Ministère de L’enseignement supérieur et de la recherche scientifique

Direction Générale des Etudes Technologiques

Institut Supérieur Des Etudes Technologiques de Bizerte

Département Technologies de l’informatique

Report of

END-OF-STUDY PROJECT

In order to obtain:

Applied License in Information Systems Development « DSI »

Developed by: Oussama Bejaoui and Hassen Kaabouri

Framed by: Mr Bilel Zemzem and Mr Ramzi Chridi

Internship period: Still***

Host organization: INTUITIV GROUP

- **Address:** Still***
- **Tel / Fax:** Still***
- **Email:** contact@intuitiv.fr

University Year: 2021/2022

Appreciation

It is with great respect and gratitude that we express our sincere thanks to our pedagogical supervisor Mr. Bilel Zemzem, teacher at the Higher Institute of Technological Studies of Bizerte, for his attentive supervision, his remarks, his guiding ideas, his pedagogy, his support throughout our project, his enthusiasm, his personal and professional qualities and his constructive and relevant suggestions which have been of invaluable help in the development of this work.

We would also like to express our sincere thanks to our professional supervisor Mr Ramzi Chridi for welcoming us to the INTUITIV group.

We would also like to express our gratitude to all the teachers of the computer science department of the ISET of Bizerte.

Plan

Table of Contents

Figures

General Introduction

In this age of technology, everything is digital. More and more companies are investing a huge amount of money and resources to digitalize their work and manage it through virtual solution is order to win time, lower their outcomes and to most importantly stay in the race of technology. We have to admit also that when it comes to managing human resources it is a lot more complex and delicate then we think it’s a domain that doesn’t accept mistakes. Some of the main complex tasks for a human resources team are:

- Legal Compliance
- Organizational Recruitment and Selection
- Human Resource Planning and Development

When it comes to our enterprise “INTUITIV”, it is very similar to the global situation.

It is investing their time and money in order to facilitate and improve the HR team process and avoid its mistake especially in its complex tasks as we mentioned above.

In the next few pages of our internship record, we will touch the majority of these main complex tasks in order to present and create a suitable solution.

In this occasion, we divided the report to chapters. First chapter we will present the company, the issues, used methodology, Scrum and the used technologies. Next, we will have 3 chapters which will be particularly similar, each will contain the sprint’s backlog, process and advancement with the addition of the specific diagrams.

Chapter 1: Working Context

Introduction

In this chapter, we will introduce the company "INTUITIV", and then we will present the problem and describe the project to be carried out. Finally, we will define the Scrum agile framework as a framework for carrying out our mission.

I. Presentation of the company

Intuitiv-technology is a company with a dual culture: both an INNOVATIVE ESN and an INTERACTIVE AGENCY, it bases its success on the reactivity, innovation, reliability and quality of its services, the company has taken advantage of new technologies to offer ever more innovative solutions in development engineering, and in the integration of all the solutions offered by the group.

The referencing of Intuitiv-Technology with large national and international groups reinforces our conviction of the quality and efficiency of service structures on a human scale.

Intuitiv-Technology stands out as much by its levels of expertise as by its human-sized structure, where real values can really be expressed...

II. Issue of the « PFE »

1. Description of the problem

By making an overview of the field of Human Resources Management Systems (HRMS) in the market, we've discovered that it's growing and developing very quickly, thanks to its great usefulness for companies.

H.R.M.S is a type of information system (IS) that is designed to manage an organization's computerized and automated human resource (HR) processes.

An H.R.M.S is therefore highly recommended for any company especially big companies, regardless of the field of application in which it operates, since it allows you to:

- Manage employee records.
- Manage attendance.
- Manage paperwork digitally.
- Improve the efficiency of your HR team.
- Improve your employee experience.
- Save you money and time.
- Help you make better decisions.
- Improve your regulatory compliance.

Our project is for the benefit of the INTUITIV group company. One of its main activities of it is to organize the paperwork of their employees for different situations in order to stay organized.

Among the problems encountered by the INTUITIV group with the absence of H.R.M.S, we cite:

- The difficulty of tracking employee's attendances.
- The difficulty of tracking projects and their teams.
- The difficulty of exploring employee's records and history with the company.
- Higher risk when it comes to making wrong decisions.
- Higher risk when it comes to getting scammed from the employees.

2. Proposed solutions

Our solution here is to develop a Human resource management system in both web and mobile versions to facilitate the management of employees for the Human resources team and for project managers to manage their team and to track the actions such as:

- Managing enterprise projects
- Managing Employees Requests
- Consoling employee’s job history
- Managing enterprise side employee’s information
- Managing upcoming hire requests

- Managing Employees Documents Digitally

III. Used methodologies

1. Project management approach

In our project, we opted for the agile approach to manage the entire project.

We went with this choice because we found that the specifications include needs that are not very detailed and not clear. We have also predicted that the needs will evolve over time and this is already specified in the specifications.

All its indicators led us to choose an agile approach to manage the project, on the one hand to succeed in implementing the maximum possible of the requested functionalities, and on the other hand to succeed in resisting changes and new needs.

An agile approach can also save us time since it avoids the trap of the tunnel effect, which is one of the most famous and serious drawbacks of a classic project management approach.

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments.[1]

The agile approach is characterized by its flexibility and adaptability, unlike traditional project management approaches. Agile methods involve the requestor (client) as much as possible and allow great responsiveness to their requests.

Agile concepts and tools come from IT but Agile practices are now available for all trades: from innovative services to R&D in heavy industries.

There are several agile methods namely: Scrum, XP, RAD, DSDM.

2. Presentation of the used framework

We have defined the characteristics of agile methods. A study of the characteristics of the most used agile methods led us to choose the Scrum method. In fact, scrum indicates that the size of the team can be reduced and this is our case, another reason is that scrum is flexible in terms of the duration of the sprint (between 2 and 4 weeks).

Scrum is a framework that helps teams work together. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.[2]

Scrum belongs to the family of iterative and incremental methodologies and is based on agile principles and values.

Most often, Scrum experts, even its founders, describe it as a project management-oriented process framework or pattern that can incorporate different engineering methods or practices.

IV. Presentation and application of Scrum

In this part, we will present the product backlog and the list of actors acting in our project.

We will also present the definition of done, one of the artefacts of the Scrum method.

1. Product backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome. The product backlog is the single authoritative source for things that a team works on. [3]

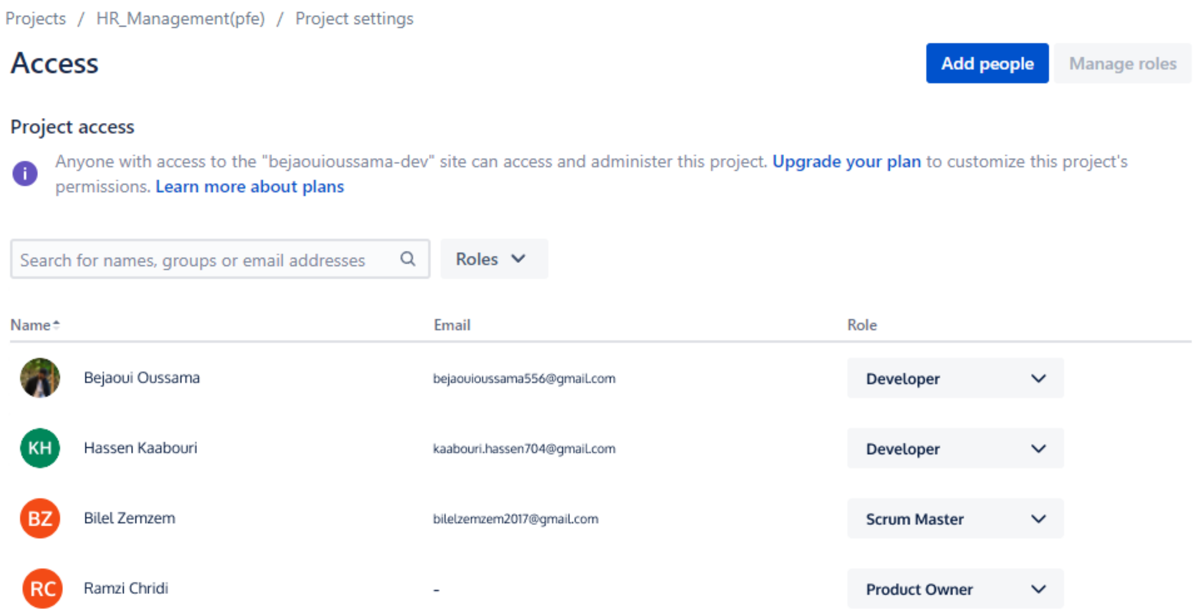
ID	AS	USER STORY	PRIORITY	DURATION
1	Administrator or manager	can manage Employees and Project managers	High	5 days
2	Administrator	can manage enterprise related settings for an employee (Contract Type/Salary/...)	High	4 days
3	Administrator	can manage teams and assign project manager and developers to a team	High	4 days
4	Manager	can manage his teams	High	2 days
5	Administrator	can manage any user's role	High	3 days
6	Employee	can manage his document space	High	5 days
7	Administrator	can manage any employee’s documents	High	3 days
8	Employee	can create leave requests	High	2 days
9	Administrator	can manage any employee’s leave requests	High	4 days
10	Employee or manager	can create specific requests	Medium	2 days
11	Administrator	can manage any employee’s specific requests	Medium	3 days
12	Manager	can approve or refuse his team member’s leave requests	Medium	3 days
13	Any user	can manage his personal informations	Medium	3 days

14	Administrator	can manage projects and affect them to managers	Medium	3 days
15	Administrator	can manage recruitment requests and respond to them	Medium	20 days
16		preparation of a graphic prototype for the web and mobile app	Medium	5 days
17		Mobile app development	Medium	20 days

We chose to work with Planning Poker, which is an agile estimating and planning technique, which is consensus-based. In the poker planning session, the product owner or customer reads an agile user story or describes a feature to the estimators, and then Each estimator will give the ideal days number that he thinks it will take to do that feature depending on its complexity. We have used the poker method to ensure the best estimation time for our stories and features.

2. List of actors

The scrum approach requires a project team made up of several actors. Each has a well-defined role. In our case, the role of Product Owner was assigned to Mr Ramzi Chridi (professional supervisor). A product owner expresses the client's needs and sets the deadlines for the delivery of deliverables. The role of Scrum master was assigned to Mr Bilel Zemzem (educational supervisor). A Scrum master is the leader of a team that applies Scrum. The development team consists of us and it is responsible for transforming the customer's needs into a functional product delivered on time.



Projects / HR_Management(pfe) / Project settings

Access

Project access

Anyone with access to the "bejaouioussama-dev" site can access and administer this project. Upgrade your plan to customize this project's permissions. Learn more about plans

Name	Email	Role
Bejaoui Oussama	bejaouioussama556@gmail.com	Developer
KH Hassen Kaabouri	kaabouri.hassen704@gmail.com	Developer
BZ Bilel Zemzem	bilelzemzem2017@gmail.com	Scrum Master
RC Ramzi Chridi	-	Product Owner

Figure 1: List of actors in the jira project

3. Artifact done by Scrum

This is one of the artifacts of the scrum Method. The development team needs to agree on the conditions for tasks and stories to move to the done state. In this sense, we agreed in collaboration with the product owner on the definition of done before starting the implementation of the first sprint. This definition requires that a task only switch to the done state if it has been integrated with the previously developed features and it works wonderfully. In addition, the task must be validated with the Product Owner.

4. Global use case diagram

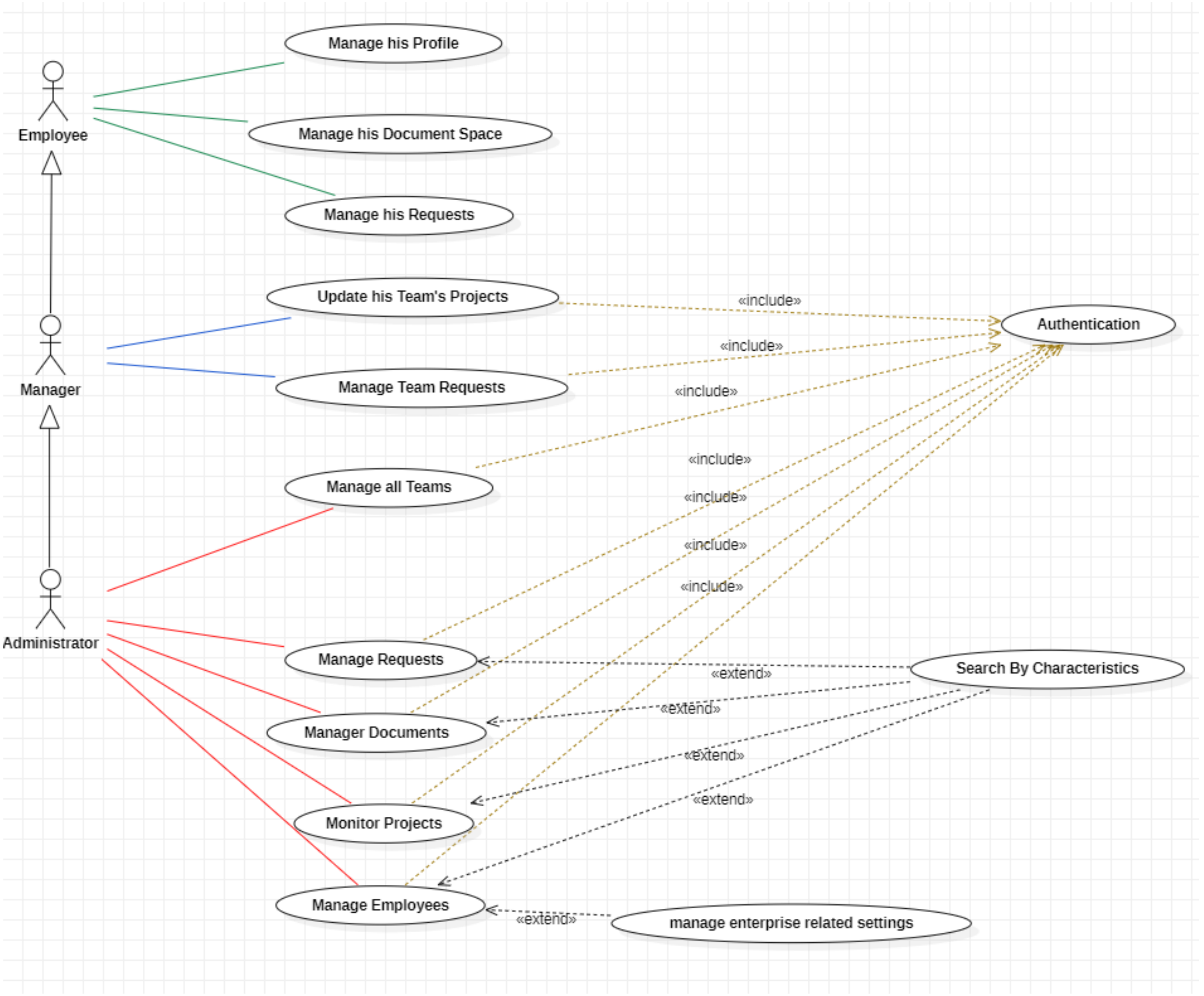
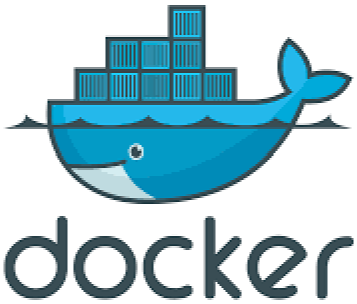


Figure 2: Global use case diagram

V. Presentation of the used technologies

In our project, there were requirements in terms of technologies to be used. They were required by the nature of the project.



Docker is a set of platforms as a service (PaaS) product that use OS-level virtualization to deliver software in packages called containers. The software that hosts the containers is called Docker Engine It was first started in 2013 and is developed by Docker, Inc. [3]

We choose to work with docker to ensure a safer, rapid and optimized environment for our project And to avoid compatibility problems between machines.

Nest (NestJS) is a framework for building efficient, scalable Node.js server-side applications. It uses progressive JavaScript, is built with and fully supports TypeScript (yet still enables developers to code in pure JavaScript) and combines elements of OOP (Object Oriented Programming), FP (Functional Programming), and FRP (Functional Reactive Programming).[4]

We went with NestJS for its quick and efficient development process and becaus it’s Highly scalable and easy to maintain applications. So, it’s the suitable framework for our backEnd.

StarUML is a UML tool by MKLab. The software was licensed under a modified version of GNU GPL until 2014, when a rewritten version 2.0.0 was released under a proprietary license. After being abandoned for some time, the project had a revival to move from Delphi to Java, before stopping again.[5]

We decided to work with StarUML for its simplicity and usefullness in our projects to help up create the best and most well-structured diagrams.

MySQL is an open-source relational database management system (RDBMS) that organizes data into one or more data tables in which data types may be related to each other. these relations help structure the data.[6]

We went with MySQL to ensure a well-structured and organized database in order to help us store our project’s data

Jira Software is an agile project management tool that supports any agile methodology, be it scrum, kanban, or your own unique flavor. From agile boards, backlogs, roadmaps, reports, to integrations and add-ons you can plan, track, and manage all your agile software development projects from a single tool. Pick a framework to see how Jira Software can help your team release higher quality software, faster.[7]

We choose Jira to help organize our sprints and tasks and get to most agility we could in our workflow.

Angular is a TypeScript-based free and open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations.[8]

We chose to use angular as our front-end framework due due to it's popularity and strength with the management type of application

Conclusion

This chapter is a general presentation of the framework. We first presented the company, and then we focused on the problem and the project to end up presenting the chosen project management method.

The next chapter will present the first sprint, which will contain some chosen features with a common goal. We will present the process and the advancement of the sprint features.

Chapter 2: Sprint 1:

Introduction

In this chapter, we will present our first sprint. This sprint's objectives are creating the prototype of the web application and implementing it, as long as building the employee management module for the admin and employees.

We have created the web application prototype to ensure a clean interface before starting coding it as long as implementing the option to update personal account information for users, the possibility to manage the employee's and project manager's enterprise information, and roles for administrators and managing teams in addition to assigning employees and project manager to a team.

I. Sprint Objectives

The objective for this first sprint is to:

- Prepare a graphic prototype for the web application.
- Implement a web interface where the administrator can manage employees, project managers, every enterprise related setting for the employees as long as their roles.

- Implement a web interface where any user can manage his personal informations(first name, last name, birth date, password and profile picture).
- Implement a web interface for managing teams and assigning team manager and members.

