3.a The user clicks the cancel button.

2.1.2 Use case « Password reset »

Table 10: Use case description “Password reset”

Title	Password reset
Actor	User
Goal	User can reset their password.
Pre-conditions	User must be registered in the system.
Nominal scenario	<ol style="list-style-type: none"> 1. The user enters the email and clicks the button to reset the password; 2. The system checks the email entered; 3. The system sends an email to reset password; 4. The user opens the email sent and clicks the link found; 5. The system redirects the user to the reset password interface; 6. The user enters the new password and confirm password; 7. The system updates the password; 8. The system redirects the user to the login interface.
Alternative scenario	2.a The email does not exist 6.a The new password does not match the confirm password
Exceptional path	5.a The JWT expires.

2.1.3 Use case « Authentication »

Table 11: Use case "Authentication"

Name	Authentication
------	----------------

Actor	User
Goal	User can log into his account.
Pre-conditions	User must be registered first.
Nominal scenario	<ol style="list-style-type: none"> 1. User must open the application; 2. User must enter the email and password provided; 3. The system verifies the information entered; 4. The system redirects to the main page.
Alternative scenario	3.a Fields are not valid.

3. Conception

3.1 Sequence diagrams

3.1.1 Sequence diagram “Password update”

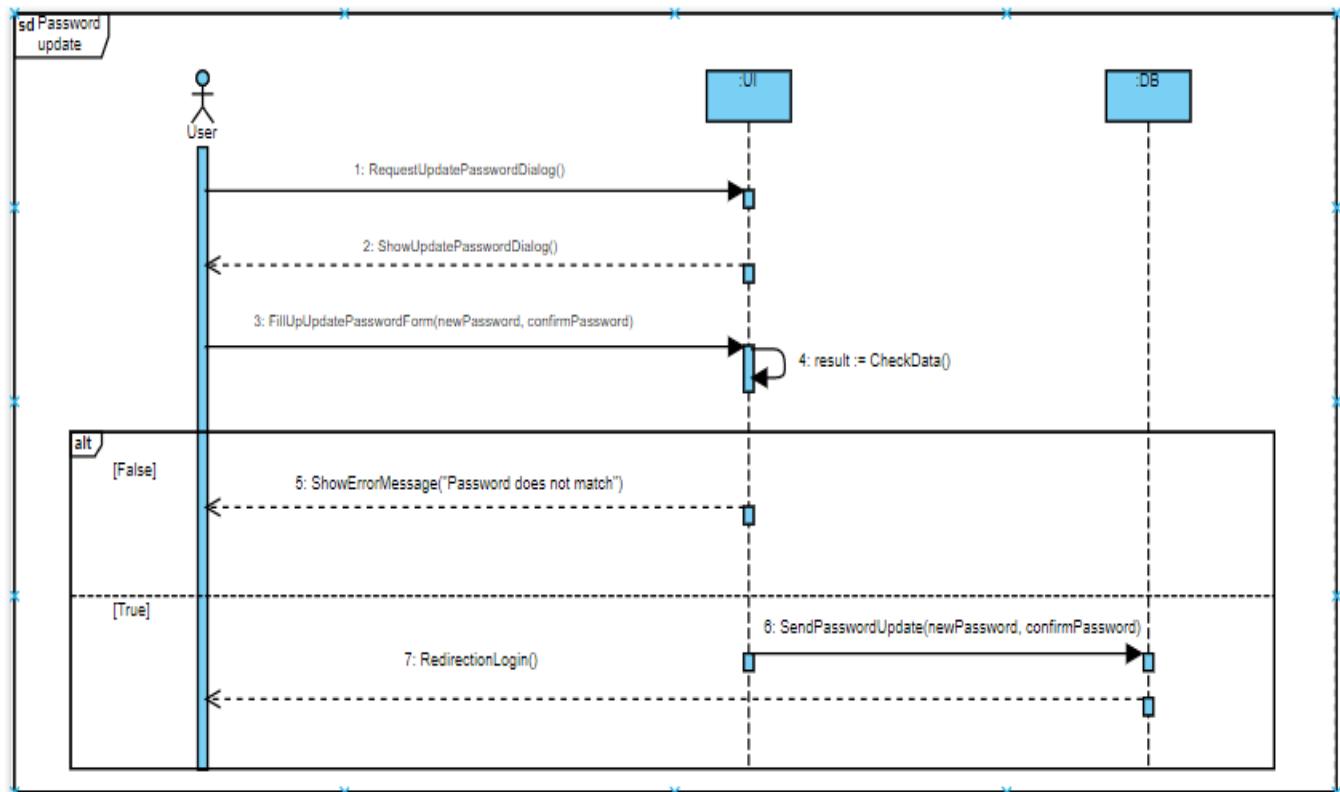


Figure 12: Sequence diagram “Password update”

3.1.2 Sequence diagram “Reset password”

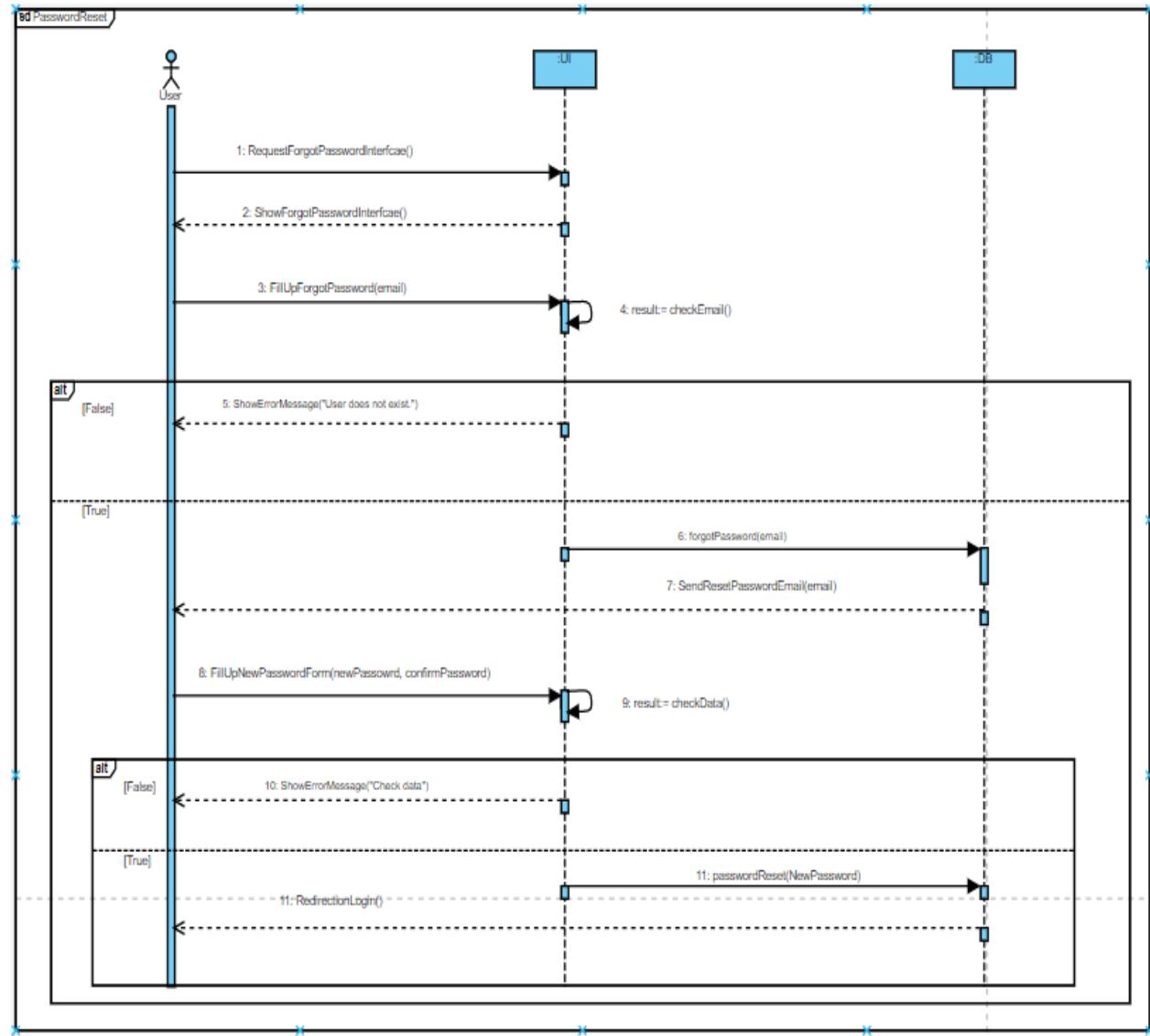


Figure 13: Sequence diagram “Reset password”

3.1.3 Sequence diagram “Authentication”

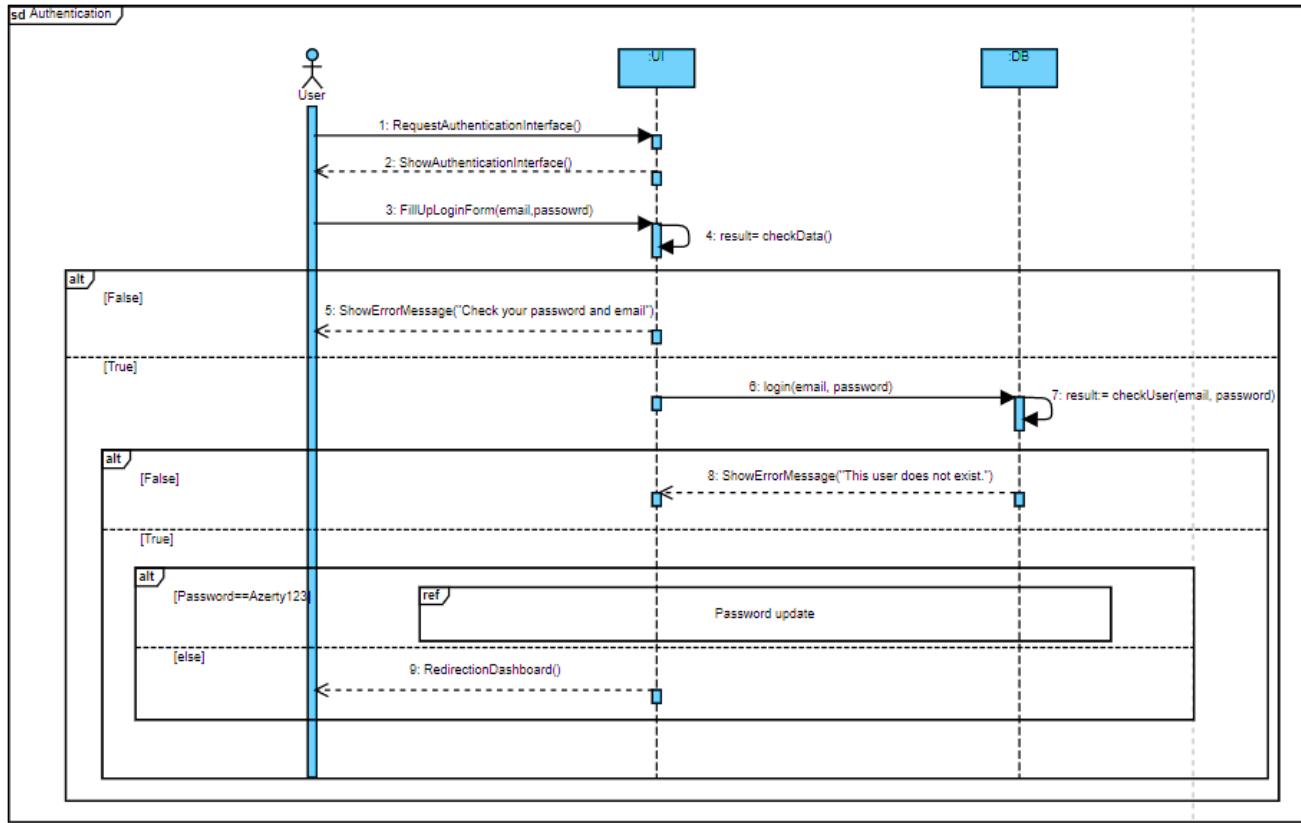


Figure 14: Sequence diagram “Authentication”

4. Realization and testing

4.1 Testing execution and results

4.1.1 Testing Authentication

The following is a representation of the authentication errors that can occur when the data is invalid.

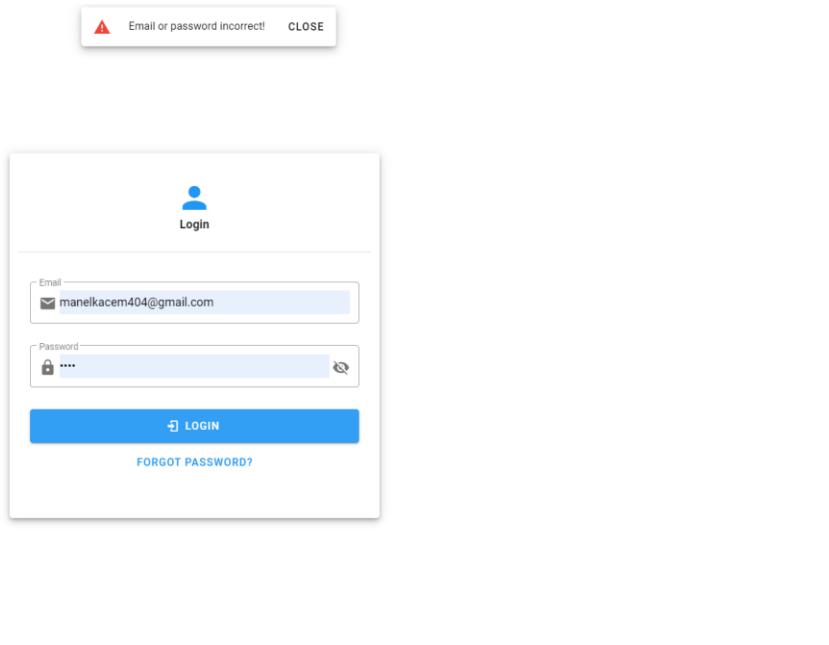


Figure 15: Authentication error – data not valid

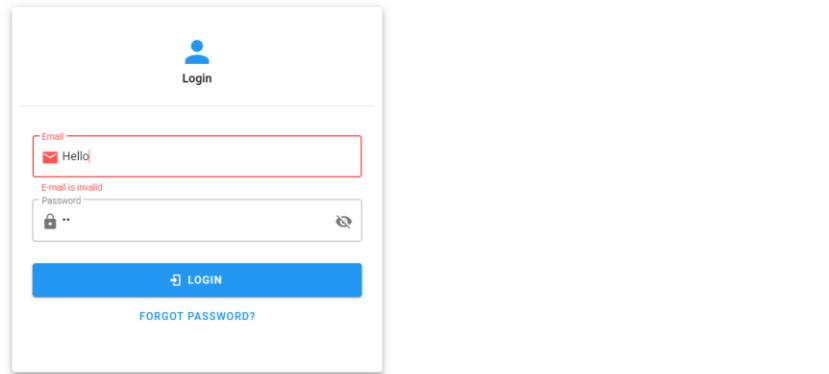


Figure 16: Authentication error – email is invalid

4.1.2 Testing Reset Password

For the reset password we have choose to use the following steps:

1. The user enters their email;
2. The user clicks the link in the mailbox;
3. The user enters the new password.

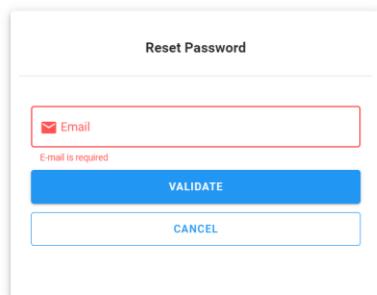
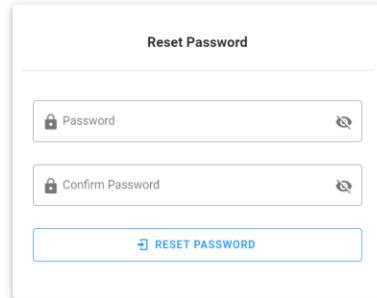


Figure 17: The interface "Forgot password"



Figure 18: Email with the reset link



The image shows a clean, modern 'Reset Password' interface. It features a light gray header with the title 'Reset Password'. Below the header are two input fields: 'Password' and 'Confirm Password', both preceded by a lock icon. Each field has a small circular icon with a question mark or exclamation mark to its right. A blue rectangular button labeled 'RESET PASSWORD' with a small arrow icon is positioned below the input fields.

Figure 19: Reset password interface

4.1.3 Testing password update

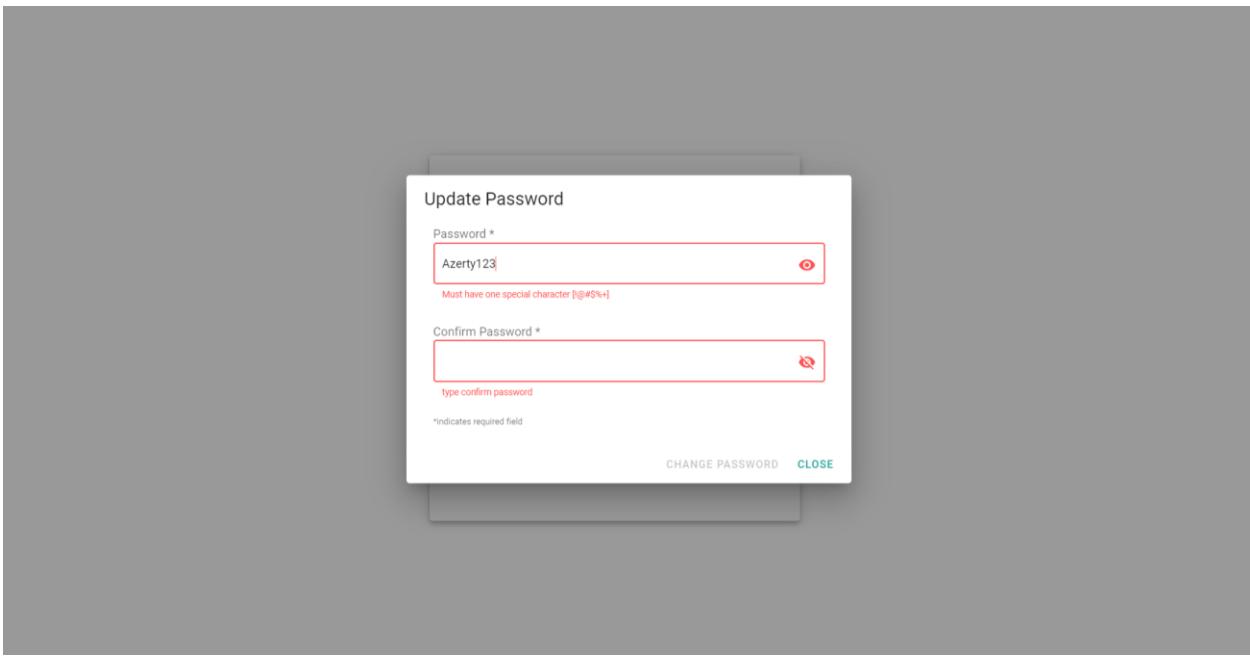


Figure 20: Update password interface

Conclusion

In this chapter, we looked at the first sprint of our project. For this, we have completed the specification, analysis, conception, and testing. In the next chapter, we will focus on our second sprint.

