



Tunisian Republic
Ministry of Higher Education
and Scientific Research
University of Carthage
Higher Institute of Information
Technologies and Communication



GRADUATION PROJECT REPORT

Presented with the goal of obtaining the
LICENCE IN COMPUTER SCIENCE

Mention: Computer Science
Speciality: Computer Science

Customer Relationship Management Platform Ebuild

By
FIRAS AYARI

ALA EDDINE GTARI

Realized within EBUILD

EBUILD

Academic year: 2021 - 2022

Dedications

A special feeling of gratitude to my dear parents, Lotfi Ayari and Hayet Abidi whose words of encouragement and push for determination still ring in my ears. I will forever be thanking you for all the sacrifices you made for me to be who I am today.

To my sisters Hanin and Mayara Ayari who have always been by my side and motivated me to do my best.

To my friends who always got my back and heard my voice when silence seemed the only answer. I will be forever grateful to my best friend and partner Hiba Hmida who always lighted me with wisdom and lifted me up in my dark days.

To my pair Ala Gtari who has been the shoulder I can always depend on and who have supported me throughout our process.

To Mrs. Wafa TEBOURSKI, doctor of computer science, professor at the Higher Institute of Information and Communication Technologies, and our academic supervisor for her consistent determination to help us achieve our best and reach our fullest potential.

Dedications

As well as everything that I do, I would be honor to dedicate this compilation to my parents Alouen Gtari and Dalila Aouini , who gave me the necessary tools and values to be where I am standing today.

To My pair Firas Ayari who supported me on every step I make, and decision I take during my first professional experience.

I will never finish thanking Mrs. Wafa TEBOURSKI, doctor of computer science, professor at the Higher Institute of Information and Communication Technologies, and our academic supervisor for helping me to improve my career prospects by upgrading my skills.

I am so grateful to my friends for trusting me that I would do a good job in the university, and letting me come to achieve a higher education.

Contents

Dedication	i
Dedication	ii
General Introduction	1
1 Project Context	2
1.1 Introduction	2
1.2 Hosting Company	2
1.3 Project Context	2
1.3.1 Criticism of the existing	3
1.4 Presented Solution	3
1.5 Project Management Methodology	4
1.5.1 Scrum	4
1.5.2 The Scrum Team	4
1.5.3 Scrum Events	5
1.5.4 Scrum Artifacts	5
1.6 Technological choices	5
1.6.1 Frontend	6
1.6.2 Backend	6
1.6.3 Software	7
1.6.4 Teamwork Tools	8
1.7 Conclusion	8
2 Requirements specification	9
2.1 Introduction	9
2.2 Requirements Specification	9
2.2.1 Functional requirements	9
2.2.2 Non-Functional requirements	10
2.2.3 Actors Identification	10
2.3 Global use case diagram	12
2.4 Product backlog	12
2.4.1 Project management	13
2.4.2 Project Follow-up	14
2.5 Conclusion	15

3 Release 1 : Implementation Of The Project Management	16
3.1 Introduction	16
3.2 Sprint 0	16
3.2.1 Identifying the sprint0 backlog	16
3.2.2 Refinement of sprint 0	17
3.2.3 Deployment diagram	25
3.2.4 Design of sprint 0	25
3.2.5 Implementation of sprint 0	33
3.3 Sprint 1	49
3.3.1 Identifying the Sprint1 backlog	49
3.3.2 Refinement of sprint 1	50
3.3.3 Design of sprint 1	59
3.4 Implementation of sprint 1	67
3.5 Conclusion	75
4 Release 2 : Implementation Of The Project Follow-Up	76
4.1 Introduction	76
4.2 Sprint 2	76
4.2.1 Identifying the release2 backlog	76
4.2.2 Refinement of sprint 2	77
4.2.3 Design of sprint 2	81
4.3 Implementation of sprint 2	84
4.4 Sprint 3	86
4.4.1 Identifying the release2 backlog	86
4.4.2 Refinement of sprint 2	88
4.4.3 Design of sprint 2	94
4.5 Global class diagram	99
4.6 Implementation of sprint 3	100
4.7 Conclusion	108
Bibliography	110

List of Figures

1.1	Logo E-build	2
1.2	Rest API model [15]	6
2.1	global use case diagram	12
3.1	use case diagram “authenticate”	17
3.2	use case diagram “manage users”	18
3.3	use case diagram “manage documents”	21
3.4	use case diagram “manage teams”	23
3.5	Deployment diagram	25
3.6	class diagram of the user story “authenticate”	25
3.7	sequence diagram of the user story “authenticate”	26
3.8	class diagram of the administrator story “manage staff”	26
3.9	sequence diagram of the administrator story “manage staff”	27
3.10	class diagram of the administrator story “manage customers”	28
3.11	sequence diagram of the administrator story “manage customers”	29
3.12	class diagram of the administrator story “manage documents”	30
3.13	sequence diagram of the administrator story “manage documents”	31
3.14	class diagram of the administrator story “manage teams”	32
3.15	sequence diagram of the administrator story “manage teams”	33
3.16	Authenticate Interface	34
3.17	Authentication Error Notification	34
3.18	Manage Staff Interface	35
3.19	Add Staff Interface	36
3.20	Update Staff Interface	36
3.21	Updated Staff Interface	37
3.22	Deleted Staff Interface	37
3.23	Manage Customers Interface	38
3.24	Add Customer Interface	39
3.25	Update Customer Interface	39
3.26	Updated Customer Interface	40
3.27	Deleted Customer Interface	40
3.28	Manage Documents Interface	41
3.29	Add Document Interface	42

LIST OF FIGURES

3.30 Added Document Interface	42
3.31 Appraisal PDF Format Example	43
3.32 Bills PDF Format Example	43
3.33 Update Document Interface	44
3.34 Updated Document Interface	44
3.35 Deleted Document Interface	45
3.36 Manage Teams Interface	46
3.37 Add Team Interface	46
3.38 Added Team Interface	47
3.39 Update Team Interface	47
3.40 Updated Team Interface	48
3.41 Deleted Staff Member from Team Interface	48
3.42 Deleted Team Interface	49
3.43 use case diagram “manage projects”	50
3.44 use case diagram “manage tasks”	52
3.45 use case diagram “manage subtasks”	54
3.46 use case diagram “manage tickets”	56
3.47 class diagram of the administrator story “manage projects”	59
3.48 sequence diagram of the administrator story “manage projects”	60
3.49 class diagram of the customer story “manage tasks”	61
3.50 sequence diagram of the customer story “manage tasks”	62
3.51 class diagram of the administrator story “manage subtasks”	63
3.52 sequence diagram of the administrator story “manage subtasks”	64
3.53 class diagram of the customer story “manage tickets”	65
3.54 sequence diagram of the customer story “manage tickets”	66
3.55 Manage Projects Interface	67
3.56 Add Projects Interface	68
3.57 Added Project Interface	68
3.58 Manage Tasks Interface	69
3.59 Add Task Interface	70
3.60 Added Subtask Interface	71
3.61 Manage Tickets Interface	72
3.62 Add Ticket Interface	72
3.63 Added Ticket Interface	73
3.64 Update Ticket Interface	73
3.65 Updated Ticket Interface	74
3.66 Deleted Ticket Interface	74
4.1 use case diagram “consult tickets”	77
4.2 use case diagram “consult projects”	78
4.3 use case diagram “consult tasks”	79
4.4 use case diagram “consult subtasks”	80
4.5 class diagram of the staff story “consult tickets”	81

4.6	sequence diagram of the staff story “consult tickets”	82
4.7	class diagram of the staff story “consult projects”	82
4.8	sequence diagram of the staff story “consult projects”	82
4.9	class diagram of the staff story “consult tasks”	83
4.10	sequence diagram of the staff story “consult tasks”	83
4.11	class diagram of the staff story “consult subtasks”	84
4.12	sequence diagram of the staff story “consult subtasks”	84
4.13	Consult Tickets Interface	84
4.14	Consult Projects Interface	85
4.15	Consult Tasks Interface	85
4.16	Consult Subtasks Interface	86
4.17	use case diagram “manage responses”	88
4.18	use case diagram “manage comments”	90
4.19	use case diagram “consult documents”	92
4.20	use case diagram “consult responses”	93
4.21	class diagram of the staff story “manage responses”	94
4.22	sequence diagram of the staff story “manage responses”	95
4.23	class diagram of the staff story “manage comments”	96
4.24	sequence diagram of the staff story “manage comments”	97
4.25	class diagram of the customer story “consult documents”	98
4.26	sequence diagram of the customer story “consult documents”	98
4.27	class diagram of the customer story “consult responses”	99
4.28	sequence diagram of the customer story “consult responses”	99
4.29	Global class diagram	100
4.30	Manage Responses Interface	101
4.31	Add Response Interface	101
4.32	Added Response Interface	102
4.33	Updated Response Interface	102
4.34	Deleted Response Interface	103
4.35	Manage Comments Interface	104
4.36	Update Task Comment Interface	104
4.37	Update Subtask Comment Interface	105
4.38	Updated Subtask Comment Interface	105
4.39	Deleted Task Comment Interface	106
4.40	Deleted Subtask Comment Interface	106
4.41	Consult Documents Interface	107
4.42	Consult Documents Interface	107