II. Sprint planning

In this part, we will present all the user stories and activities that will be treated by this sprint in a sprint backlog and we will present a schedule of the progress of this sprint.

1. Sprint backlog

In this part, we will present all the user stories and the activities that will be covered by this sprint in a sprint backlog and we will present a schedule for the progress of this sprint.

ID	User Story	Task ID	Task	Estimation
1	**preparation of a graphic prototype for the web application**	1.1	preparation of a graphic prototype for the web	5 days
2	**Administrator can manage employees and project managers**	2.1	Implementation of the user interface (UI)	4 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	

3	**Administrator can manage enterprise related settings for an employee (Contract Type/Salary/...)**	2.1	Implementation of the user interface (UI)	3 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	
4	**Administrator can manage any user's role**	2.1	Task automatization within the enterprise related settings functions in the backend	2 days
5	**Any user can manage his personal informations**	2.1	Implementation of the user interface (UI)	1 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	
6	**Administrator can manage teams and assign project manager and employees to a team**	2.1	Implementation of the user interface (UI)	4 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	
7	**Manager can manage his teams**	2.1	Implementation of the user interface (UI)	2 days

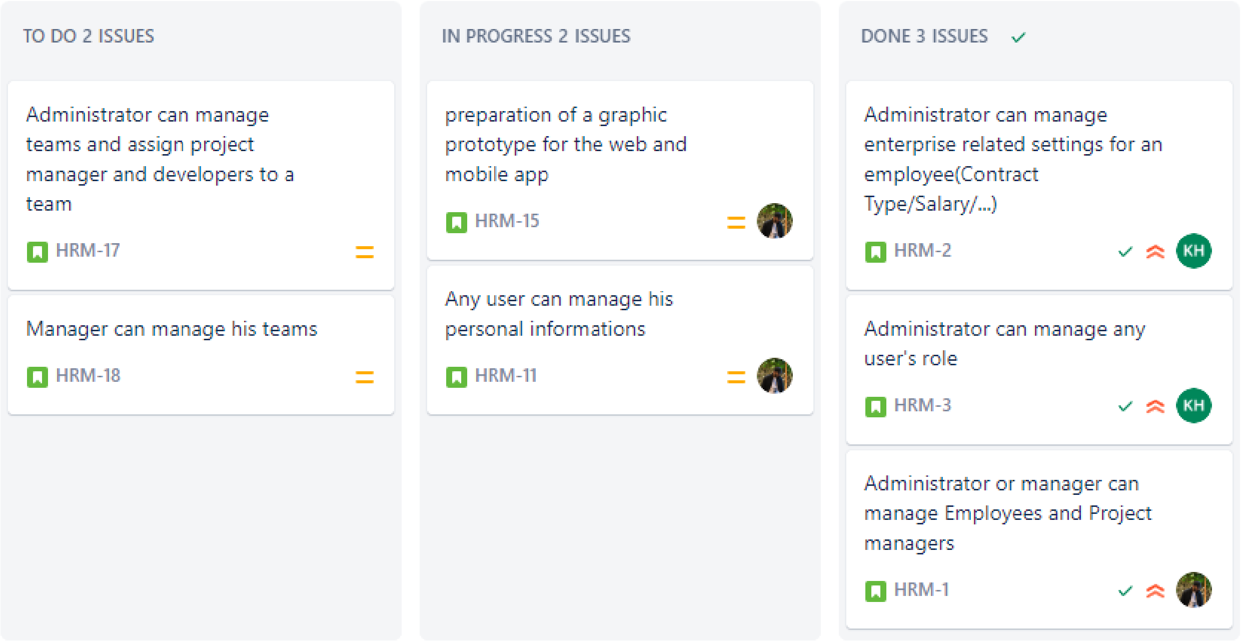
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	

2. Job description

This sprint addresses two of highest priority user stories, which are: the possibility of tracking and managing the employees where he can edit their enterprise informations and roles in addition to that, the option of managing teams. These two user stories belong to the WEB application. The objective of this sprint is to set up the "dashboard" web interface and display the employees and teams with the possibility to search with different features such as name and project managers and also implementing the "CRUD" interfaces.

3. SCRUM Board

To be able to share and track the tasks to be done, an agile project management software seems indispensable. We opted for a tool called Jira. It is a proprietary project management tool and an issue tracking product developed by Atlassian that allows bug tracking and agile project management. It also offers a scrum board that allows the monitoring of the life cycle of tasks. In this part, we will present the scrum board of sprint 1 which contains its user stories, each user story contains a set of related tasks.



TO DO 2 ISSUES

- Administrator can manage teams and assign project manager and developers to a team — HRM-17
- Manager can manage his teams — HRM-18

IN PROGRESS 2 ISSUES

- preparation of a graphic prototype for the web and mobile app — HRM-15
- Any user can manage his personal informations — HRM-11

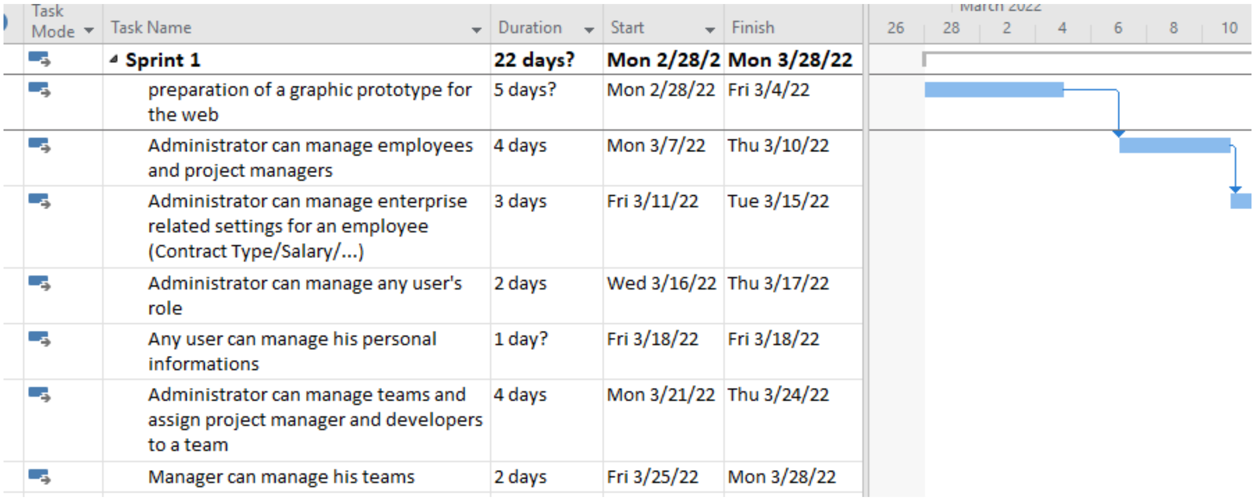
DONE 3 ISSUES ✓

- Administrator can manage enterprise related settings for an employee(Contract Type/Salary/...) — HRM-2 ✓ KH
- Administrator can manage any user's role — HRM-3 ✓ KH
- Administrator or manager can manage Employees and Project managers — HRM-1 ✓

Figure 3: Scrum Board sprint 1

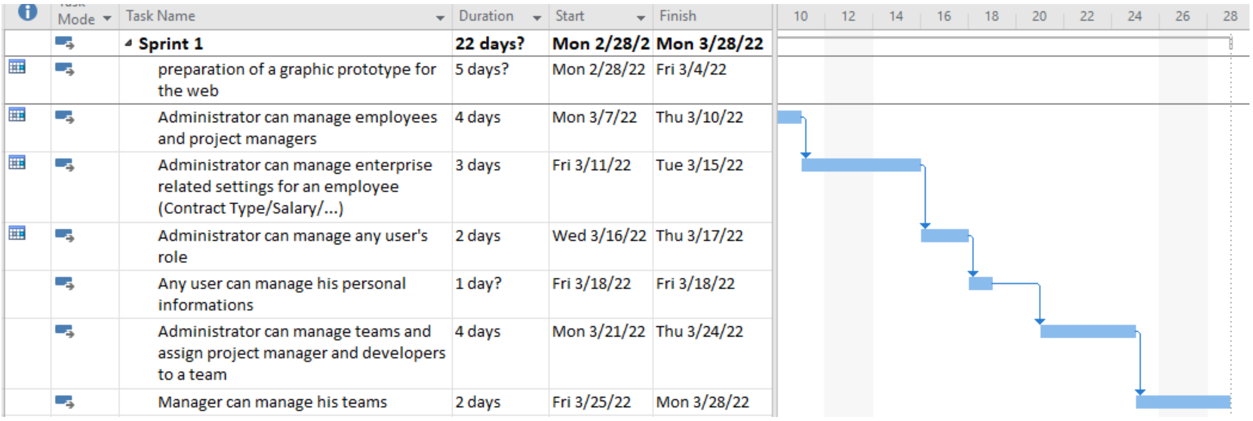
4. Presentation of the sprint process

In what follows, we will present the provisional schedule for the first sprint. We start by presenting a Gantt chart that represents the chronological sequence of tasks.



Task Mode	Task Name	Duration	Start	Finish
	Sprint 1	**22 days?**	**Mon 2/28/2**	**Mon 3/28/22**
	preparation of a graphic prototype for the web	5 days?	Mon 2/28/22	Fri 3/4/22
	Administrator can manage employees and project managers	4 days	Mon 3/7/22	Thu 3/10/22
	Administrator can manage enterprise related settings for an employee (Contract Type/Salary/...)	3 days	Fri 3/11/22	Tue 3/15/22
	Administrator can manage any user's role	2 days	Wed 3/16/22	Thu 3/17/22
	Any user can manage his personal informations	1 day?	Fri 3/18/22	Fri 3/18/22
	Administrator can manage teams and assign project manager and developers to a team	4 days	Mon 3/21/22	Thu 3/24/22
	Manager can manage his teams	2 days	Fri 3/25/22	Mon 3/28/22

Figure 4: Gantt chart of Sprint 1 tasks [1/2]



Task Mode	Task Name	Duration	Start	Finish
	Sprint 1	**22 days?**	**Mon 2/28/2**	**Mon 3/28/22**
	preparation of a graphic prototype for the web	5 days?	Mon 2/28/22	Fri 3/4/22
	Administrator can manage employees and project managers	4 days	Mon 3/7/22	Thu 3/10/22
	Administrator can manage enterprise related settings for an employee (Contract Type/Salary/...)	3 days	Fri 3/11/22	Tue 3/15/22
	Administrator can manage any user's role	2 days	Wed 3/16/22	Thu 3/17/22
	Any user can manage his personal informations	1 day?	Fri 3/18/22	Fri 3/18/22
	Administrator can manage teams and assign project manager and developers to a team	4 days	Mon 3/21/22	Thu 3/24/22
	Manager can manage his teams	2 days	Fri 3/25/22	Mon 3/28/22

Figure 5: Gantt chart of Sprint 1 tasks [2/2]

This diagram contains all the tasks related to Sprint 1, their estimated durations and the dependencies that link them.

III. Non-formal requirements specification

In what follows, we will analyze and specify the functional and non-functional requirements that must be met by the application at the end of this iteration.

1. Functional needs

The functional requirements of this sprint are summarized in the implementation of the web interface. We will present the admin dashboard interfaces for managing the employees by viewing (multiple searches are implemented such as searching by name, projects manager or start date), editing, deleting or adding them to teams which will bring us to the second part which is managing the teams, the admin will be able to create new teams assign them to managers he could also edit and delete them and last but not least every user in the platform has the access to change his personal informations.

2. Non-Functional needs

The non-functional needs of this sprint are to ensure a scalable and optimized project on the server side as well as the client side.

To ensure the confidentiality of the stored data on server side, authentication is required in addition to restricting access to users depends on their role and permissions.

To make our client-side project scalable we created well organized structure that contains 2 main module the core module which contains every core element of the app such as the guards and interceptors and the shared module which contains all shared components or services between the admin, manager and employee as well as UI library containing all the UI elements to make it easier and faster to upgrade the application.

IV. Semi-formal specification of needs

Before starting the design part, a more in-depth study of the functional needs that our application must meet is essential. In this part, we will describe the requirements using a set of UML diagrams. We have chosen to present use case diagrams, system sequence diagrams and some textual descriptions.

1. Modeling Language

UML, short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object-oriented software and the software development process.[10]

2. Use case diagram

Below, we present the use cases which present a semi-formal description of the functional needs corresponding to this iteration.

We begin with use case diagram which describe the management of employees and the employee’s enterprise related settings, the management of teams for the administrator, the management of the manager’s teams and the management of personal profile.

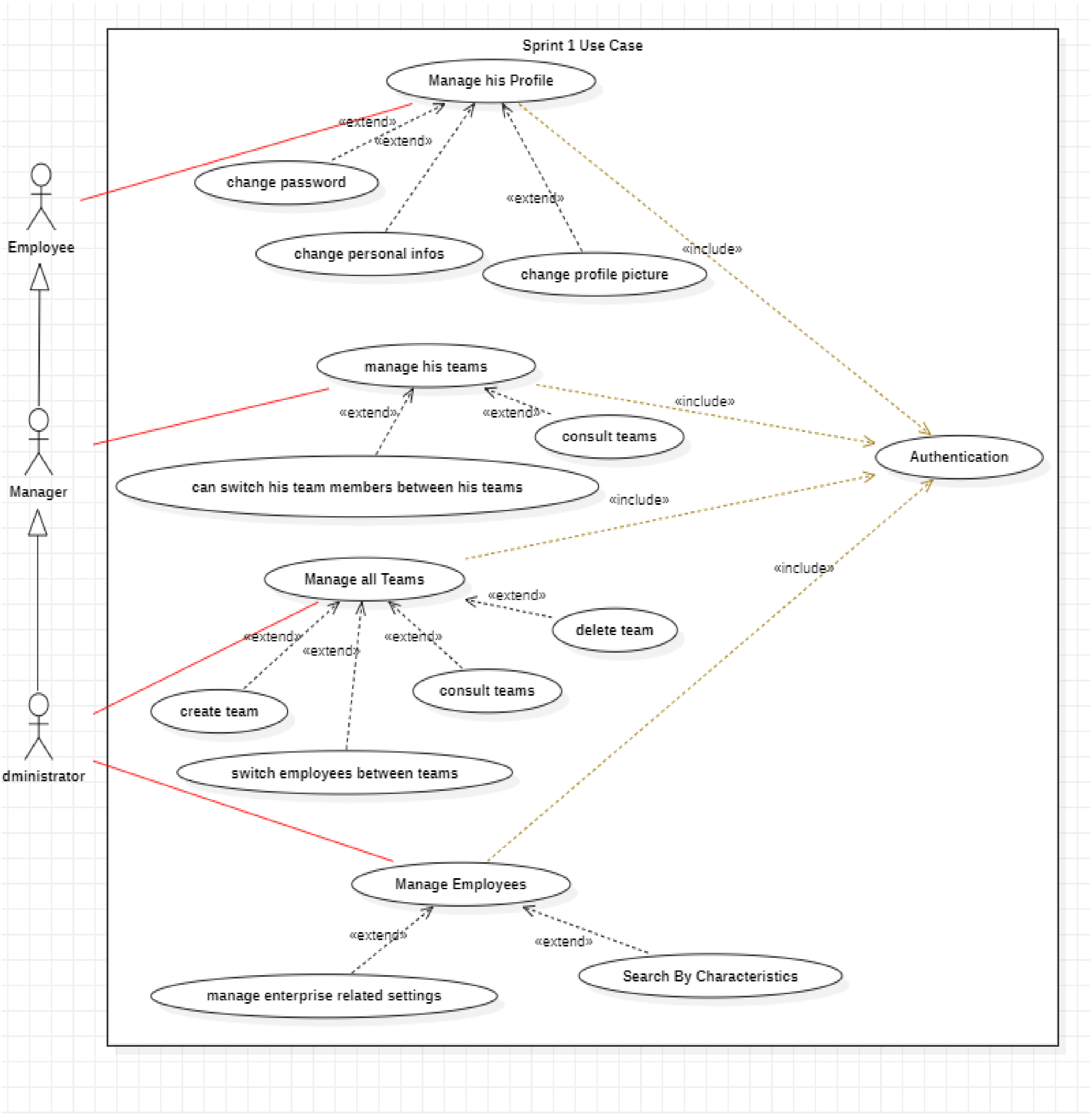


Figure 6: Sprint 1 use case diagram

The use case diagram above represents the functional needs of the management of employees and the employee’s enterprise related settings, the management of teams for the administrator, the management of the manager’s teams and the management of personal profile.

❖ Textual Description of use case « change password »:

In the following, we present the textual description associated with the use case of changing password.

Use Case	-Change password.
Actors	Employee, Manager and Administrator.
Precondition	- Functional server - Authenticated user
Post-condition	-User’s profile is modified (personal informations, password or profile picture).
Basic scenario description	1-User click on “settings”. 2-User click on change password. 3- The system shows an empty form inside a module. 4-User type the old password and new password twice to confirm it 5-Click submit button. 6-Server saves modifications and return the result.

	7-Session logout. 8-User login using new password.
Exception	- The server is down, in this case the user is warned.

The textual description above describes the use case diagram " change password " by indicating the actors, the post conditions, the pre conditions and the exceptions that can be refocused.

❖ Textual Description of use case « create team »:

In the following, we present the textual description associated with the use case of creating a team.

Use Case	-Create a team.
Actors	Administrator.
Precondition	- Functional server - Authenticated user
Post-condition	-Team created.
Basic scenario description	1-User clicks on “teams” on the sidebar. 2-User clicks on “create a team”. 3- The system shows an empty form inside a module. 2-User type the team’s name.

	3-Select the team manager. 4-Select the team members. 5-Click the submit button. 6-Server saves the new team and modify the team of the selected members. 7-the system refreshes the page and shows a success message.
Exception	- The server is down, in this case the user is warned.

The textual description above describes the use case diagram " create a team " by indicating the actors, the post conditions, the pre conditions and the exceptions that can be refocused.

- **Textual Description of use case « delete a team »:**

In the following, we present the textual description associated with the use case of creating a team.

Use Case	-Delete a team.
Actors	Administrator.
Precondition	- Functional server - Authenticated user
Post-condition	-Team deleted.

Basic scenario description	1-User clicks on “teams” on the sidebar. 2-User clicks on the button “view team”. 3-The system shows the team. 4-click on “delete team” 5- The system shows a model to confirm the deletion of the team. 6-click on “Yes, Delete it.” 7-Server deletes the team. 8-The system redirect to the teams list.
Exception	- The server is down, in this case the user is warned.

The textual description above describes the use case diagram "create a team " by indicating the actors, the post conditions, the pre-conditions and the exceptions that can be refocused.