Chapter 4 : Study and execution of Sprint 2

Introduction

The fourth chapter includes four main parts. The first one contains the sprint backlog, which will display a list of user stories that will be completed during this sprint. The second part is the analysis, we will identify the use case diagram and the description of the related scenarios. The third part will be dedicated to concepts, mainly we will identify sequence diagrams. Finally, we will cover the testing of each feature of the second sprint.

1. Sprint Backlog

In this sprint, our goal is to satisfy the following user stories described in the table below.

Table 12: Sprint2 Backlog

Feature name	User Story	Difficulty	Estimation
Activity management	As a user , if I have the right role for it, I want to be able to manage the activities.	Medium	6
Role management	As a user , if I have the right role for it, I want to be able to manage the roles.	Medium	6
User management	As a user , if I have the right role for it, I want to be able to manage the users.	Medium	6

2. Analysis

The use-case diagrams model the behavior of a system and help to capture the requirements of the system. These diagrams help identify the interactions between the system and its actors. In the following diagram we present the use case diagram for the second sprint:

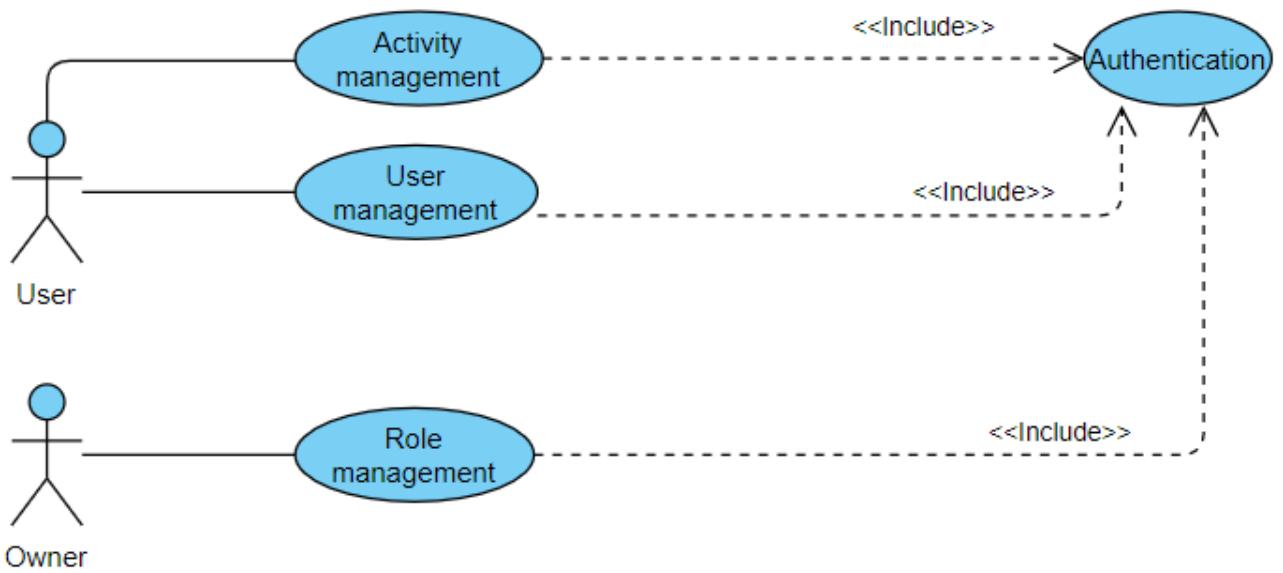


Figure 21: Use case diagram Sprint2

2.1 Use case diagrams

2.1.1 Use case «Activity management»

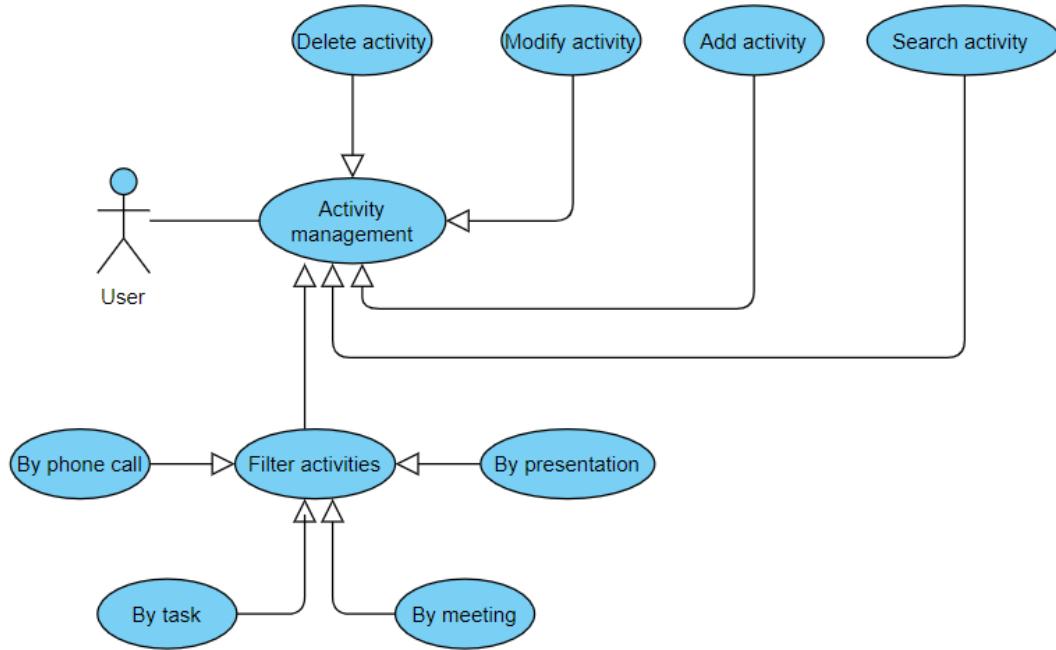


Figure 22: Use case "Activity management"

Table 13: Use case description " Add activity"

Name	Add activity
Actor	User
Goal	Add new activity.
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User clicks on activity management found in the navbar; 2. The system shows the page of activity management; 3. The user clicks on the add activity button; 4. The user fills the form; 5. The user clicks on the add button; 6. The system displays a success message.
Alternative scenario	4.a. Invalid fields 5.a The user clicks on the cancel button.

Table 14: Modify activity

Name	Modify activity
Actor	User
Goal	Modify activity.
Pre-conditions	The user must be authenticated and have the right role.

Nominal scenario	<ol style="list-style-type: none"> 1. User clicks on activity management found in the navbar; 2. The system shows the page of activity management; 3. The user clicks on the activity they want to modify; 4. The user completes the form; 5. The user clicks on the save button; 6. The system displays a success message.
Alternative scenario	4.a invalid fields 5.a the user clicks on the cancel button

Table 15: Use case description "Delete activity"

Title	Delete activity
Actor	User
Goal	Delete activity
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User clicks on activity management found in the navbar; 2. The system shows the page of activity management; 3. The user clicks on the activity they want to delete; 4. The user clicks the “Trash icon” to delete the activity; 5. The system displays an alert to check the choice of the user; 6. The user clicks the confirm button; 7. The system displays a success message.
Alternative scenario	6.a The user clicks on the cancel button

2.1.2 Use case « Role management »

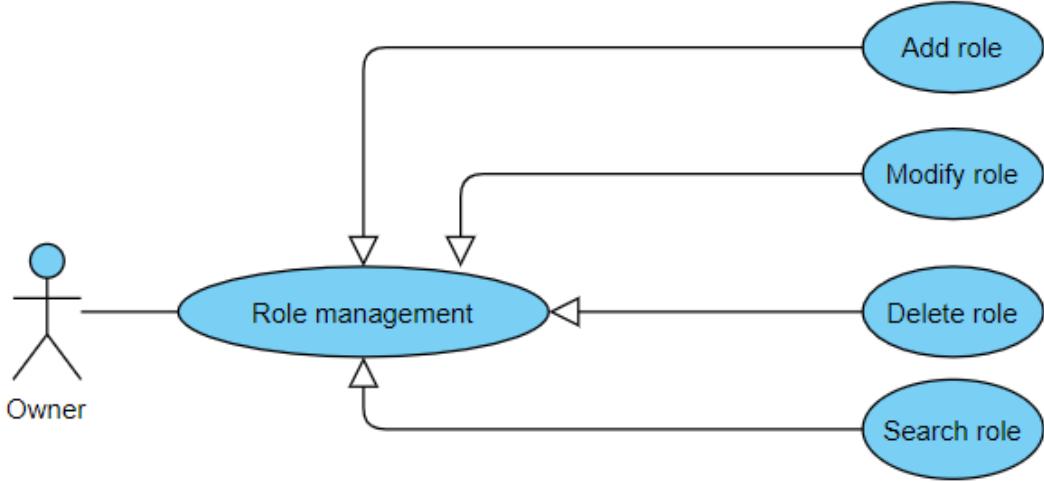


Figure 23: Use case description “Role management”

Table 16: Use case description “Add role”

Title	Add role
Actor	Owner
Goal	Add role
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. Owner select role management found in the navbar; 2. The system shows the page role management; 3. The owner clicks on the add role button; 4. The owner fills in the role name and functionalities associated to it, then clicks on the add button; 5. The system adds the new role; 6. The system displays a success message.

Alternative scenario	4.a The owner clicks the cancel button; 5.a The name of the department exists already in the system.
----------------------	---

Table 17: Use case description “Modify role”

Title	Modify role
Actor	Owner
Goal	Modify role
Pre-condition	The owner must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. Owner select role management found in the navbar; 2. The system shows the page role management; 3. The owner clicks on the “Pencil icon” to edit the specific role; 4. The owner edits the role name and functionalities associated to it, then clicks on the save button; 5. The system saves the changes; 6. The system displays a success message.
Alternative scenario	4.a The owner clicks on the cancel button.

Table 18: Use case description “Delete role”

Title	Delete role
Actor	Owner
Goal	Delete role
Pre-conditions	The user must be authenticated and have the right role.

Nominal scenario	<ol style="list-style-type: none"> 1. Owner selects role management found in the navbar; 2. The system shows the page role management; 3. The owner clicks on the “Trash icon” to delete the specific role; 4. The system shows a confirmation window; 5. The owner clicks on “Ok” button to confirm the delete; 6. The system displays a success message.
Alternative scenario	5.a The owner clicks on the “Cancel” button

2.1.3 Use case « User management »

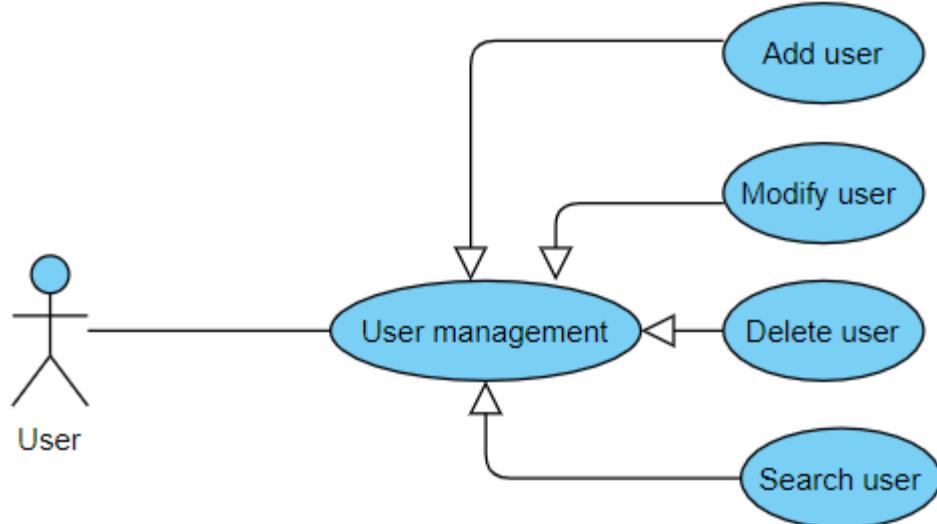


Figure 24: Use case "User management"

Table 19: Use case description "Add user"

Title	Add user
Actor	User
Goal	Add user

Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User select user management found in the navbar; 2. The system shows the page user management; 3. The user clicks the button “New user” to add new user; 4. The user enters the email and select the role for the new user; 5. The user clicks on the save button; 6. The system validates the email field; 7. The system adds the new user; 8. The system displays a success message.
Alternative scenario	5.a The user clicks the cancel button; 6.a. The email entered exists already in the system

Table 20: Use case description "Modify user"

Title	Modify user
Actor	User
Goal	Modify user
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User select user management found in the navbar; 2. The system shows the page user management; 3. The user clicks the button “Pencil” in the actions column to edit the selected user; 4. The user selects the new role from the list; 5. The user clicks the save button;

	<ol style="list-style-type: none"> 6. The system saves the changes made; 7. The system displays a success message.
Alternative scenario	5.a The user clicks the cancel button

Title	Delete user
Actor	User
Goal	Delete user
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User select user management found in the navbar; 2. The system shows the page user management; 3. The user clicks the button “trash icon” in the actions column to delete the selected user; 4. The systems show a confirmation window; 5. The user clicks the “Ok” button; 6. The system displays a success message.
Alternative scenario	5.a The user clicks the cancel button

Figure 25: Use case description "Delete user"

4. Conception

4.1 Sequence diagrams

4.1.1 Sequence diagram “Add activity”

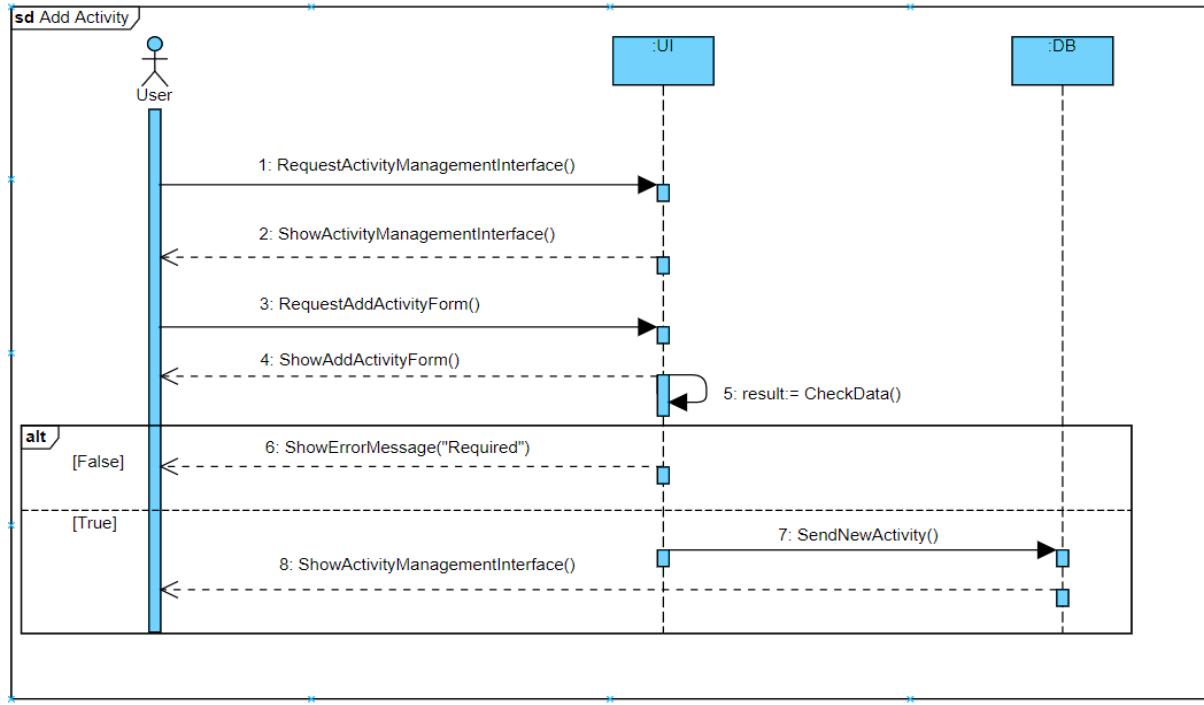


Figure 26: Sequence diagram “Add activity”

4.1.2 Sequence diagram “Modify activity”

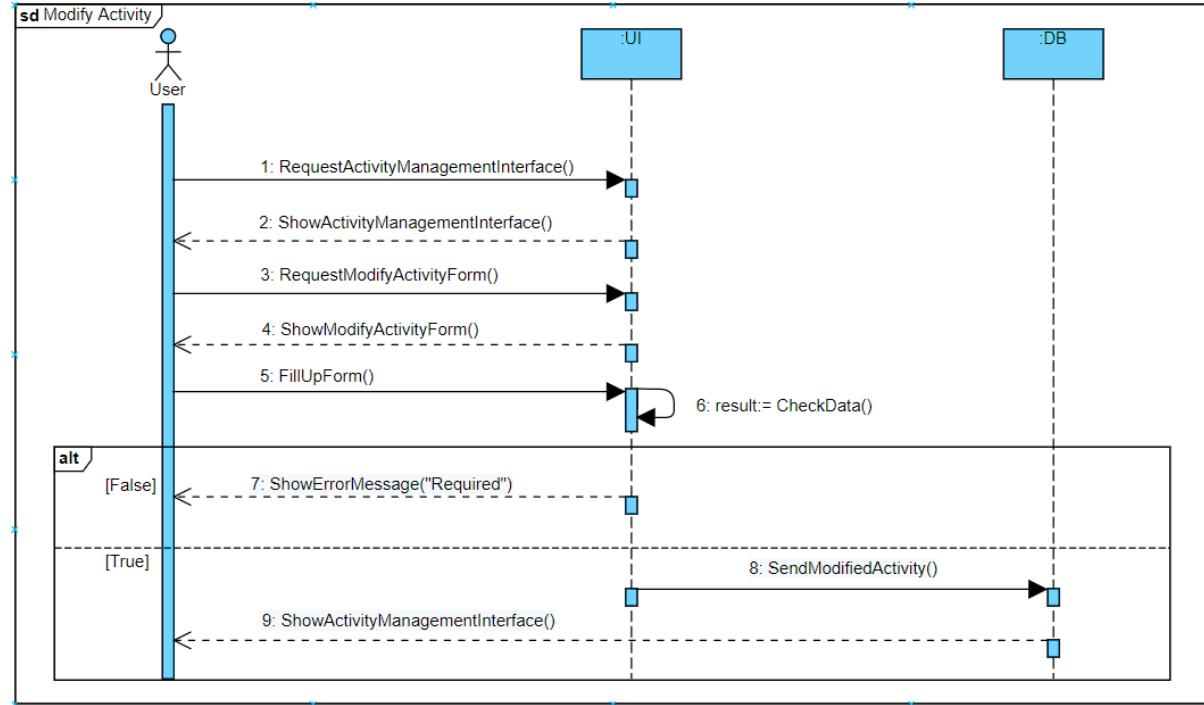


Figure 27: Sequence diagram “Modify activity”