List of Tables

1.1	Technological choices frontend	6
1.2	Technological choices backend	6
1.3	Software	7
1.4	Teamwork Tools	8
2.1	Detailed description of the actors	11
2.2	Product backlog Release 1	13
2.3	Product backlog Release 2	14
3.1	Product backlog Release 1 sprint 0	16
3.2	Detailed description of the "Authentication" use case	18
3.3	Detailed description of the "manage users" use case	19
3.4	Detailed description of the "manage documents" use case	21
3.5	Detailed description of the "manage teams" use case	23
3.6	Product backlog Release 1 sprint 1	49
3.7	Detailed description of the "manage projects" use case	51
3.8	Detailed description of the "manage tasks" use case	53
3.9	Detailed description of the "manage subtasks" use case	55
3.10	Detailed description of the "manage tickets" use case	57
4.1	Product backlog Release 2 sprint 2	76
4.2	Detailed description of the "consult tickets" use case	78
4.3	Detailed description of the "consult projects" use case	79
4.4	Detailed description of the "consult tasks" use case	80
4.5	Detailed description of the "consult subtasks" use case	81
4.6	Product backlog Release 2 sprint 3	86
4.7	Detailed description of the "manage responses" use case	88
4.8	Detailed description of the "manage comments" use case	90
4.9	Detailed description of the "consult documents" use case	92
4.10	Detailed description of the "consult reponses" use case	93

General Introduction

Technology and economy have always been among the most impactful pillars laying a solid foundation to the human evolution over the centuries. As a matter of fact, their contributions always go hand in hand, and since the business world never ceased to be the main driver of the global economy, it has witnessed a huge innovation regarding its aspects throughout the history, remarkably after becoming influenced by modern technology.

One of these most indispensable aspects of business is customer relationship, as it is considered a long-term key for an organization to nurture its stability in the current fast-pace blooming market. Therefore, Technology brought up an upright concept to ideate strong personal bonding within customers that drives businesses to new levels of success. This strategy is what is well-known today as Customer Relationship management or CRM.

In the current era of digitization, CRM systems are the ideal collection centralizing all data sources under an organization, providing a real time atomistic vision of customers growing needs and demands therefore introducing the sophisticated strategies that can be implemented to serve them in a better way and guarantee a fruitful business, what was the main aim of E-build, a lately-established start-up looking to make a name based on a solid foundation.

Our PFE-internship project consists of developing a CRM system dedicated to E-build's working space, that provides an easy-to-use virtual environment that does not only build a personal and emotional linkage with customers only, but with the working staff too.

During the developing phase of our project, we pursued Scrum as a working methodology and therefore had an aspectual approach by splitting the project into the following four chapters:

- The first chapter entitled as "Project Context" shows the exploratory study.
- The second chapter entitled as "Requirements Specification" explains the requirement specification.
- The third chapter entitled as "Release 1: Implementation of the Project Management" demonstrates the implementation of the project management on our customer relationship management platform.
- The fourth chapter entitled as Project Follow-Up "Release 2: Implementation of the Project follow-up" demonstrates the implementation of the project follow-up on our customer relationship management platform.

Eventually, we finalize our report with a general conclusion and a viewpoint of the project and beyond.

Chapter 1

Project Context

1.1 Introduction

In the first chapter, we will focus on addressing the general context of our project. We will start with the hosting company presentation to move then we will go deeper into the project context. Next, we will proceed to criticize the existing solutions, so that we could explain the added value of our proposed solution. Finally, we will analyse the work methodology we have chosen for our project.

1.2 Hosting Company

E-build is a digital agency providing various services which sustainably support those looking to build their brands and have a professional companion in their journey. These proffered services are mainly websites development, graphic design, social media strategy, community management assistance, implementation of web marketing levers (Google ADS/SEA, SEO/SEO, etc), targeted advertising, content creation,etc....



Figure 1.1: Logo E-build

1.3 Project Context

E-build is a new company targeting customers that are looking to build their brands with the innovative ways and technologies, going with the flow and trends of the modern era. And since they are newly-established, they were looking for a system that boosts their relationship with the customers, keep track of the recent projects accomplished and collects the customers data to better understand the needs of the current market and have a long vision on the future. These systems are well-known today as Customer Relationship Management systems (CRM). A CRM system allows business owners to keep in touch with their customers by always updating their contacts, tracking their interactions and maintaining their accounts. Commonly, all these data were saved into analogue devices or unconnected media, or just kept in people's heads which increases the risk of its lost and forgetfulness. In fact, this is why CRM systems exist, they help businesses improve their Customer Lifetime Value (CLV),which is very vital regarding the amount

CHAPTER 1. PROJECT CONTEXT

of daily generated data, by addressing challenges like these as they transform the collected data into effective insights that grant business owners a better view over their customers growing and ever-changing interests. Having a better understanding of the customers exigencies is so critical in today's market, as if someone's business is not providing the services that meet these needs gets left behind in the race of targeting new clients and keeping the acquired ones. That is why for a newly-established company, there was no better way to kick off the business than to lay a solid foundation for the customer relationships management, since the number of customers appreciating the offered services is increasing and so are the risks of having their data lost forever. And this triggered the search for a solution to this matter.

1.3.1 Criticism of the existing

When E-build has freshly started, it did not have a CRM system of its own. The initial solution was to buy a CRM system template and then just add the already collected data. However, as we were going through the existing CRM systems on the internet and the purchasable one, we have come to notice that they did not cover really all the needs of e-build and their expectations for the services they want to include in their system. The most common disadvantages were the following:

- Most CRM systems are either customer-oriented or too invested in just providing a sort of a dynamic aspect that only stores and collects the data. Most of the times, the possible information and details to add into the system can only be accessed by the admin where they can organize it as they please, and that is due to having the system composed of only admin interfaces, that the customer can't really interact with them or have the privileges to use them to follow the progress of their projects.
- Most CRM systems that are heavily customer-oriented don't make the client as a part of the project as they should be. They are only based on generating projects documents containing the required details of the payments and the operations that should be completed.
- There is not really a CRM that manages to be a platform for both the clients and company personnel at the same time.
- The customer relationship services provided are not really building the expected personal and emotional linkage and are not solidifying the communication with the customer. Many CRMs lack email services, or the clients are deprived of the tickets submitting. Moreover, the direct interaction with the staff working on their project by supervising the tasks or even making some changes in the process with the help of comments or the client feedback, is not really supported by the existing CRM systems.

1.4 Presented Solution

With the growing number of customers and potential customers, the urge of developing an own solution for the customer relationship management kept going higher and higher, with each new customer or project acquired, E-build became more and more determined to build its CRM system. As a starting point of our project, we challenged ourselves to come with an innovating idea that revolutionizes the aspect of what a CRM system consisted of. After taking the time needed to go through the existing well-known CRM systems, and brainstorming until getting an initial refined idea. Eventually, we proposed the final solution to E-build and it consisted of developing a system that comes with a new way to interact with customers, providing them with the possibility to be contributing as much as they are willing to do even during the process of developing their projects, by submitting tickets and checking on the developers' comments on each step or task completed. Moreover, the customer will have all the information about the project

CHAPTER 1. PROJECT CONTEXT

once done, including the bills payments required and even the operations going through, all in one place. Our solution takes it a step further even in organizing the company workspace, with services that help manage the teams and their assigned projects and tasks, therefore building in addition to the customer relationship, a staff relationship based on a professional personal and emotional linkage.

- Client space includes all the services the customer is looking for once they are logged into the platform such as Tickets services as a way to consult all the details of the project life-time or to make some changes, in addition to Documents Services that allow them to keep track of the required payments and ongoing transactions.
- Personnel Space includes all the services the staff members need to access in order to be able to consult their projects and get notified with the latest tasks assigned to them. Moreover, they have the required privileges to update their assignments status and add comments for both the administrator and the customer associated with the project they are working on.
- The admin has access to all data and information related to all clients and personnel members. In addition to that, they are the only one able to add clients and personnel to the platform. Also, they are responsible for adding projects based on the customer tickets

1.5 Project Management Methodology

For this project, we have chosen to work with the Agile methodology: Scrum.

1.5.1 Scrum

Scrum is an agile project management framework that allows teams to work together in order to generate a better value. It is based on empiricism and lean thinking; For the empiricism, it asserts that knowledge is developed and evolves with decision making and experience of what is observed. For the lean thinking, it increases productivity by reducing the waste and prioritizing the essentials. These are the steps followed when working with a scrum methodology:

- Product Owner instructs the Scrum Team to partition the work to be done in a Product Backlog.
- The Scrum Team works to accomplish the first task in the Product Backlog within the predetermined period of time or what is known as a Sprint.
- The Scrum Team and its stakeholders check out the results and adapt them for the next Sprint.
- Redo until all tasks in the Product Backlog have been achieved.

1.5.2 The Scrum Team

The Scrum Team is composed of one Scrum Master, one Product Owner, and Developers, each one of them concentrates on the same product, so it continues to be agile while having a significant amount of work to achieve within a Sprint. These members share the required skills to come up with value in each Sprint by collaborating their efforts verification, maintenance, operations, experimentation and researches. Yet, there has to be no sub-teams or hierarchies.