3. Conception

In this section, we are going to present the class, the communication and the sequence diagrams of this sprint.

A. Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code.[11]

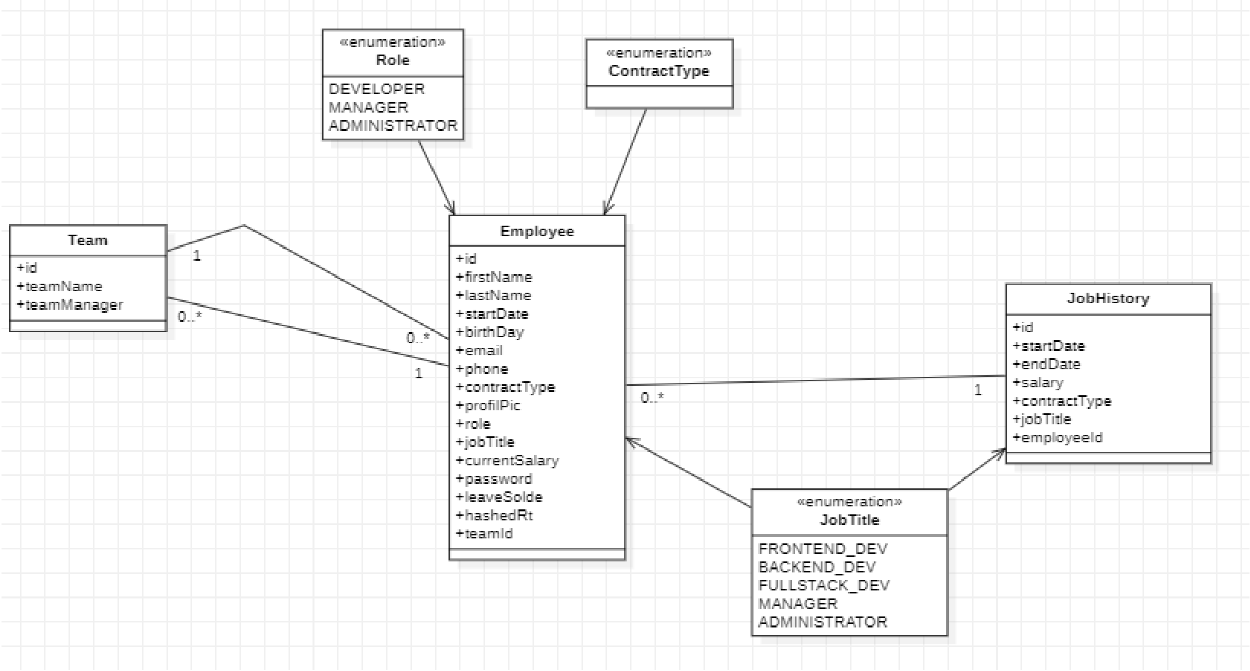


Figure 7: Sprint 1 class diagram

B. Communication Diagram

Communication diagrams, formerly known as collaboration diagrams, are almost identical to sequence diagrams in UML, but they focus more on the relationships of objects—how they associate and connect through messages in a sequence rather than interactions. [12]

The figure below represents the communication diagram of the use case «Manage employees »

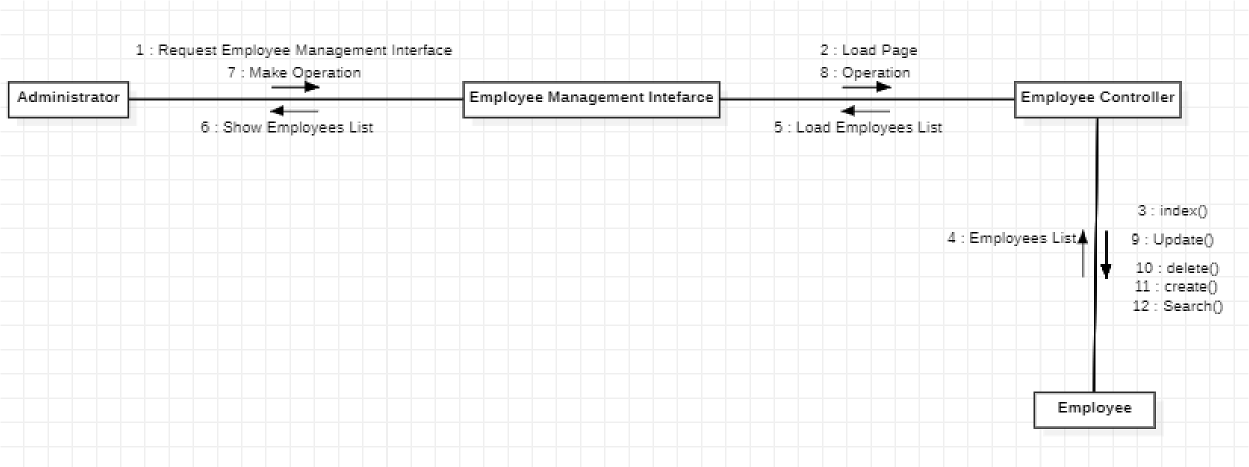


Figure 8: Communication diagram «Manage employees»

C. *Sequence Diagrams*

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when. [13]

- **Sequence diagram of use case « Add Employee »**

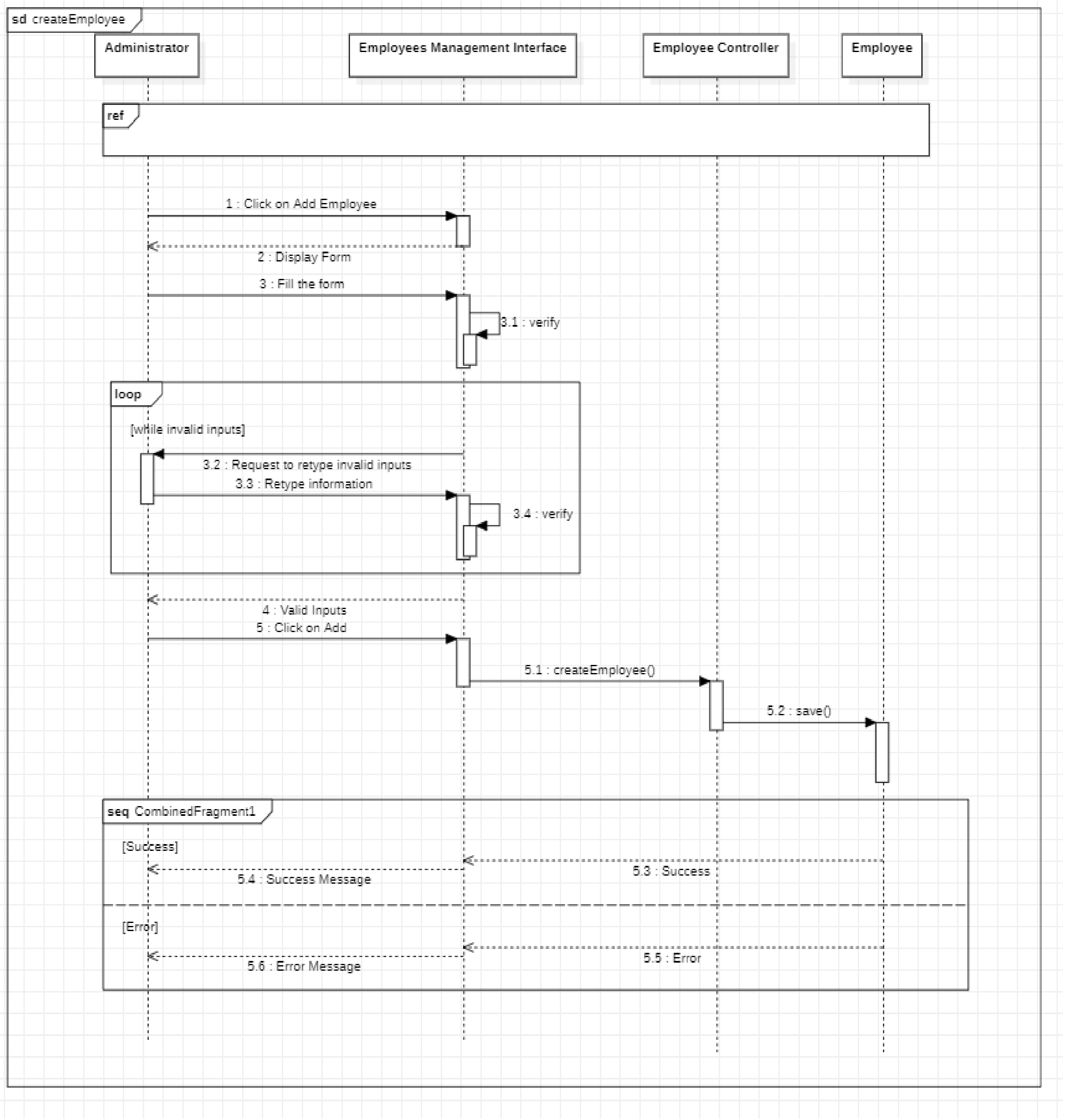


Figure 9: Sequence diagram «Add employee»

❖ **Sequence diagram of use case « Delete Team »**

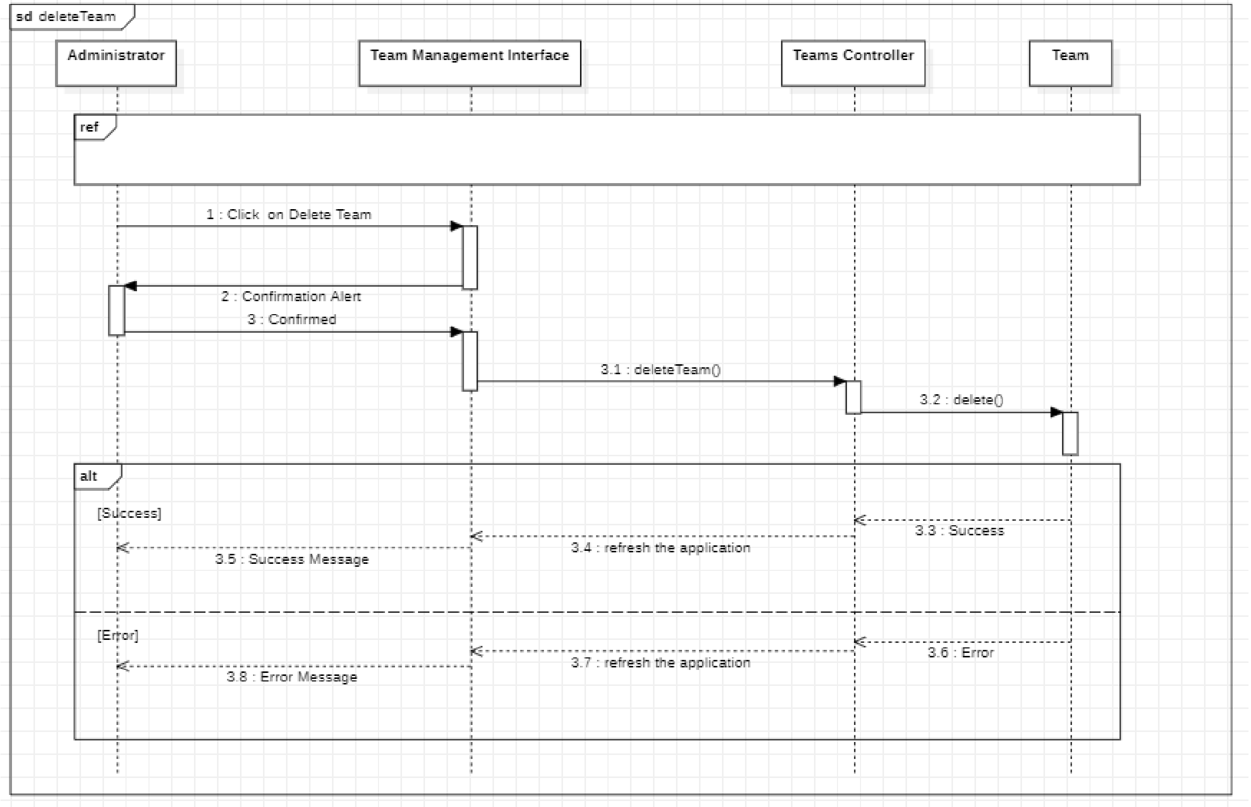


Figure 10: Sequence diagram «Delete employees»

V. Wireframe and prototyping

Prototyping is a very useful task in the development of IT projects. It helps to avoid falling into the tunnel effect, one of the problems that are often encountered when working with a traditional project management method. An interface prototype is discussed with the client and the product owner and this is how you avoid the tunnel effect and ensure that the work is going in the right direction.

In the sake of having the best result possible, we used figma, which is a platform used to create user interfaces with the need and use of the tailwind design system to create our web application UI.

During this section, we will show our wireframe (low fidelity prototype) and our main prototype as long as explain how we accomplished them and what tools we used.

1. Wireframe

Before we start creating our UI, we need to create a wireframe or what we call a low fidelity prototype which is a very simple beta version of the UI. It’s so important because it will help us get a much clear vision about the final product.

In the figures below, we will present some of the main pages wireframes

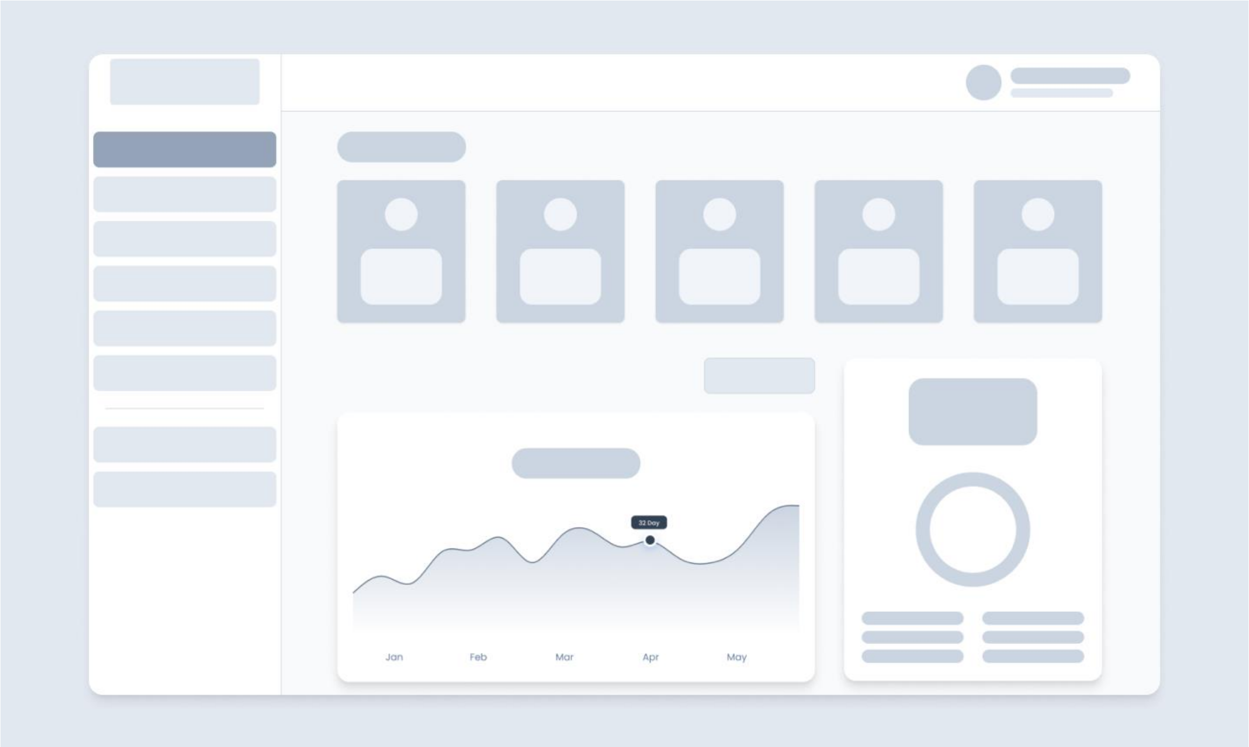


Figure 11: General interface for administrator «Wireframe»

