Republic of Tunisia

The Honoris United Universities

The Private Higher School of Engineering and Technology

END-OF-YEAR PROJECT REPORT

Submitted in Partial Fulfillment of the Requirements to pass the 3rd year of the computer science and communication engineering cycle

Field of Study : Computer Science and Communication Engineering

CTN - Web System

By
Mohamed Dhia Jebali

Conducted within "CTN - Tunisian Navigation Company"

Publicly distributed in September 30 2023 with approval of :
Professional Supervisor : Mrs. Asma BEN HAMED, CTN Data Analyst

College Year: 202-2023

Republic of Tunisia
The Honoris United Universities
The Higher Institute of Information and Communication Technologies

END-OF-YEAR PROJECT REPORT

Submitted in Partial Fulfillment of the Requirements to pass the 3rd year of the computer science and communication engineering cycle
Field of Study : Computer Science and Communication Engineering

CTN - Web System

By
Mohamed Dhia Jebali

Conducted within "CTN - TUnisian Navigation Company"

Authorization of project report submission :

Professional Supervisor :
Mrs. Asma BEN HAMED

Dedications

To my family, the train of my life.
* My father, the wisest of fathers who taught me everything i knew, the captain my train.
* My mother, the kindest of mothers who was my gateway and guide thorough every chapter in my life, the engine of my train.
* My brother, the closet, person to me in this vast world who always keeps me happy everytime i think of him.
* My sisters, the sweetest angels of my road.
* My grandparents, who always took care of me and always think of me and since my birth.
* My aunts and uncles, who supported me with their praises and motivations.
I want to express my deepest love and appreciations to you, you are everything in my life and you made me into the man I'm today.

And to all my friends, who supported me thorough my journey, the so called Life, from the deepest of my heart, thank you so much.

Specially

Yours Truly — Mohamed Dhia Jebali

Acknowledgements

Before presenting our work, i want to take this opportunity to thank every person who contributed in this work. Specially **Mrs. Asma BEN HAMED** who welcomed me with open arms in CTN and who supervised me and helped me in the entirety of the internship duration and who provided me with valuable lessons and support.

Finally, We want to indicate our profound gratitude to all the staff at CTN and to the Director of the Private Higher School of Engineering and Technology **Mr. Taher BELAGTHER** and all the administration staff for their hard work and efforts that made such occasions possible.

Contents

Dedications i

Acknowledgements ii

General Introduction 1

1 Project Framework 2
- 1.1 Introduction . . . 2
- 1.2 Presentation of the Host Organization . . . 2
  - 1.2.1 Presentation . . . 2
  - 1.2.2 It's History . . . 3
- 1.3 System Context . . . 3
- 1.4 Criticism of the Existing . . . 3
  - 1.4.1 Description . . . 3
- 1.5 Considered Solution . . . 3
  - 1.5.1 Identification . . . 3
  - 1.5.2 Goal . . . 3
- 1.6 Methodology of Work . . . 4
  - 1.6.1 Agile Methodology . . . 4
  - 1.6.2 Agile Values . . . 4
  - 1.6.3 Agile Principles . . . 5
- 1.7 Development Requirements . . . 5
  - 1.7.1 Development Team . . . 5
  - 1.7.2 Development Tools . . . 5
  - 1.7.3 Accessibility and Requirements . . . 6
- 1.8 Conclusion . . . 8

2 Analysis and Use Cases 9
- 2.1 Introduction . . . 9
- 2.2 Identification of Use Cases . . . 9
  - 2.2.1 Functional Use Cases . . . 9
  - 2.2.2 Non-Functional Use Cases . . . 10
- 2.3 Actors Identification . . . 10
- 2.4 Global Diagrams . . . 11
  - 2.4.1 Global Use Case Diagram . . . 12
  - 2.4.2 Global Class Diagram . . . 13
- 2.5 Product Backlog . . . 14
- 2.6 Conclusion . . . 14

3 Conception **15**
3.1 Introduction . . . 15
3.2 First Sprint (Sprint 0) . . . 15
3.2.1 Refinement of the Use Case - "Authenticate" . . . 15
3.2.2 Refinement of the Use Case - "Command Movements" . . . 17
3.2.3 Refinement of the Use Case - "Manage Movements" . . . 18
3.2.4 Refinement of the Use Case - "Manage Users" . . . 19
3.3 Second Sprint (Sprint 1) . . . 20
3.3.1 Refinement of the Use Case - "Consult Movements" . . . 20
3.3.2 Refinement of the Use Case - "Evaluate Movements" . . . 21
3.3.3 Refinement of the Use Case - "Consult Evaluations" . . . 22
3.4 Conclusion . . . 22

4 Realization **23**
4.1 Introduction . . . 23
4.2 First Sprint (Sprint 0) . . . 23
4.2.1 Use Case - "Authenticate" . . . 23
4.2.2 Use Case - "Command Movements" . . . 24
4.2.3 Use Case - "Manage Movements" . . . 25
4.2.4 Use Case - "Manage Users" . . . 25
4.3 Second Sprint (Sprint 1) . . . 26
4.3.1 Use Case - "Consult Movements" . . . 26
4.3.2 Use Case - "Evaluate Movements" . . . 26
4.3.3 Use Case - "Consult Evaluations" . . . 26

General Conclusion **27**

List of Figures

1.1 CTN logo 2
1.2 "Agile Scrum" Schema 4

2.1 Actors 11
2.2 Diagram of the global use case 12
2.3 Global class Diagram 13

3.1 Sequence diagram of "Authenticate" 16
3.2 Sequence diagram of "Command Movements" 17
3.3 Sequence diagram of "Manage Movements" 18
3.4 Activity diagram of "Manage Users" 19
3.5 Sequence diagram of "Consult Movements" 20
3.6 Sequence diagram of "Evaluate Movements" 21
3.7 Sequence diagram of "Consult Evaluations" 22

4.1 Realization screenshot of "Authenticate" | Main Interface 23
4.2 Realization screenshot of "Command Movement" | Users List 24
4.3 Realization screenshot of "Command Movement" | Command Movement . 24
4.4 Realization screenshot of "Manage Movements" 25
4.5 Realization screenshot of "Manage Users" 25
4.6 Realization screenshot of "Evaluate Movement" 26
4.7 Realization screenshot of "Consult Evaluations" 26

List of Tables

1.1 Development team table 5
1.2 Development operation system table 6
1.3 Development programs table 6
1.4 Development programming languages table 7
1.5 Development frameworks table 7

2.1 Actors roles 11
2.2 Table of product backlog 14

3.1 Refinement table of "Authenticate" 16
3.2 Refinement table of "Command Movements" 17
3.3 Refinement table of "Manage Movements" 18
3.4 Refinement table of "Manage Users" 19
3.5 Refinement table of "Consult Movements" 20
3.6 Refinement table of "Evaluate Movements" 21
3.7 Refinement table of "Consult Evaluations" 22

General Introduction

This is a practical web project for the CTN company made in Agile(Scrum) Methodology, dedicated to establish and provide main functions and clear communication system between the navigators, navigation managers, navigation commandants and user managers.

The Navigator (Navigant) can check all of his navigations, old and new ones as well as their evaluation results.

The Navigation Commander (Commandant) can register a navigator in a navigation.

The Navigation Manager (Gestionnaire) can monitor navigators presence in the ship and evaluate his performance.

The User Manager (G. Utilisateur) can create user profiles with their appropriate functions in the system.

This report is a resume of my work with the methodology used to get the desired realization, and it's divided into four chapters:

The first chapter aims to present the host company CTN, the project presentation, as well as the project requirements and the proposed solution.

The second chapter is devoted to the presentation of the methodology used as well as the main use cases, requirements and project . Then, the main use case diagram.

The third chapter is dedicated to the realisation of the first release and explaining its conception and the realization.

The fourth chapter is dedicated to the realisation of the second release and explaining its conception and realization.