Developers :

Their responsibilities are mainly setting up a plan for the Sprint and the Sprint Backlog, in addition to adjusting it each day of the sprint to reach the established Goals (Sprint Goal), without disregarding the fact that they can hold each other accountable.[1]

CHAPTER 1. PROJECT CONTEXT

Product Owner :

The Product Owner is the member who is responsible for setting up the Product Backlog, and establish that it is translucent, clear and understood. This responsibility includes also representing the needs of stakeholders.[2]

Scrum Master :

The Scrum Master is the member of the Scrum Team who is the leader ensuring and training the Scrum adoption, in addition to arranging the Scrum implementations within the organization and loosening up the barriers set between stakeholders and Scrum Teams.[3]

1.5.3 Scrum Events

These are the following Scrum events:

The Sprint :

A sprint is a period of time of one month or less which during it there can be no changes made to alter the Sprint Goal, yet the Product Backlog can be processed as required. As a matter of fact, Scope? may be redefined too and discussed with the Product Owner.[4]

Sprint Planning :

Sprint Planning is an event where the Scrum Team have meetings of eight hours for a one-month Sprint to debate over some topics like why is this Sprint valuable? what can be accomplished in this Sprint? how is the instructed work going to be done? [5]

Daily Scrum :

The Daily Scrum guarantees the daily inspection of the progress so far to that day towards the Sprint Goal, adjusting of the Sprint Backlog as required, as well as the upcoming scheduled due dates. It takes a maximum of 15 minutes per day. [6]

Sprint Review :

The Sprint Review assures the evaluation of the outcome of the Sprint, and resolving to future adaptations. It lasts about four hours for a one-month Sprint.[7]

Sprint Retrospective :

The Sprint Retrospective assures the analysis of the latest Sprint. The Scrum Team examines what went well and was satisfied during the Sprint, what conflicts it bumped into, and if those problems were solved or not. It lasts three hours for a one-month Sprint.[8]

1.5.4 Scrum Artifacts

These are the following Scrum Artifacts:

Product Backlog :

The Product Backlog is a list of tasks set to be finished each within a Sprint in order to accomplish the Product Goal. [9]

Sprint Backlog :

The Sprint Backlog is consisted of the Sprint Goal, the set of Product Backlog items chosen for the Sprint, as well as a plan for delivering the Increment. [10]

Increment :

An Increment is a step towards the Product Goal. Each Increment is an add-on to all prior Increments and thoroughly checked, reassuring that all Increments work together. [11]

1.6 Technological choices

For this part, we will be presenting our technological choices that we thought about and followed during the process of developing the system. These technological choices can be classed into front-end technologies and back-end technologies.

1.6.1 Frontend

The front-end technologies are responsible for anything displayed on a website, desktop or a mobile application, which the user can interact with when looking at the things represented on those platforms.

Table 1.1: Technological choices frontend



Vu.js is an open-source front end JavaScript framework maintained for building user interfaces and single-page applications. It aims to make both the development and the testing of such applications easier and simpler by providing a framework for client-side model-view-view model architectures (MVVM).[12]

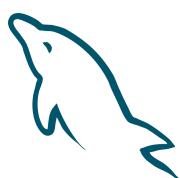
1.6.2 Backend

Back-end technologies are responsible for anything that is not displayed on a website, desktop or a mobile application and which the user can't interact with directly.

Table 1.2: Technological choices backend



Laravel is a free, open-source PHP web framework, created by Taylor Otwell and based on Symfony. It is dedicated to developing web applications following the model–view–controller (MVC) architectural pattern, yet recently it has become possible to work on micro-services architecture for faster and more complex applications using Laravel.[13]



MySQL is an open-source relational database management system (RDBMS) that allows users to interact with databases either directly by using SQL, or more often with other programs to implement applications that require relational database capabilities.[14]

REST API

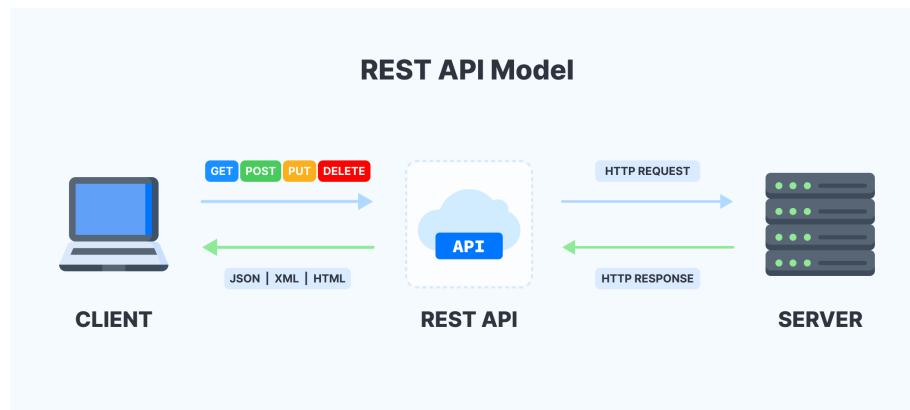


Figure 1.2: Rest API model [15]

REST stands for Representational State Transfer, which is a type of software architecture that sets constraints for web services therefore have the communications between them become easier. As a matter of fact, It's often based upon existing system protocols like Hypertext Transfer Protocol HTTP. On the other hand, API is the abbreviation for Application Programming Interface, which represents the intermediary layer between systems allowing them to exchange information and functionality without having them to identify how each one was implemented by setting up a collection of definitions and protocols.

1.6.3 Software

Table 1.3: Software



Visual Studio Code is Microsoft's source-code editor for every existing operating system. It is considered the most popular developer environment tool for the features it offers for free such as smart code completion, snippets, debugging, embedded Git, etc [16]



Postman is an API platform for developers to create and send different HTTP requests and receive their responses. Therefore, it is used mainly as a tool to test the functionalities of RESTful APIs during the development process.[17]



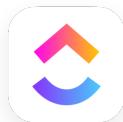
StarUML is a UML tool developed by MKLab. It supports nearly all the diagram types listed in UML 2.0.[18]



Xampp is a free open-source cross-platform web server stack package containing The Apache HTTP Server, MariaDB Database and interpreters for PHP and Perl scripts. It eases the process of transitioning from a local server for testing to a live server.[19]

1.6.4 Teamwork Tools

Table 1.4: Teamwork Tools



Click Up is a user-friendly solution made for any business owner to manage their projects. It employs a hierarchy of views to organize the business's projects: Teams / Spaces / Projects / Lists / Tasks / Subtasks. [20]



Git and GitHub Git is a control system that is distributed by GitHub, which is a code hosting platform. It is a software developed to keep track of the changes made in files over time, what helps users always have the latest version of each file in a repository, compare those changes and revert them to a specific state if needed later.[21]

1.7 Conclusion

Throughout this first chapter, we took up the general project context by introducing the hosting company, discussing the project context, then analyzing the existing solutions and criticizing their inconveniences so that we can explain our proposed solution and eventually address the methodology we worked with. In the following chapter, we will go through the details of requirements specification.

Chapter 2

Requirements specification

2.1 Introduction

Throughout this chapter, we will go through the study of the requirements specification of our project, by identifying our functional, non-functional requirements and actors. Then we will advance to the global use case diagram which will clarify all of that. And eventually, still within this context, we will proceed to the backlog product presentation.

2.2 Requirements Specification

Requirements' specification establishes the main foundation for the product development. It defines how a developed system is set to function, as well as the constraints to be implemented on its design. Laying out this foundation on solid bases guarantees that the team working on the project development creates a product satisfying the customers' needs. Therefore, it characterizes the following functional and non-functional requirements:

2.2.1 Functional requirements

Functional requirements, known also as the functional specification, are a collection of requirements that outlines the main goal of the system and the purpose of its offerings to all users. It features the systems functionalities that the team must fulfil during the process of development, what helps them to keep track of their progress.

- Authenticate
- Consult Document
- Manage Ticket
- Consult response
- Consult Ticket
- Consult Project
- Consult Task
- Consult Subtask
- Manage Response

- Manage Comments
- Manage Users
- Manage Teams
- Manage Projects
- Manage Tasks
- Manage Subtasks
- Manage Documents

2.2.2 Non-Functional requirements

The non-functional requirements describe the developed system qualities when it is performing one of its use cases. In fact, these requirements are more of a set that defines the manners to hold the functional ones on a line that is bound to the software system judgements followed by its security, portability, usability etc.... Therefore, the system won't meet the expectations of neither the owner nor the user unless these requirements are fulfilled. Yet, even if the requirements are not respected, the system would still work.

- Security
- Usability
- Ergonomics
- Maintainability
- Extensibility
- Modularity

2.2.3 Actors Identification

An actor is the role fulfilled by a user interacting with the developed system. This compartment actors are the following

Table 2.1: Detailed description of the actors

Actor	Roles
Administrator	<ul style="list-style-type: none"> ● Authenticate ● Manage Users ● Manage Documents ● Manage Teams ● Manage Projects ● Manage Tasks ● Manage Subtasks ● Consult Tickets
Staff	<ul style="list-style-type: none"> ● Authenticate ● Consult Tickets ● Consult Projects ● Consult Tasks ● Consult Subtasks ● Manage Responses ● Manage Comments
Customer	<ul style="list-style-type: none"> ● Authenticate ● Consult Documents ● Manage Tickets ● Consult Responses

2.3 Global use case diagram

The following figure showcases the global use case diagram of our project.