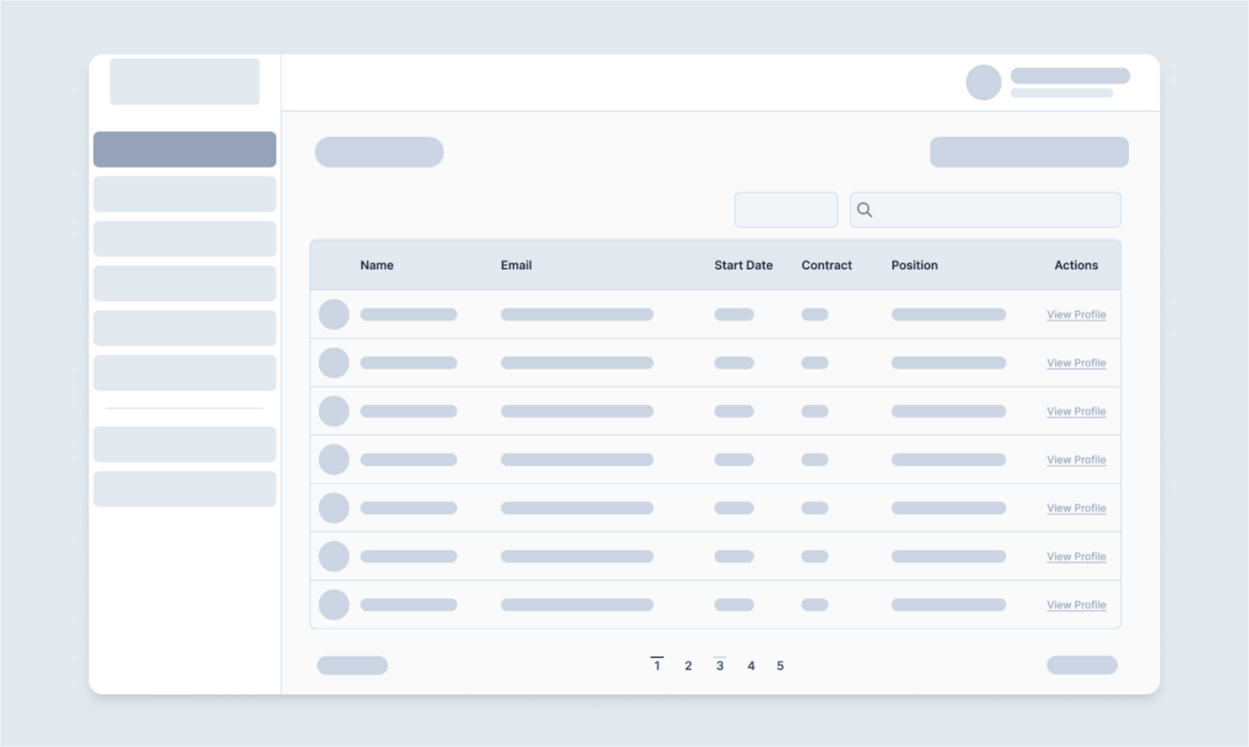
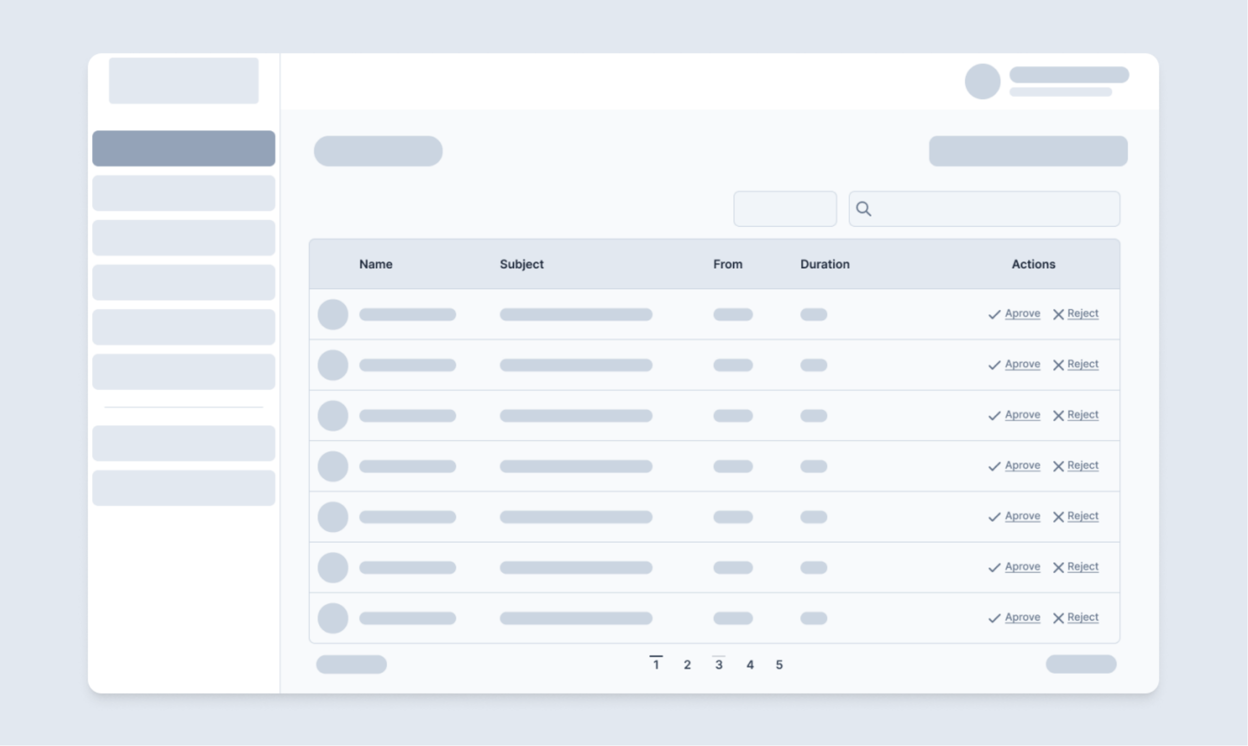


Figure 12: Employees table for administrator «Wireframe»



Name	Subject	From	Duration	Actions
				✓ Aprove ✕ Reject
				✓ Aprove ✕ Reject
				✓ Aprove ✕ Reject
				✓ Aprove ✕ Reject
				✓ Aprove ✕ Reject
				✓ Aprove ✕ Reject
				✓ Aprove ✕ Reject

Figure 13: Leave requests table for administrator and manager «Wireframe»

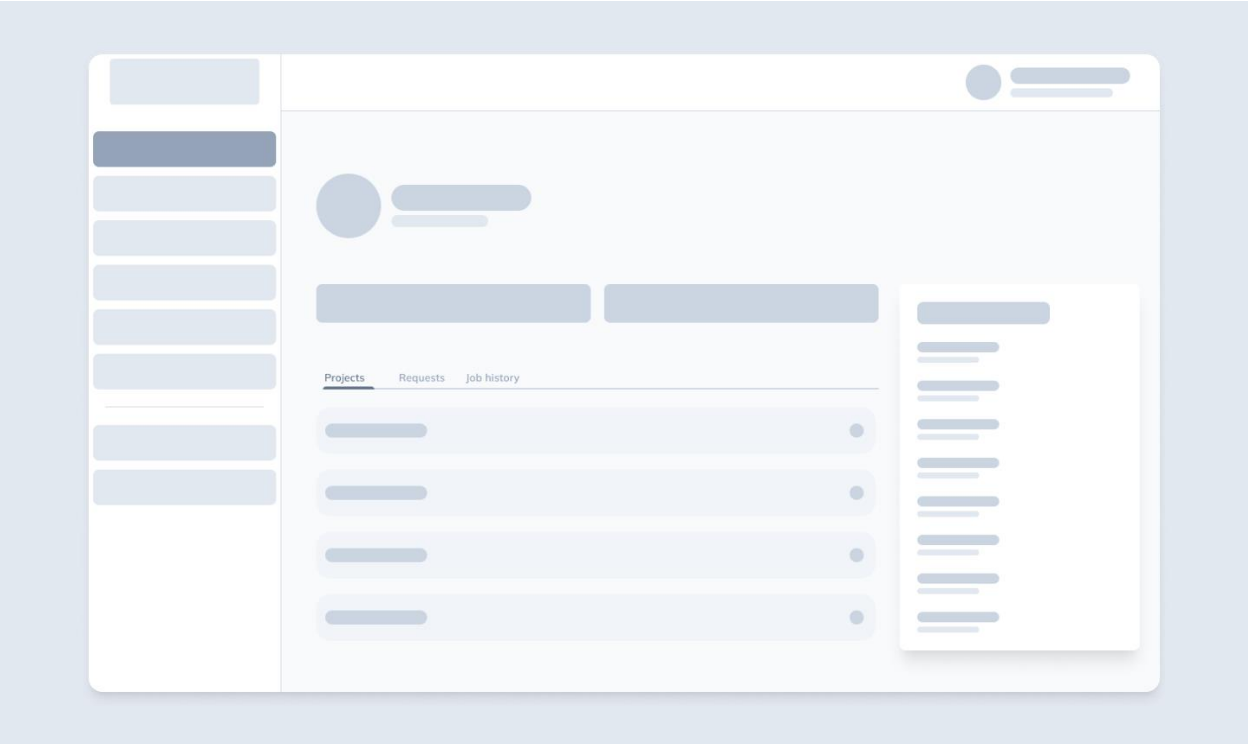


Figure 14: Employee profile «Wireframe»

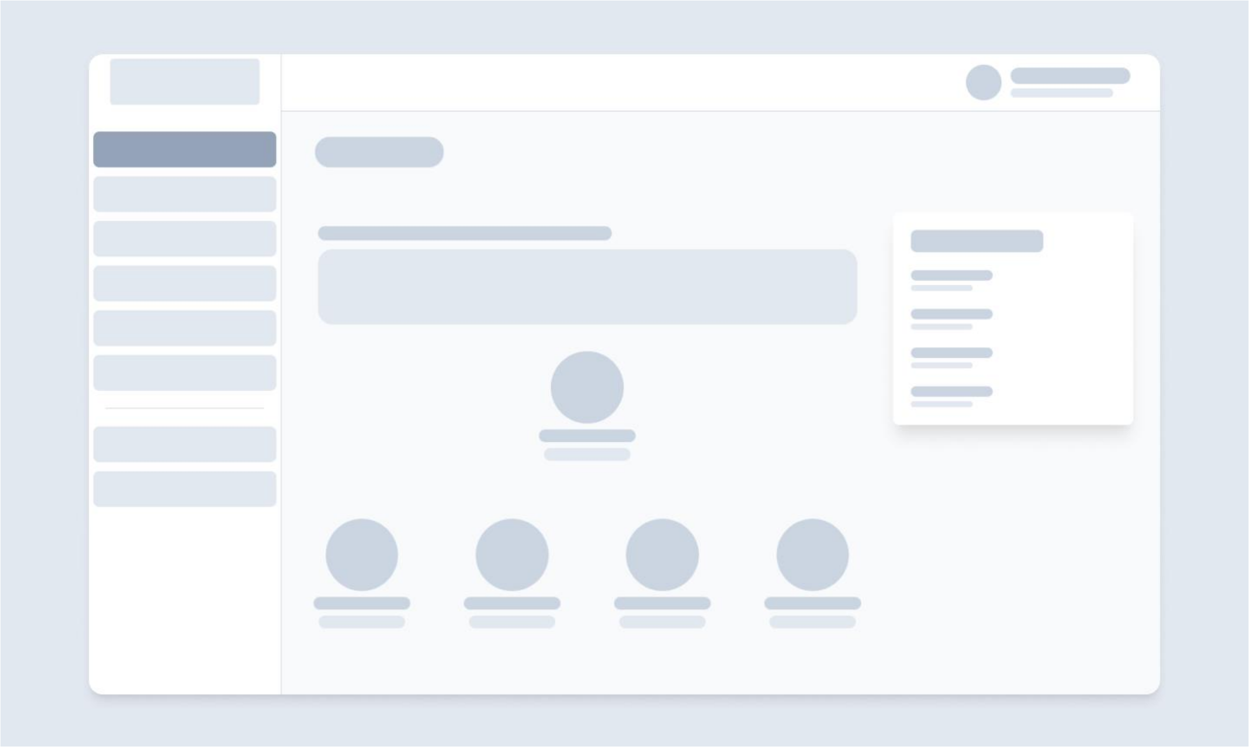


Figure 15: Project profile «Wireframe»

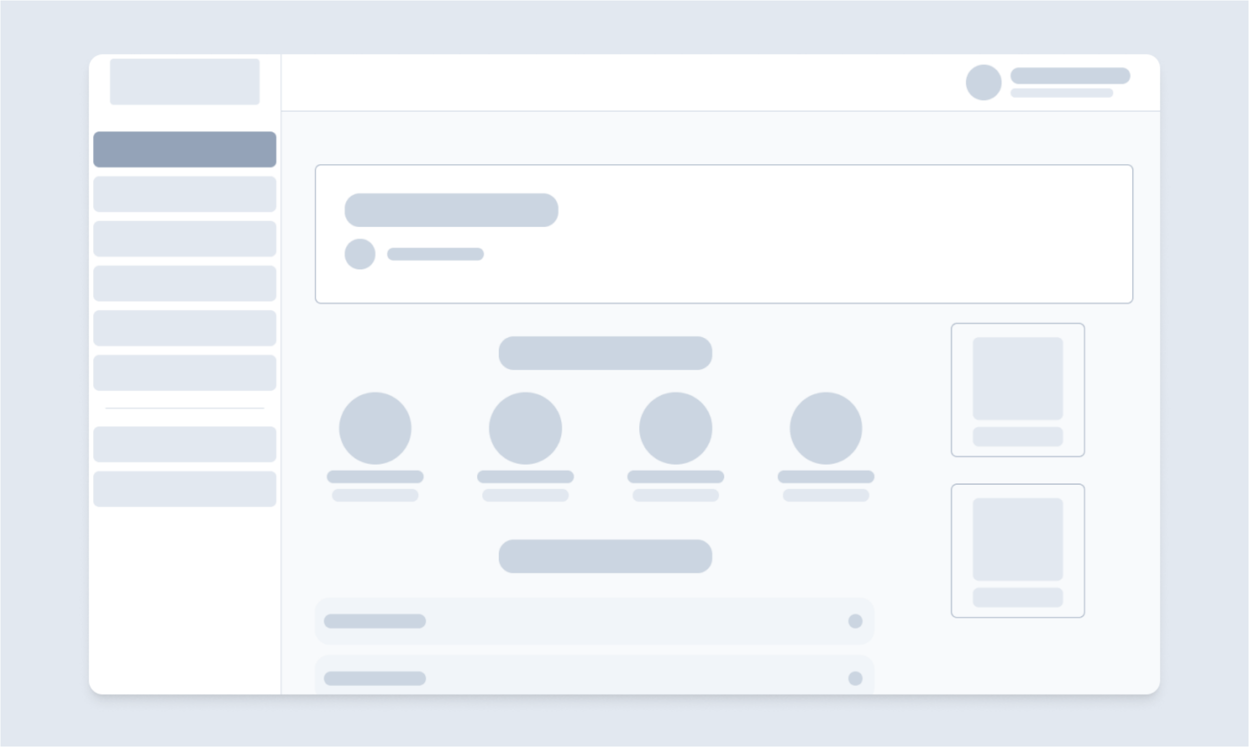


Figure 16: Team profile «Wireframe»

Now that we have highlighted some of the main wireframes, we will move to the next step, which is designing them to the final result.

2. Prototype

Prototyping is the final step before we start implementing our ui because it will be the guide during the client side implementation to prevent us from any time waste or conflicts

During the figures bellow, we will present to you some of the main final interfaces

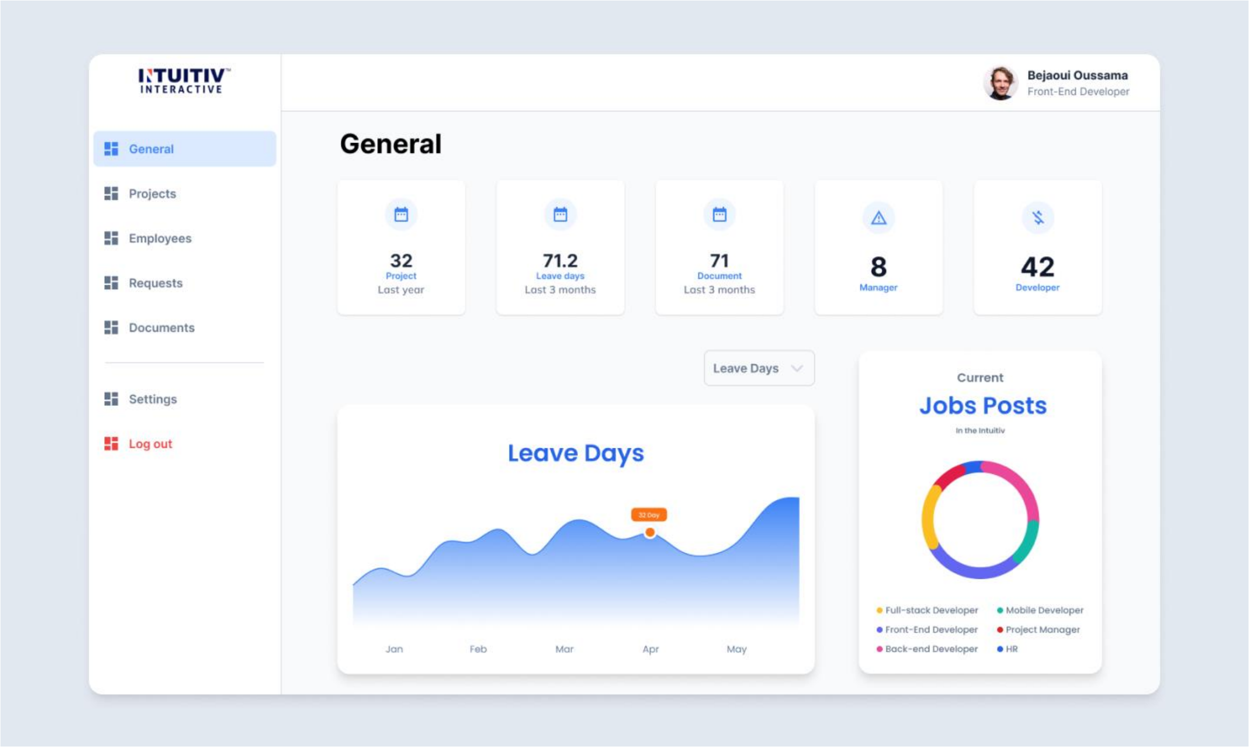
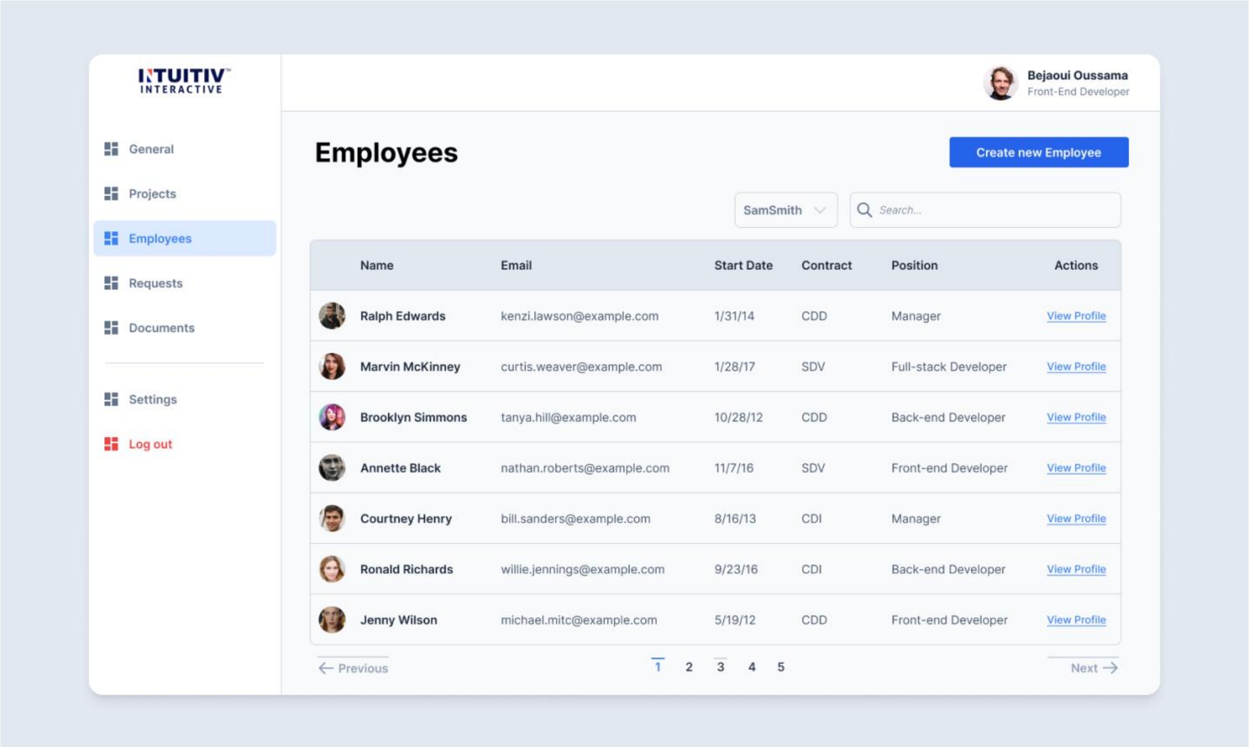