4.1.3 Sequence diagram “Delete activity”

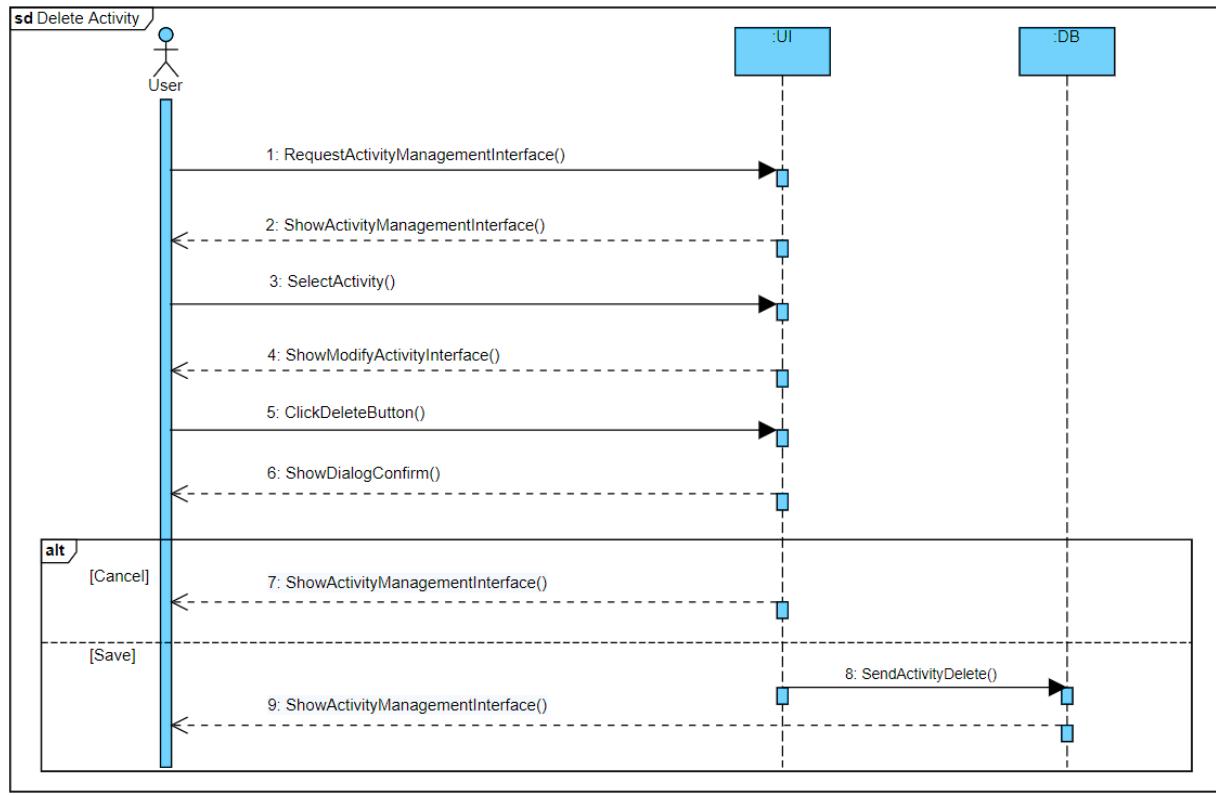


Figure 28: Sequence diagram “Delete activity”

4.1.4 Sequence diagram “Add role”

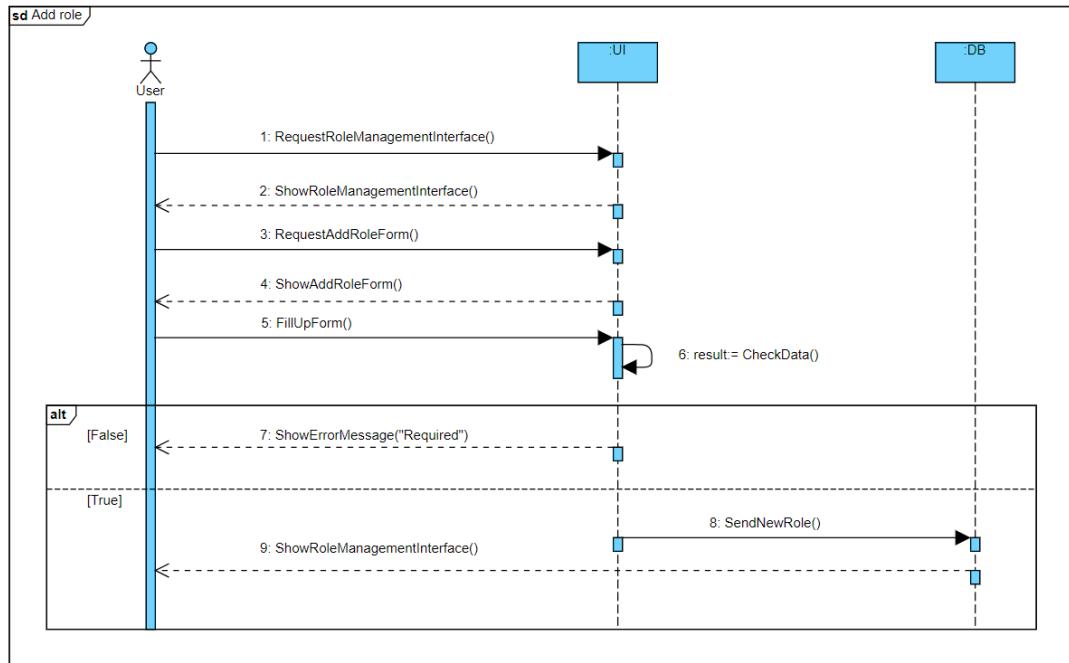


Figure 29: Sequence diagram “Add role”

4.1.5 Sequence diagram “Modify role”

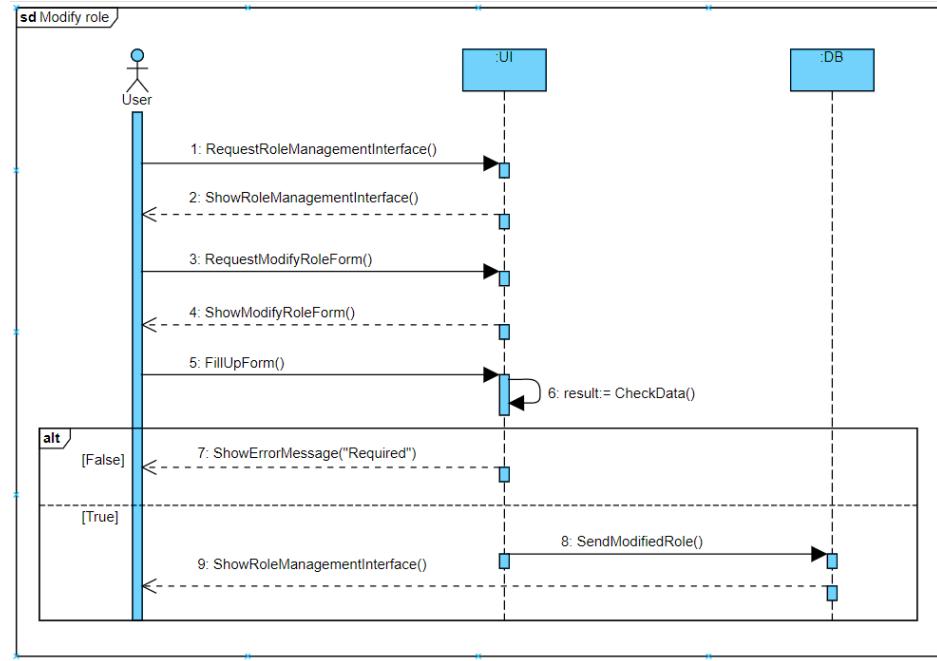


Figure 30: Sequence diagram “Modify role”

4.1.6 Sequence diagram “Delete role”

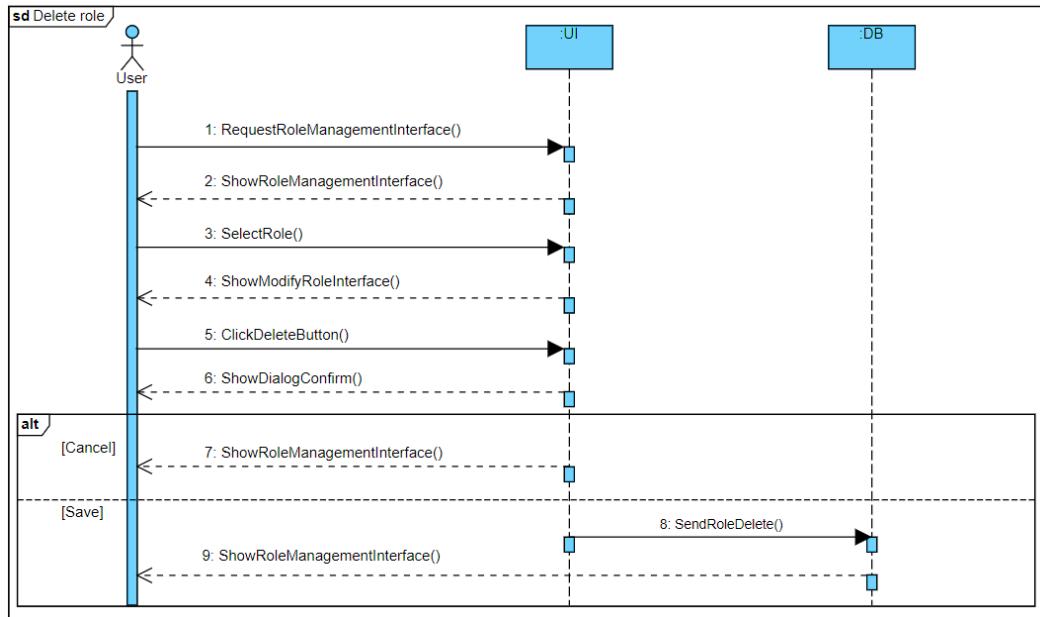


Figure 31: Sequence diagram “Delete role”

4.1.7 Sequence diagram “Add user”

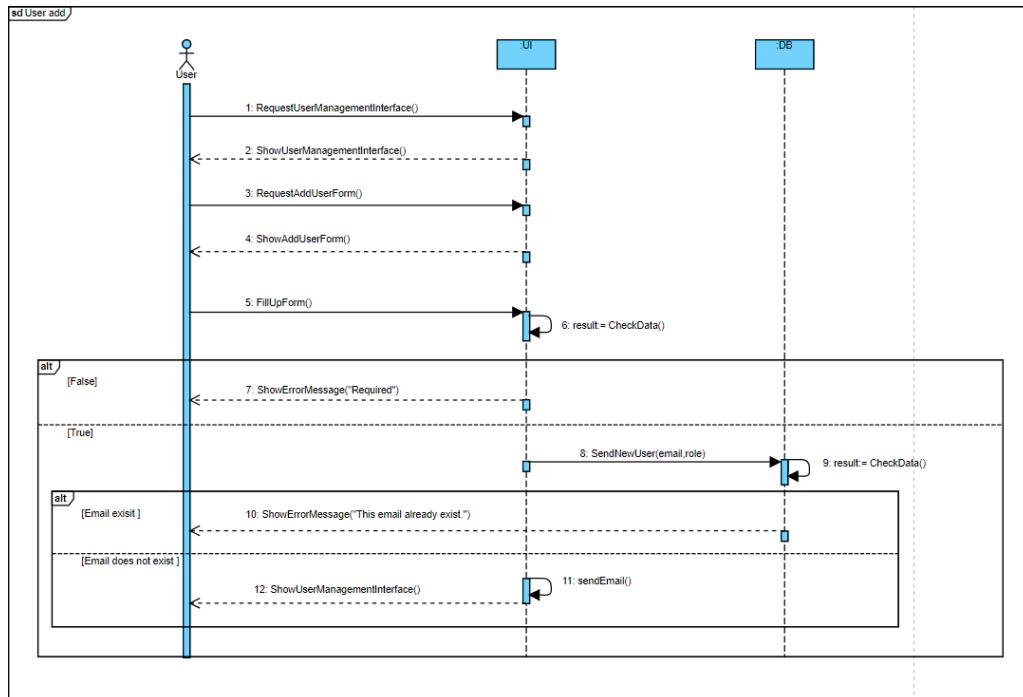


Figure 32: Sequence diagram “Add user”

4.1.8 Sequence diagram “Modify user”

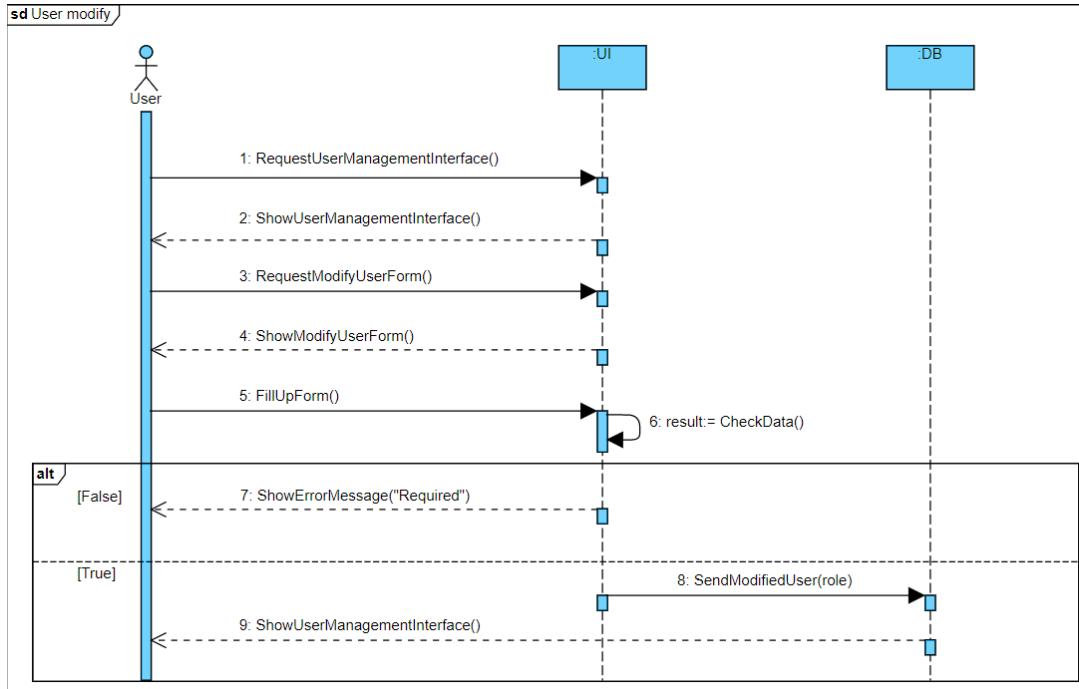


Figure 33: Sequence diagram “Modify user”

4.1.9 Sequence diagram “Delete user”

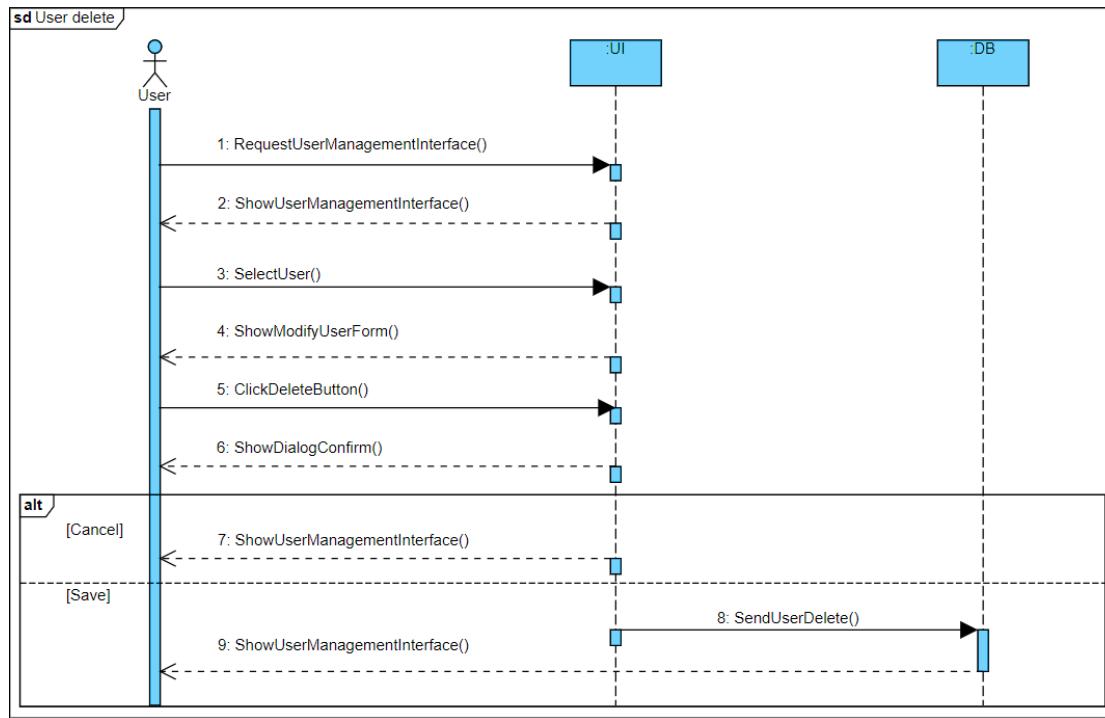


Figure 34: Sequence diagram “Delete user”

5. Realization and testing

5.1 Testing execution and results

5.1.1 Testing Activity management

The screenshot shows a user interface for managing activities. On the left is a sidebar with a user profile icon and the name "manelkacem12 HR". Below the profile are several menu items: Dashboard, Profile Management, Activity Management (which is highlighted in grey), Calendar, User Management, Contact Management, and Deals Management. At the bottom of the sidebar is a "LOGOUT" button. The main area is titled "Activities" and contains a single activity card. The card has a red border and displays the title "Meeting with the sal...", type "Meeting", start date "2021-06-12", end date "2021-06-12", and time "21:05". Above the card is a toolbar with icons for refresh, search, and add activity. To the right of the card is a search bar with placeholder text "Search" and a green "ADD ACTIVITY" button.

Figure 35: User interface "Activity management"

■ Add activity:

The screenshot shows the same user interface as Figure 35, but with a modal window open over the "Activities" list. The modal is titled "New Activity" and contains fields for "Type" (set to "Meeting"), "Time" (set to "09:00 AM"), "Start Date" (set to "2021-07-01"), "End Date" (set to "2021-07-01"), "People" (set to "mariem@gmail.com"), "Title" ("Marketing brainstorming"), and "Description" ("Marketing brainstorming meeting with the mariem at 9.00AM"). At the bottom of the modal are "CANCEL" and "SAVE" buttons. The background of the main screen is dimmed to indicate the modal is active.