Finally, we will close our report with a general conclusion.

Chapter 1

Project Framework

1.1 Introduction

Let's begin this chapter by presenting the company of our project which is called **"CTN - Tunisian Navigation Company"**, then we will present the system context and the considered solution as well as the used methodology of our work and finally the development requirements of our project.

1.2 Presentation of the Host Organization

We introduce in this part the presentation of the company "CTN - Tunisian Navigation Company" , its activities and its adopted strategy then we present the professional solutions launched by this company.

1.2.1 Presentation

The "Compagnie Tunisienne de Navigation" (CTN or COTUNAV) is a Tunisian shipping line, providing regular passenger ferry connections between Tunisia and the ports of Marseille and Genoa, as well as freight transport to Barcelona and Livorno.

Figure 1.1: CTN logo

1.2.2 It's History

Founded on 7 March 1959, it initially concentrated on the development of regular shipping links between Tunisia and its principal trading partners, essentially serving only Marseille and Rouen. However, with the drive to diversify Tunisian overseas trade, it rapidly expanded its route network to encompass ports in Italy, Spain, Germany and the Benelux countries.

1.3 System Context

In a general context where gain and benefit are becoming user requirements, several companies want to be able to offer applications that can be adapted to their needs.
With the aim of mastering and deepening my knowledge, the company welcomed me and proposed to me to develop **a web application** for **Their staff members and navigators**. As this application help the user perform their main functions of monitoring navigation

1.4 Criticism of the Existing

1.4.1 Description

They already have an application that works fine enough, but it have some points that need some improvements :
- It uses outdated design and technologies - It lack some functions like the ability fora simple navigator to check their navigations and evaluations.
- Some limited accessibility like user can only login in when they have a navigation to manage, therefore they cannot check their profile and other information if he doesn't have an active navigation.

1.5 Considered Solution

I simply proposed a simple solution that rectify the previous mentioned points. Utilizing a more modern design with modern web technologies and added functionalities.

1.5.1 Identification

CTN Web System is a **Web Symfony based application** conceived in 2023 and was in development for one month and 30 days started in August of the same year. It have four sides : **a navigator side**, **a commandant side**, **a nav manager side**, and **a user manager side**,

1.5.2 Goal

As stated before the goal of this application is to give these mentioned users of CTN the ability to perform their intented functions accordingly and smoothly.
The Navigator (Navigant) can check all of his navigations, old and new ones as well as

their evaluation results.

The Navigation Commander (Commandant) can register a navigator in a navigation.

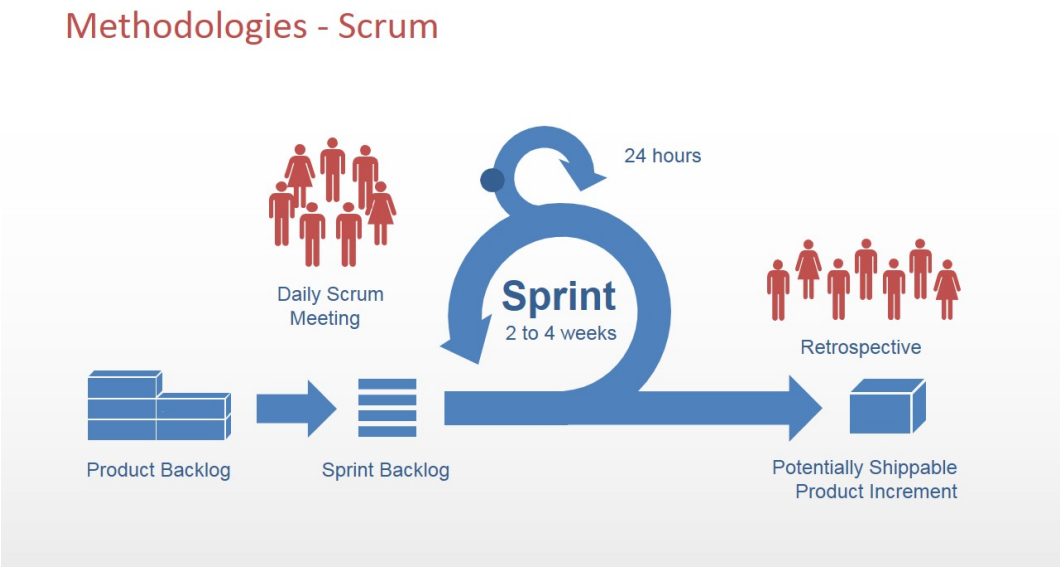
The Navigation Manager (Gestionnaire) can monitor navigators presence in the ship and evaluate his performance.

The User Manager (G. Utilisateur) can create user profiles with their appropriate functions in the system.

1.6 Methodology of Work

1.6.1 Agile Methodology

The Agile methodology is a way to **manage** a project by **breaking it up** into several **phases**. It involves constant collaboration with stakeholders and continuous improvement at every stage.
Once the work begins, teams cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders.[2]



[3]

Figure 1.2: "Agile Scrum" Schema

1.6.2 Agile Values

Based on their combined experience of developing software and helping others do that, the seventeen signatories to the manifesto proclaimed that they value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan [4]

1.6.3 Agile Principles

The Manifesto for Agile methodology is based on **twelve** principles:

1/ Customer satisfaction by early and continuous delivery of valuable software.
2/ Welcome changing requirements, even in late development
3/ Deliver working software frequently (weeks rather than months)
4/ Close, daily cooperation between business people and developers
5/ Projects are built around motivated individuals, who should be trusted
6/ Face-to-face conversation is the best form of communication (co-location)
7/ Working software is the primary measure of progress
8/ Sustainable development, able to maintain a constant pace
9/ Continuous attention to technical excellence and good design
10/ Simplicityâthe art of maximizing the amount of work not doneâis essential
11/ Best architectures, requirements, and designs emerge from self-organizing teams
12/ Regularly, the team reflects on how to become more effective, and adjusts accordingly [5]

1.7 Development Requirements

This section will showcase every entity needed for the development of the application

1.7.1 Development Team

The team members founding the project

Members	Role
Mohamed Dhia Jebali	Developing the project.
Asma Ben Hamed	Supervising the project.

Table 1.1: Development team table

1.7.2 Development Tools

In order to create our application in the best possible way. We chose these tools for the development.

Operation System :

	**Microsoft Windows 11**	Windows 11 is the latest major release of Microsoft's Windows NT operating system, released in October 2021. It can run every tool we need for the project perfectly and then some.[6]

Table 1.2: Development operation system table

Programs :

Icon	Title	Description
	**Visual Studio Code**	VSC is a code editor and in this regard our code editor of choice, features extremely customizable and upgradable core style with seemingly endless extensions, adorning a beautiful interface making it very easy to use without sacrificing any depth.[7]
	**Microsoft Edge**	It is a chromium based web browser created by Microsoft in 2015 (2019 with Blink engine), it is used to test our application and for all research and learning needed for it's making.[8]
	**GitHub**	It is an online platform for basic to complex internet hosting for software development using Git. it serves as the hosting base for the project development repository.[11]
	**Git**	It is a software for tracking changes in any set of files, it is the center of GitHub service as it is needed for the project cooperatives to know, pull and push changes of the application files.[12]
	**GNU Image Manipulation Program**	It is a free and open-source raster graphics editor used for image manipulation (retouching) and image editing, free-form drawing, transcoding between different image file formats, and more specialized tasks. We used it to design most visual graphical pictures of the project.[13]
	**XAMPP**	It is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.[14]

Table 1.3: Development programs table

1.7.3 Accessibility and Requirements

All it needs is a HTML5 capable Web browser to run. And of-course the database of the company.