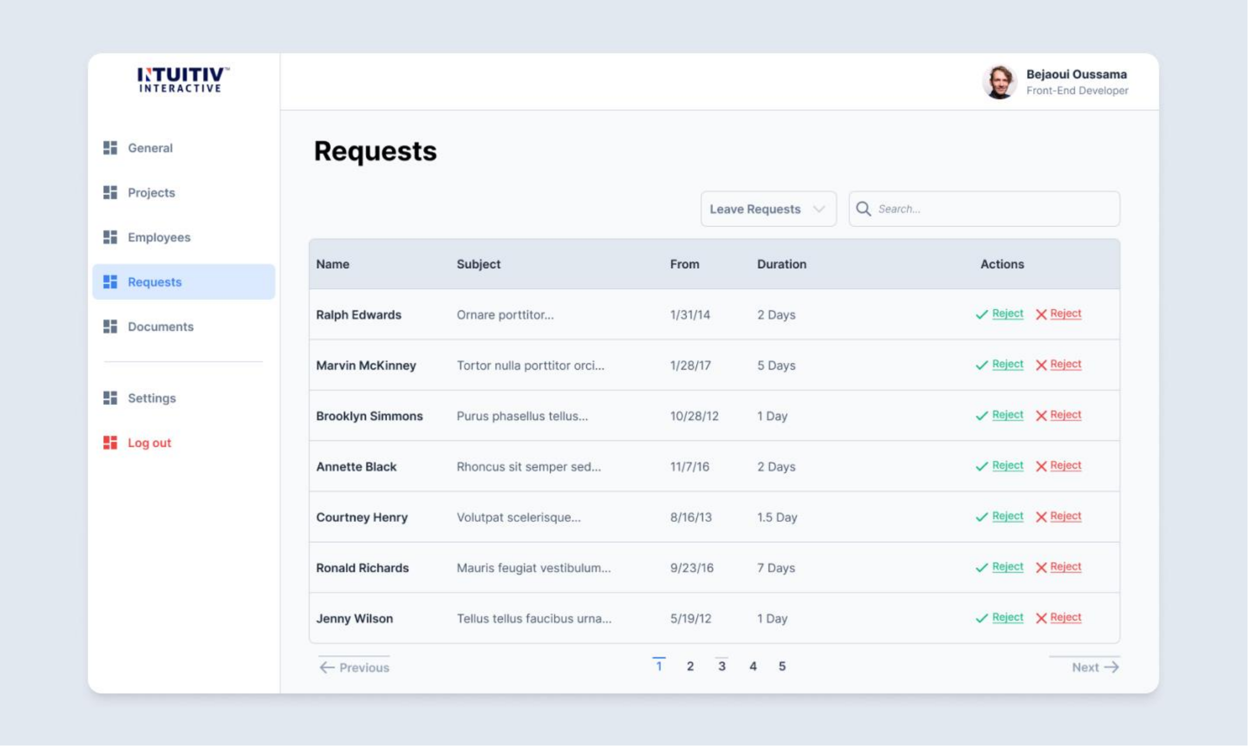
Figure 17: General interface for administrator «Prototype»



Employees

Name	Email	Start Date	Contract	Position	Actions
Ralph Edwards	kenzi.lawson@example.com	1/31/14	CDD	Manager	View Profile
Marvin McKinney	curtis.weaver@example.com	1/28/17	SDV	Full-stack Developer	View Profile
Brooklyn Simmons	tanya.hill@example.com	10/28/12	CDD	Back-end Developer	View Profile
Annette Black	nathan.roberts@example.com	11/7/16	SDV	Front-end Developer	View Profile
Courtney Henry	bill.sanders@example.com	8/16/13	CDI	Manager	View Profile
Ronald Richards	willie.jennings@example.com	9/23/16	CDI	Back-end Developer	View Profile
Jenny Wilson	michael.mitc@example.com	5/19/12	CDD	Front-end Developer	View Profile

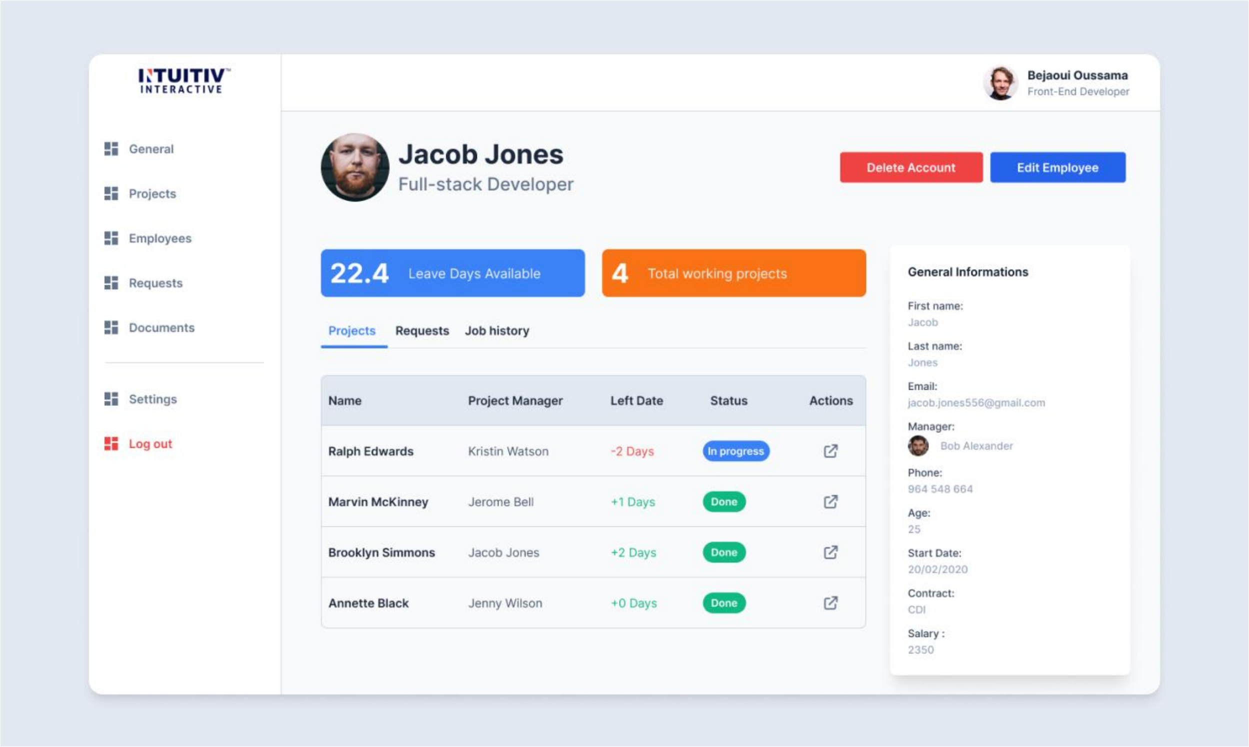
Figure 18: Employees table for administrator «Prototype»



Requests

Name	Subject	From	Duration	Actions
Ralph Edwards	Ornare porttitor...	1/31/14	2 Days	✓ Reject ✗ Reject
Marvin McKinney	Tortor nulla porttitor orci...	1/28/17	5 Days	✓ Reject ✗ Reject
Brooklyn Simmons	Purus phasellus tellus...	10/28/12	1 Day	✓ Reject ✗ Reject
Annette Black	Rhoncus sit semper sed...	11/7/16	2 Days	✓ Reject ✗ Reject
Courtney Henry	Volutpat scelerisque...	8/16/13	1.5 Day	✓ Reject ✗ Reject
Ronald Richards	Mauris feugiat vestibulum...	9/23/16	7 Days	✓ Reject ✗ Reject
Jenny Wilson	Tellus tellus faucibus urna...	5/19/12	1 Day	✓ Reject ✗ Reject

Figure 19: Leave requests table for administrator and manager «Prototype»



Bejaoui Oussama
Front-End Developer

Jacob Jones
Full-stack Developer

22.4 Leave Days Available

4 Total working projects

Name	Project Manager	Left Date	Status	Actions
Ralph Edwards	Kristin Watson	-2 Days	In progress	
Marvin McKinney	Jerome Bell	+1 Days	Done	
Brooklyn Simmons	Jacob Jones	+2 Days	Done	
Annette Black	Jenny Wilson	+0 Days	Done	

General Informations

First name:
Jacob

Last name:
Jones

Email:
jacob.jones556@gmail.com

Manager:
Bob Alexander

Phone:
964 548 664

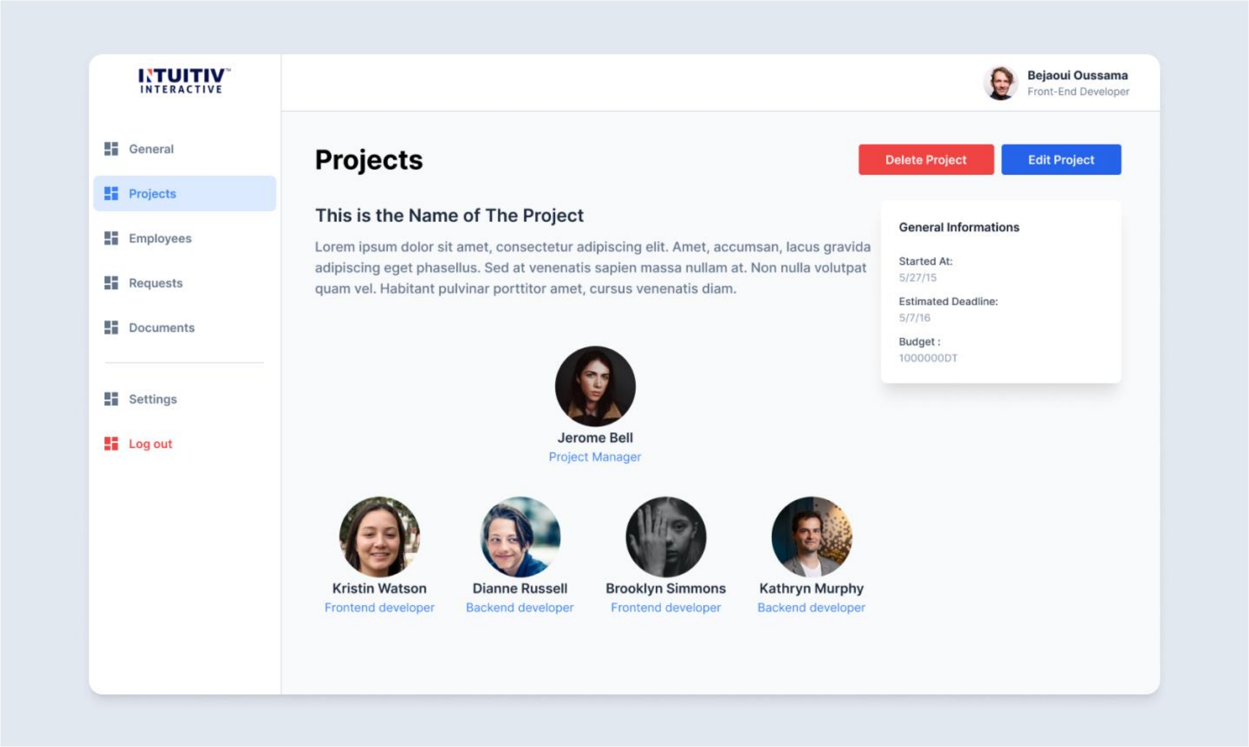
Age:
25

Start Date:
20/02/2020

Contract:
CDI

Salary :
2350

Figure 20: Employee Profile «Prototype»



Projects

This is the Name of The Project

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Amet, accumsan, lacus gravida adipiscing eget phasellus. Sed at venenatis sapien massa nullam at. Non nulla volutpat quam vel. Habitant pulvinar porttitor amet, cursus venenatis diam.

General Informations

Started At:
5/27/15

Estimated Deadline:
5/7/16

Budget :
1000000DT

Jerome Bell
Project Manager

Kristin Watson
Frontend developer

Dianne Russell
Backend developer

Brooklyn Simmons
Frontend developer

Kathryn Murphy
Backend developer

Figure 21: Project profile «Prototype»

Throughout this section, we presented some of the main final user interfaces from out prototype to get a better vision, in the next section we will explain the implementation of the UI's of this chapter sprint.

VI. Implementation and testing

In this section, we will start a presentation of the developed functionalities and the tools used to accomplish them.

1. Development tools used

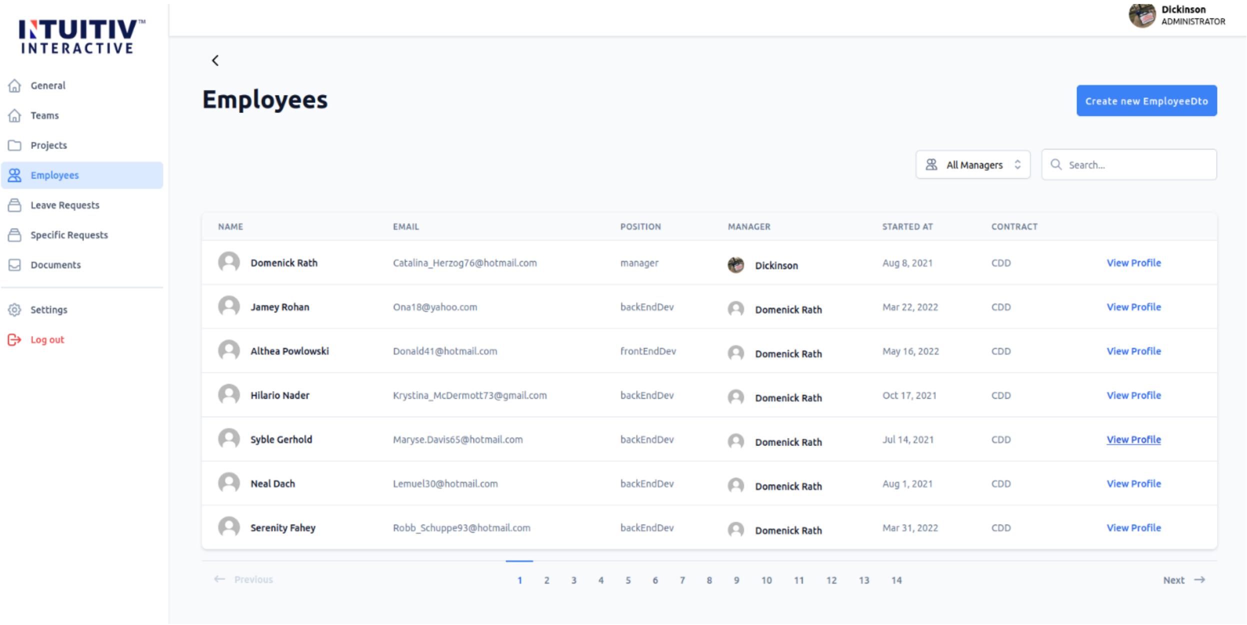
To accomplish this iteration, a set of technologies was used which will be detailed in the following. For the implementation of the GUI, we used Angular, which is a typescript front-end framework in addition to the css library tailwindui to be able to implement our design. To ensure reliable and fast communication between the client and the server, we opted for the …--

2. Description of the implementation

To implement this sprint, we started with the requirements analysis and specification phase. We drew up use case diagrams and system sequence diagrams for a better understanding of the problem. Then we moved on to the design phase. The class diagram was developed after trying to normalize the SQL data model. Then we created a prototype of the interface. This prototype was discussed with the Product Owner Mr. Chridi Ramzi and it was confirmed. Then, we spent some time self-training on NestJs, Angular and Tailwind technologies.Then we've built the docker image, Then we build the api calls related to this sprint. After that, we implemented the user interface using Angular and TailwindUI. We got stuck in the UI implemation section since we had to create our own custom UI libray for Angular and tailwind since the tailwindUI doesn't support Angular, at the same time we kept on preparing the services and some of the core funcionalities of the project to avoid any waste of time. And finally, we merged both side and tested them.