Figure 36: Test "Add activity"

The screenshot shows a user interface for managing activities. On the left is a sidebar with a user profile (manelkacem12, HR) and links for Dashboard, Profile Management, Activity Management (which is selected), Calendar, User Management, Contact Management, and Deals Management. At the bottom of the sidebar is a 'LOGOUT' button. The main area is titled 'Activities' and displays two cards for recently added meetings:

- Marketing brainstor...** (Meeting)
 - Start Date : 2021-07-01
 - End Date : 2021-07-01
 - Time : 09:00
- Meeting with the sal...** (Meeting)
 - Start Date : 2021-06-12
 - End Date : 2021-06-12
 - Time : 21:05

A success message 'Activity added!' with a checkmark icon is shown at the top right. There are also 'CLOSE', 'Search', and 'ADD ACTIVITY' buttons.

Figure 37: Test add activity successfully

5.1.2 Testing Role management

The screenshot shows a 'Role Management' page. The sidebar includes a user profile (manelkacem12, Owner) and links for Dashboard, Profile Management, Activity Management (selected), Calendar, Role Management (which is selected), User Management, Contact Management, and Deals Management. A 'LOGOUT' button is at the bottom. The main content is a table titled 'Role Management' with a 'NEW ROLE' button. The table has columns for Department, Role management, User management, Activity management, Contact management, Deals management, and Actions. It lists four roles: Sales, Dev, HR, and Owner, each with specific permissions marked by green or red circles.

Department	Role management	User management	Activity management	Contact management	Deals management	Actions
Sales	⊖	⊖	⊕	⊕	⊕	⊕
Dev	⊖	⊖	⊕	⊖	⊖	⊕
HR	⊖	⊕	⊕	⊕	⊕	⊕
Owner	⊕	⊕	⊕	⊕	⊕	⊕

Figure 38: User interface "Role management"

■ **Add role:**

The screenshot shows a user interface for 'Role Management'. On the left is a sidebar with a profile picture and the name 'manelkacem12 Owner'. Below it are links for Dashboard, Profile Management, Activity Management, Calendar, Role Management (which is selected), User Management, Contact Management, and Deals Management. At the bottom of the sidebar is a 'LOGOUT' button. The main area has a header 'Role Management' with a search bar and a 'NEW ROLE' button. Below the header is a table with columns: Department, Role management, User management, Activity management, Contact management, Deals management, and Actions. Rows represent Sales, Dev, HR, and Owner. A modal window titled 'New Role' is open, containing a text input field with 'Marketing', and several checkboxes for permissions: 'User Management' (unchecked), 'Activity Management' (checked), 'Contact Management' (checked), and 'Deals Management' (unchecked). At the bottom of the modal are 'CANCEL' and 'SAVE' buttons.

Figure 39: Test "Add role"

This screenshot is similar to Figure 39 but shows the result of adding a new role. The 'Role Management' table now includes a new row for 'Marketing'. A success message 'Role added!' with a green checkmark icon is displayed above the table. The rest of the interface is identical to Figure 39, including the sidebar and the 'NEW ROLE' button.

Figure 40: Test role added successfully

5.1.3 Testing User management

The screenshot shows the 'User management' section of a software application. On the left is a vertical sidebar menu with the following items:

- Dashboard
- Profile Management
- Activity Management
- Calendar
- Role Management
- User Management** (highlighted in grey)
- Contact Management
- Deals Management

At the bottom of the sidebar is a 'LOGOUT' button.

The main content area has a header 'User management' with a search bar and a 'NEW USER' button. Below the header is a table with columns: Email, Role, Profile, and Actions. Two users are listed:

Email	Role	Profile	Actions
manelkacem12@gmail.com	Owner	PROFILE	EDIT
mariem@gmail.com	Sales	PROFILE	EDIT

A circular navigation bar with arrows and the number '1' is centered below the table.

Figure 41: User interface "User management"

■ Add user

The screenshot shows the 'User management' section with a 'New User' dialog box overlaid. The dialog box has the following fields:

- Email: achref@gmail.com
- Role: Dev

At the bottom of the dialog box are 'CANCEL' and 'SAVE' buttons.

The background shows the same sidebar and user list as Figure 41.

Figure 42: Test "Add user"

The screenshot shows a user management interface. On the left is a sidebar with a user profile (manelkacem12, Owner) and various management options: Dashboard, Profile Management, Activity Management, Calendar, Role Management, User Management (selected), Contact Management, and Deals Management. At the bottom of the sidebar is a Logout button. The main area is titled "User management" and displays a table of users. A modal window at the top right says "User added!" with a green checkmark icon. The table has columns for Email, Role, Profile (with a "PROFILE" button), and Actions (with edit and delete icons). The data in the table is:

Email	Role	Profile	Actions
achref@gmail.com	Dev	PROFILE	
manelkacem12@gmail.com	Owner	PROFILE	
mariem@gmail.com	Sales	PROFILE	

At the bottom center of the main area is a page number "1".

Figure 43: Test user added successfully

Conclusion

In this chapter, we looked at the second sprint of our project. For this, we have completed the specification, analysis, conception, and testing. In the next chapter, we will focus on our third sprint.

Chapter 5 : Study and execution of Sprint 3

Introduction

This chapter includes four main parts. The first part contains the sprint backlog. The second part is dedicated for the analysis, in which we will identify the use case diagram and the description of the related scenarios. In the third part will focus on the conception, mainly to identify the sequence diagrams. Finally, we will introduce the testing of each feature tackled in the third sprint.

1. Sprint Backlog

In this sprint, our goal is to satisfy the following user stories described in the table below.

Table 21: Sprint3 Backlog

Feature name	User Story	Difficulty	Estimation
Contact management	As a user , if I have the right role for it, I want to be able to manage the contacts.	Medium	6
Deals management	As a user , if I have the right role for it, I want to be able to manage the deals.	Medium	6
Profile management	As a user , if I have the right role for it, I want to be able to manage the profile.	Medium	4
View Dashboard	As a user , I want to be able to view the dashboard.	Medium	6
View Calendar	As a user , I want to be able to view the calendar.	Easy	2

2. Analysis

The use-case diagrams model the behavior of a system and help to capture the requirements of the system. These diagrams help identify the interactions between the system and its actors.

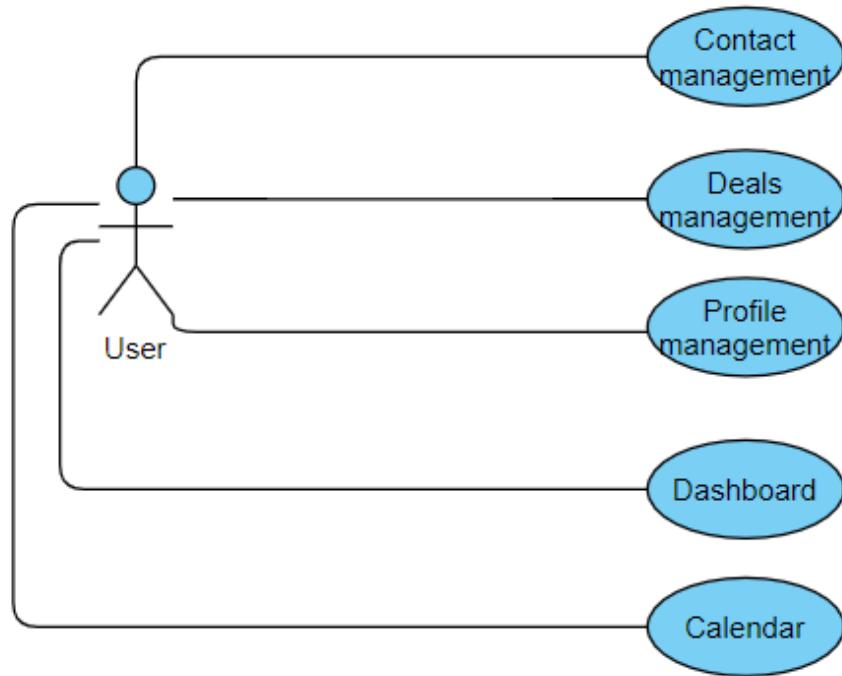


Figure 44: Use case diagram Sprint3

2.1 Use case diagrams

2.1.1 Use case « Contact management »

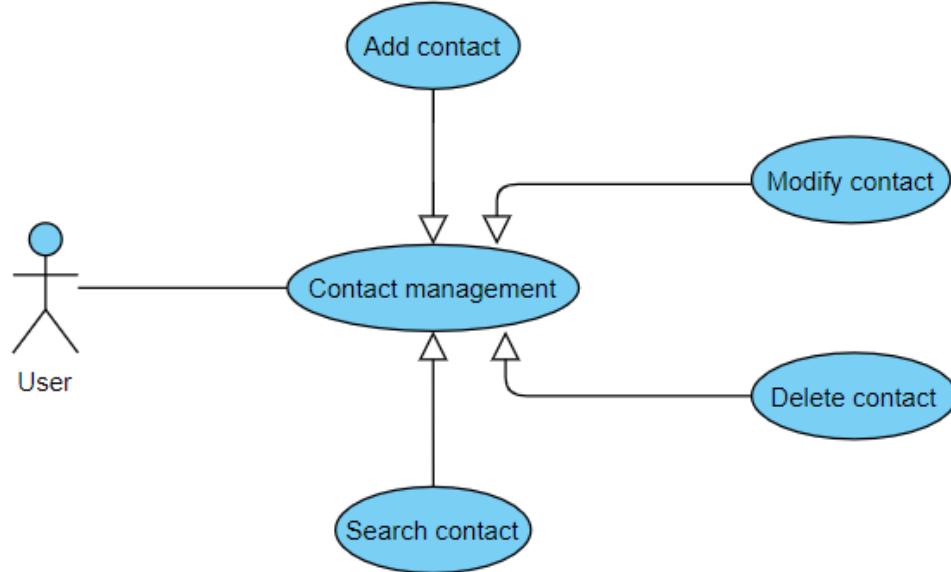


Figure 45: Use case "Contact management"

Table 22: Use case description " Add contact"

Name	Add contact
Actor	User
Goal	The user wants to add a new contact
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User clicks on contact management found in the navbar; 2. The system shows the page of contact management; 3. The user clicks on the “+ Add contact” button; 4. The user fills the form; 5. The user clicks on the save button; 6. The system validates the fields; 7. The system saves the new contact; 8. The system displays a success message.
Alternative scenario	5.a The user clicks on the cancel button. 6.a. Invalid fields

Table 23: Modify activity

Name	Modify contact
Actor	User
Goal	The user wants to modify a contact
Pre-conditions	The user must be authenticated and have the right role.

Nominal scenario	<ol style="list-style-type: none"> 1. User clicks on contact management found in the navbar; 2. The system shows the page of contact management; 3. The user clicks on the edit button; 4. The user fills the form; 5. The user clicks on the save button; 6. The system displays a success message.
Alternative scenario	4.a Invalid fields 5.a The user clicks on the cancel button

Table 24: Use case description "Delete activity"

Title	Delete contact
Actor	User
Goal	The user wants to delete a contact
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User clicks on contact management found in the navbar; 2. The system shows the page of contact management; 3. The user clicks on the delete button; 4. The system displays an alert to check the choice of the user; 5. The user clicks the confirm button; 6. The system displays a success message.
Alternative scenario	4.a The user clicks on the cancel button

2.1.2 Use case « Deals management »

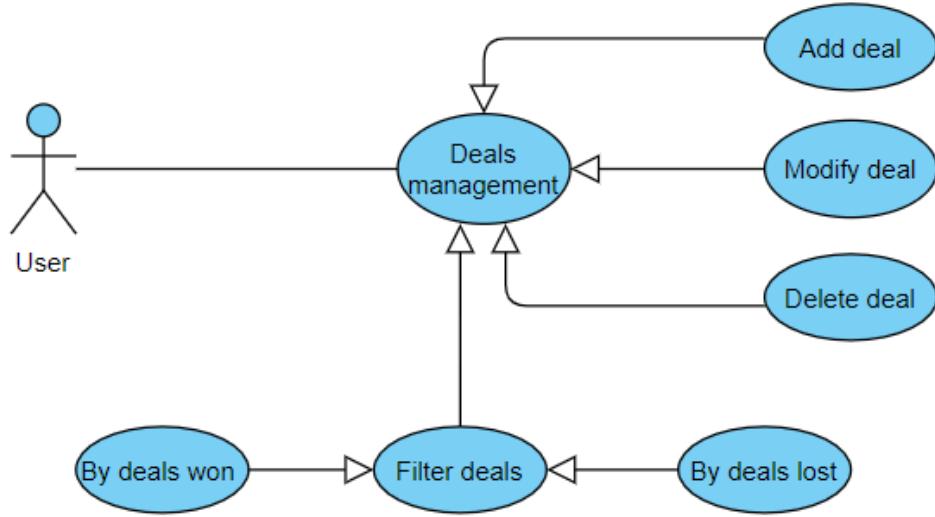


Figure 46: Use case description “Deal management”

Table 25: Use case description “Add deal”

Title	Add deal
Actor	User
Goal	The user wants to add a deal
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User select deals management found in the navbar; 2. The system shows the page deals management; 3. The user clicks on the add deal button; 4. The user fills in the form; 5. The user clicks the save button; 6. The system adds the new deal; 7. The system displays a success message.

Alternative scenario	5.a The user clicks the cancel button;
----------------------	--

Table 26: Use case description “Modify deal”

Title	Modify deal
Actor	User
Goal	The user wants to modify a deal
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User select deal management found in the navbar; 2. The system shows the page deal management; 3. The user clicks on the specific deal to modify; 4. The user fills the form; 5. The user clicks the save button; 6. The system saves the changes; 7. The system displays a success message.
Alternative scenario	5.a The user clicks on the cancel button;

Table 27: Use case description “Delete deal”

Title	Delete deal
Actor	User
Goal	The user wants to delete a deal
Pre-conditions	The user must be authenticated and have the right role.

Nominal scenario	<ol style="list-style-type: none"> 1. User select deal management found in the navbar; 2. The system shows the page deal management; 3. The user clicks on the deal to delete; 4. The user clicks on the “Trash icon” to delete the specific deal; 5. The system shows a confirmation window; 6. The user clicks on “Ok” button to confirm the delete; 7. The system displays a success message.
Alternative scenario	5.a The user clicks on the “Cancel” button

2.1.3 Use case « Profile management »

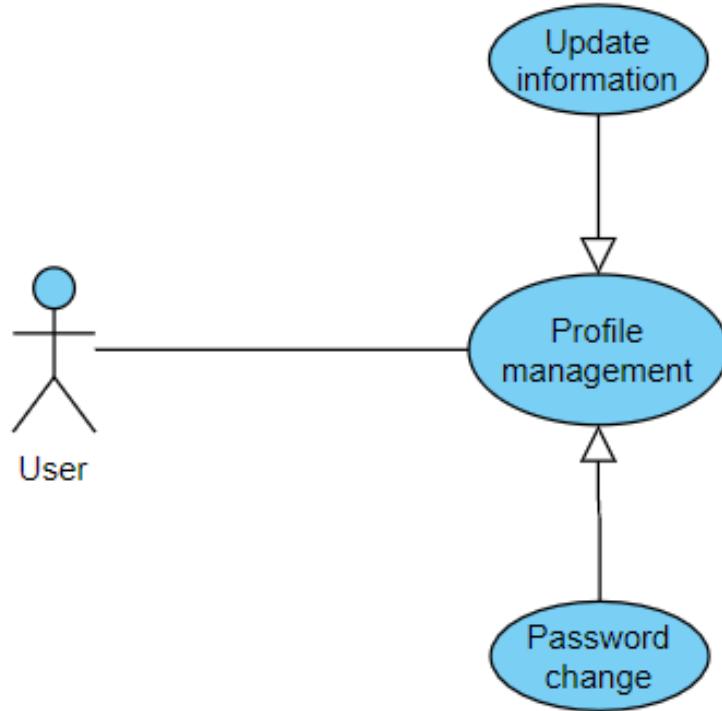


Figure 47: Use case "Profile management"

Table 28: Use case description "Update information"

Title	Update information
Actor	User
Goal	The user wants to update their data
Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none"> 1. User select profile management found in the navbar; 2. The system shows the interface profile management; 3. The user clicks the button “Edit”; 4. The system shows the “Edit profile” dialog; 5. The user enters the current password; 6. The user clicks on the “Save” button; 7. The system validates the password field; 8. The user edits the information; 9. The user clicks the save button; 10. The system displays a success message.
Alternative scenario	<p>6.a The user clicks the cancel button;</p> <p>7.a. The password entered is incorrect.</p>

Table 29: Use case description "Password change"

Title	Password change
Actor	User
Goal	The user wants to change their password

Pre-conditions	The user must be authenticated and have the right role.
Nominal scenario	<ol style="list-style-type: none">1. User select profile management found in the navbar;2. The system shows the interface profile management;3. The user clicks the change password button;4. The user enters both the current password and new password;5. The user clicks the save button;6. The system checks the fields;7. The system saves the changes;8. The system displays a success message.
Alternative scenario	<p>4.a The current password is incorrect;</p> <p>5.a The user clicks the cancel button.</p>

4. Conception

4.1 Sequence diagrams

4.1.1 Sequence diagram “Add Contact”

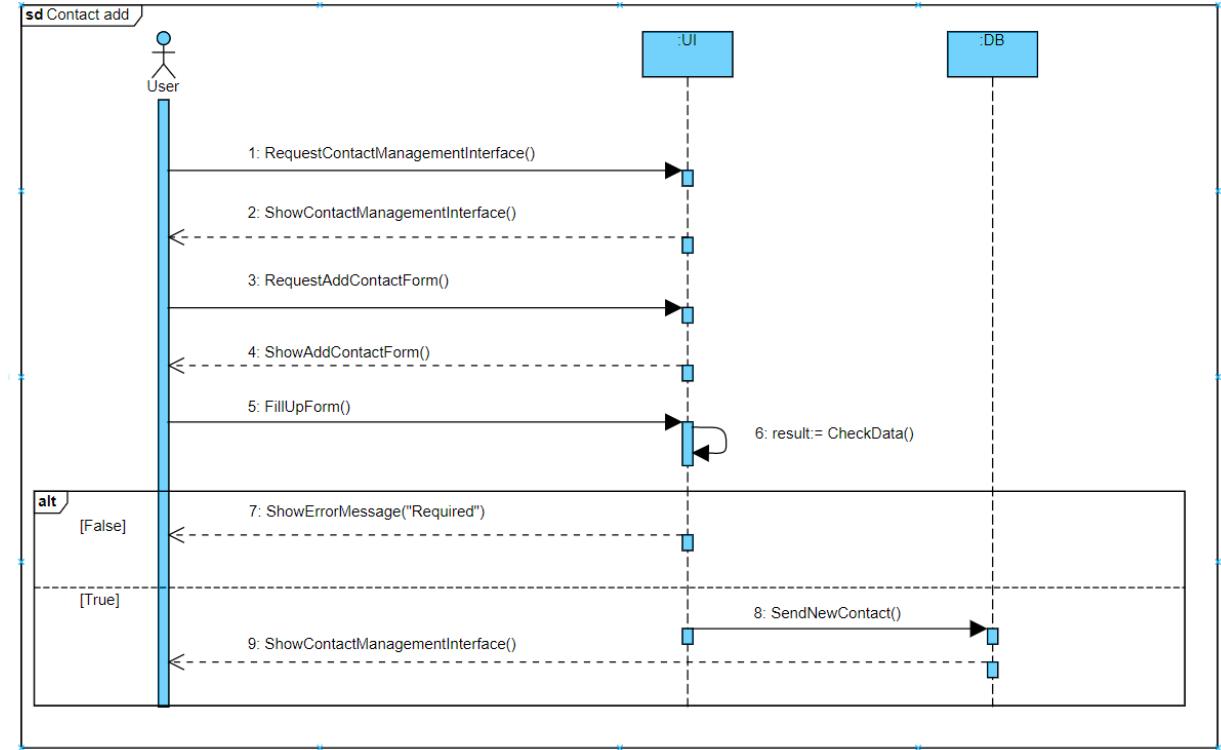


Figure 48: Sequence diagram "Add contact"