Programming Languages :

Icon	Title	Description
	**Hypertext Markup Language revision 5**	It is a markup language for web development started in 2008, structuring and presenting content on the World Wide Web. It is the fifth and final major HTML version that is a World Wide Web Consortium (W3C) recommendation. It contain three major components, one of them is the base HTML but upgraded.[15]
	**Cascading Style Sheets**	It is a style sheet programming language to pcontrol in the specification of presentation characteristics.[16]
	**JavaScript**	It is a standard programming language and one of HTML5's fundamental components, while html and css focus on the interface, javascript focus on performing our typical main programmings. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles.[17]
	**PHP**	It is a versatile scripting language that is especially suited for web development. It was created by Rasmus Lerdorf in 1993 and released in 1995. PHP is now maintained by the PHP Group..[18]

Table 1.4: Development programming languages table

Frameworks :

Icon	Title	Description
	**Symfony**	Symfony is a free and open-source PHP web application framework, as well as a set of reusable PHP component libraries. It was published as free software on October 18, 2005, and released under the MIT License. Symfony is a full-stack framework, meaning that it can be used to build both the front-end and back-end of a web application. It provides a wide range of features.[19]
	**Bootstrap**	It is a frontend toolkit library of HTML, CSS and JS components, it provide our app with it's theme and format.[22]

Table 1.5: Development frameworks table

1.8 Conclusion

To summarize this chapter, we introduced our host organization, the "SSS Innovation" startup and presented it's activities and goal, we discussed the main system context of our project and the previous solution that our app is based on and we mentioned all the tools needed to develop the app and everything the app require to launch and which platform can run it.
The next chapter will focus on the use cases of our project, We will be reviewing the general use cases, actors and main sprints.

Chapter 2

Analysis and Use Cases

2.1 Introduction

In this chapter ,We will be listing every functional and non-functional use case of the project, the actors interacting with these use cases and finally we will break down the global diagrams and backlog product of the project. These information will be more then effective to put us on the track of fully understanding the "goals" of the project.

2.2 Identification of Use Cases

In this section, we will capture all the functional and non-functional use cases.

2.2.1 Functional Use Cases

In this project we have mainly 14 functional use cases :

- **Consult movements**
The system must enable its users to consult ship movements.
- **Manage movements**
The system must enable its users to manage navigators ship movements state and their details.
- **Command movements**
The system must enable its users to command movements to navigators.
- **Evaluate movements**
The system must enable its users to evaluate navigators performance in their movements.
- **Consult evaluations**
The system must enable its users to consult their ship movements evaluations.
- **Manage users**
The system must enable its users to manage their users.
- **Authenticate**
The system must enable its users to authenticate their access to their user account.

2.2.2 Non-Functional Use Cases

Non-functional use case define internal primordial requirements for the system to be executed properly such as :

- **Ergonomics**
The application must adapt an ergonomic modern interface, it have to be simple but without compromising depth and options.
- **Usability**
The application must be easy to comprehend, easy and to use and easy to blend in.
- **Performance**
The application must output good human-acceptable performance, it should be fast and available at any time.
- **Extensibility**
The application must be capable of being expandable, maintainable and upgradable by receiving new updates and functions without the need rebuilding it from the ground.
- **Security**
The application must be secure to use and must respect online privacy rules (we used SHA-256 for hashing passwords and Firebase build-in security rules).

2.3 Actors Identification

There are only three actors in our application :

- **Navigator**
Navigators are ship movement attenders who will embark into a commanded ones and perform their duties in the vicinity. They simply can login to their account and check their movements and evaluations.
- **Commander**
Commanders are siege workers who command navigators and movement managers into ship movements.
- **Movement Manager**
Movement Managers are ship movement attenders too but they regulate other navigators movement state as well as evaluating them.
- **User Manager**
User Managers are simply tasked with managing everyone system accounts from creating to modifying then deleting.

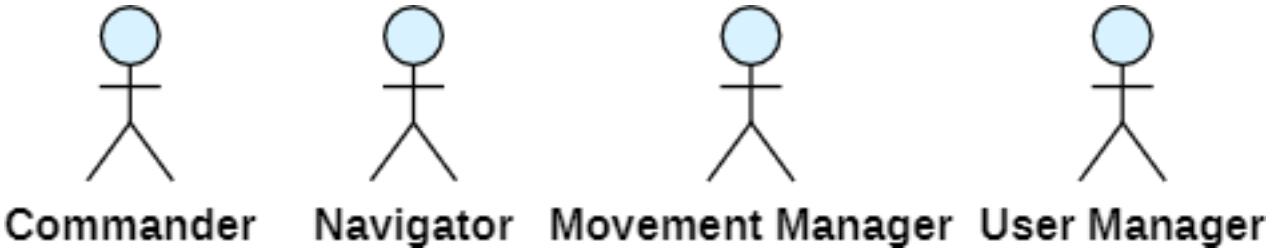


Figure 2.1: Actors

Actors	Role
Commander	- Consult movements - Command movements
Movement Manager	- Consult movements - Manage movements - Evaluate movements - Consult evaluations
User Manager	- Manage users
Navigator	- Consult movements - Consult evaluations

Table 2.1: Actors roles

2.4 Global Diagrams

In this section, we will illustrate the global sequence diagram and the class diagram allowing us to have an idea on the global specification of our application.

2.4.1 Global Use Case Diagram

We will be showcasing the global use case diagram of our project, visualizing what we explained so far in an elaborate figure :