3. Interfaces implementation

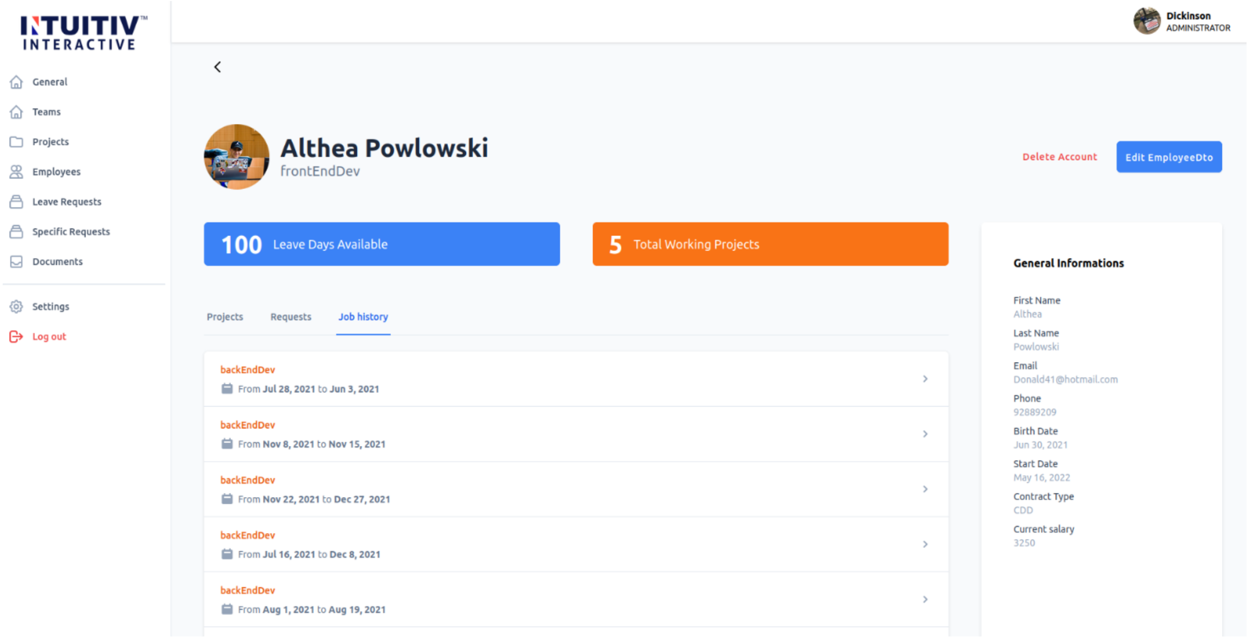
In the following part, we will present to you the implemented web interfaces during this sprint



Employees

NAME	EMAIL	POSITION	MANAGER	STARTED AT	CONTRACT	
Domenick Rath	Catalina_Herzog76@hotmail.com	manager	Dickinson	Aug 8, 2021	CDD	View Profile
Jamey Rohan	Ona18@yahoo.com	backEndDev	Domenick Rath	Mar 22, 2022	CDD	View Profile
Althea Powlowski	Donald41@hotmail.com	frontEndDev	Domenick Rath	May 16, 2022	CDD	View Profile
Hilario Nader	Krystina_McDermott73@gmail.com	backEndDev	Domenick Rath	Oct 17, 2021	CDD	View Profile
Syble Gerhold	Maryse.Davis65@hotmail.com	backEndDev	Domenick Rath	Jul 14, 2021	CDD	View Profile
Neal Dach	Lemuel30@hotmail.com	backEndDev	Domenick Rath	Aug 1, 2021	CDD	View Profile
Serenity Fahey	Robb_Schuppe93@hotmail.com	backEndDev	Domenick Rath	Mar 31, 2022	CDD	View Profile

Figure 22: Employee table page



Althea Powlowski
frontEndDev

100 Leave Days Available

5 Total Working Projects

backEndDev
From Jul 28, 2021 to Jun 3, 2021

backEndDev
From Nov 8, 2021 to Nov 15, 2021

backEndDev
From Nov 22, 2021 to Dec 27, 2021

backEndDev
From Jul 16, 2021 to Dec 8, 2021

backEndDev
From Aug 1, 2021 to Aug 19, 2021

General Informations

First Name
Althea

Last Name
Powlowski

Email
Donald41@hotmail.com

Phone
92889209

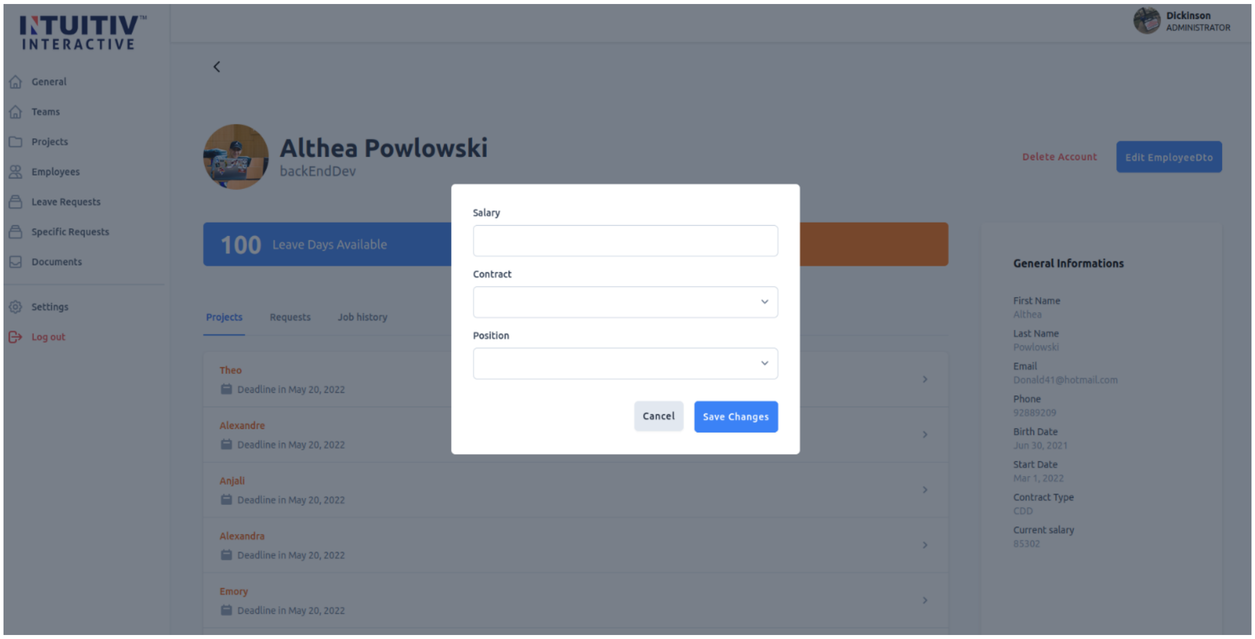
Birth Date
Jun 30, 2021

Start Date
May 16, 2022

Contract Type
CDD

Current salary
3250

Figure 23: Employee profile page (job history section)



INTUITIV INTERACTIVE

General
Teams
Projects
Employees
Leave Requests
Specific Requests
Documents
Settings
Log out

Dickinson
ADMINISTRATOR

Althea Powlowski
backEndDev

Delete Account
Edit EmployeeDto

100 Leave Days Available

Projects Requests Job history

Theo
Deadline in May 20, 2022

Alexandre
Deadline in May 20, 2022

Anjali
Deadline in May 20, 2022

Alexandra
Deadline in May 20, 2022

Emory
Deadline in May 20, 2022

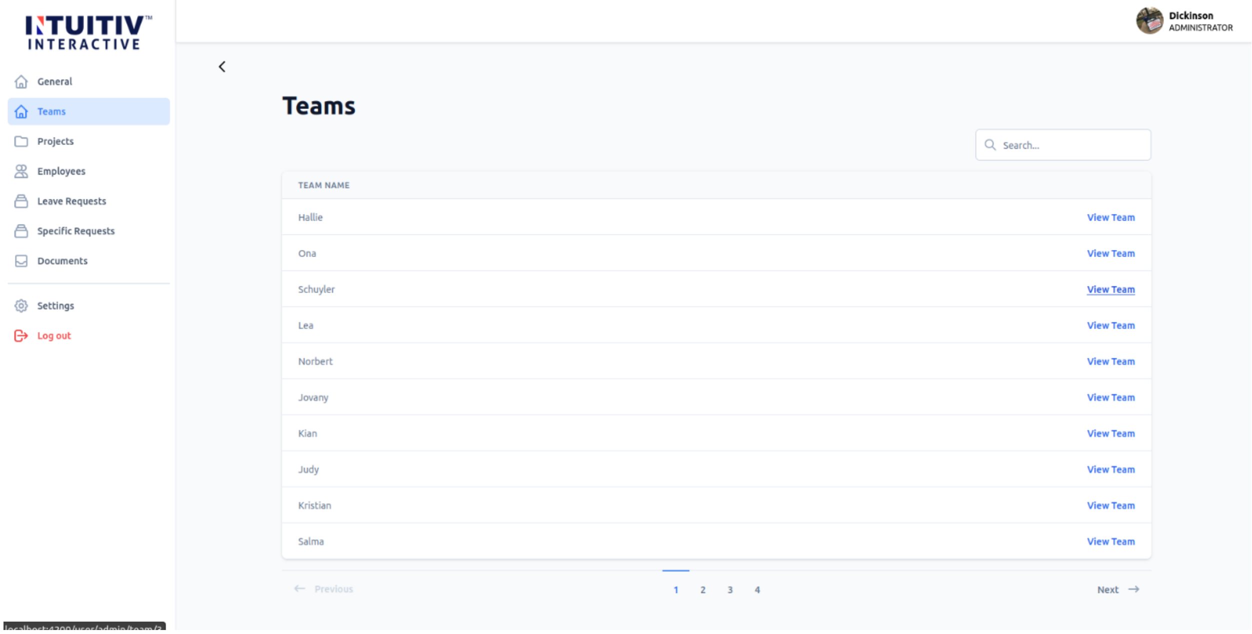
General Informations

First Name
Althea
Last Name
Powlowski
Email
Donald41@hotmail.com
Phone
92889209
Birth Date
Jun 30, 2021
Start Date
Mar 1, 2022
Contract Type
CDD
Current salary
85302

Salary
Contract
Position

Cancel
Save Changes

Figure 24: Admin's edit employee page



INTUITIV INTERACTIVE

General
Teams
Projects
Employees
Leave Requests
Specific Requests
Documents
Settings
Log out

Dickinson
ADMINISTRATOR

Teams

Search...

TEAM NAME	
Hallie	View Team
Ona	View Team
Schuyler	View Team
Lea	View Team
Norbert	View Team
Jovany	View Team
Kian	View Team
Judy	View Team
Kristian	View Team
Salma	View Team

← Previous 1 2 3 4 Next →

localhost:4200/users/admin/team/3

Figure 25: Admin's teams table page

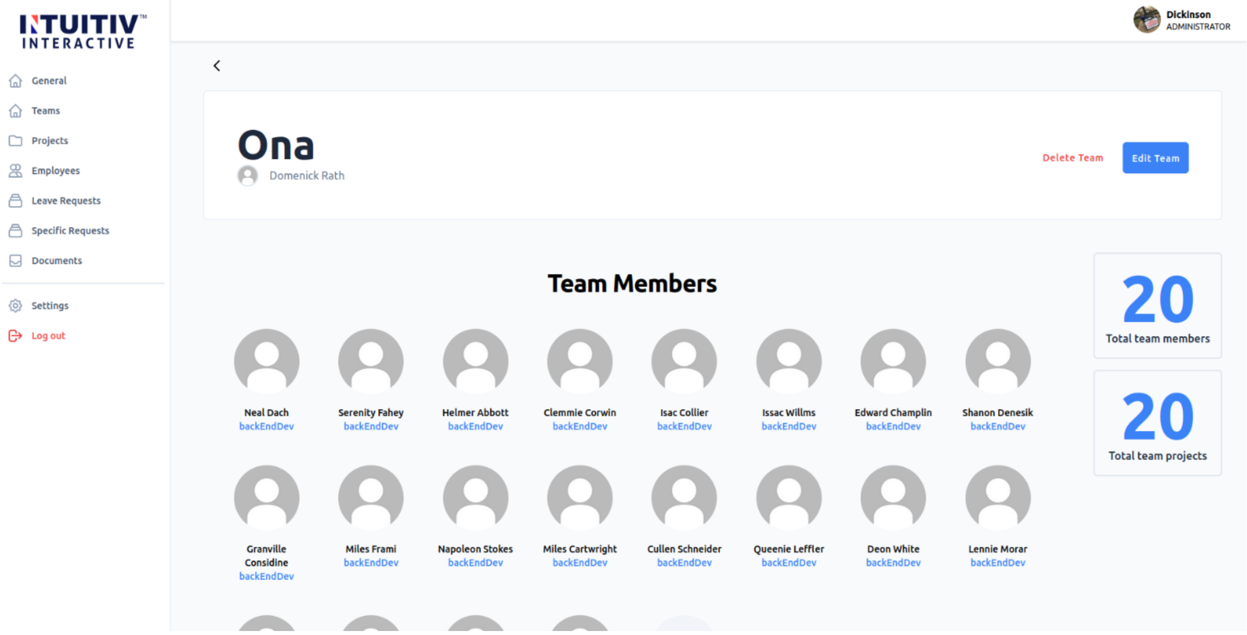


Figure 26: Team profile page

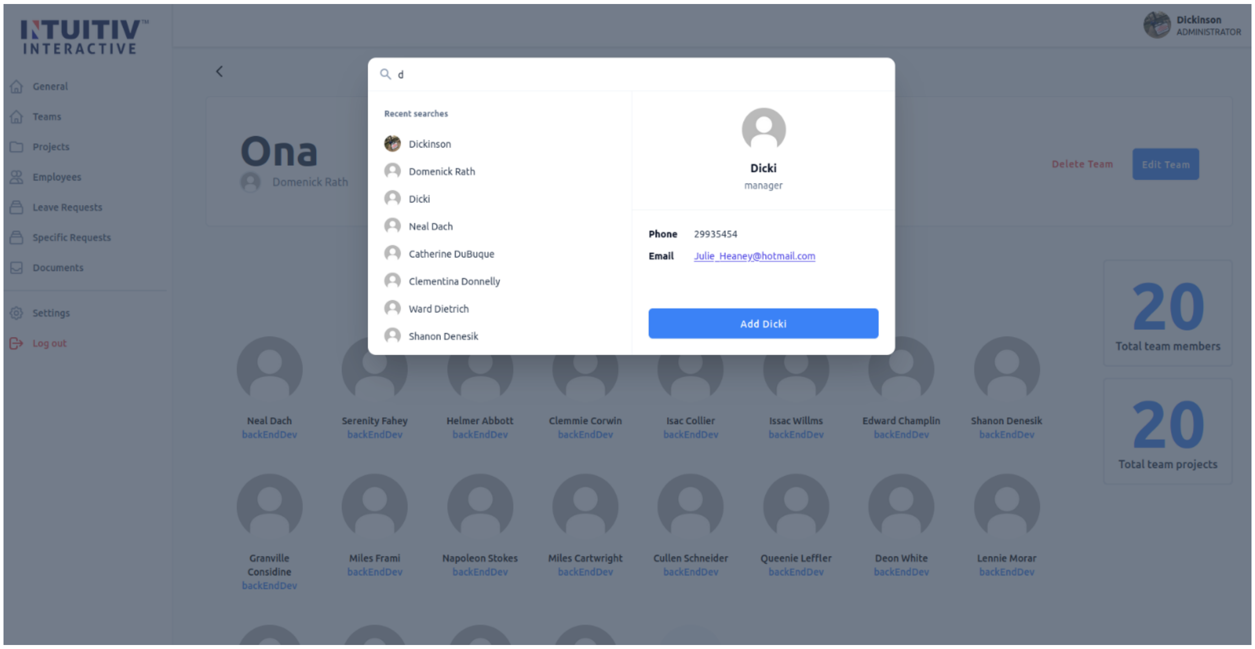
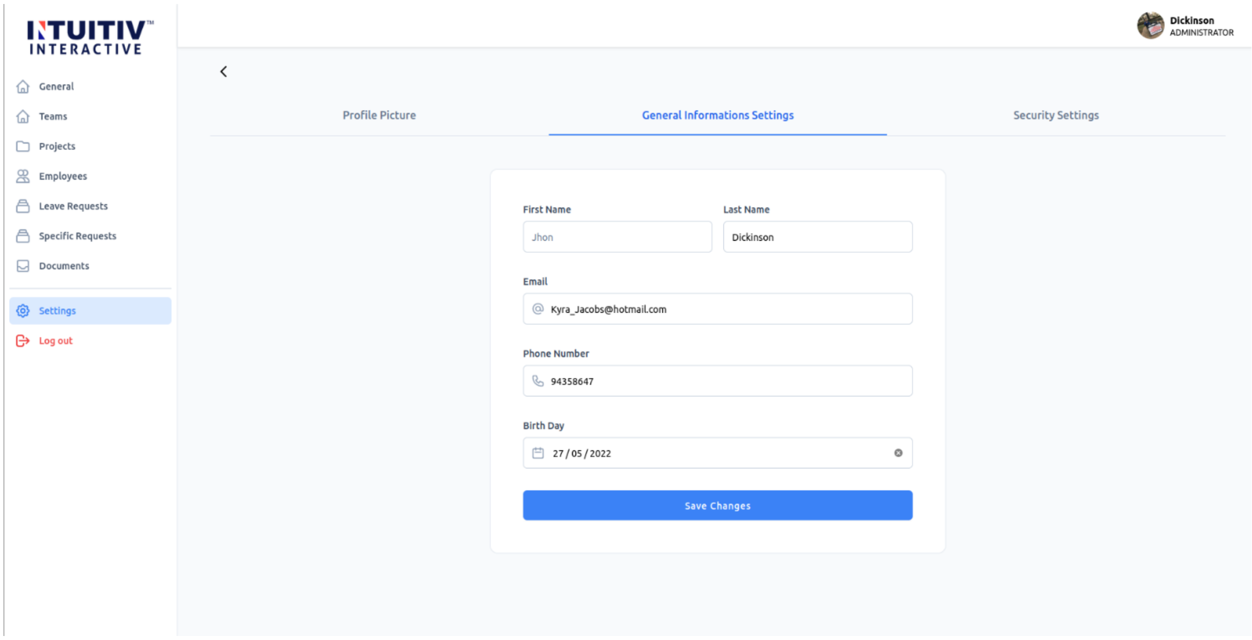


Figure 27: Assign Employee to a team page



First Name: Jhon
Last Name: Dickinson
Email: Kyra_Jacobs@hotmail.com
Phone Number: 94358647
Birth Day: 27 / 05 / 2022

Figure 28: Edit profile informations page

In the figures above, we presented to you most of this sprint interfaces which contains the profile editing interface for employee and admin interfaces for managing the employees as well as the teams.

VII. Sprint review

In this sprint, we managed to set up the desired functionalities such as the management of employees and the employee’s enterprise related settings, the management of teams for the administrator, the management of the manager’s teams and the management of personal profile.

During this sprint review, we presented the completed work to the product owner Mr Chridi Ramzi. The work has been validated and we are ready to start the next iteration.

VIII. Sprint retrospective

In this sprint, we managed to implement all the expected features and it is thanks to the effort put in by all the team members. There are no problems encountered and everyone is satisfied. Consequently, we will continue to work with the same practices.

Chapter 3: Release 1: Sprint 2

Introduction

In the previous chapter, we have presented the first sprint which contained the employee’s management module and teams’ management module. These are the main modules that our next steps are built on.

For this chapter, we are presenting release 1, our objectives for this sprint are to create multiple modules such as leave requests management module, specific requests management module, documents space management module and projects management module.

I. Sprint Objectives

The objective for this second sprint is to:

- Implement a web interface where the administrator can manage any employee’s leave request and manager can approve or reject the leave requests of his teams members.

- Implement a web interface where the administrator can manage the specific requests.
- Implement a web interface where any employee can store some documents and administrator can manage them.
- Implement a web interface for managing projects.

II. Sprint planning

In this part, we will present all the user stories and activities that will be treated by this sprint in a sprint backlog and we will present a schedule of the progress of this sprint.

1. Sprint backlog

In this part, we will present all the user stories and the activities that will be covered by this sprint in a sprint backlog and we will present a schedule for the progress of this sprint.