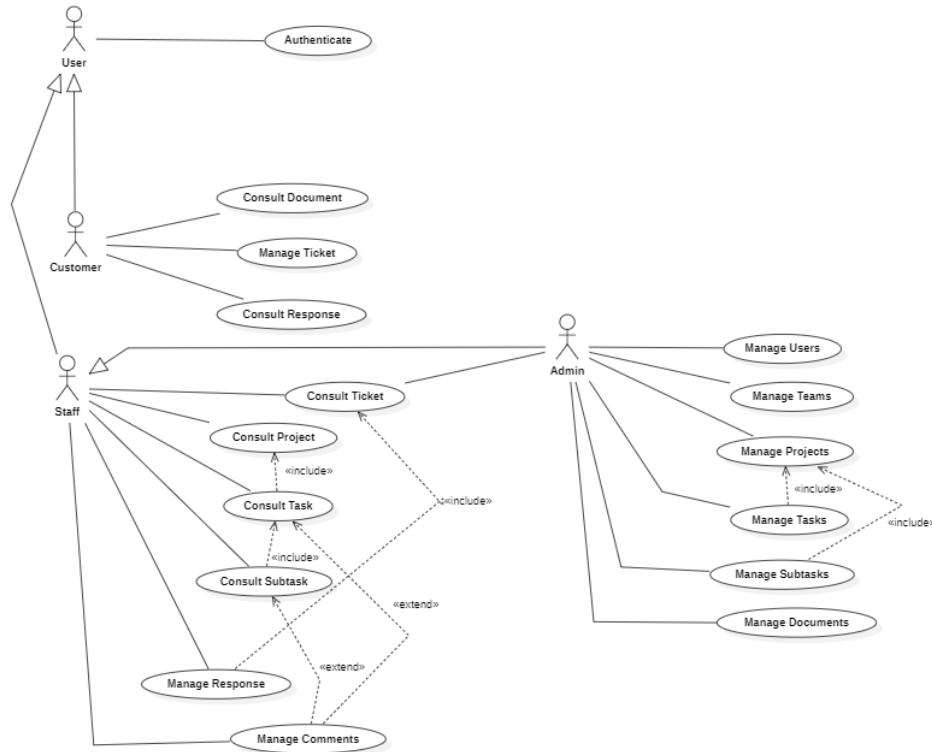


Figure 2.1: global use case diagram

2.4 Product backlog

The backlog product is list based on prioritizing the tasks of developing the project following levels bound to fulfil the system requirements. Therefore, the most important task will have the top level so that the development team identifies easily what to deliver first

2.4.1 Project management

Table 2.2: Product backlog Release 1

Priority	User story	Sprint	Estimation	Release
1	As a user, I can authenticate	Sprint 0	Medium	1
1	As an administrator, I can manage users	Sprint 0	High	1
1	As a administrator, I can manage documents	Sprint 0	High	1
1	As a administrator, I can manage teams	Sprint 0	High	1
2	As an administrator, I can manage projects	Sprint 1	High	1
2	As an administrator, I can manage tasks	Sprint 1	High	1
2	As an administrator, I can manage subtasks	Sprint 1	Medium	1
2	As an customer, I can manage tickets	Sprint 1	High	1

2.4.2 Project Follow-up

Table 2.3: Product backlog Release 2

Priority	User story	Sprint	Estimation	Release
3	As a staff member, I can consult tickets	Sprint 2	Medium	2
3	As a staff member, I can consult projects	Sprint 2	Medium	2
3	As a staff member, I can consult tasks	Sprint 2	Medium	2
3	As a staff member, I can consult subtasks	Sprint 2	Medium	2
4	As a staff member, I can manage responses	Sprint 3	High	2
4	As a staff member, I can manage comments	Sprint 3	High	2
4	As a customer, I can consult documents	Sprint 3	medium	2
4	As a customer, I can consult Responses	Sprint 3	Medium	2

2.5 Conclusion

All over this chapter, we have gone through the requirements specification, analysing the functional non-functional ones and the actors interacting with this system. We proceeded then to the main functionalities going through by the global use case diagram. And eventually, we have concluded this chapter with the backlog product.

Chapter 3

Release 1 : Implementation Of The Project Management

3.1 Introduction

Now that we set up a perfect understanding of our system by analyzing its use cases, classifying them by priorities into 4 sprints of 2 releases, and identifying their actors relatively, we proceed to the next chapter and the first phase of development. We will be going through combined use cases of our system, which have the highest priority, by explaining their backlogs, elaborating their user stories with the refinements, then eventually showcasing them by implementing the class and sequence diagrams in addition to the final reveal of the developed interfaces.

3.2 Sprint 0

We will be presenting the following user stories in the first sprint:

- Authenticate
- Manage users
- Manage documents
- Manage teams

3.2.1 Identifying the sprint0 backlog

The following table contains the backlog elements that are realised during the sprint 0:

Table 3.1: Product backlog Release 1 sprint 0

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Priority	User story	Sprint	Estimation	Release
1	As a user, I can authenticate	Sprint 0	Medium	1
1	As an administrator, I can manage users	Sprint 0	High	1
1	As an administrator, I can manage documents	Sprint 0	High	1
1	As a administrator, I can manage teams	Sprint 0	High	1

3.2.2 Refinement of sprint 0

In this section, we analyze the different use-case scenarios of the first sprint.

Refinement of the user story "authenticate"

The following figure showcases the use case .

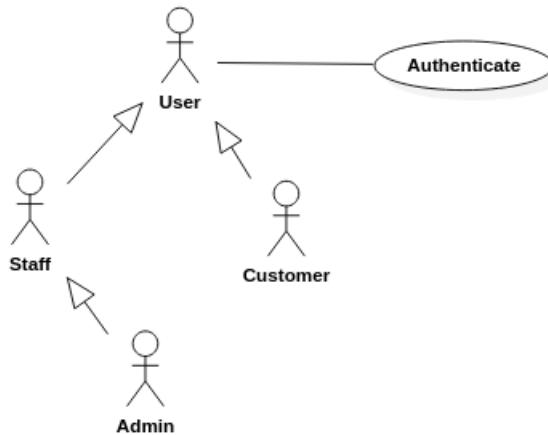


Figure 3.1: use case diagram “authenticate”

The following table elaborates on this user story with textual description.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Table 3.2: Detailed description of the "Authentication" use case

Use Case Scenario	As a user, I can authenticate
Actors	User
Pre-Conditions	The User must have an account
Post-Conditions	Authenticate
Main Scenario	<ul style="list-style-type: none"> • The user types the email and password. • The user clicks on the login button. • The system verifies whether the credentials exist in the database or not. • If they do: The system redirects the user to their specified Dashboard. • If not: the system will notify the user that their credentials are wrong so that they type their credentials again.

Refinement of the administrator story “manage users”

The following figure showcases the use case .

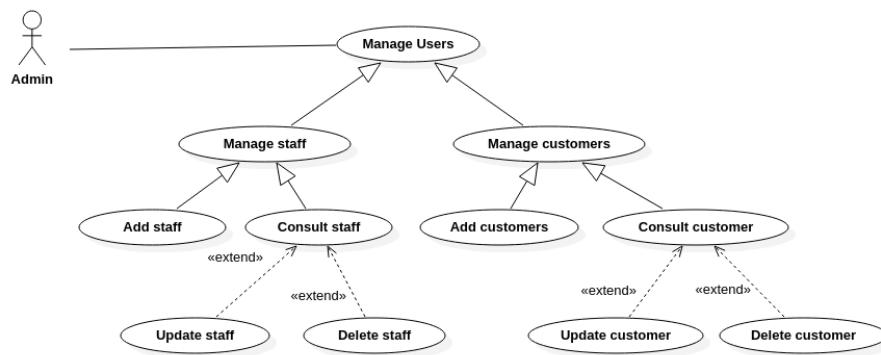


Figure 3.2: use case diagram “manage users”

The following table elaborates on this user story with textual description.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Table 3.3: Detailed description of the "manage users" use case

Use Case Scenario	As an administrator, I can manage users
Actors	Administrator
Pre-Conditions	The administrator must be connected
Post-Conditions	Users managed
Extensions	<ul style="list-style-type: none">• Add• Consult• Update• Delete
Main Scenario	<ul style="list-style-type: none">• The administrator chooses whether to add new customer or new staff member• The administrator clicks on the Client space button• The administrator clicks on the personnel space

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Secondary Scenario

- Add User:
 - The administrator clicks the add button
 - The system displays the desired form
 - The administrator types the required information about the user
 - The administrator clicks the validate button
 - The system adds a new user to the database with the input data and sends a notification to the user email
- Consult User:
 - The system displays the list of the users specific to the selected space
 - The administrator selects the user that they want to delete or update their information
- Update User:
 - The system displays the desired form
 - The administrator fills the form with the new data
 - The administrator clicks the validate button
 - The system updates the selected user in the database with the new input data
- Delete User:
 - The administrator clicks on the delete button
 - The system drops the selected user from the database

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Refinement of the administrator story “manage documents”

The following figure showcases the use case .