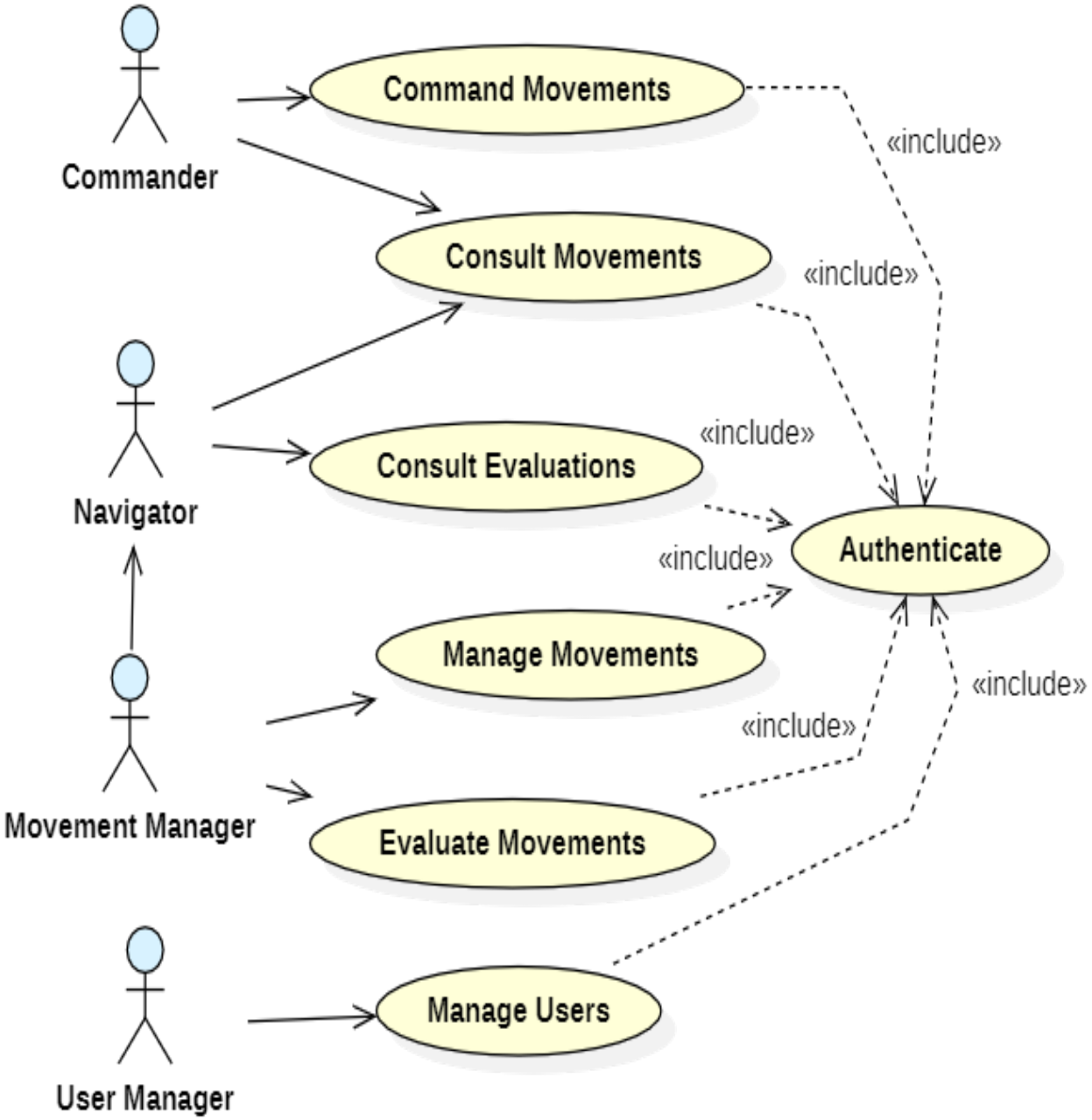


Figure 2.2: Diagram of the global use case

2.4.2 Global Class Diagram

We will be showcasing the global class diagram of our project, visualizing every entity and their interactions in the system

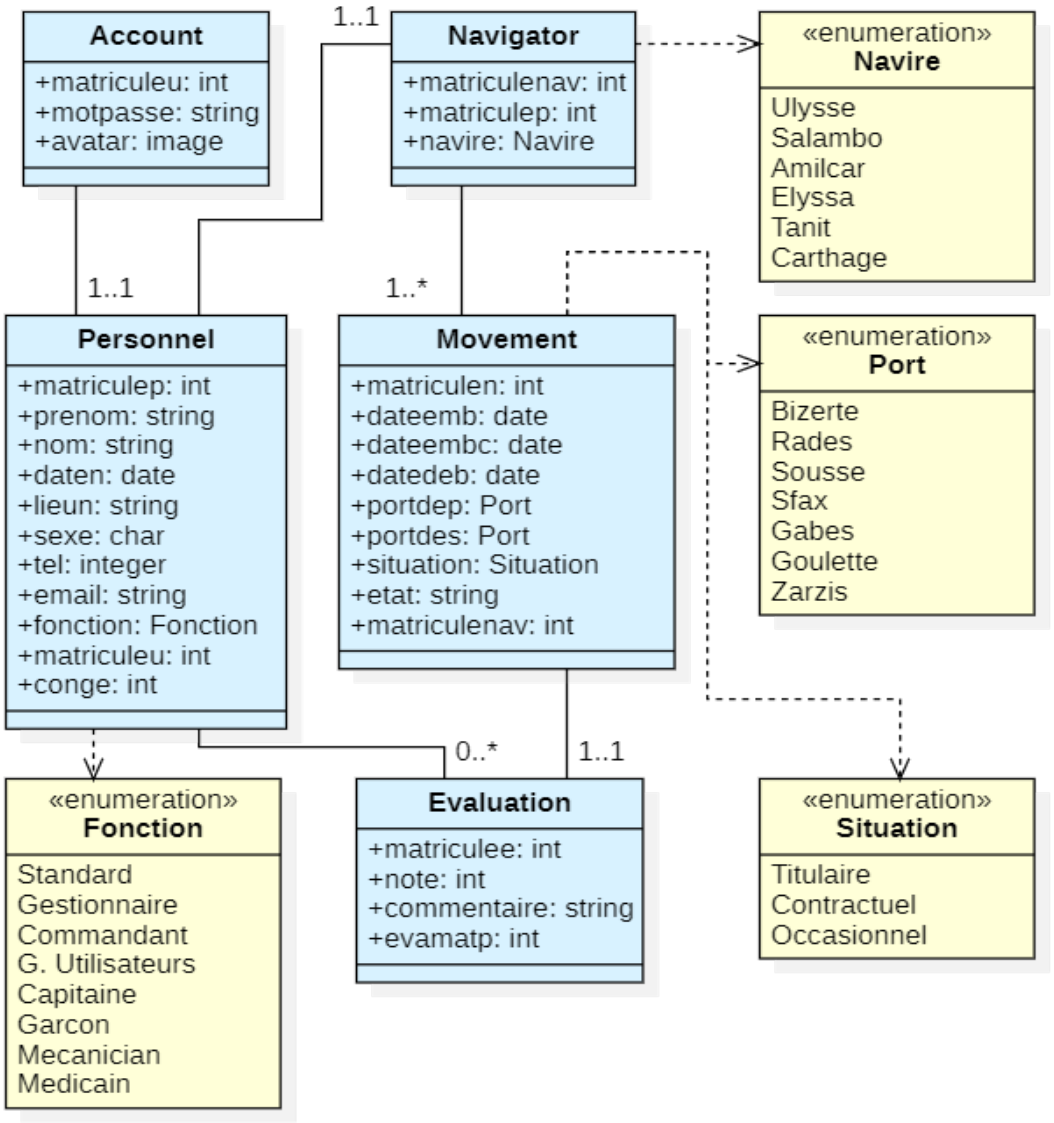


Figure 2.3: Global class Diagram

2.5 Product Backlog

In this section, we will present the table of the product backlog which classify the direction of the project into sprints (but one release), containing the use cases with high estimation for all of them like this :

Back Log of the sprint	Priority	Sprint
As a **commander**, i can **authenticate**	1	0
As a **commander**, i can **command movements**	1	0
As a **commander**, i can **consult movements**	3	1
As a **movement manager**, i can **authenticate**	1	0
As a **movement manager**, i can **manage movements**	1	0
As a **movement manager**, i can **evaluate movements**	2	1
As a **movement manager**, i can **consult movements**	3	1
As a **movement manager**, i can **consult evaluations**	1	2
As a **user manager**, i can **authenticate**	1	0
As a **user manager**, i can **manage users**	2	0
As a **navigator**, i can **authenticate**	2	0
As a **navigator**, i can **consult movements**	2	1
As a **navigator**, i can **consult evaluations**	3	1

Table 2.2: Table of product backlog

2.6 Conclusion

In this chapter, we presented our project primordial elements based on scrum structure, the functional and non-functional requirements use cases, specified the general use case and general class diagram and we finished by showcasing the backlog of the product . In The next chapter, we will elaborate on the realization of our solution.

Chapter 3

Conception

3.1 Introduction

In this chapter, we will focus on the conception of our project with the most important use cases of our 3 sprints.

3.2 First Sprint (Sprint 0)

Use Cases of the first sprint

This sprint puts on the way to develop these cases of our project :

- **Authenticate**
- **Command movements**
- **Manage movements**
- **Manage users**

Now we will refine all the use cases of sprint 0 by expressing all the scenarios corresponding with every case :

3.2.1 Refinement of the Use Case - "Authenticate"

We will extract the description, the use case diagram and the refinement table of this use case.

Description :

Any user (Commander, Movement Manager, User Manager or Navigator) can sign in by inserting their credentials which is email/phone number and password then their respective interface will load up (can also select the forgot password to reset their password securely).

Refinement table :

This is the refinement table of our use case complete with the actors, pre-condition, post-condition, a full description of the principal scenario and the exception :

Use case	***Authenticate***
Actors	Every Actor
Pre-condition	The system is working.
Post-condition	The user is authenticated.
Principal scenario description	1/ After opening the application the user insert **Email/Phone** and **Password** then the **Connecter** button to sign in the their right interface. 2/ The user can select the **Mot de passe oublie** button and then insert their **Email/Phone** in the new textbox then confirm to receive email resetting the password.
Exception	- Launch problem - Connection problem - Wrong inserted information problem

Table 3.1: Refinement table of "Authenticate"

Sequence diagram :
We're going to use the sequence diagram that shows object interactions arranged in time sequence.