ID	User Story	Task ID	Task	Estimation
1	**Administrator can manage any employee’s leave requests**	2.1	Implementation of the user interface (UI)	4 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	
2	**Manager can approve or refuse his team**	2.1	Implementation of the user interface (UI)	3 days

	member's leave requests	2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	
3	**Employee can create leave requests**	2.1	Implementation of the user interface (UI)	2 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	
4	**Administrator can manage any employee's specific requests**	2.1	Task automatization within the enterprise related settings functions in the backend	3 days
5	**Employee can create specific requests**	2.1	Implementation of the user interface (UI)	2 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	
6	**Administrator can manage any employee's documents**	2.1	Implementation of the user interface (UI)	3 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	
7	**Administrator can manage projects and affect them to managers and teams**	2.1	Implementation of the user interface (UI)	3 days
		2.2	Develop RestFul APIs to Create, Update, View, Delete, Search a regulatory text	

2. Job Presentation

This sprint deals with mainly four user stories that express a set of needs, namely:

- The need of admin and manager to manage leave requests.
- The need of admin to manage documents.
- The need of admin to manage specific requests.
- The need of employee and manager to manage their own document space.

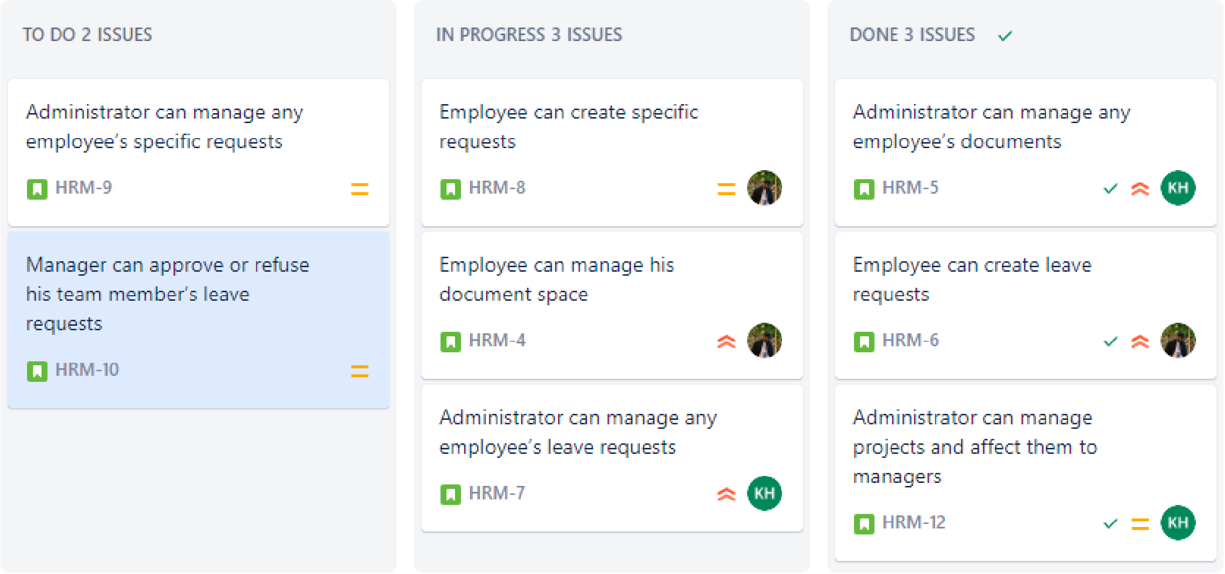
We will also deal with the case of creating a new leave request and a new specific request.

The above-mentioned needs require authentication, which must be implemented in the web and mobile application. Currently, only the web application will be addressed.

For users to be able to make leave requests they must have enough leave days score that gets incremented every month. A more detailed description of this part will be presented in the implementation section.

3. Scrum board

In this part, we will present the scrum board of the second sprint. This sprint contains a several user stories. Each user story contains a set of relative tasks.



TO DO 2 ISSUES

Administrator can manage any employee's specific requests
HRM-9

Manager can approve or refuse his team member's leave requests
HRM-10

IN PROGRESS 3 ISSUES

Employee can create specific requests
HRM-8

Employee can manage his document space
HRM-4

Administrator can manage any employee's leave requests
HRM-7 KH

DONE 3 ISSUES ✓

Administrator can manage any employee's documents
HRM-5 ✓ KH

Employee can create leave requests
HRM-6 ✓

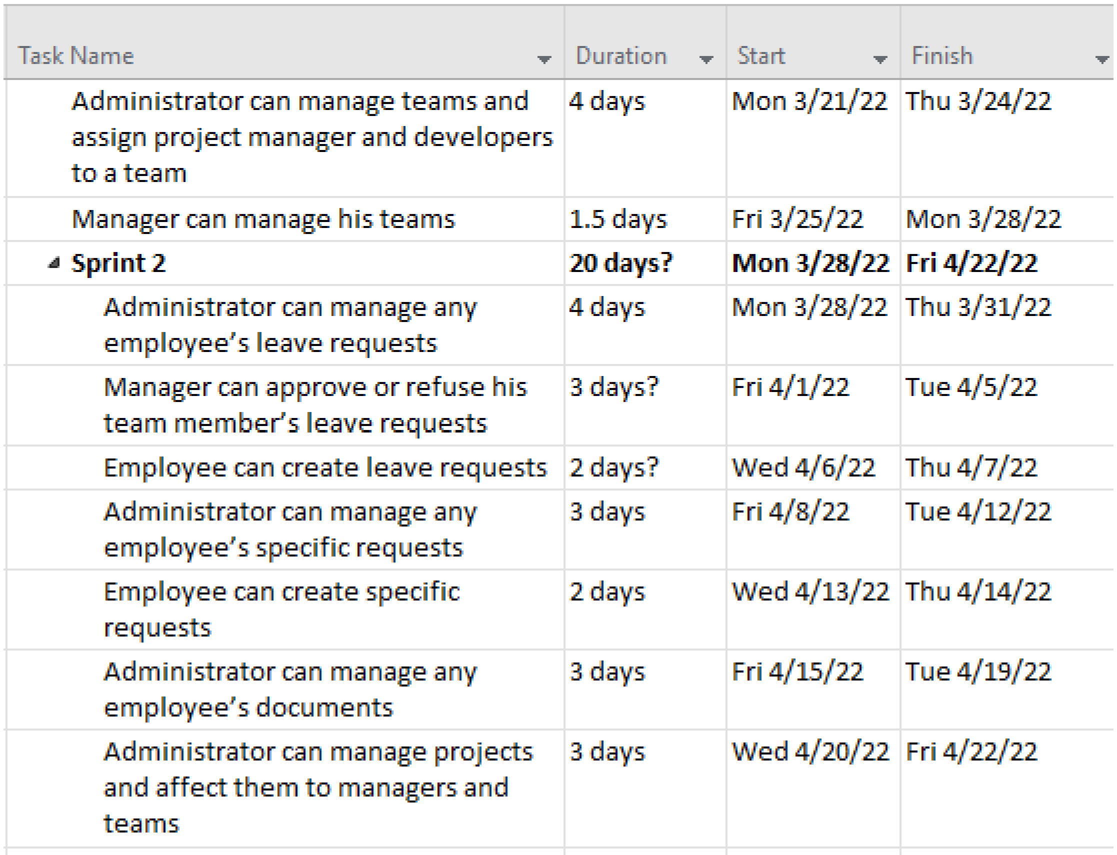
Administrator can manage projects and affect them to managers
HRM-12 ✓ KH

Figure 29: Scrum board sprint 2

This scrum board was captured during the implementation. A sharing of these tasks to the different human resources must take place. We applied the principle of self-organisation (as required by Scrum). Each time, one of us chooses the tasks that suit him.

4. Presentation of the sprint process

In the following, we will present a provisional schedule for the development of this second sprint. This diagram contains all the tasks to be carried out during this iteration as well as their chronological sequence. We start by presenting a Gantt chart which represents the chronological sequence of tasks.



Task Name	Duration	Start	Finish
Administrator can manage teams and assign project manager and developers to a team	4 days	Mon 3/21/22	Thu 3/24/22
Manager can manage his teams	1.5 days	Fri 3/25/22	Mon 3/28/22
Sprint 2	**20 days?**	**Mon 3/28/22**	**Fri 4/22/22**
Administrator can manage any employee’s leave requests	4 days	Mon 3/28/22	Thu 3/31/22
Manager can approve or refuse his team member’s leave requests	3 days?	Fri 4/1/22	Tue 4/5/22
Employee can create leave requests	2 days?	Wed 4/6/22	Thu 4/7/22
Administrator can manage any employee’s specific requests	3 days	Fri 4/8/22	Tue 4/12/22
Employee can create specific requests	2 days	Wed 4/13/22	Thu 4/14/22
Administrator can manage any employee’s documents	3 days	Fri 4/15/22	Tue 4/19/22
Administrator can manage projects and affect them to managers and teams	3 days	Wed 4/20/22	Fri 4/22/22

Figure 30: Gantt diagram sprint 2 [1/2]

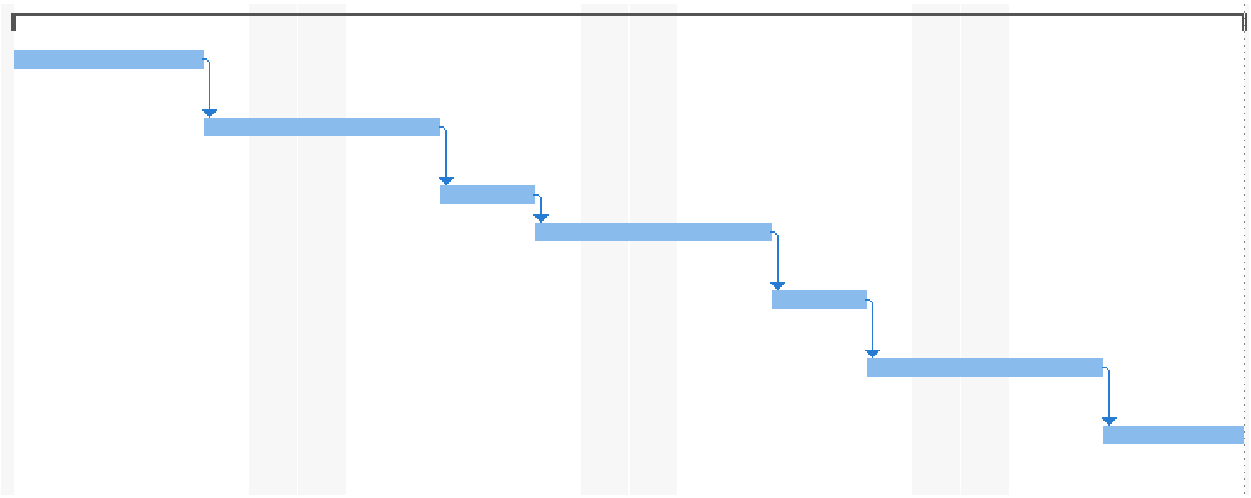


Figure 31: Gantt diagram sprint 2 [2/2]

This diagram contains all the tasks belonging to Sprint 2 and their estimated durations. We will start with the requirements analysis and specification phase. Then we will move on to the design phase.

In fact, use case diagrams, textual descriptions and system sequence diagrams will be developed for a better understanding of the problem. A class diagram will be designed to present a relational view of the architecture of our problem (SQL architecture).

III. Non-formal requirements specification

In the following, an analysis of the functional and non-functional requirements that must be satisfied by the application at the end of this iteration will be performed.

3. Functional needs

The functional requirements of this sprint are separated by roles:

- For administrator:
  - The approve or rejection of the leave requests.
  - The management of employee’s specific requests.
  - The management of employee’s documents.
  - The management of projects.
- For managers:
  - The approve or rejection of his teams leave requests.
- For employees:
  - The creation of leave requests.

- The creation of specific requests.
- The management of his document space.

In this iteration, we will be implementing the automatic mail sending for the handle of the leave requests, specific requests and document spaces.

4. Non-Functional needs

The non-functional needs of this chapter’s sprint are the same as the previous, which is maintaining a scalable and optimized web application.

IV. Semi-formal specification of needs

In this section, we will describe the requirements using a set of diagrams.

We have chosen to present the use case diagram, communication diagram, the system sequence diagram and some textual descriptions.

1. Use case diagram

Below, we present the use cases which present a semi-formal description of the functional needs corresponding to this iteration.

We begin with use case diagram which describe the management of documents (create for employee and delete for administrator), the management of leave requests (create for employee and approve or refuse for manager and administrator), the management of specific requests and the management of projects (update for manager and create, update and delete for administrator).

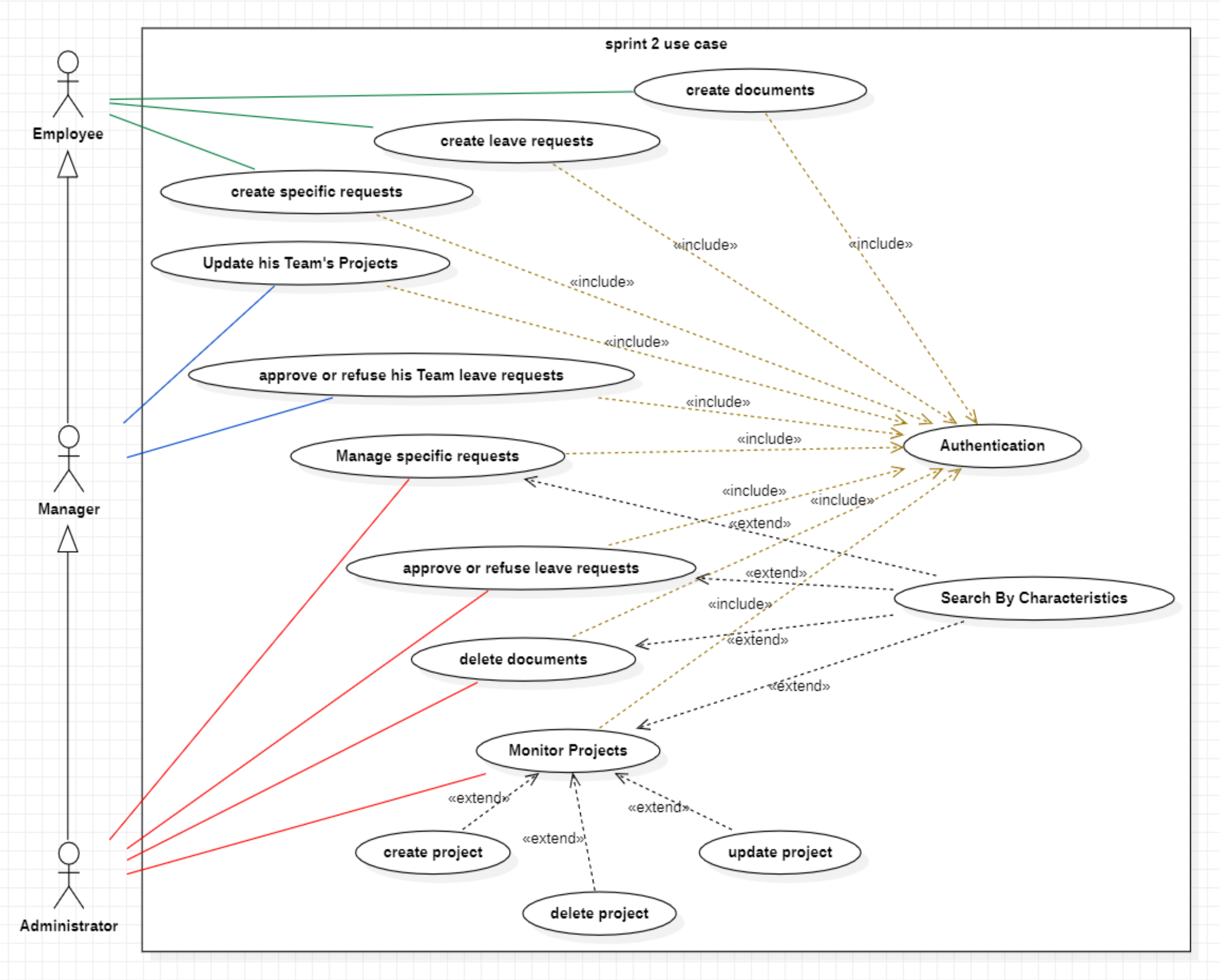


Figure 32: Sprint 2 use case diagram

❖ Textual Description of use case « create document »:

In the following, we present the textual description associated with the use case of creating document.

Use Case	-Create document.
Actors	Employee, Manager and Administrator.

Precondition	- Functional server - Authenticated user
Post-condition	-Document created.
Basic scenario description	1-User click on “Document Space” on sidebar. 2-User click on “Add a document”. 3- The system shows an empty form inside a module. 4-User type the document name and description. 5-User click on “Upload a File” to select the file. 6-The system shown a window where the user can select the file. 7-User clicks on "create document" to submit the form. 8- Server saves the new document and the file is uploaded to the folder “uploads/documentSpace”. 9- The system redirects the user to his document space.
Exception	- The server is down, in this case the user is warned.

The textual description above describes the use case diagram " create document " by indicating the actors, the post conditions, the pre conditions and the exceptions that can be refocused.

❖ Textual Description of use case « delete a team »:

In the following, we present the textual description associated with the use case of creating a team.

Use Case	-Create a leave request.
Actors	Employee, Manager and Administrator.
Precondition	- Functional server - Authenticated user
Post-condition	-Leave request created.
Basic scenario description	1-User clicks on “My Leave Requests” on the sidebar. 2-User clicks on the button “Create a leave request”. 3- The system shows an empty form inside a module. 4-click on the submit button. 5- Server saves the new leave requests and send a mail to the user’s manager.

	6-The system redirects the user to his leave requests list.
Exception	- The server is down, in this case the user is warned.

The textual description above describes the use case diagram "create a leave request " by indicating the actors, the post conditions, the pre-conditions and the exceptions that can be refocused.

- **Textual Description of use case « update project »:**

In the following, we present the textual description associated with the use case of creating a team.

The manager can only update his team's project and administrator can update all the projects.

Use Case	-Update project.
Actors	Manager and administrator.
Precondition	- Functional server. - Authenticated user.
Post-condition	-Project updated.
Basic scenario description	1-User clicks on “Projects” on the sidebar. 2-User clicks on the project he wants to update.

	3- User clicks on “Edit project” 4- The system shows a full form inside a module already filled with the project informations. 5-User changes the fields he wants to update 6-Click the submit button. 7-Server saves the changes. 8-the system redirects the user to the projects list.
Exception	- The server is down, in this case the user is warned.

The textual description above describes the use case diagram " update a project " by indicating the actors, the post conditions, the pre conditions and the exceptions that can be refocused.

2. Conception

In this section, we are going to present the class, the communication and the sequence diagrams of this sprint.

A. Class Diagram

We tried to standardize the model of the data used at the level of this sprint in order to be able to develop a class diagram.

In what follows, we will present the class diagram containing the entities handled in this sprint.