4.1.2 Sequence diagram “Modify Contact”

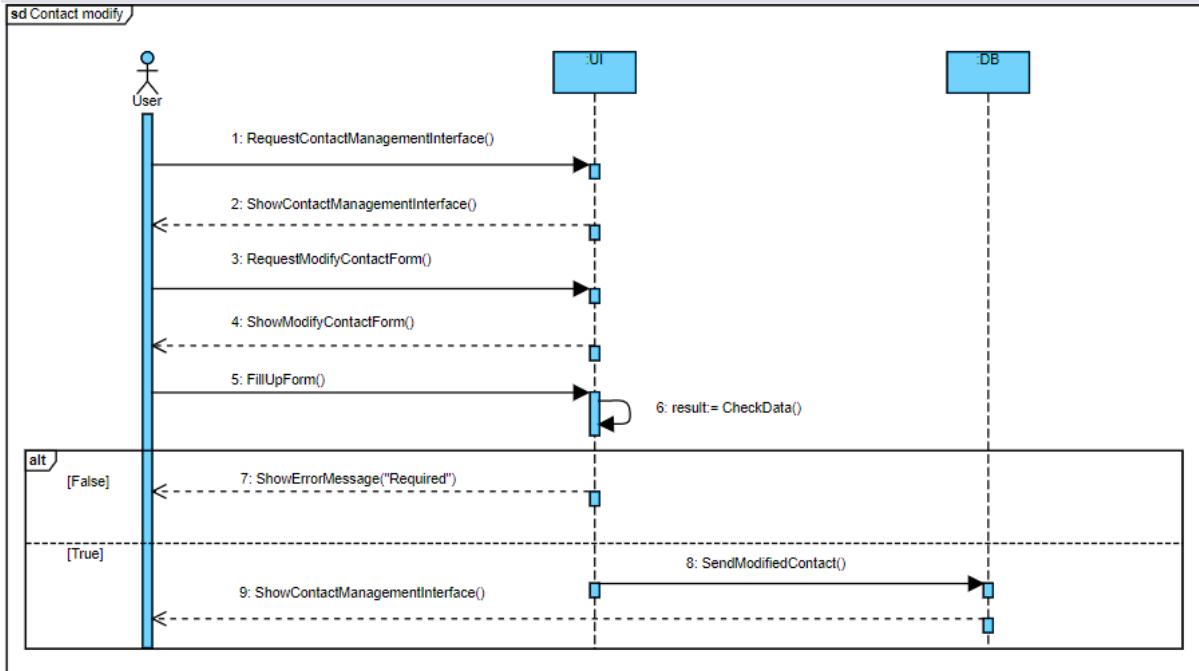


Figure 49: Sequence diagram "Modify contact"

4.1.3 Sequence diagram “Delete Contact”

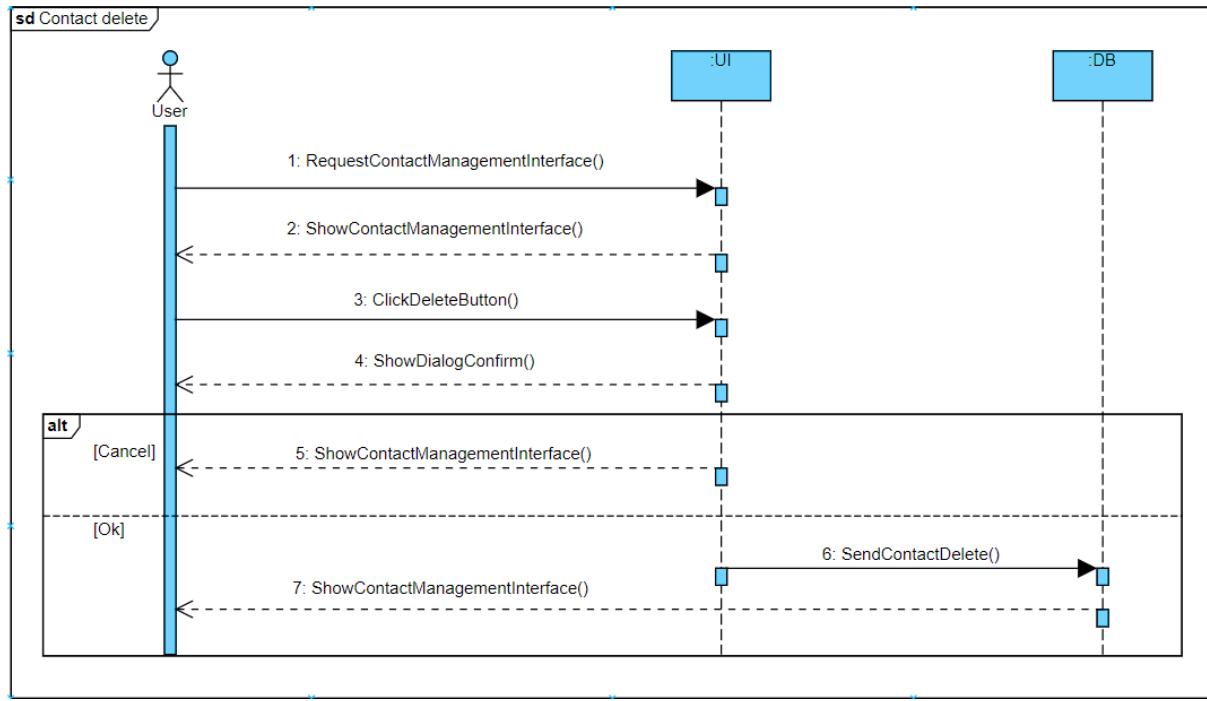


Figure 50: Sequence diagram "Delete contact"

4.1.4 Sequence diagram “Add deal”

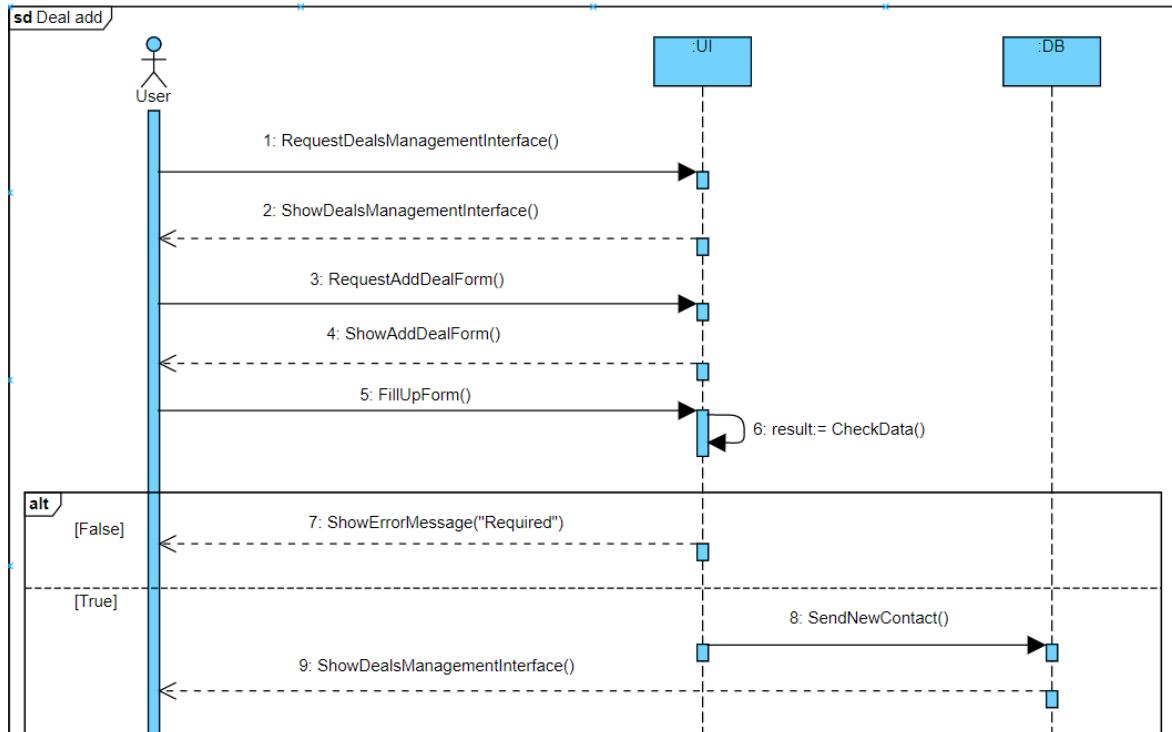


Figure 51: Sequence diagram “Add deal”

4.1.5 Sequence diagram “Modify deal”

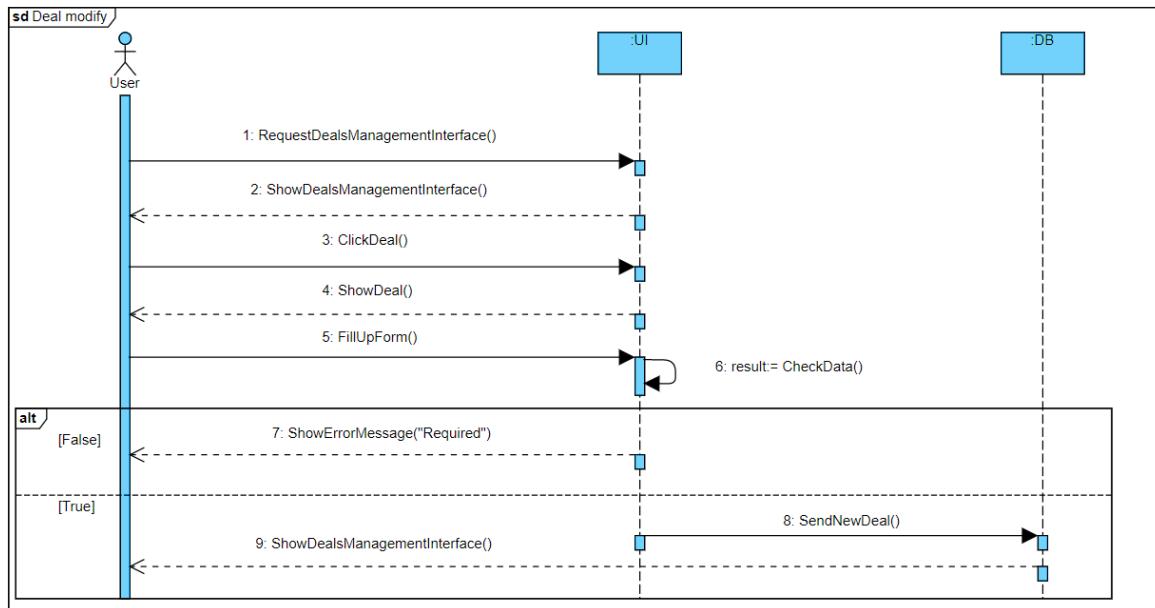


Figure 52: Sequence diagram “Modify deal”

4.1.6 Sequence diagram “Delete deal”

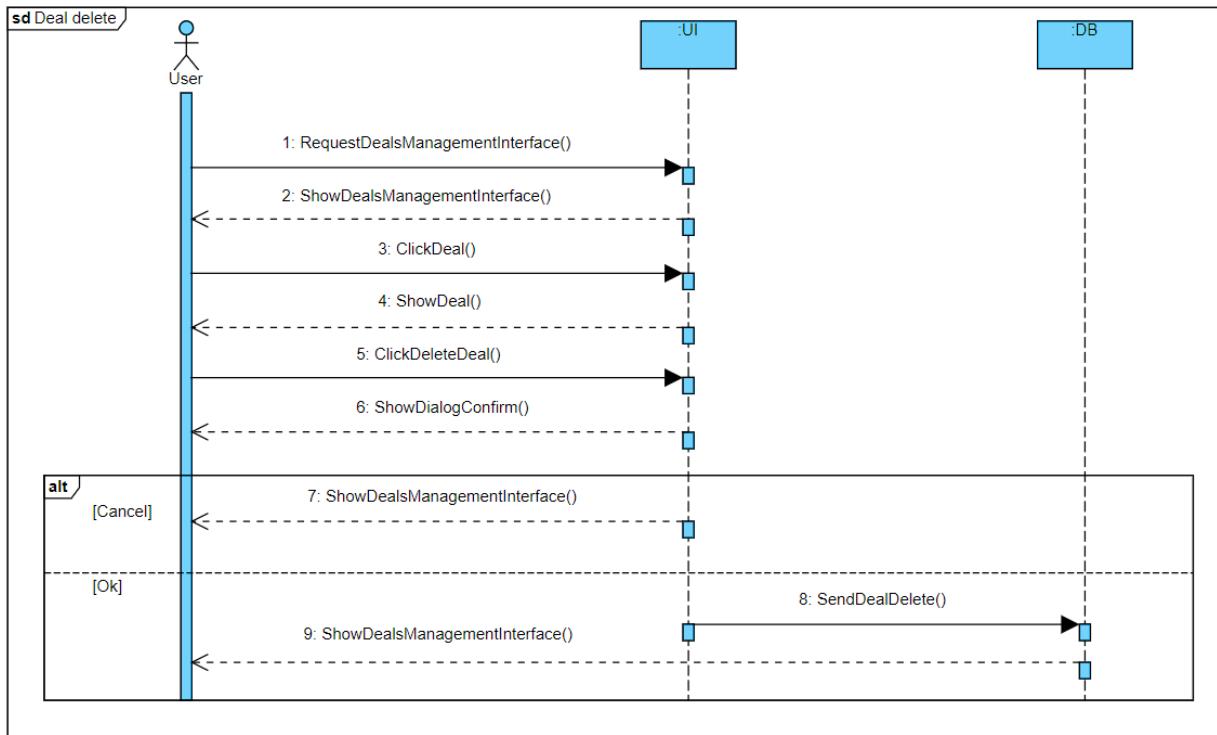


Figure 53: Sequence diagram “Delete deal”

4.1.7 Sequence diagram “Update information”

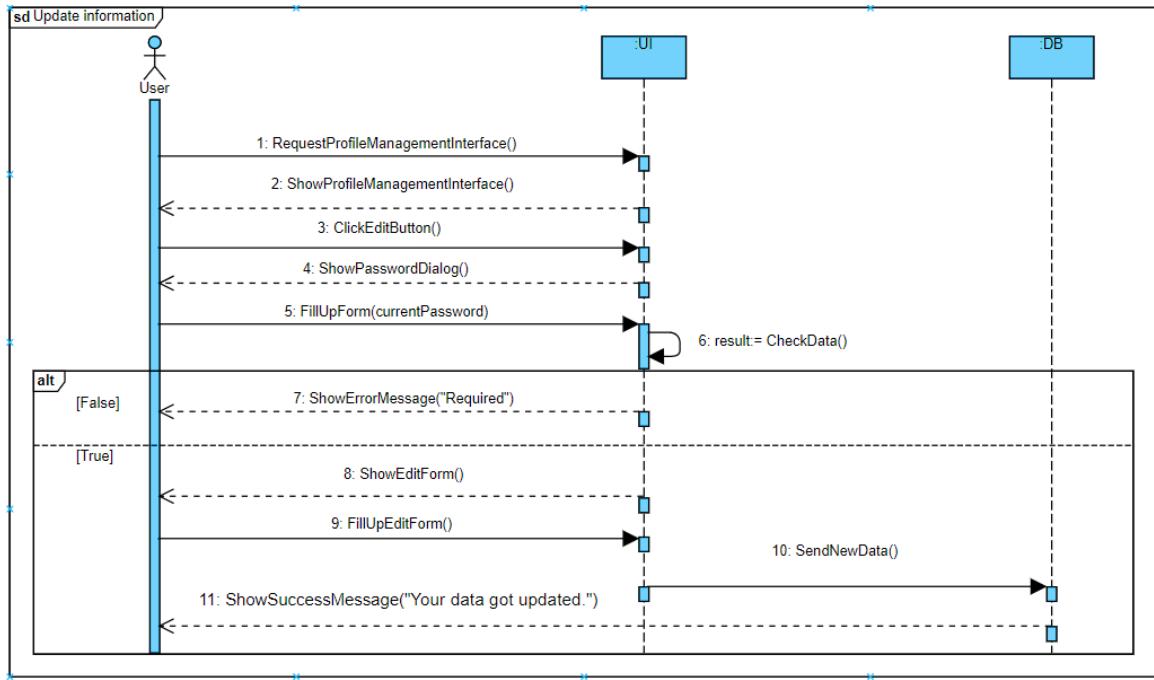


Figure 54: Sequence diagram “Update information”

4.1.8 Sequence diagram “Change password”

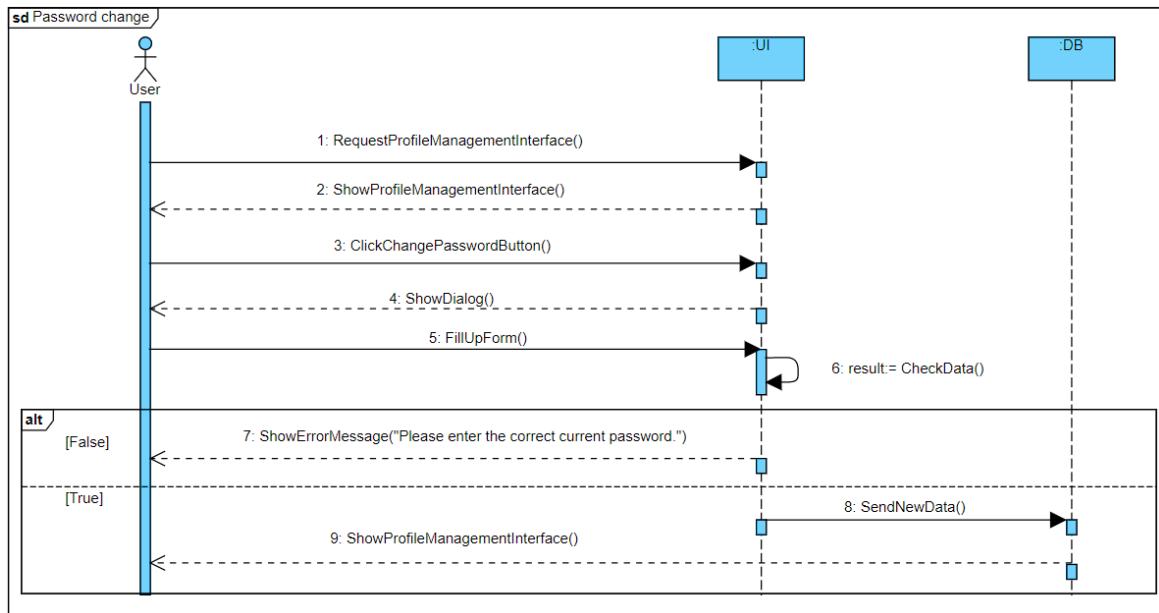


Figure 55: Sequence diagram “Change password”

5. Realization and testing

5.1 Testing execution and results

5.1.1 Testing Contact management

The screenshot shows a user interface for managing contacts. On the left is a sidebar with a user profile (manekacem12, Owner) and a navigation menu including Dashboard, Profile Management, Activity Management, Calendar, Role Management, User Management, Contact Management (which is selected and highlighted in grey), and Deals Management. At the bottom of the sidebar is a green 'LOGOUT' button. The main area is titled 'Contacts' and shows a single contact entry for 'Hadil Ben' (hadil@gmail.com). The contact details are: Type: Client, Phone Number: 55 555 555, Address: Tunisia. Below the contact card are 'EDIT' and 'DELETE' buttons. A search bar and a '+ ADD CONTACT' button are at the top right. At the bottom right, it says 'Page 1 of 1' with navigation arrows. The footer has 'Items per page' set to 8.