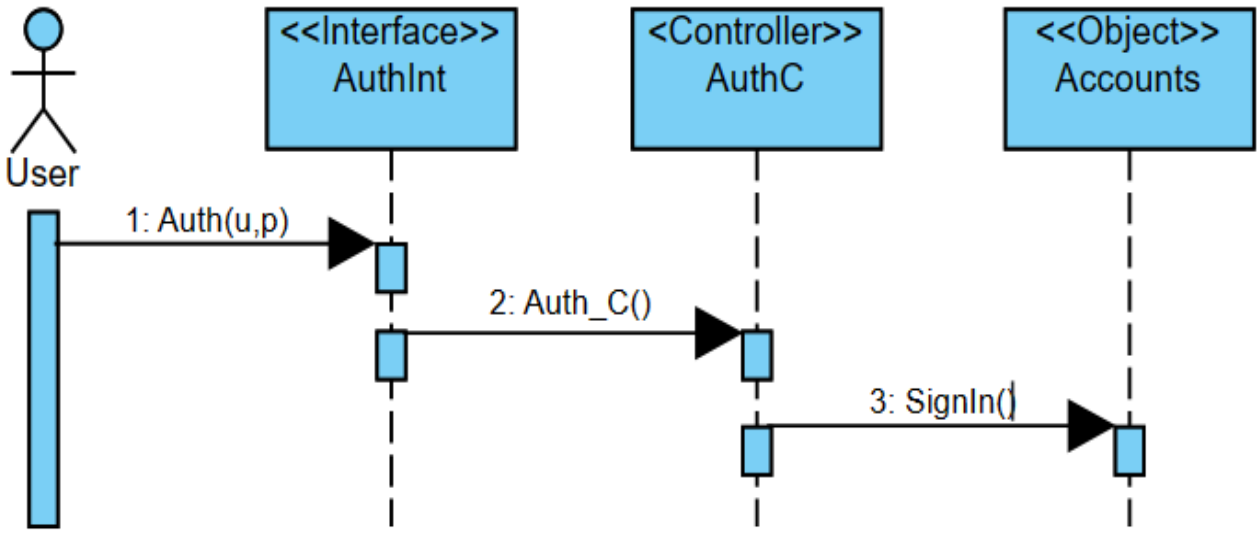


Figure 3.1: Sequence diagram of "Authenticate"

3.2.2 Refinement of the Use Case - "Command Movements"

Description :
The commander can issue to a navigator (or a movement manager) a ship movement with embark date and potential debark date, from a port to another.
Refinement table :

Use case	***Command Movements***
Actors	The commander
Pre-condition	The system is working and the user is authenticated
Post-condition	The command issue is sent successfully.
Principal scenario description	1/ After authentication the system will show a list of navigators, the commander press **Red** dot of a navigator line will show options, then the commander have to press **Ajouter Mouvement**. 2/ The commander **fill up** the form of the new movement which include the debark date and port and expected debark date and port. 3/ The commander press *Ajouter movement* to analyze the form and issue the new movement command to the selected navigator if the inserted info are correct. 4/ The user is also capable of filtering the the list or generating PDF/XLS with filter/generation panel.
Exception	- Launch problem - Connection problem - Wrong inserted information problem

Table 3.2: Refinement table of "Command Movements"

Sequence diagram :

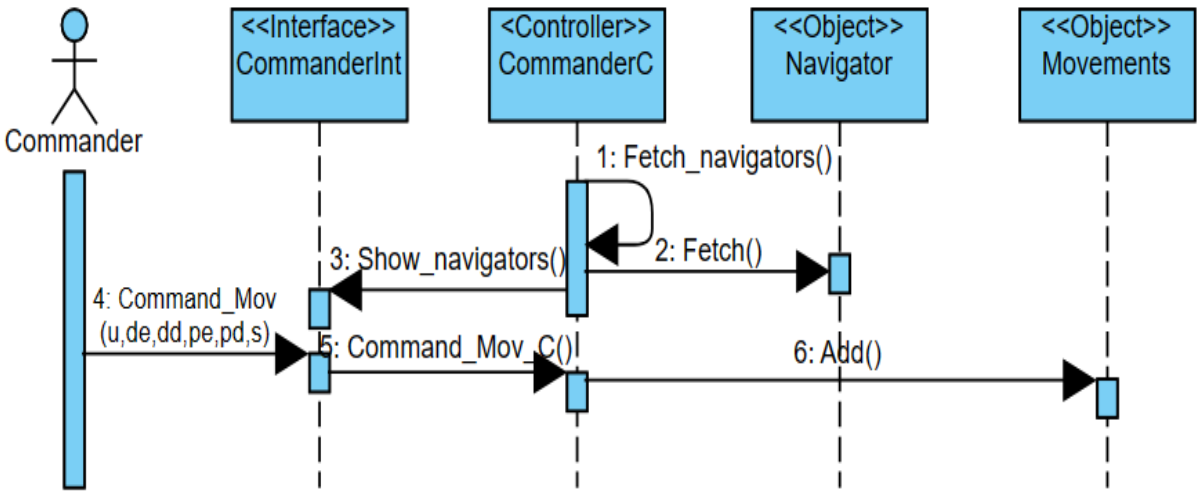


Figure 3.2: Sequence diagram of "Command Movements"

3.2.3 Refinement of the Use Case - "Manage Movements"

Description :
The movement manager can check movements of their ship travel only, when inside the ship they can mark the time when navigator enter and leave the ship.
Refinement table :

Use case	***Manage Movements***
Actors	The movement manager
Pre-condition	The system is working and the user is authenticated
Post-condition	The navigator's mv. arrival/debark time/state is saved.
Principal scenario description	1/ After authentication the system will show a list of movements with their states of the user's ship, the manager can insert the **Embark** in the embarquer date column of the desired programmed movement line then confirm by pressing the **Red dot** followed by selecting **Embarquer** (or cancel or absent). 2/ The manager can insert the **Debark** date in the debarquer date column of an arrived navigator after the shift is done then confirm by pressing the **Red dot** followed by selecting **Debarquer** (or Evaluer). 3/ The user is also capable of filtering the the list or generating PDF/XLS with filter/generation panel.
Exception	- Launch problem - Connection problem - Wrong inserted information problem

Table 3.3: Refinement table of "Manage Movements"

Sequence diagram :

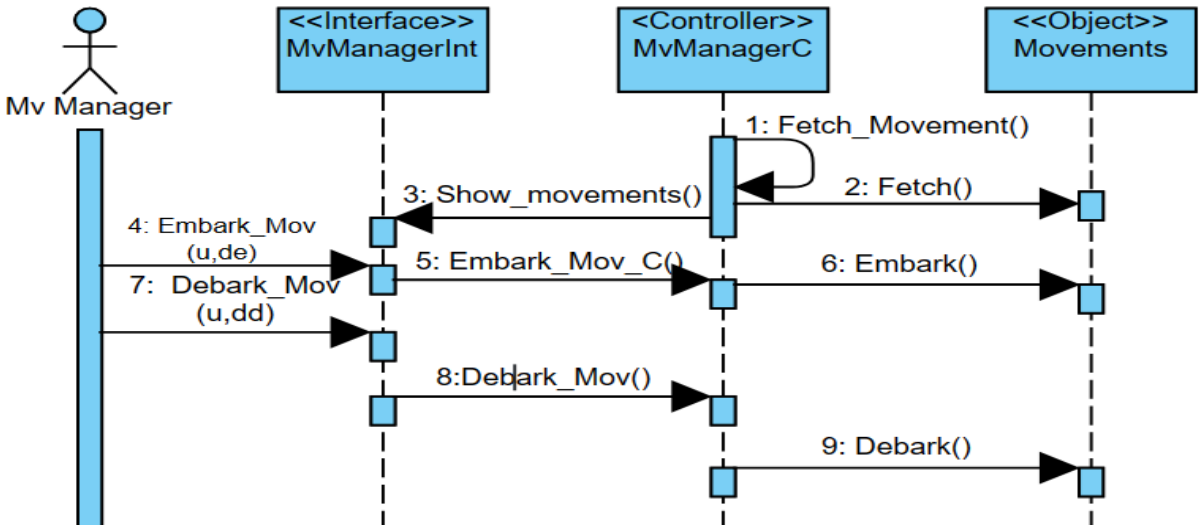


Figure 3.3: Sequence diagram of "Manage Movements"