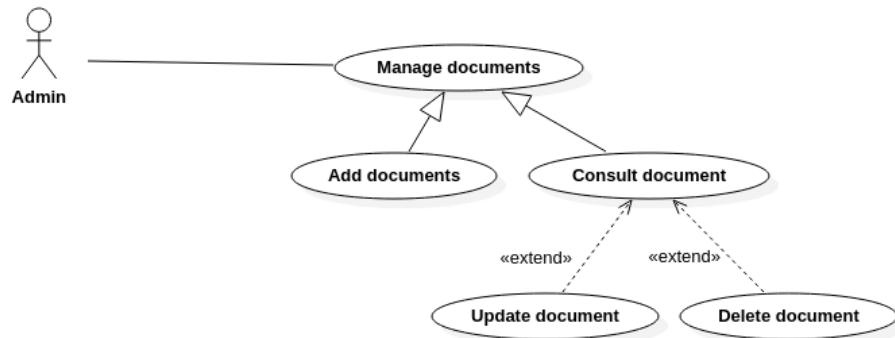


Figure 3.3: use case diagram “manage documents”

The following table elaborates on this user story with textual description.

Table 3.4: Detailed description of the ”manage documents” use case

Use Case Scenario	As an administrator, I can manage documents
Actors	Administrator
Pre-Conditions	The administrator must be connected
Post-Conditions	documents managed
Extensions	<ul style="list-style-type: none">• Add• Consult• Update• Delete

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Main Scenario	<ul style="list-style-type: none">● The administrator clicks on the documents' space button● The system displays the list of documents
Secondary Scenario	<ul style="list-style-type: none">● Add document:<ul style="list-style-type: none">– The administrator clicks the add button– The system displays the desired form– The administrator types the required information about the document– The administrator clicks the validate button– The system adds a new document to the database● Consult document:<ul style="list-style-type: none">– The system displays the list of the documents specific to the selected space– The administrator selects the document that they want to delete or update their information● Update document:<ul style="list-style-type: none">– The system displays the desired form– The administrator fills the form with the new data– The administrator clicks the validate button– The system updates the selected document in the database with the new input data● Delete document:<ul style="list-style-type: none">– The administrator clicks on the delete button– The system drops the selected document from the database

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Refinement of the administrator story “manage teams”

The following figure showcases the use case .

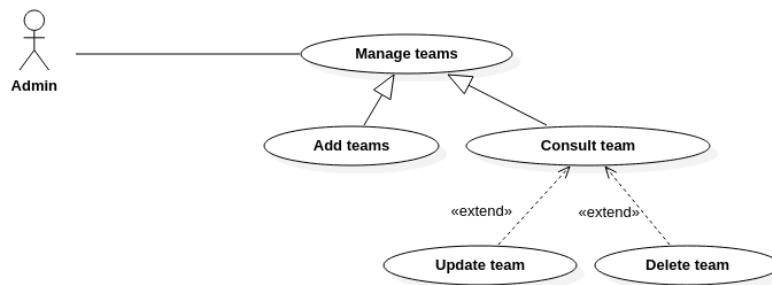


Figure 3.4: use case diagram “manage teams”

The following table elaborates on this user story with textual description.

Table 3.5: Detailed description of the ”manage teams” use case

Use Case Scenario	As an administrator, I can manage teams
Actors	Administrator
Pre-Conditions	The administrator must be connected
Post-Conditions	Teams managed
Extensions	<ul style="list-style-type: none">• Add• Consult• Update• Delete

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Main Scenario	<ul style="list-style-type: none">● The administrator clicks on the teams' space button● The system displays the list of teams
Secondary Scenario	<ul style="list-style-type: none">● Add team:<ul style="list-style-type: none">– The administrator clicks the add button– The system displays the desired form– The administrator types the required information about the team– The administrator clicks the validate button– The system adds a new team to the database● Consult team:<ul style="list-style-type: none">– The system displays the list of the teams specific to the selected space– The administrator selects the team that they want to delete or update their information● Update team:<ul style="list-style-type: none">– The system displays the desired form– The administrator fills the form with the new data– The administrator clicks the validate button– The system updates the selected team in the database with the new input data● Delete team:<ul style="list-style-type: none">– The administrator clicks on the delete button– The system drops the selected team from the database

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

3.2.3 Deployment diagram

A deployment diagram is a UML diagram that illustrates the global architecture of a system by covering both software and hardware executing it, and showcases the connection between them. The following figure is a demonstration of the deployment diagram of our project.

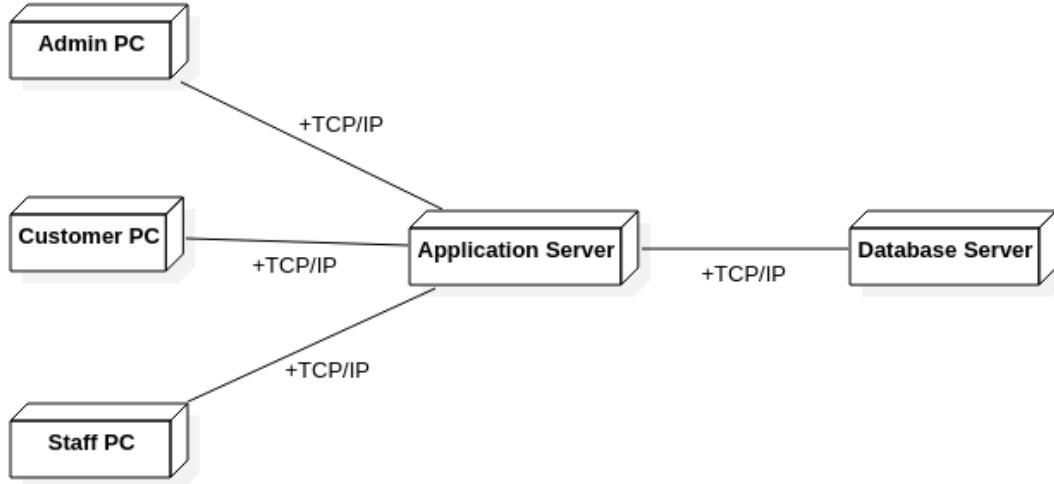


Figure 3.5: Deployment diagram

3.2.4 Design of sprint 0

We will present the class and sequence diagrams of the different use-case scenarios that we implemented in the last section of their refinements.

Design of the user story “authenticate”

Class diagram

The following figure represents the class diagram of this use-case.

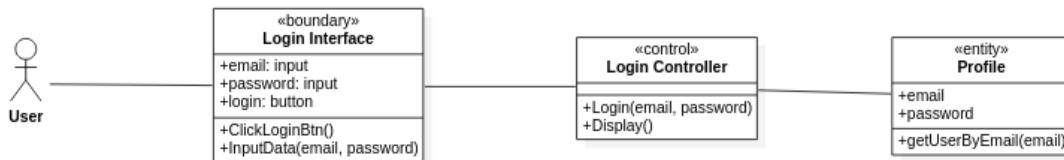


Figure 3.6: class diagram of the user story “authenticate”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

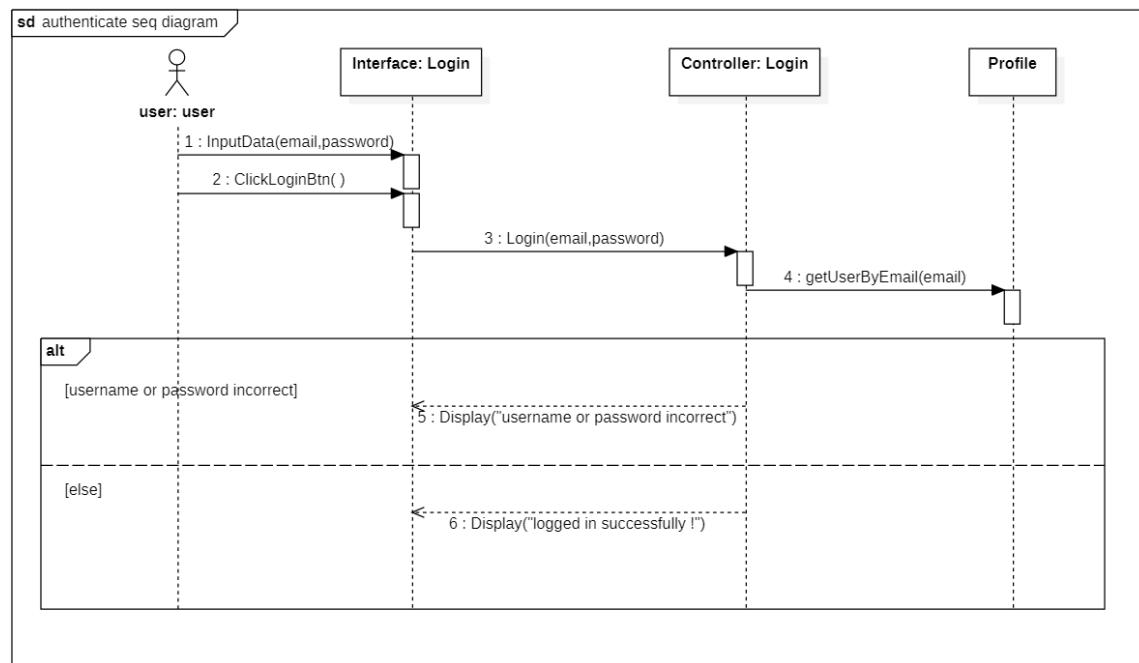


Figure 3.7: sequence diagram of the user story “authenticate”

Design of the administrator story “manage users”

Manage staff

Class diagram

The following figure represents the class diagram of this use-case.