Figure 56: User Interface "Contact management"

■ Add contact

The screenshot shows the same user interface as Figure 56, but with a modal window open over the 'Contacts' list. The modal is titled 'New Contact' and contains fields for entering contact information: Contact Type (dropdown), Email (input field with a mail icon), Address (input field with a location pin icon), Phone Number (dropdown with a red phone icon and an input field 'Enter a Phone N...'), Gender (dropdown with a female icon), Date of birth (input field with a calendar icon), Country of residence (input field with a flag icon), Time Zone (input field with a clock icon), and Nationality (input field with a globe icon). At the bottom of the modal are 'CANCEL' and 'SAVE' buttons. The background shows the same contact list as Figure 56, with the 'Contact Management' item still selected in the sidebar.

Figure 57: Test "Add contact"

The screenshot shows a user interface for managing contacts. On the left is a sidebar with navigation links: Dashboard, Profile Management, Activity Management, Calendar, Role Management, User Management, Contact Management (which is selected and highlighted in grey), and Deals Management. The main area is titled "Contacts". It displays two contact entries in cards:

- Adam Brown**: Community Leader, Adam@gmail.com, Type: Community Leader, Phone Number: (437) 886-2841, Address: Canada. Buttons: EDIT, DELETE.
- Hadil Ben aaa**: Client, hadil@gmail.com, Type: Client, Phone Number: 55 555 555, Address: Tunisia. Buttons: EDIT, DELETE.

A success message "Contact added!" is visible at the top right. A search bar and a "+ ADD CONTACT" button are also present. At the bottom, there are pagination controls: "Page 1 of 1" and arrows for navigating through pages.

Figure 58: Test add contact successfully

5.1.2 Testing Deals management

The screenshot shows a user interface for managing deals. The sidebar on the left includes: Dashboard, Profile Management, Activity Management, Calendar, Role Management, User Management, Contact Management, and Deals Management (selected). The main area is titled "Deals" and displays the following information:

- New Deal**: 6000 TND - 2 Offers
- Contacted**: 0 TND - 0 Offers
- Qualified**: 0 TND - 0 Offers
- Negotiation**: 0 TND - 0 Offers

Below these statistics are two deal cards:

- Trip to Turkey**: 2021-06-22 - 2021-06-21, 1000 TND
- Deal 1**: 2021-06-12 - 2021-06-12, 5000 TND

A "LOGOUT" button is located at the bottom left.

Figure 59: User interface "Deals management"

Chapter 5: Study and execution of Sprint 3

The screenshot shows a user interface for managing deals. On the left is a sidebar with navigation links: Dashboard, Profile Management, Activity Management, Calendar, Role Management, User Management, Contact Management, and Deals Management (which is currently selected). The main area is titled 'Deals' and contains four colored boxes: 'New Deal' (blue), 'Contacted' (orange), 'Qualified' (green), and 'Negociation' (red). Below these are two deal cards: 'Trip to Turkey' (blue card) and 'Deal 1' (red card). A modal window titled 'Deals won' is open, showing two deals: 'deal1' (Value: 5000 TND) and 'Deal with community leader Adam' (Value: 4000 TND). A 'CLOSE' button is at the bottom right of the modal. At the top right of the main area are three buttons: 'DEALS LOST' (red), 'DEALS WON' (green), and 'DEAL' (blue).

Figure 60: User interface "Deals won"

This screenshot shows the same user interface as Figure 60, but with a success message: 'Deal modified!' with a green checkmark icon, and a 'CLOSE' button above the deal cards. The deal cards are identical to Figure 60: 'Trip to Turkey' (blue) and 'Deal 1' (red). The sidebar and main navigation are also the same.

Figure 61: Test modify deal successfully

5.1.3 Testing Profile management

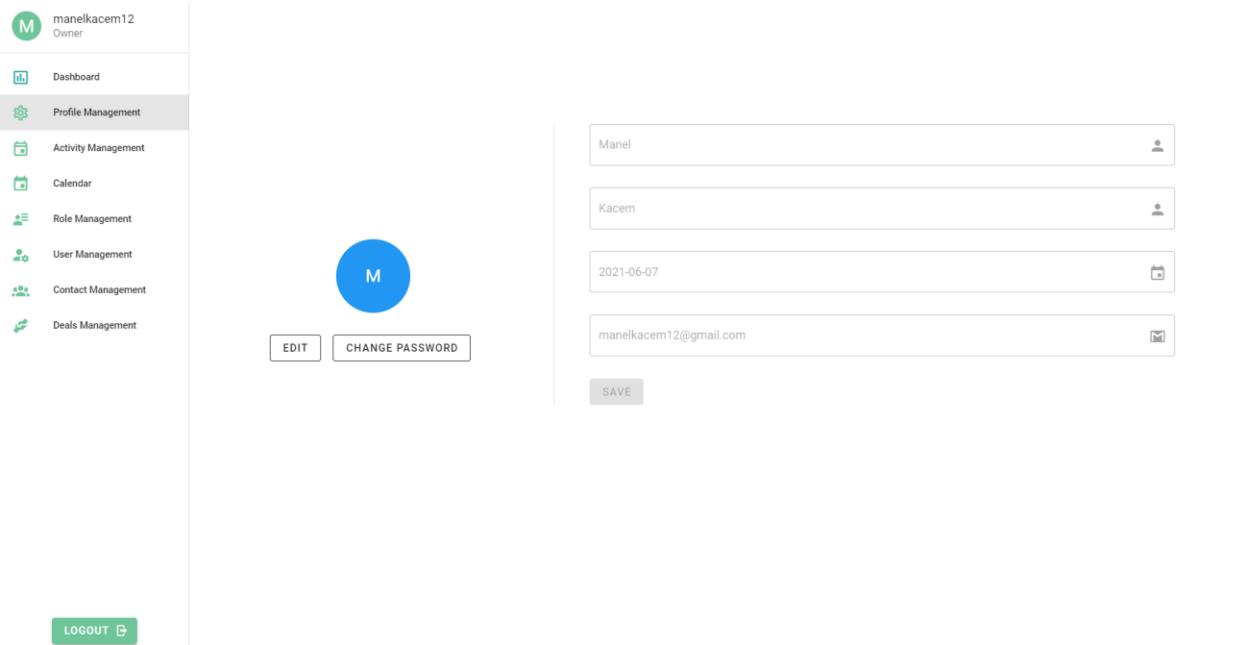


Figure 62: User interface "Profile management"

- **Edit profile**

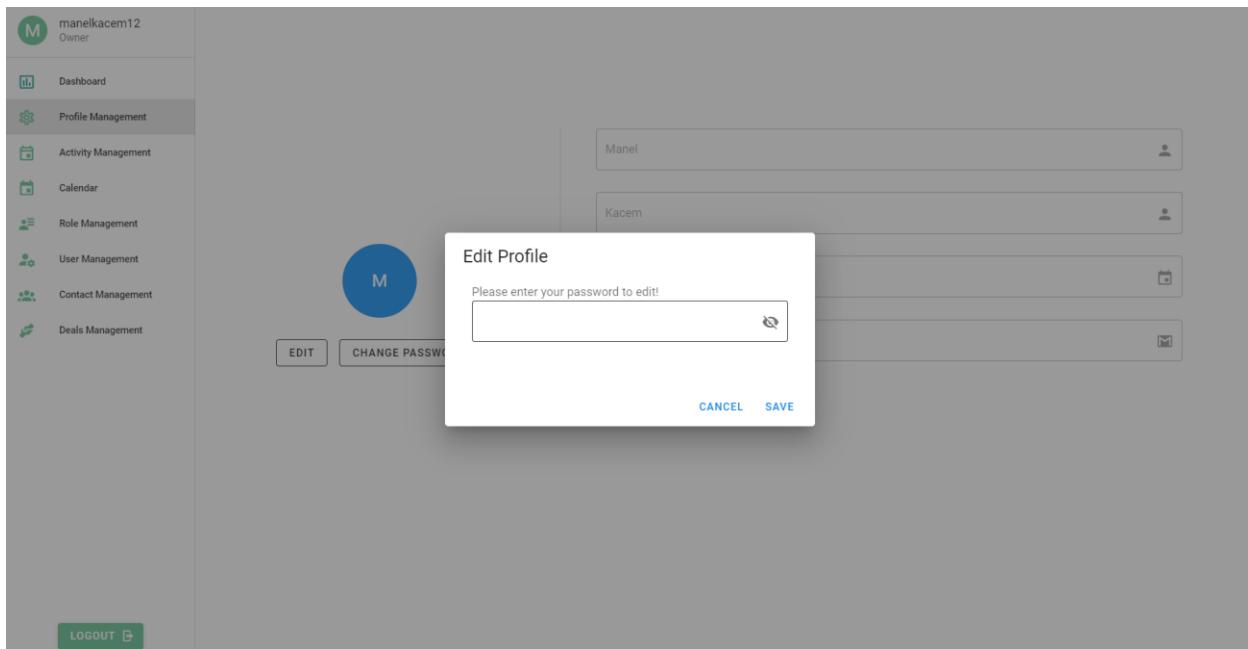


Figure 63: User interface "Edit profile"

5.1.4 Testing View dashboard

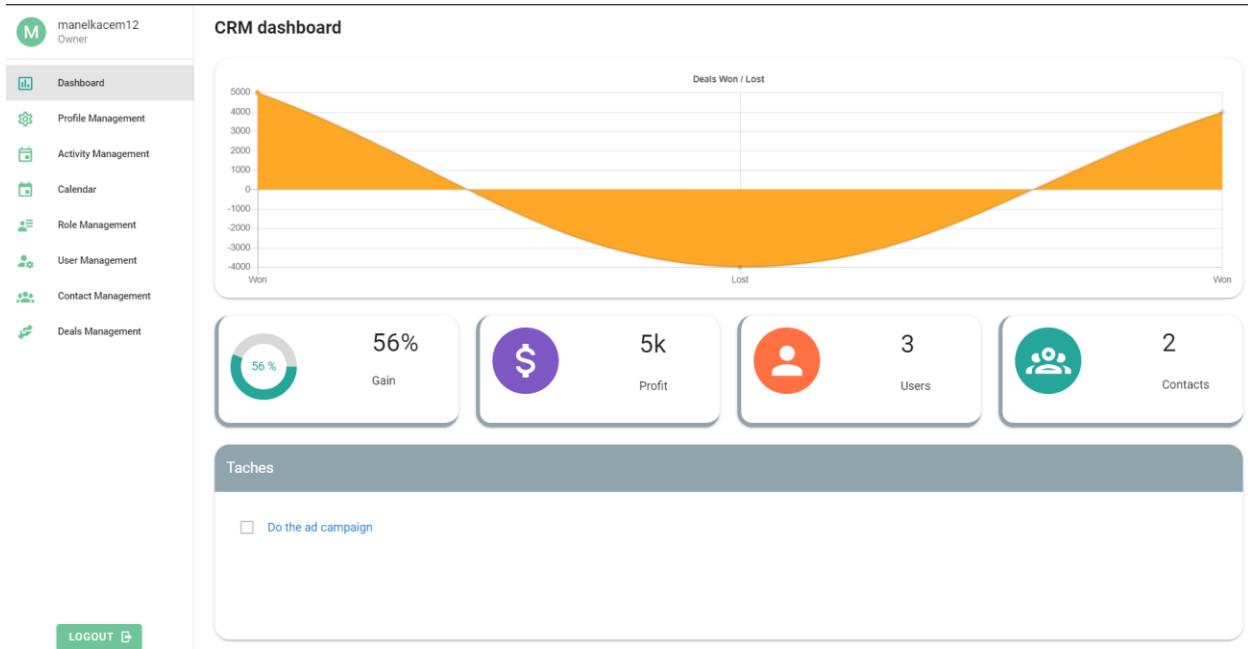


Figure 64: User interface "Dashboard"

5.1.5 Testing Calendar

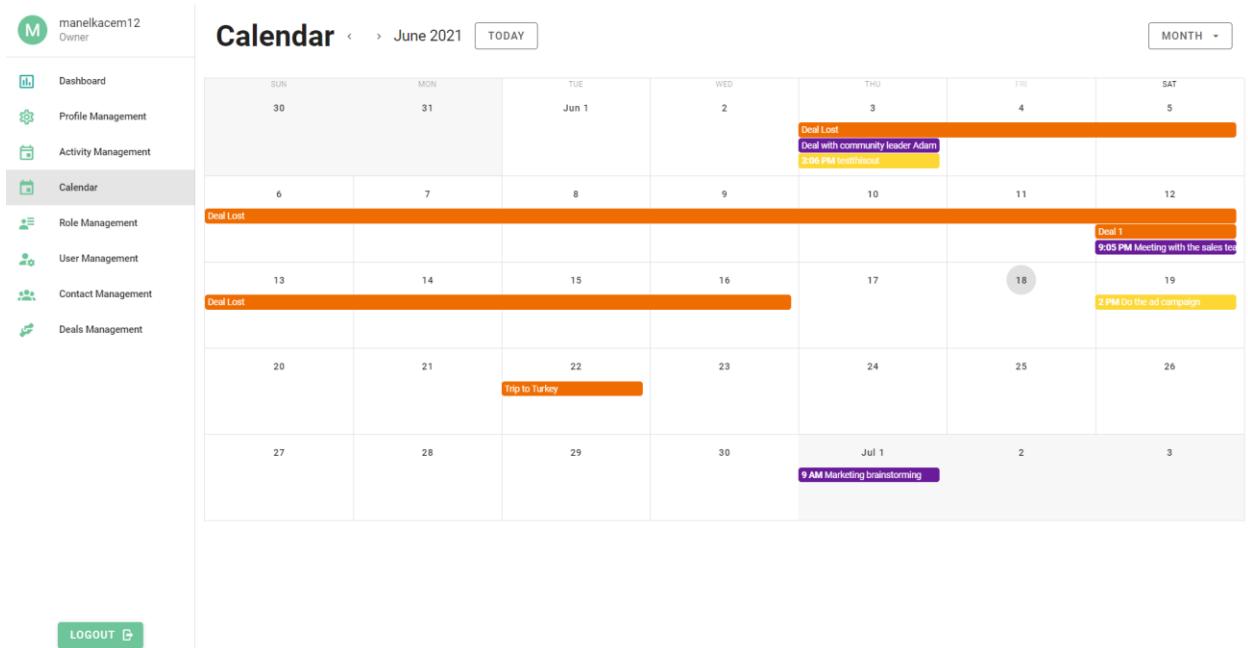


Figure 65: User interface "Calendar"

Conclusion

In this chapter, we analyzed the third sprint of our project. To this end, we have completed the specification, analysis, conception, and testing. In the next chapter, we will focus on our fourth sprint.

Chapter 6 : Study and execution of Sprint 4

Introduction

This chapter mainly focuses on the last sprint of our project which includes two main parts. The first part contains the list to tasks also known as the sprint backlog. The second part is dedicated to the testing of each feature to showcase the improvements made and realization of the fourth sprint.

1. Sprint backlog

In this sprint, our goal is to satisfy the following user stories described in the table below:

Table 30: Sprint 4 Backlog

Feature name	User Story	Difficulty	Estimation
Improve the authentication process interfaces	As a user, I want to improve the authentication process interfaces	Easy	4
Improve the booking process interfaces	As a user, I want to improve the booking process interfaces	Easy	5
Improve the trips and events interfaces	As a user, I want to improve the previous trips and events process interfaces	Easy	4
Improve the details trip interface	As a user, I want to improve the details trip interface	Easy	4
Improve the host trip process interfaces	As a user, I want to improve the host trip process interfaces	Easy	4

Improve the navbar	As a user, I want to improve the Navbar	Easy	2
--------------------	---	------	---

2. Realization and testing

2.1 Testing execution and results

2.1.1 Testing the authentication interfaces

- **Before the improvements**

The authentication is an important process for the startup WanToTrip since their aim is to have more clients, community leaders and travel agencies with accounts. In addition, it allows them to identify more the users and their identity using the data entered to connect with them and to give them a better service. Due to lack of ergonomic in the previous interfaces (as shown in the figure 66 and 67), not many of the visitors convert into users with accounts. The goal from the new improvements is to make better interfaces so that visitors turn into clients and maximize efficiency.