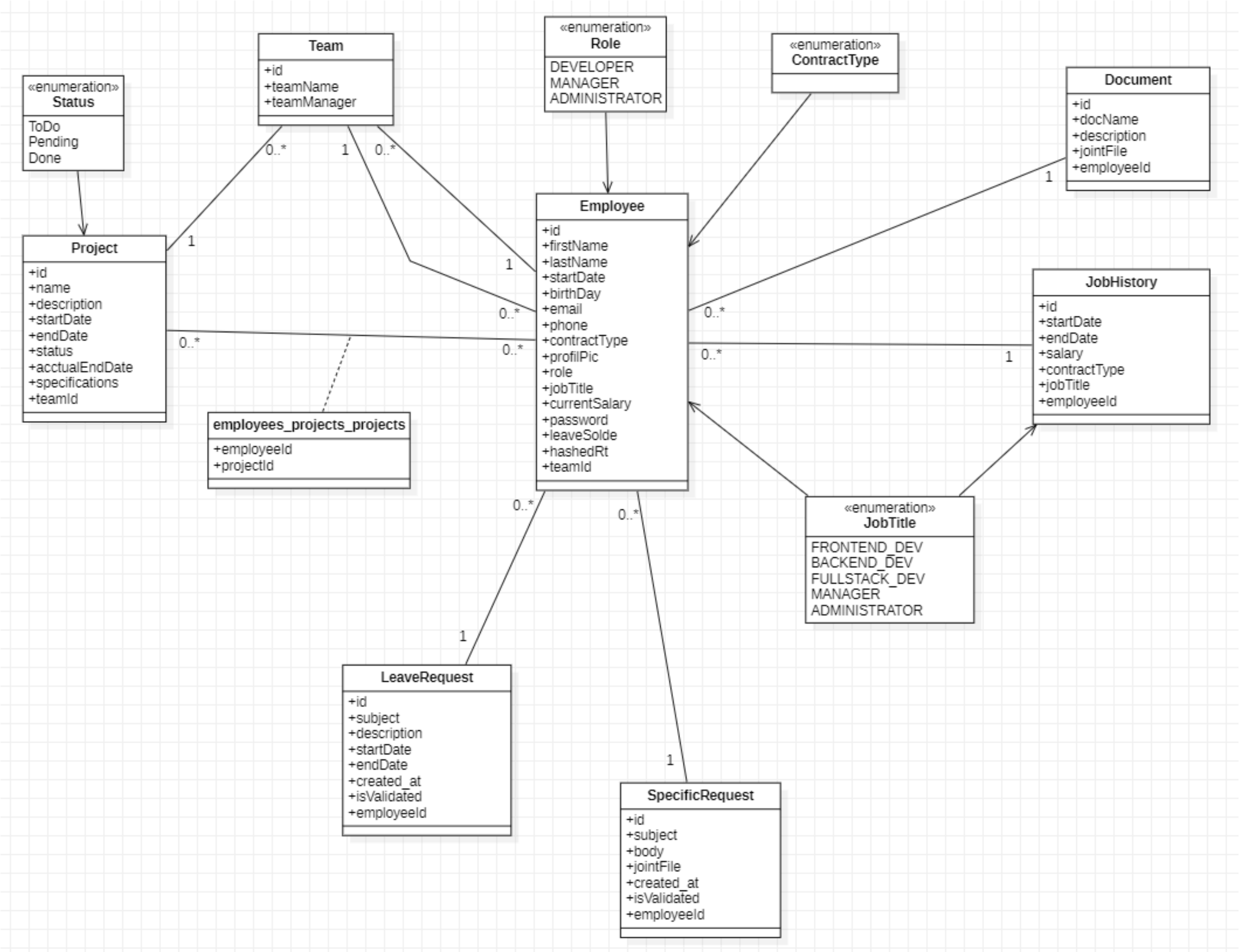


Figure 33: Sprint 2 class diagram

B. Communication Diagram

The figure below represents the communication diagram of the use case « Monitor Projects »

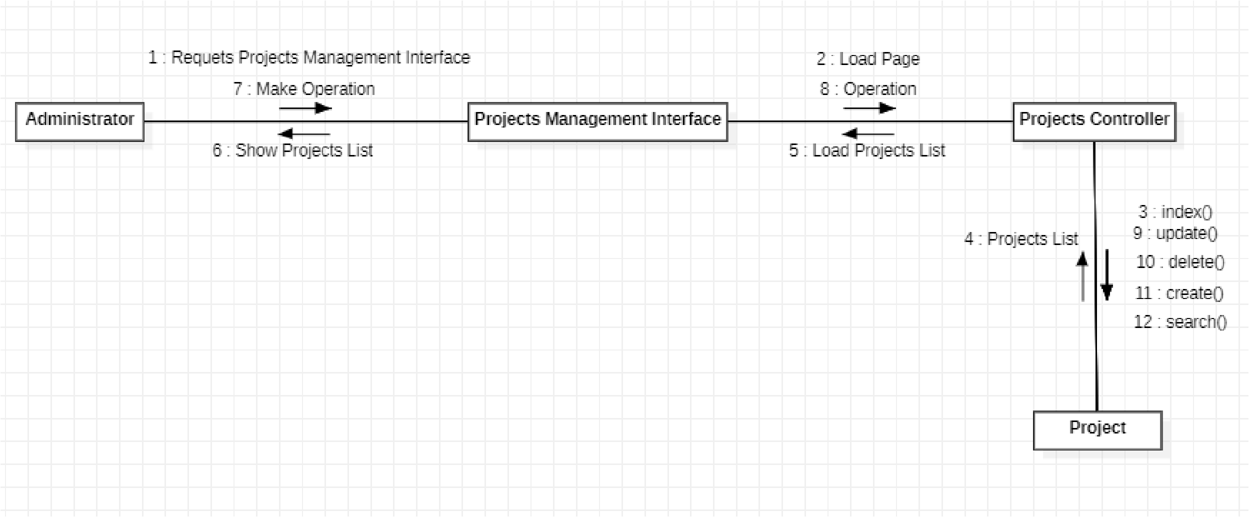


Figure 34: Communication diagram «Monitor Projects»

C. Sequence Diagrams

❖ Sequence diagram of use case « Create Document »

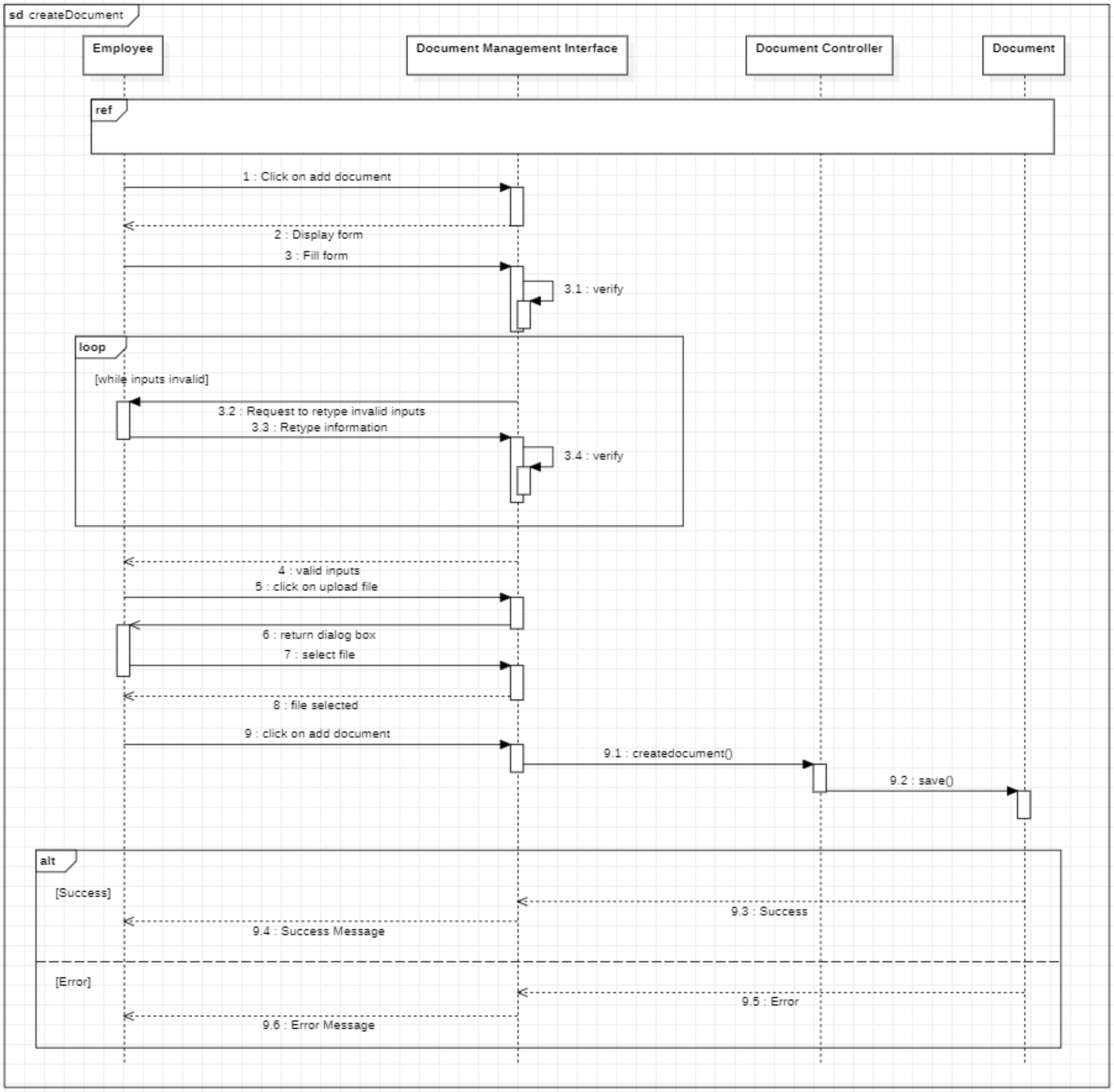


Figure 35: Sequence diagram «Create account»

❖ Sequence diagram of use case « Create project »

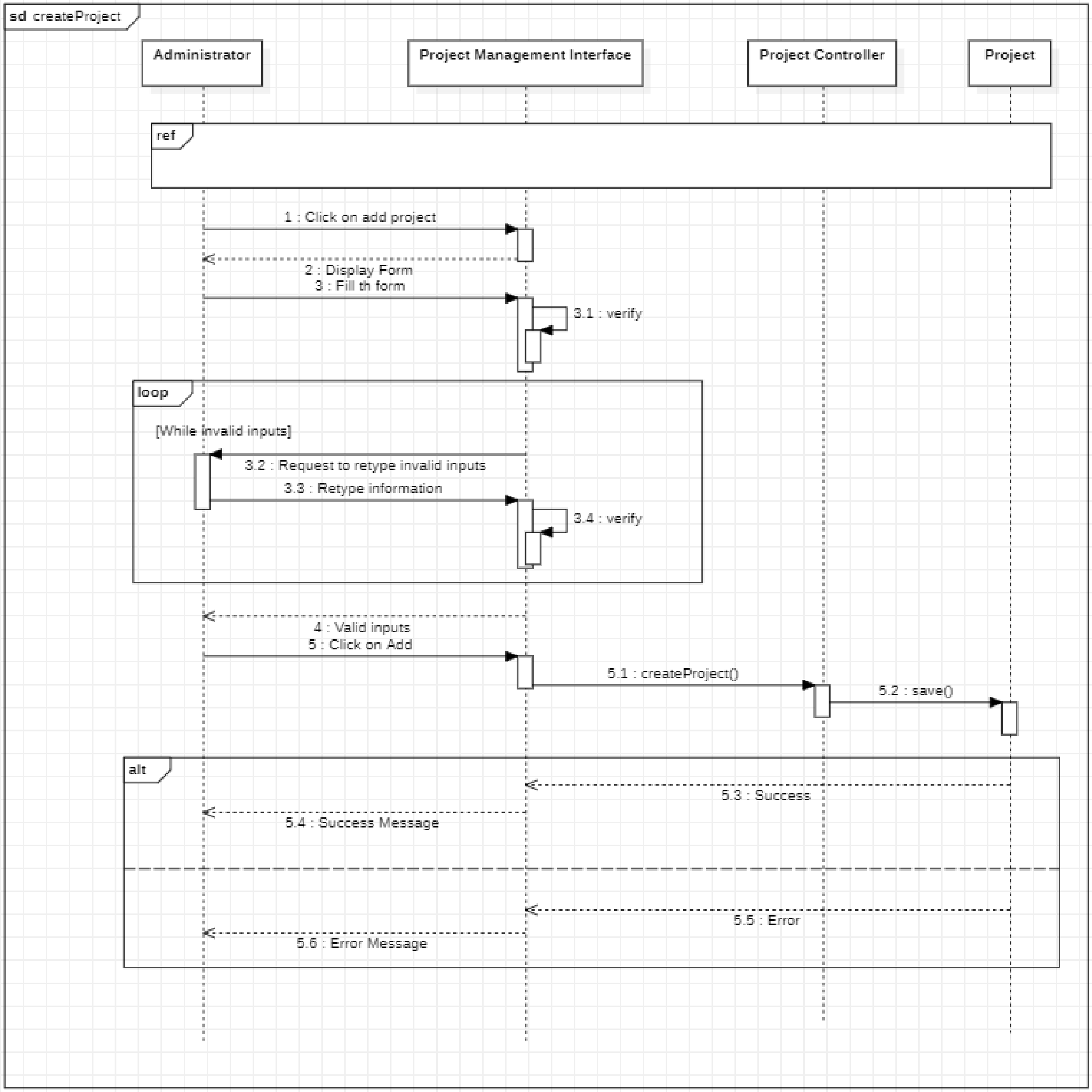


Figure 36: Sequence diagram «Create project»

❖ Sequence diagram of use case « Approve or refuse leave request »

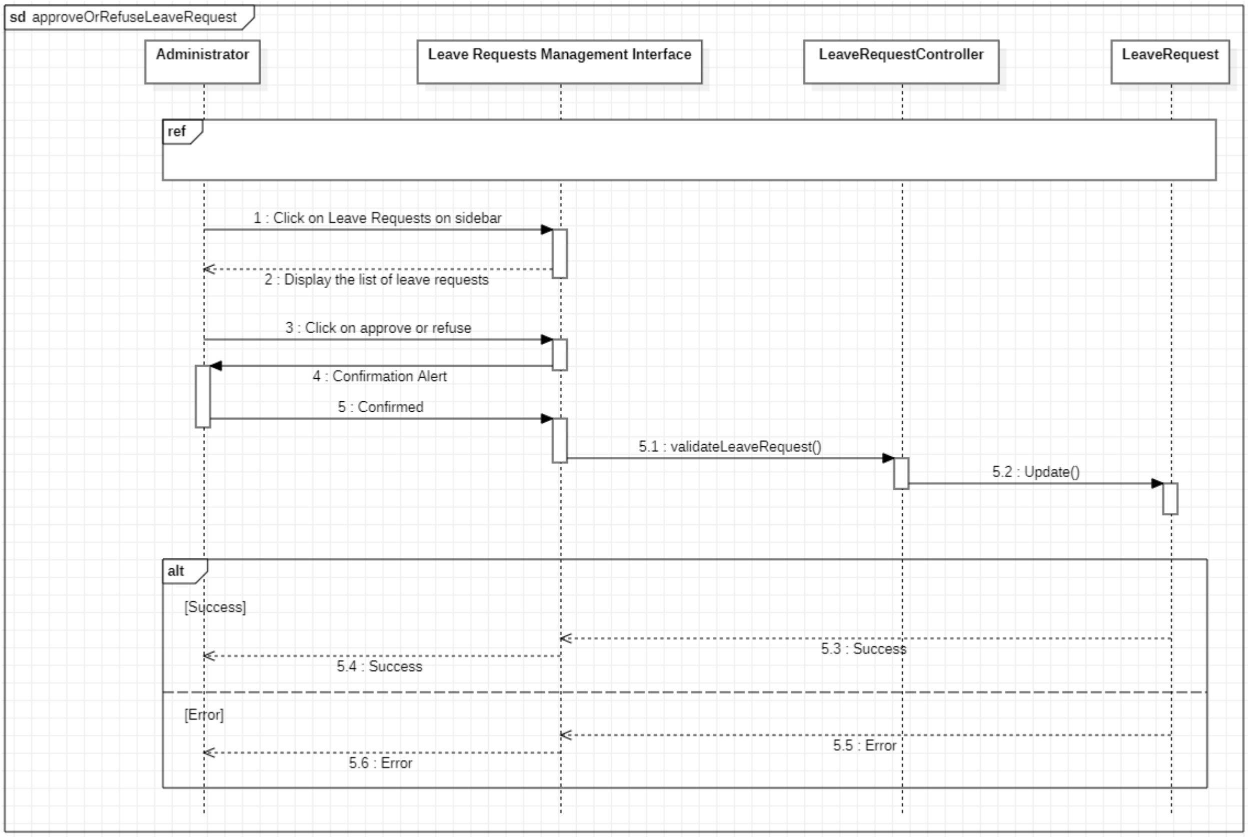

sd approveOrRefuseLeaveRequest
Administrator
Leave Requests Management Interface
LeaveRequestController
LeaveRequest
ref
1 : Click on Leave Requests on sidebar
2 : Display the list of leave requests
3 : Click on approve or refuse
4 : Confirmation Alert
5 : Confirmed
5.1 : validateLeaveRequest()
5.2 : Update()
alt
[Success]
5.3 : Success
5.4 : Success
[Error]
5.5 : Error
5.6 : Error

Figure 37: Sequence diagram «Approve/refuse leave requests»

Bibliography

[1] : https://www.atlassian.com/agile 07/04/2022

[2] : https://www.atlassian.com/agile/scrum 07/04/2022

[3] : https://www.agilealliance.org/glossary/backlog 07/04/2022

[4] : https://en.wikipedia.org/wiki/Docker_(software) 08/04/2022

[5] : https://docs.nestjs.com/ 08/04/2022

[6] : https://en.wikipedia.org/wiki/StarUML 08/04/2022

[7] : https://en.wikipedia.org/wiki/MySQL 08/04/2022

[8] : https://www.atlassian.com/software/jira/agile 08/04/2022

[9] : https://en.wikipedia.org/wiki/Angular_(web_framework) 08/04/2022

[10] : https://visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/ 18/04/2022

[11] : https://en.wikipedia.org/wiki/Class_diagram 19/04/2022

[12] : https://www.lucidchart.com/pages/uml-communication-diagram 19/04/2022

[13] : https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/ 22/04/2022