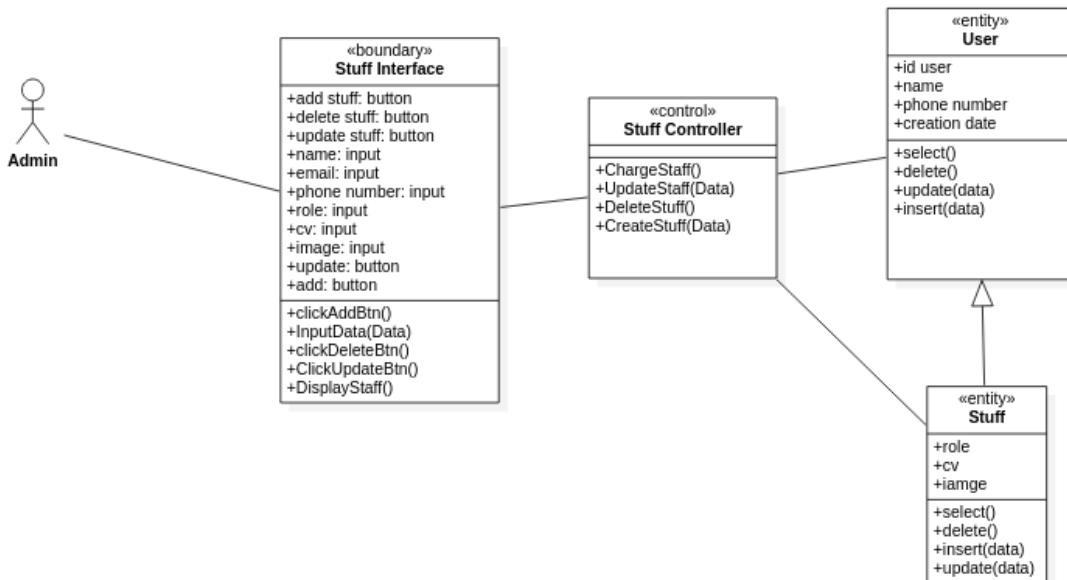


Figure 3.8: class diagram of the administrator story “manage staff”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Sequence diagram

The following figure represents the sequence diagram of this use-case.

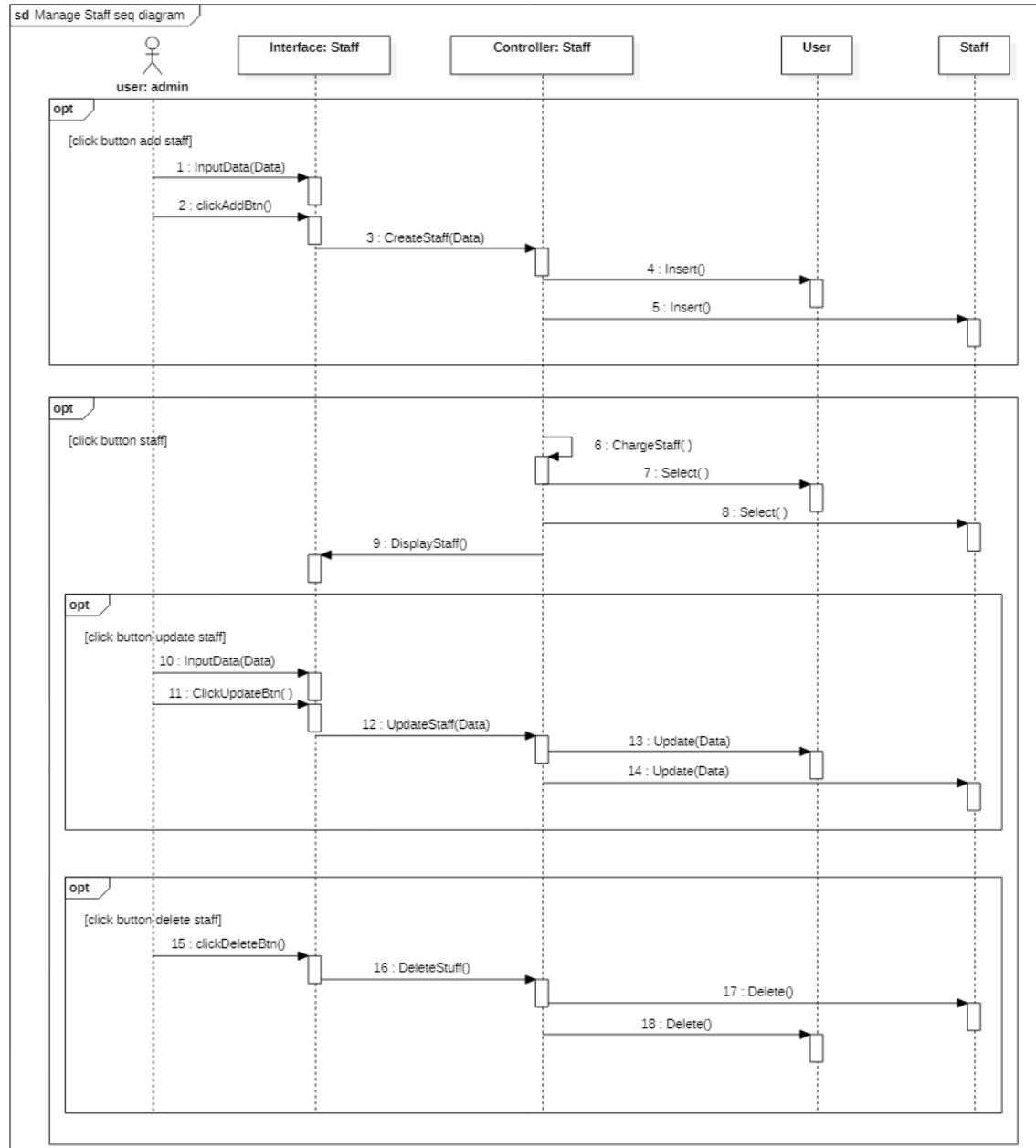


Figure 3.9: sequence diagram of the administrator story “manage staff”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Manage customers

Class diagram

The following figure represents the class diagram of this use-case.

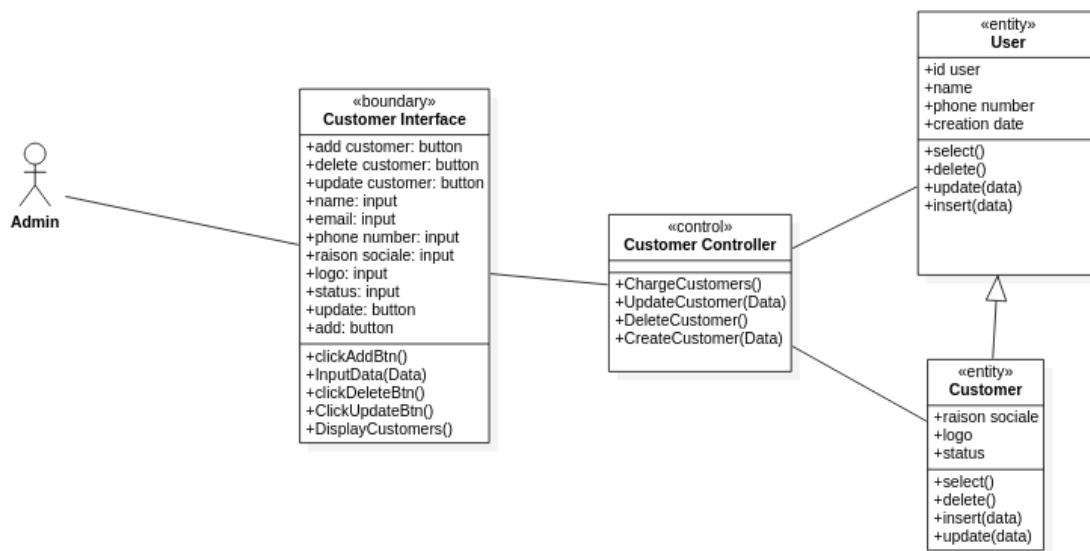


Figure 3.10: class diagram of the administrator story “manage customers”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Sequence diagram

The following figure represents the sequence diagram of this use-case.