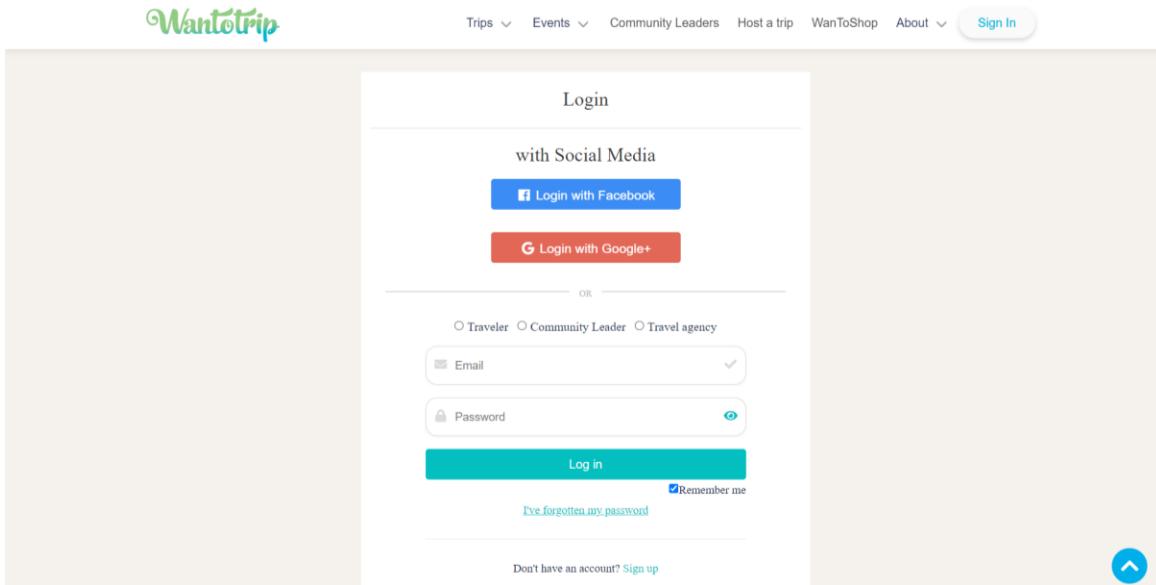


Figure 66: The login interface before the improvements

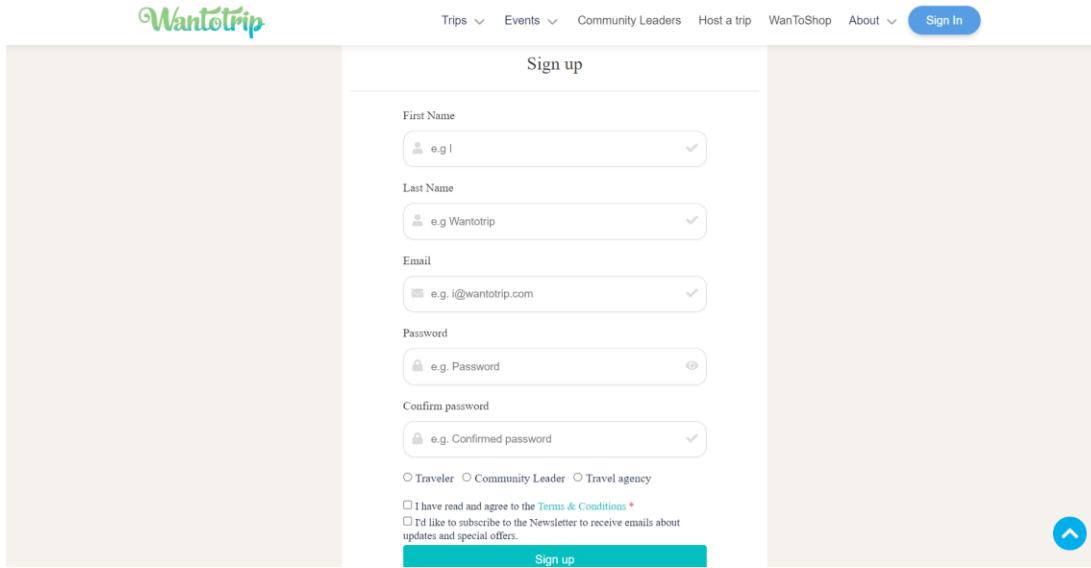


Figure 67: The sign-up interface before the improvements

▪ **After the improvements**

The following figures (68 and 69) presents both the login interface and sign-up interface after the improvements made. The new user interface presents a better visual design by implementing new elements such as fonts, colors, etc. As a result it made the pages more elegant without compromising on its function nor content.

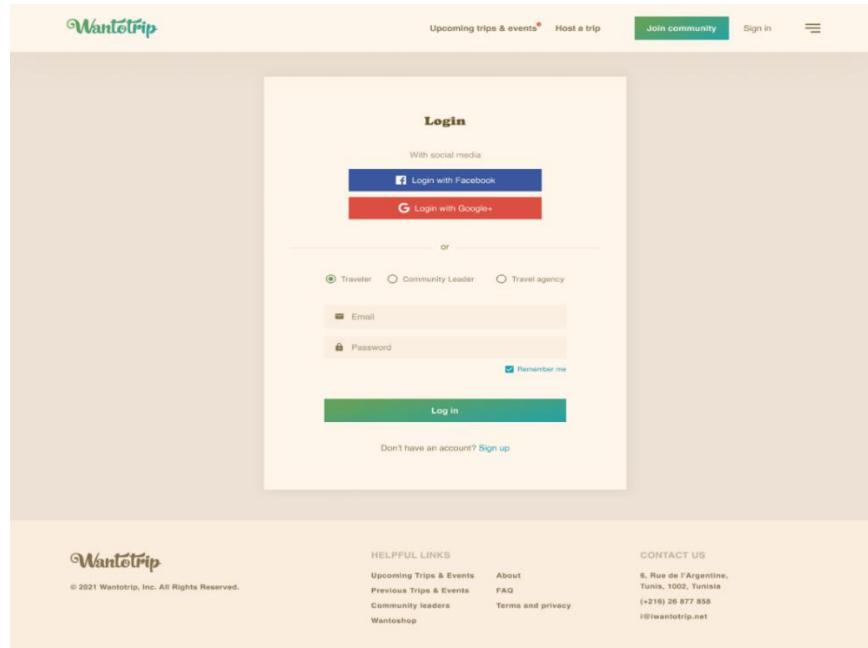


Figure 68: The login interface after the improvements

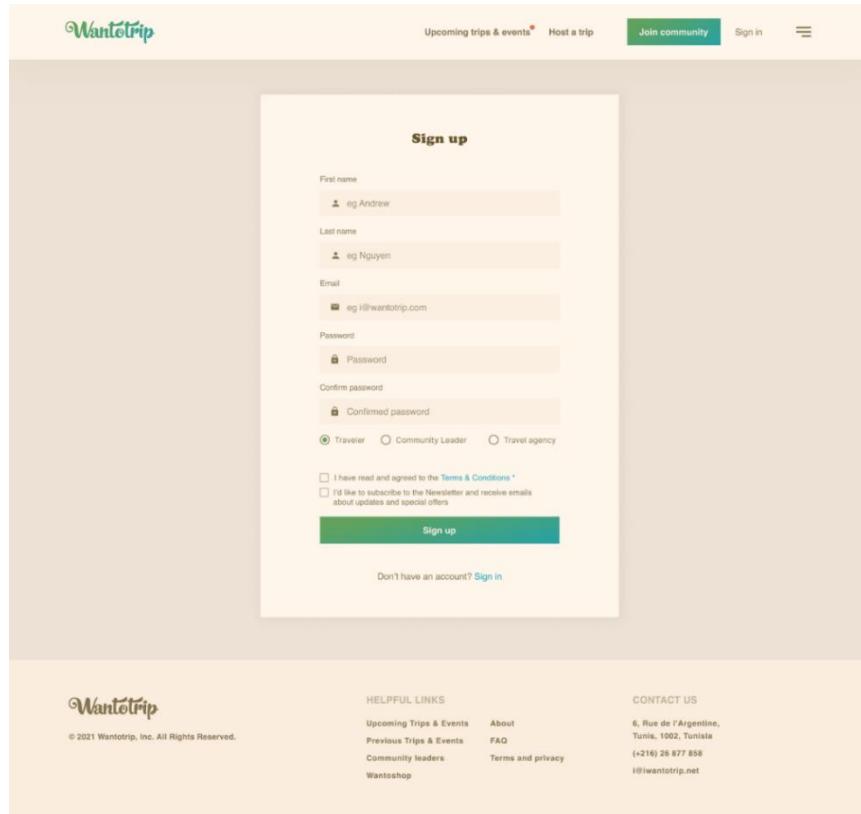


Figure 69: The sign-up interface after the improvements

2.1.2 Testing the booking process interfaces

- **Before the improvements**

As a travel agency the booking process is one important step to generate revenue. However, the existing user interfaces (as shown in figure 70 and 71) are not facilitating the user experience toward booking a trip. So as to maximize the number of clients and increase the sales we decided to make changes on the existing interfaces. The goal was to make a more interactive design which allows the user to understand each step in the booking process.

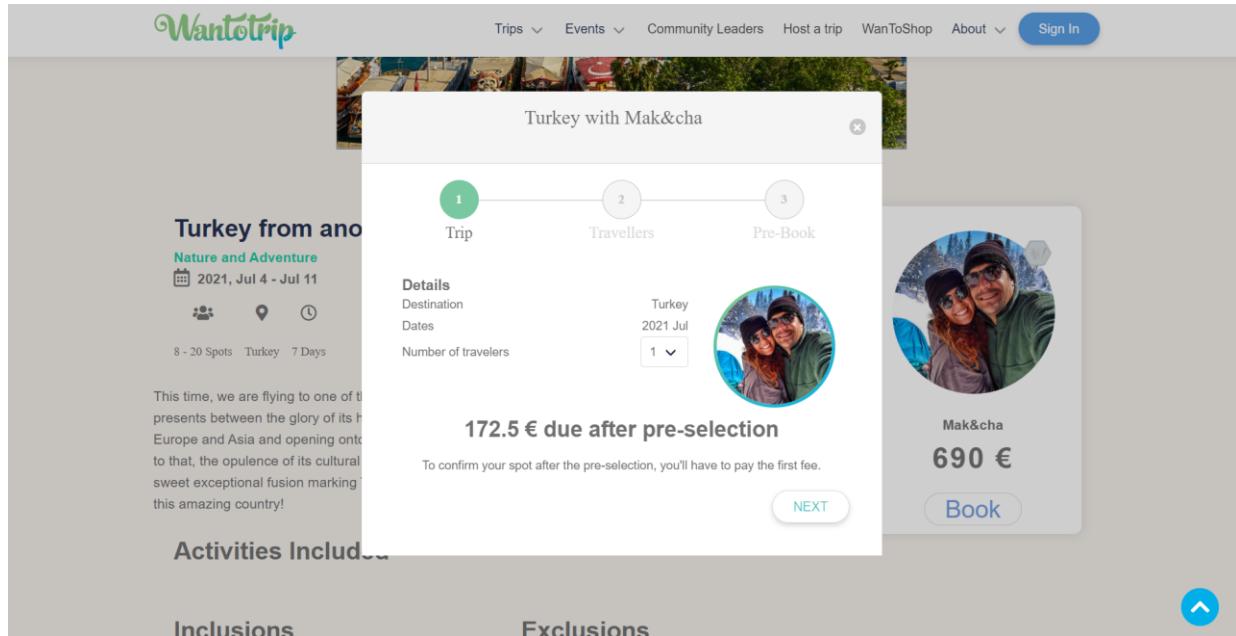


Figure 70: The booking interface 1 before the improvements

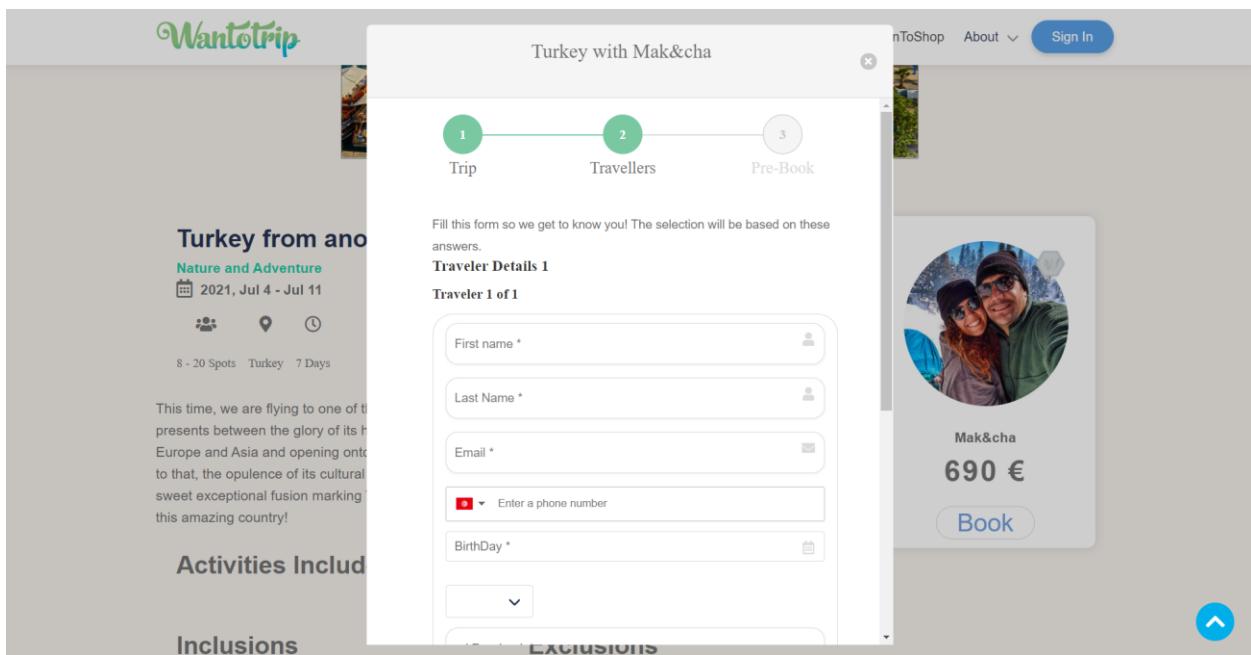


Figure 71: The booking interface 2 before the improvements

■ ***After the improvements***

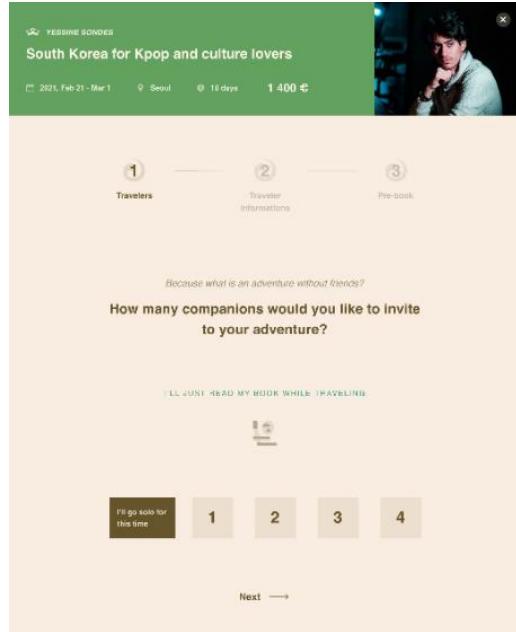


Figure 72: The first booking interface after the improvements

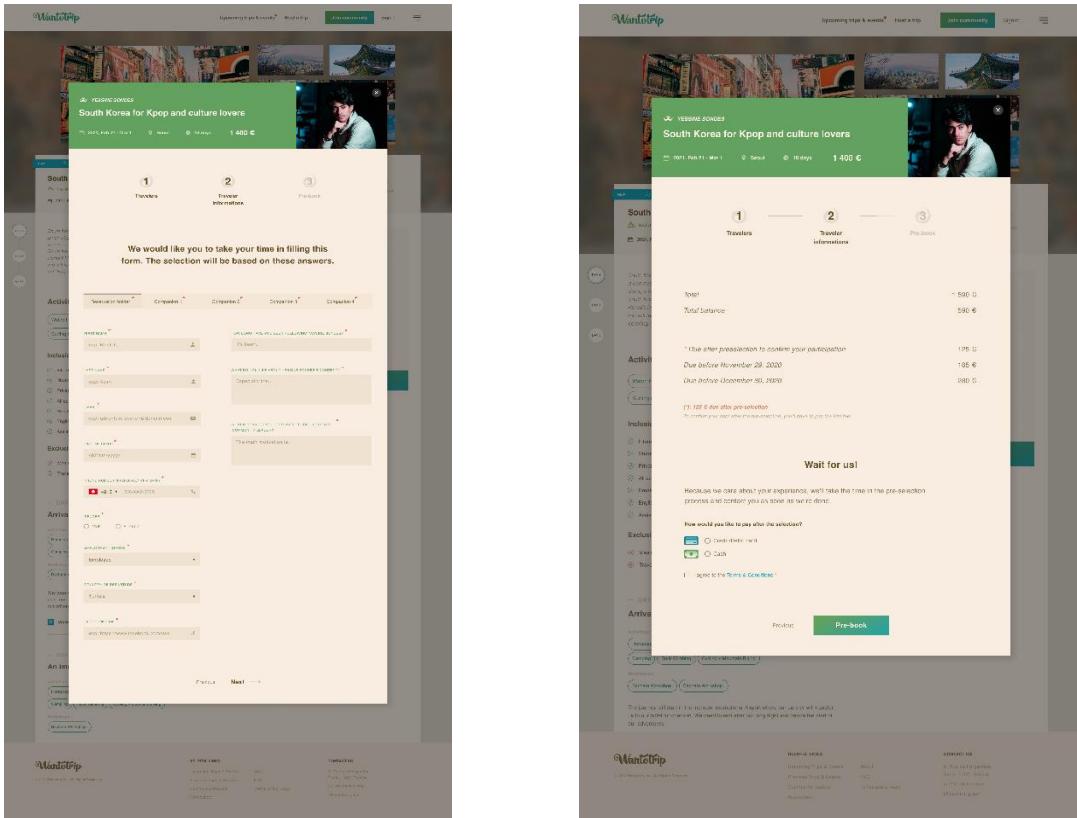


Figure 73: The booking interfaces after the improvements

2.1.3 Testing the trips and events interface

- **Before the improvements**

The following user interface highlights events and trips conducted by WanToTrip, so that visitors can better understand the services provided. Figures 74 and 75 below show the user interfaces before the upgrade.