3.2.4 Refinement of the Use Case - "Manage Users"

Description :
The user manager can check users of all types and can create more and modify or remove exiting ones.
Refinement table :

Use case	***Manage Users***
Actors	The user manager
Pre-condition	The system is working and the user is authenticated.
Post-condition	A user account is created, modified or removed.
Principal scenario description	1/ After authentication the system will show a list of users with their info, the user manager can press the **Ajouter** red button in the topbar to load the user creation form, the manager then can fill up the form and press the confirm button to add a new user. 2/ The manager can press the **Red** dot of a user list to show modify and remove options. 3/ The user is also capable of filtering the the list or generating PDF/XLS with filter/generation panel.
Exception	- Launch problem - Connection problem - Wrong inserted information problem

Table 3.4: Refinement table of "Manage Users"

Activity diagram :

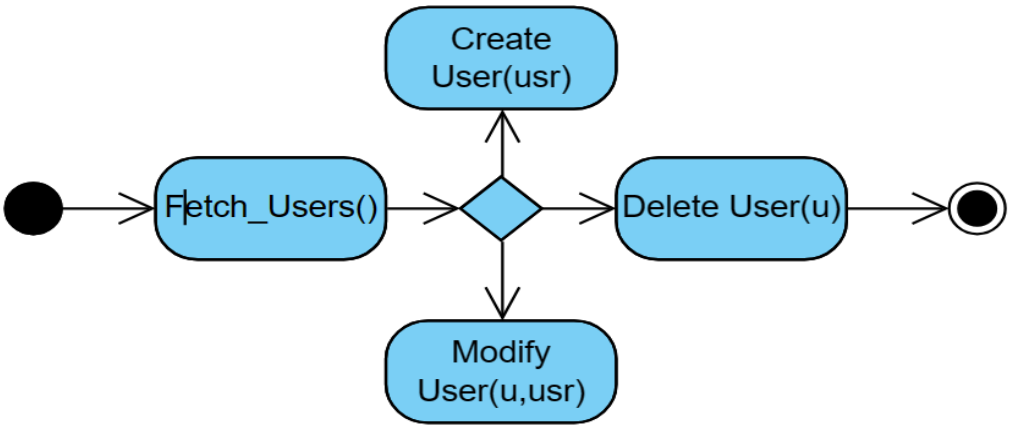


Figure 3.4: Activity diagram of "Manage Users"

3.3 Second Sprint (Sprint 1)

Use Cases of the second sprint

This sprint puts on the way to develop these cases of our project :

- **Consult movements**
- **Evaluate movements**
- **Consult evaluations**

Now we will refine all the use cases of sprint 1 by expressing all the scenarios corresponding with every case :

3.3.1 Refinement of the Use Case - "Consult Movements"

Description :
Every user except the user manager can consult (their) movements with their user account restrictions.
Refinement table :

Use case	***Consult Movements***
Actors	Commander, Movement Manager and Navigator
Pre-condition	The system is working and the user is authenticated.
Post-condition	List of movements is displayed.
Principal scenario description	1/ After authentication the system will show a list of movements. If the user is a navigator it will show their movements, if the user is a commander it will show every movement and if the user is a movement manager it will show their ship movements. 2/ The user is also capable of filtering the the list or generating PDF/XLS with filter/generation panel.
Exception	- Launch problem - Connection problem - Wrong inserted information problem

Table 3.5: Refinement table of "Consult Movements"

Sequence diagram :

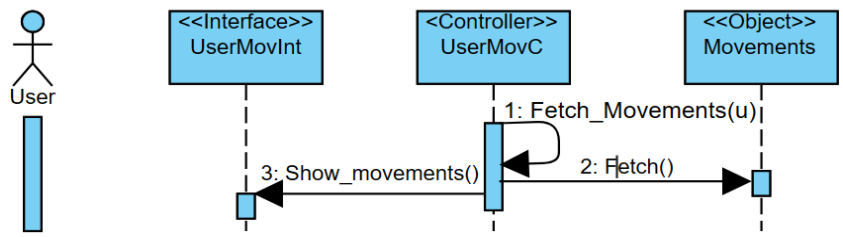


Figure 3.5: Sequence diagram of "Consult Movements"

3.3.2 Refinement of the Use Case - "Evaluate Movements"

Description :
The movement manager can evaluate navigator performance in the duration of the ship travel with a 0 to 20 points and a comment.
Refinement table :

Use case	***Evaluate Movements***
Actors	Movement Manager
Pre-condition	The system is working and the user is authenticated.
Post-condition	Movement is evaluated.
Principal scenario description	1/ After the user authenticated and embarked a movement they can press the **Red dot** of the desired movement then **Evaluer** to take the user to the evaluation page 2/ The manager then can use the **Slider** to select from 0 to 20 the score of the evaluation and insert a comment in the **Textarea** then confirm, this will mark the movement as debarked and evaluated. 2/ The manager can also reevaluated a movement by pressing the **Red dot** then **Revaluer**.
Exception - Launch problem	- Connection problem - Wrong inserted information problem

Table 3.6: Refinement table of "Evaluate Movements"

Sequence diagram :

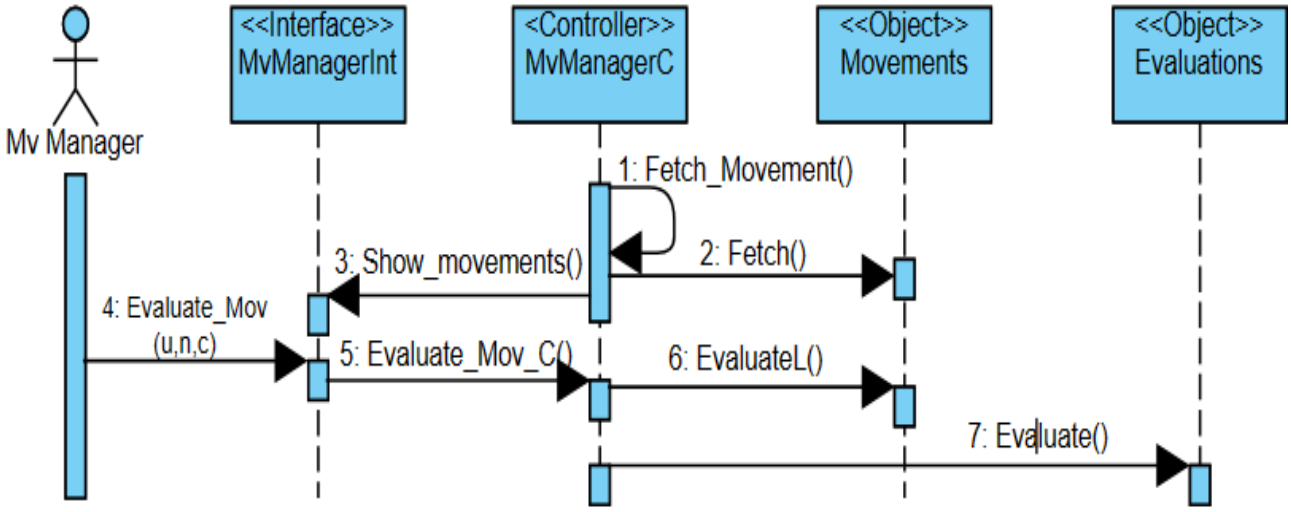


Figure 3.6: Sequence diagram of "Evaluate Movements"