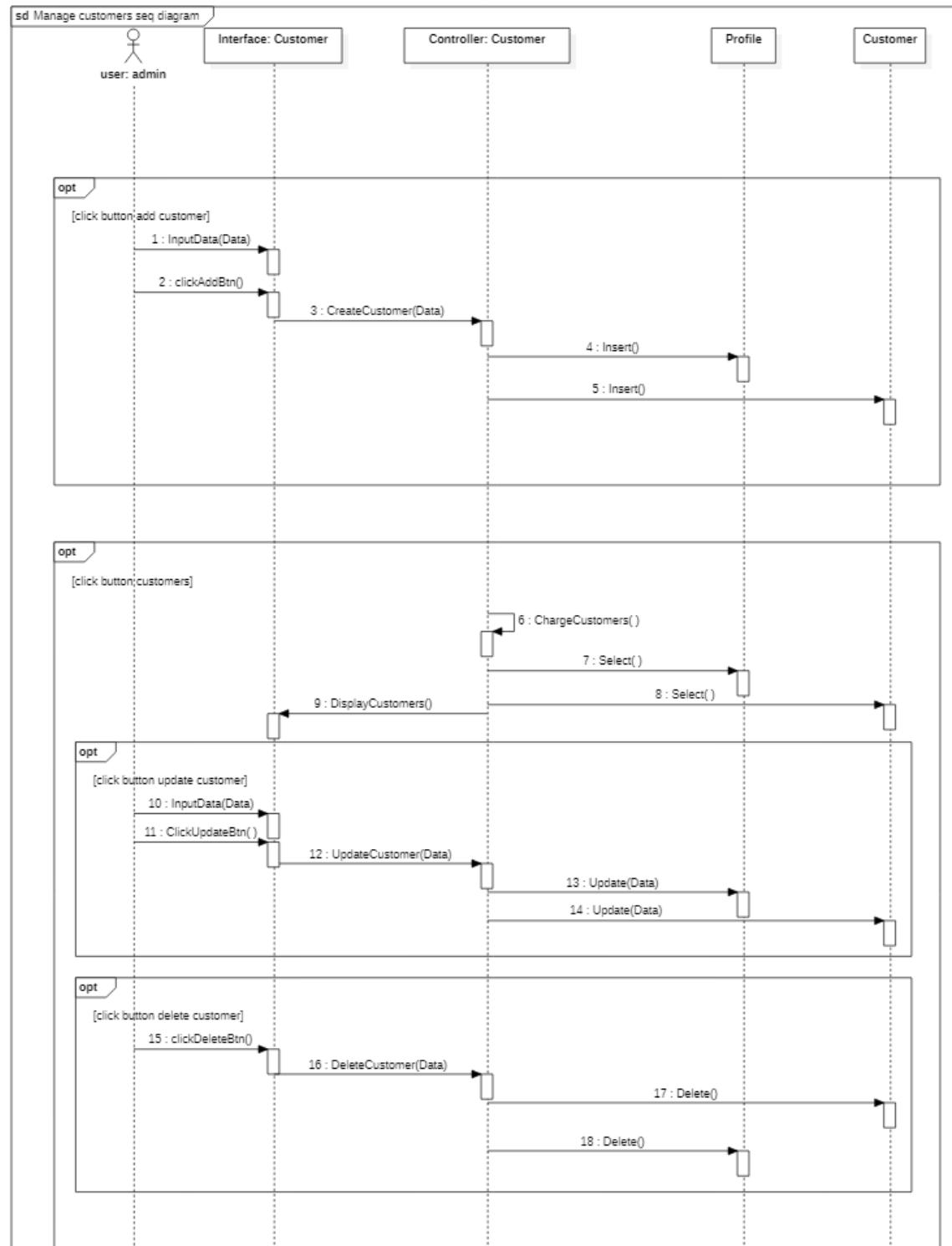


Figure 3.11: sequence diagram of the administrator story “manage customers”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Design of the administrator story “manage documents”

Class diagram

The following figure represents the class diagram of this use-case.

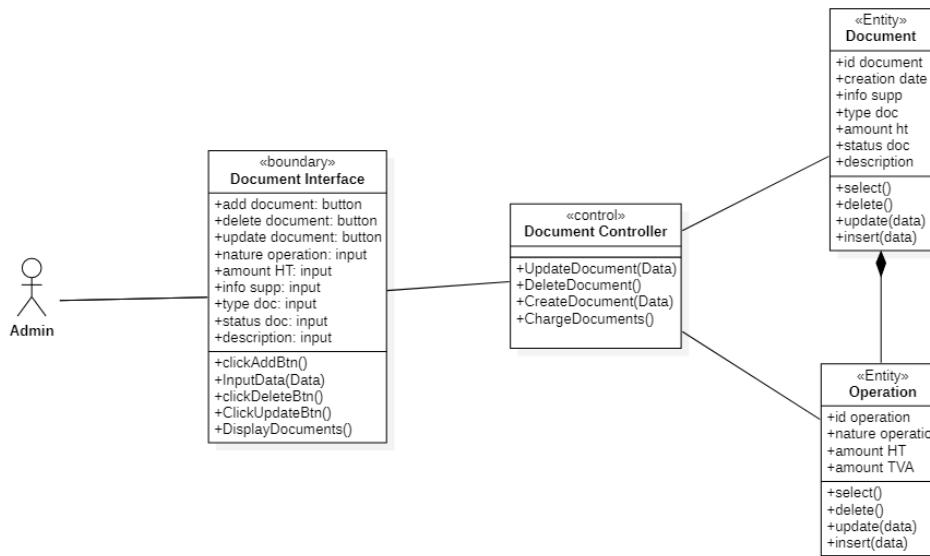


Figure 3.12: class diagram of the administrator story “manage documents”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Sequence diagram

The following figure represents the sequence diagram of this use-case.

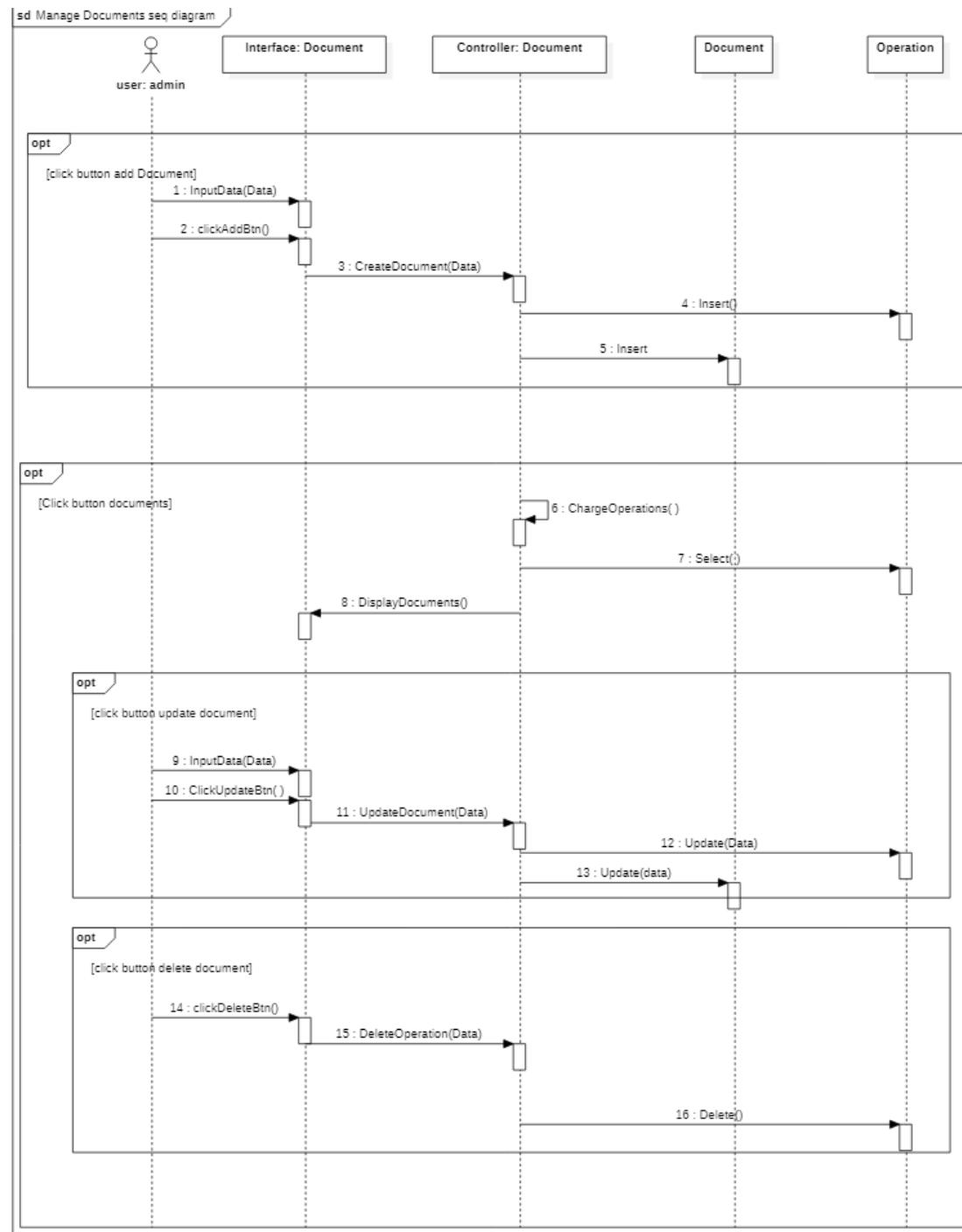


Figure 3.13: sequence diagram of the administrator story “manage documents”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Design of the administrator story “manage teams”

Class diagram

The following figure represents the class diagram of this use-case.