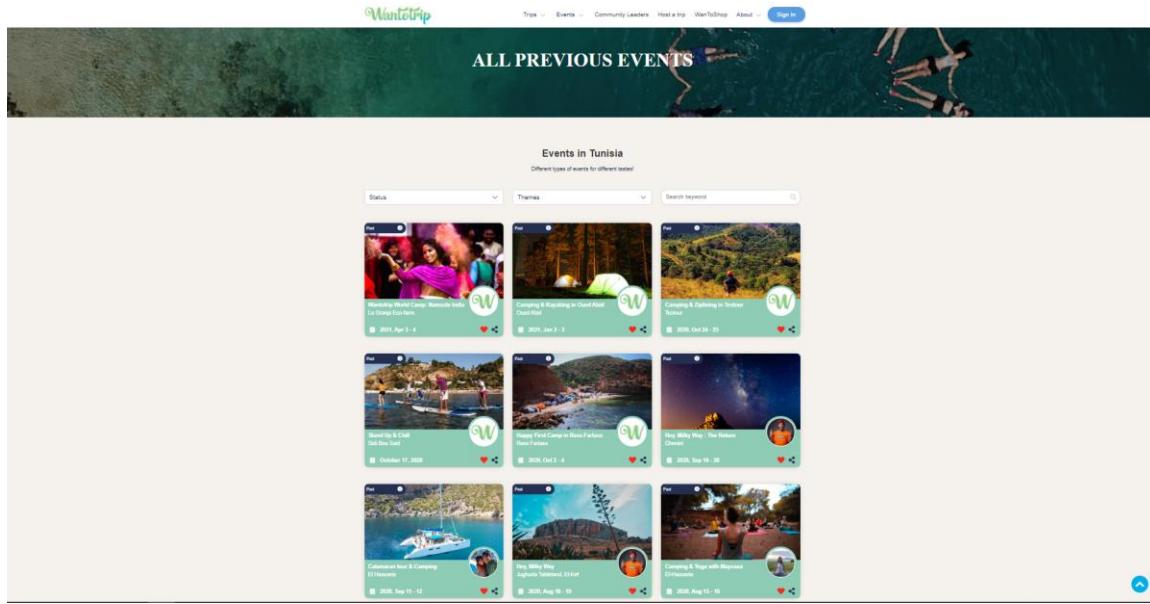


Figure 74: The previous events interface before the improvements

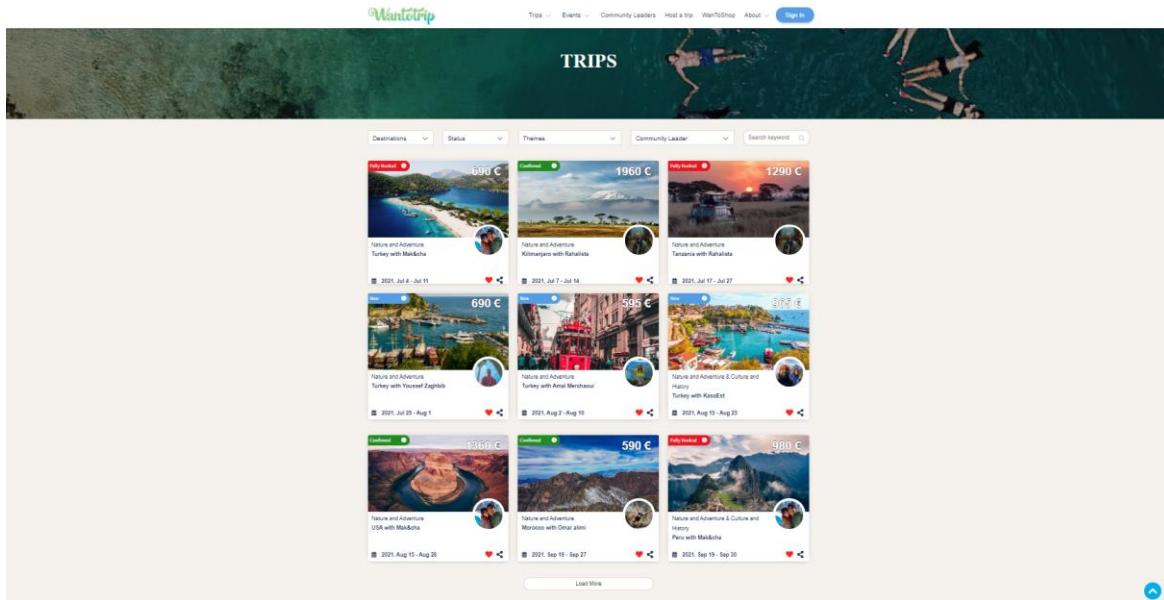


Figure 75: The trips interface before the improvements

■ **After the improvements**

The purpose from improving these interfaces is to allow visitors to stay longer on the page, view more events, travel, and interact.

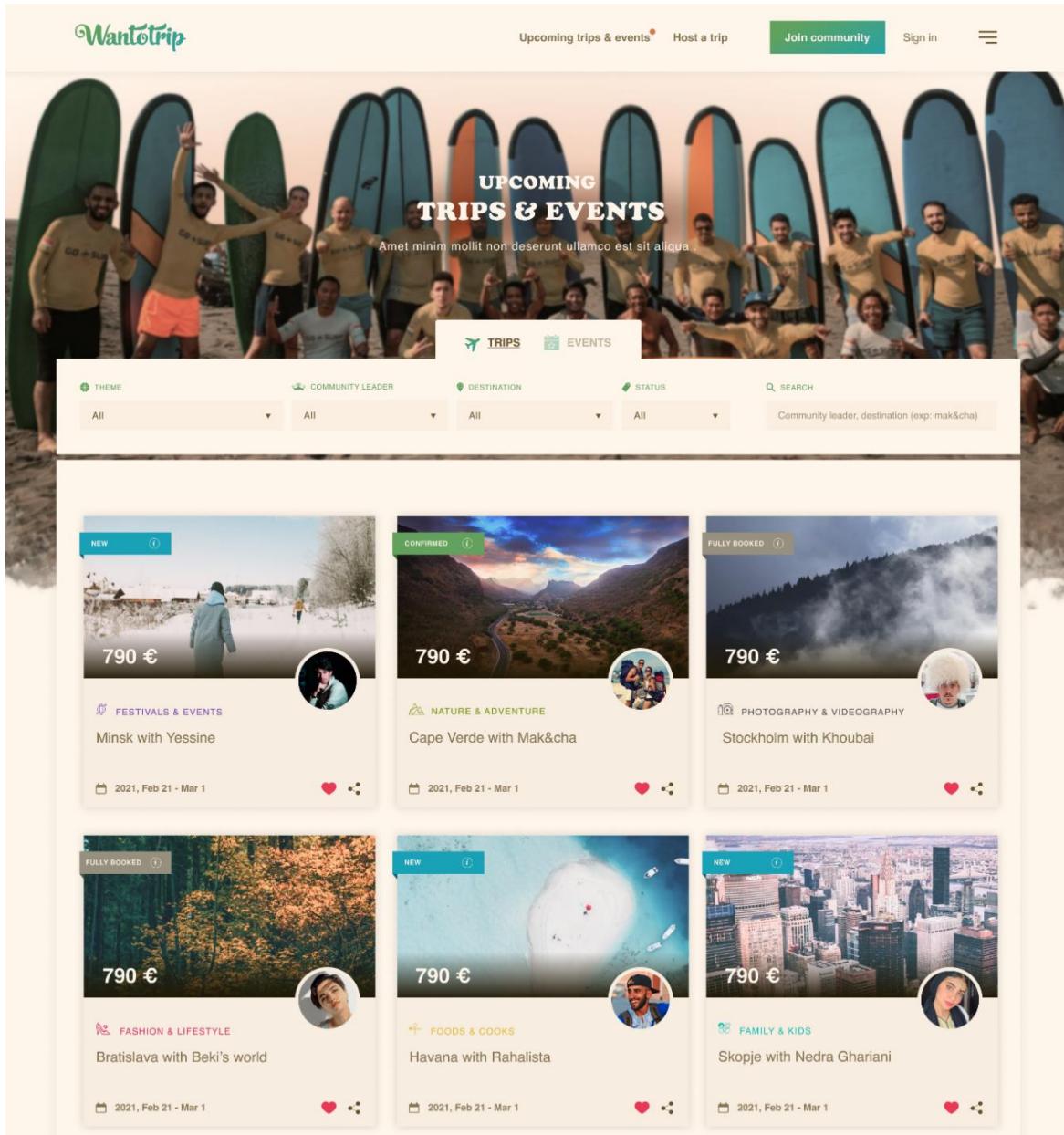


Figure 76: The upcoming trips and events interface 1 after the improvements

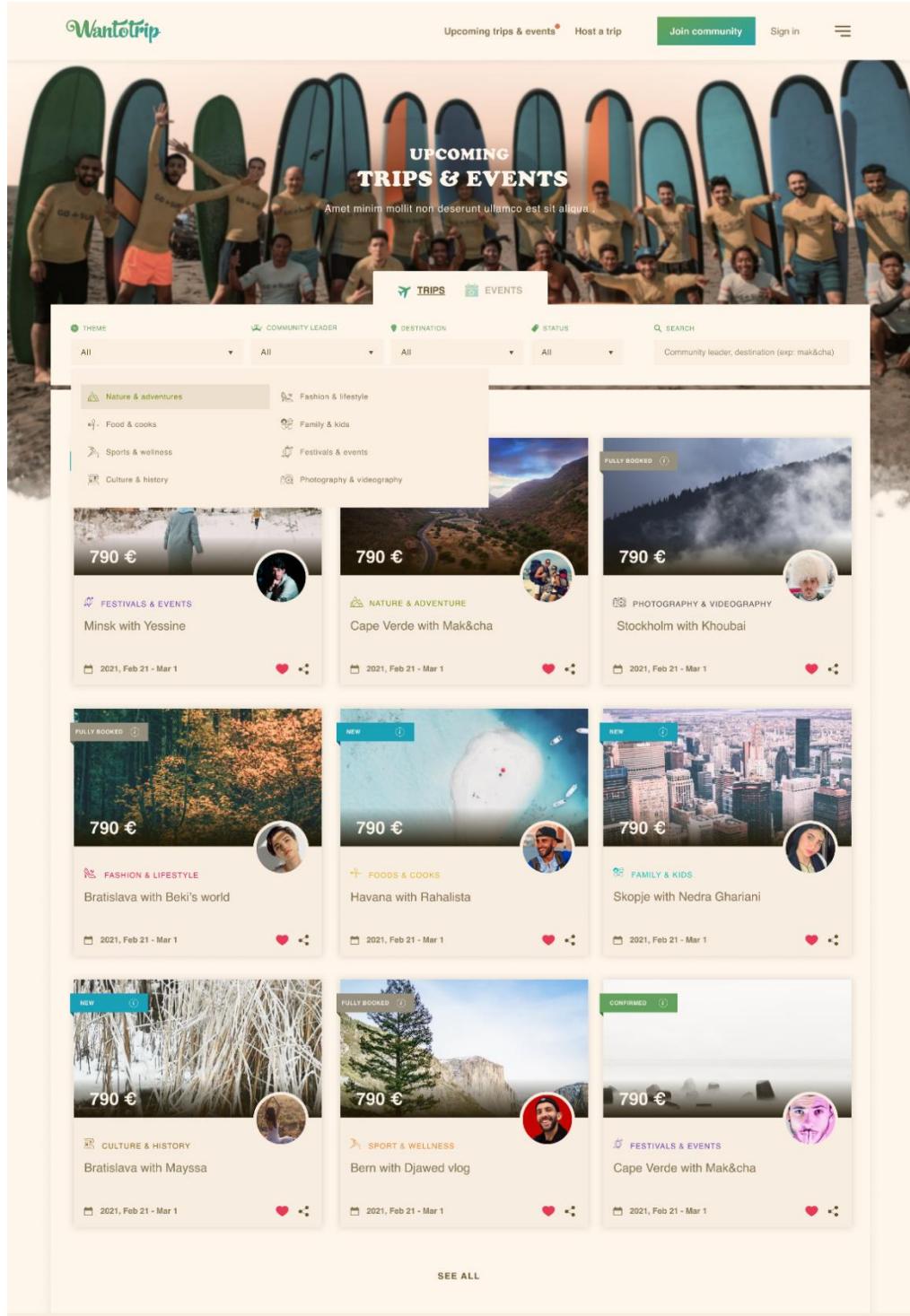


Figure 77: The upcoming trips and events interface 2 after the improvements

2.1.4 Testing the detail trip interface

- **Before the improvements**

The following figure shows the user interface trip details before making the changes.

Turkey from another perspective

Return and Adventure
2021, Jul 1 - Jul 11

This time, we are going to see one of the most interesting destinations in the world: the ancient remains, a place where the past meets the future. The blend of its landscapes and the variety of the possible, unique Europe and Asia and opening onto the Black Sea on a side, and the great Mediterranean on another. Adding to that, the splendor of the cultural heritage and the pride and sense of belonging of its citizens, and we get a wonderful opportunity to witness THE one and only Turkey! Let's go for a dive into the richness of the world of this amazing country!

Activities Included

<input type="checkbox"/> Shared accommodation (2 pax) <input type="checkbox"/> Breakfast <input type="checkbox"/> Daily round-trip airport transfers <input type="checkbox"/> Return flight <input type="checkbox"/> Kayak at lake <input type="checkbox"/> Camping <input type="checkbox"/> Hiking on the Lycian Way <input type="checkbox"/> Boat party	<input type="checkbox"/> Paragliding in Oludeniz <input type="checkbox"/> International flight tickets <input type="checkbox"/> Hotel transfers included <input type="checkbox"/> Travel insurance <input type="checkbox"/> Visa (if applicable)
--	--

Detailed itinerary

Day 1: Welcome to Antalya!
Once we arrive in Antalya, we will have the time to settle in our hotel before meeting for an opening ceremony during which we will get to know each other and discuss the details of our adventure.

Figure 78: The trips details before the improvements

▪ After the improvements

The following figure shows the user interface trip details after making the changes.

South Korea for Kpop and culture lovers

2021, MAY 10 - 5 APR 2022

South Korea is a highly developed country known for advanced technology and for having a rich cultural scene. It is also the land of K-pop and TV Dramas. Not only that; here it is also doing a tremendous job at keeping the Korean culture and tradition alive. Seoul, the capital of South Korea, is a vibrant city that offers a mix of traditional and modern elements. In this tour, we will travel around Seoul to witness the beauty of the city and its people. We will also visit some of the most famous landmarks and attractions in Seoul, such as Gyeongbokgung Palace, Namsan Tower, and Lotte World. This tour is perfect for anyone who wants to experience the best of South Korea's history and culture while staying at the latest technologies... we won't just travel. This tour will feature learning, new technologies, food, culture, fun and a lot of fun. Get ready to step up to discover the systems of what it has to offer!

Activities included

- Watch the Sunset
- Horseback Riding
- Rock Climbing
- Surfing at Bonita beach
- Camping
- Cycling + Mountain Biking

Inclusions

- International flight return (2 pax)
- Shared accommodation (2 pax)
- Private vehicle for Namdo Island day
- All activities and entrances fees mentioned in the program
- Breakfast
- English tour guide
- Assistance on how to fill the transport card during the free days

Exclusions

- Meals not mentioned
- Travel insurance

DAY 1

Arrival to Incheon International Airport

Activities

- Horseback Riding
- Watch the Sunset
- Horseback Riding
- Camping
- Rock Climbing
- Cycling + Mountain Biking

Workshops

- Beach Workshop
- Joshua Workshop

The journey will start in the Incheon International Airport where our partner will transfer us to our hotel for check-in. We need to rest after our long flight and before the start of our adventure.

Breakfast: Other Inn Camping

Figure 79: The trips details after the improvements

Conclusion

In this chapter, we introduced the last sprint of the project, focusing on the changes made to WanToTrip's existing website with the purpose of getting a better user experience and engaging the custom.

General conclusion and perspectives

To summarize, the goal as detailed and presented in this report is to create a customer relationship management web application for the hosting company WanToTrip to improve both their customer service and the efficiency of their employees.

This internship was a unique and fruitful for me since I was able to discover new technologies, environments, and the startup ecosystem in Tunisia.

In addition, this project was an exceptional opportunity to discover the professional environment in a Tunisian startup, to work in a team, to collaborate and network which is the core of the scrum methodology.

In fact, this internship presented some challenges at the beginning since I had to improve the user experience of the existing web application using specific technologies. Then created the customer relationship management system using other technologies in a duration of three months.

However, all these challenges allowed me to improve my technical skills, discover new technologies, build a good logic way of thinking and get adapted to new environment. This internship allowed me for sure to build great technical and soft skills.

Certainly, our project is open for several perspectives. We can add other features for the customer relationship management web application created such as billing management or campaign management or notification management that will make our system more efficient.

Bibliography

- [1] [Online]. Available: <https://www.tutterflycrm.com/faqs/>. [Accessed 25 April 2021].
- [2] [Online]. Available: <https://www.capterra.com/p/2501/Dolphin/#about>. [Accessed 25 May 2021].
- [3] [Online]. Available: <https://www.g2.com/products/adjetter-media-kapture-crm/reviews#details>. [Accessed 31 May 2021].
- [4] [Online]. Available: <https://www.360logica.com/blog/a-brief-introduction-to-scrum-methodology/>. [Accessed 31 May 2021].
- [5] [Online]. Available: <https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html>. [Accessed 20 May 2021].
- [6] [Online]. Available: <https://www.scrum.org/resources/what-is-a-product-backlog>. [Accessed 31 May 2021].
- [7] [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>. [Accessed 18 June 2021].
- [8] [Online]. Available: <https://vuejs.org/v2/guide/>. [Accessed 18 March 2021].
- [9] [Online]. Available: <https://stackify.com/n-tier-architecture/>. [Accessed 1 June 2021].
- [10] [Online]. Available: <https://vuetifyjs.com/en/introduction/why-vuetify/#getting-started>. [Accessed 17 May 2021].
- [11] [Online]. Available: <https://www.geeksforgeeks.org/bulma-introduction/>. [Accessed 1 June 2021].
- [12] [Online]. Available: <https://www.mongodb.com/nosql-explained>. [Accessed 31 May 2021].
- [13] [Online]. Available: <https://www.guru99.com/what-is-mongodb.html>. [Accessed 30 May 2021].
- [14] [Online]. Available: https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm. [Accessed 30 May 2021].
- [15] [Online]. Available: https://en.wikipedia.org/wiki/Visual_Studio_Code. [Accessed 30 May 2021].
- [16] [Online]. Available: <https://medium.com/@steffen.knoedler/everything-you-need-to-know-about-git-and-its-commands-c4404f1638c4>. [Accessed 31 May 2021].
- [17] [Online]. Available: <https://www.digitalcrafts.com/blog/student-blog-what-postman-and-why-use-it>. [Accessed 31 May 2021].
- [18] [Online]. Available: <https://trello.com/en/guide/trello-101>. [Accessed 18 June 2021].
- [19] [Online]. Available: https://en.wikipedia.org/wiki/Visual_Paradigm. [Accessed 1 June 2021].

ملخص

يركز مشروع تخرجي على دراسة وتصميم وتنفيذ حل ويب لإدارة علاقات العملاء
سيسمح تطبيق الويب هذا للشركة بتحسين علاقتها مع العميل ويسهل تنظيم وإدارة معلومات العميل.
الكلمات الرئيسية :تطبيق الويب ، نظام إدارة علاقات العملاء ، الشركات الحديثة

Abstract

My graduation project concentrates on the study, design, and implementation of a web solution for the management of customer relationship.

This web application will allow the company WanToTrip to improve their relationship with the client and facilitates organization and management of the customer's information.

Keywords : Web application, Customer Relationship Management System, Startup, Vue Js, Node Js

Résumé

Mon projet de fin d'étude se concentre sur l'étude, la conception et la réalisation d'une solution web pour la gestion des relations clients.

Cette application web permettra à la société WanToTrip d'améliorer sa relation avec le client ainsi que facilitera l'organisation et la gestion des informations des clients.

Mots clés : Application web, Système De Gestion De La Relation Clients, Startup, Vue Js, Node Js