3.3.3 Refinement of the Use Case - "Consult Evaluations"

Description :
The movement manager and the navigator can consult their evaluations, checking the score and comment.
Refinement table :

Use case	***Consult Evaluations***
Actors	Movement Manager and Navigator
Pre-condition	The system is working and the user is authenticated.
Post-condition	List of evaluations is shown.
Principal scenario description	1/ After the user authenticated and pressed the **Evaluations** button on the side bar it will load the page with list of evaluations. If the user is a navigator, they can see their evaluations only and if the user is a movement manager they can see their evaluations of every navigator movement they evaluated. 2/ The movement manager can reevaluate a movement by pressing the **Red dot** then **Revaluer**. 3/ The user is also capable of filtering the the list or generating PDF/XLS with filter/generation panel.
Exception	- Launch problem - Connection problem - Wrong inserted information problem

Table 3.7: Refinement table of "Consult Evaluations"

Sequence diagram :

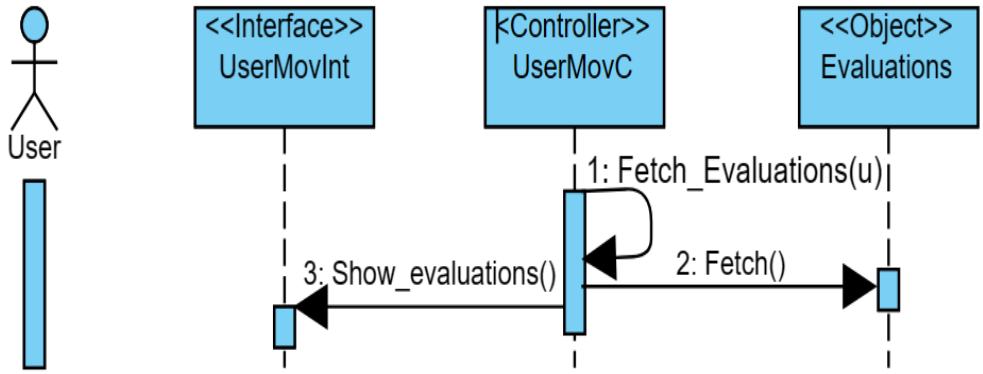


Figure 3.7: Sequence diagram of "Consult Evaluations"

3.4 Conclusion

In this chapter we showcased the conception of our project detailing each use case with description and refinement table and diagrams.]
In the next chapter, we will present the realization of our project.

Chapter 4

Realization

4.1 Introduction

In this chapter, we will focus on the realization of our project, showcasing our interfaces.

4.2 First Sprint (Sprint 0)

4.2.1 Use Case - "Authenticate"

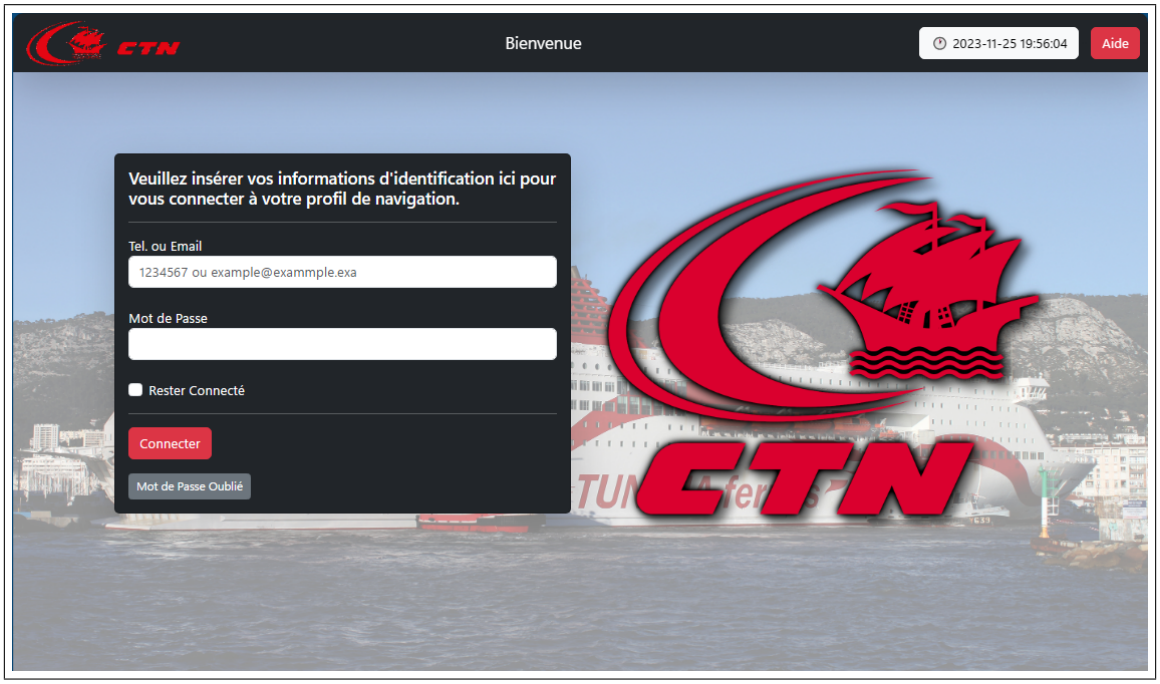


Figure 4.1: Realization screenshot of "Authenticate" | Main Interface

This is the authentification page. user have to simply insert their email or phone number and password then press Connecter.

4.2.2 Use Case - "Command Movements"

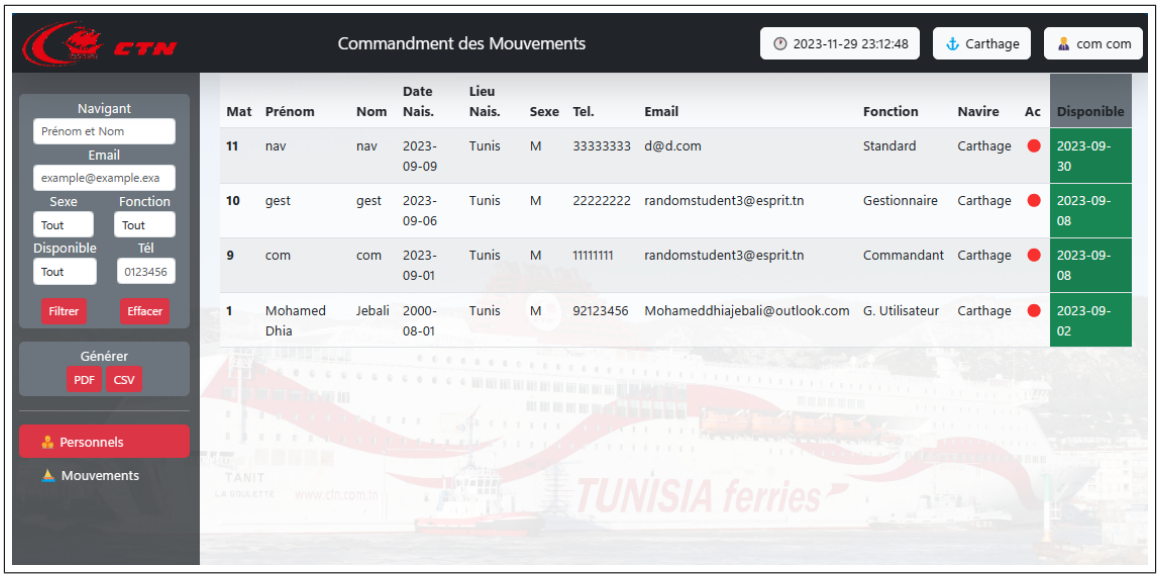


Figure 4.2: Realization screenshot of "Command Movement" | Users List

This is the page that represent the list of all navigator. commander have to simply press the red button of the navigator they want to command (the navigator have a date that he can be available) Then press Commander Mouvement.

Figure 4.3: Realization screenshot of "Command Movement" | Command Movement

Then the commander can insert all the data necessary for the movement to command.

4.2.3 Use Case - "Manage Movements"

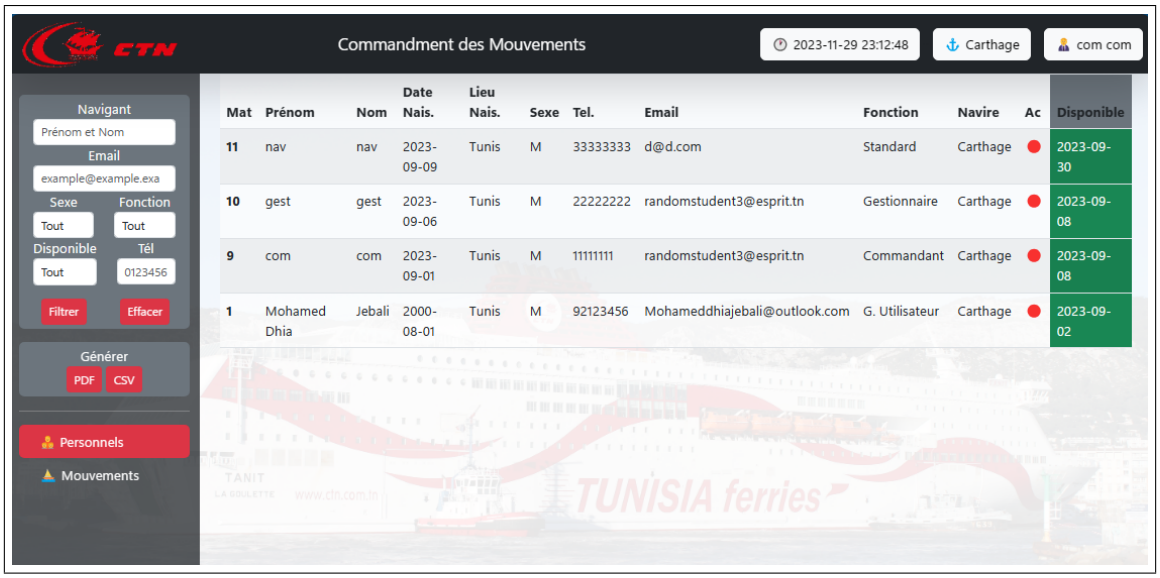


Figure 4.4: Realization screenshot of "Manage Movements"

This is the page that represent the list of all movement the movement manager can manage. They can press the red dot of a movement to embark, debark, cancel, absent, evaluate or Re valuate.

4.2.4 Use Case - "Manage Users"

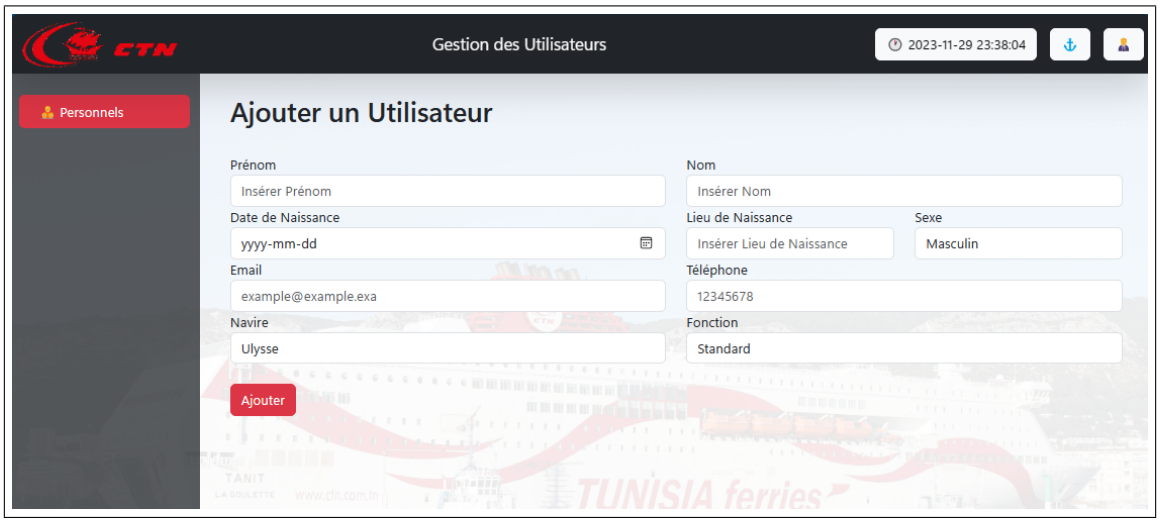


Figure 4.5: Realization screenshot of "Manage Users"

After the user manager log in it will show the list of all users like in Command Movements User List, this page represent the user adding form.

4.3 Second Sprint (Sprint 1)

4.3.1 Use Case - "Consult Movements"

It is the same interface as "Manage Movements" with different red dot commands

4.3.2 Use Case - "Evaluate Movements"

Figure 4.6: Realization screenshot of "Evaluate Movement"

The movement manager have to press the red dot of a movement then evaluate/revaluate then insert the evaluation data.

4.3.3 Use Case - "Consult Evaluations"

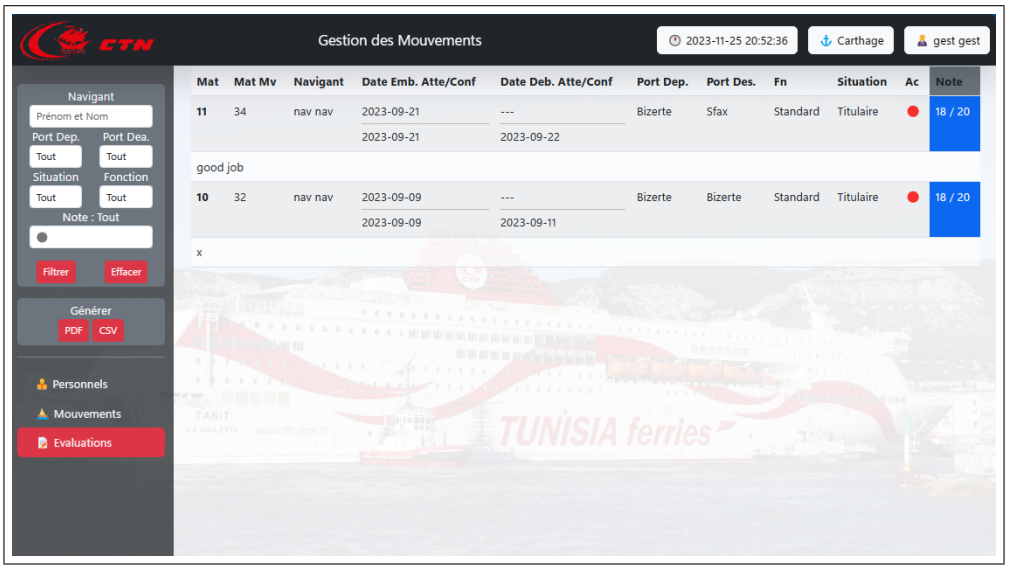


Figure 4.7: Realization screenshot of "Consult Evaluations"

Both movement manager and navigator can consult their evaluations. by pressing the evaluations button on the sidebar and it will show the lis of evaluations.

General Conclusion

And with that we reached the end of our project.

With the rapid growth of digital technology evaluations, it is a must for all companies to adapt to this rythm and update their applications accordingly and this is what this application is striving to be.
Using the latest symfony a very well reconized Web PHP Framework
I managed to create an application that outdoes the existing one and adding more functionalities.
With this report we showcased all the information, conception and realization needed to comprehend the application development.
Thank you for reading.