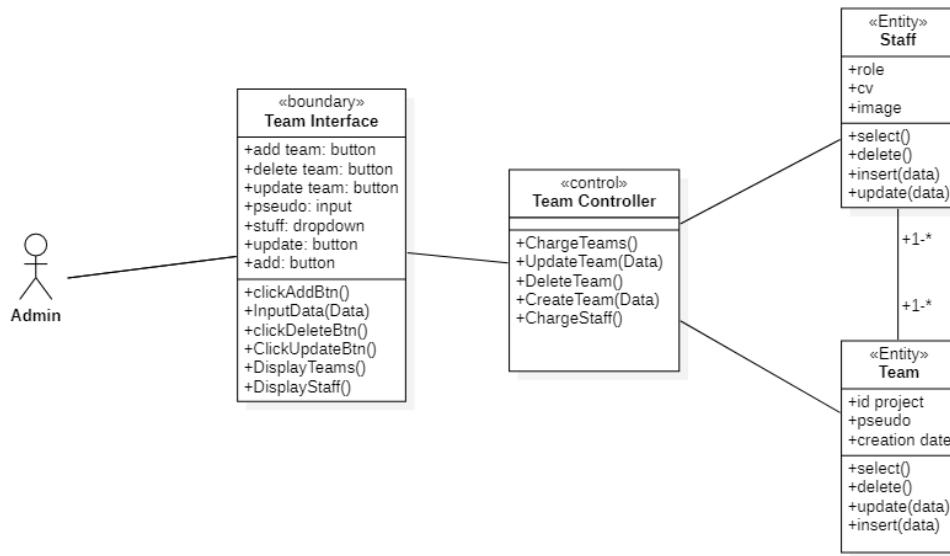


Figure 3.14: class diagram of the administrator story “manage teams”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

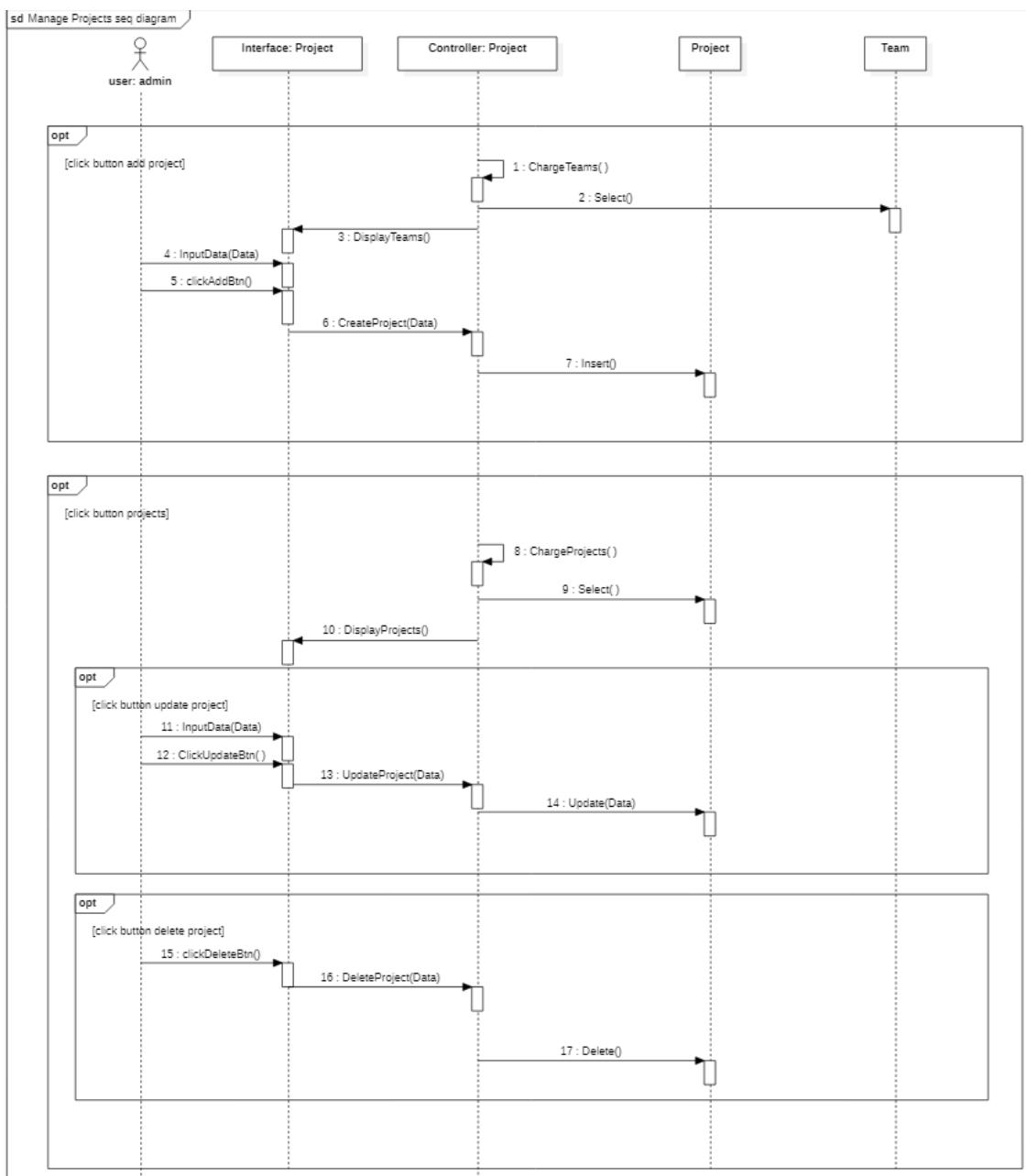


Figure 3.15: sequence diagram of the administrator story “manage teams”

3.2.5 Implementation of sprint 0

Realization of the user story “authenticate”

the following figure shows the authentication interface of all the users where they have two fields to fill with their credentials in order to authenticate and access the dashboard specified to their roles.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

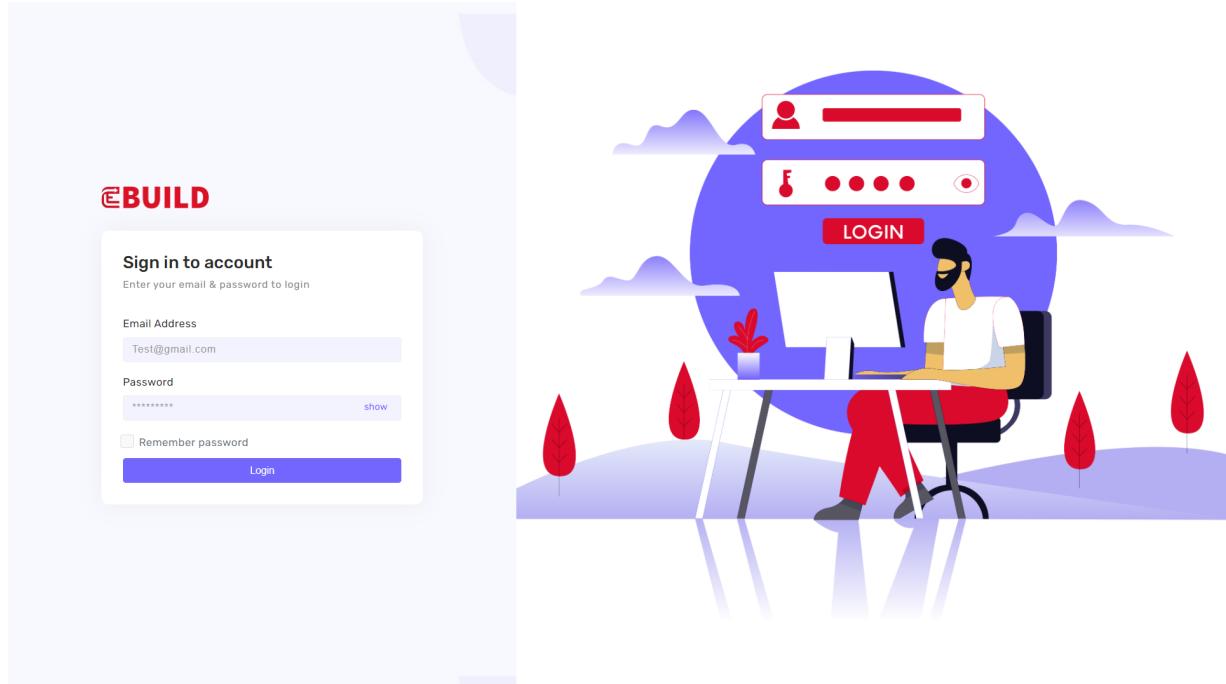


Figure 3.16: Authenticate Interface

If the credentials don't match any record in the database, or if the account was removed, the interface shows a red notification saying "The provided credentials are incorrect!"

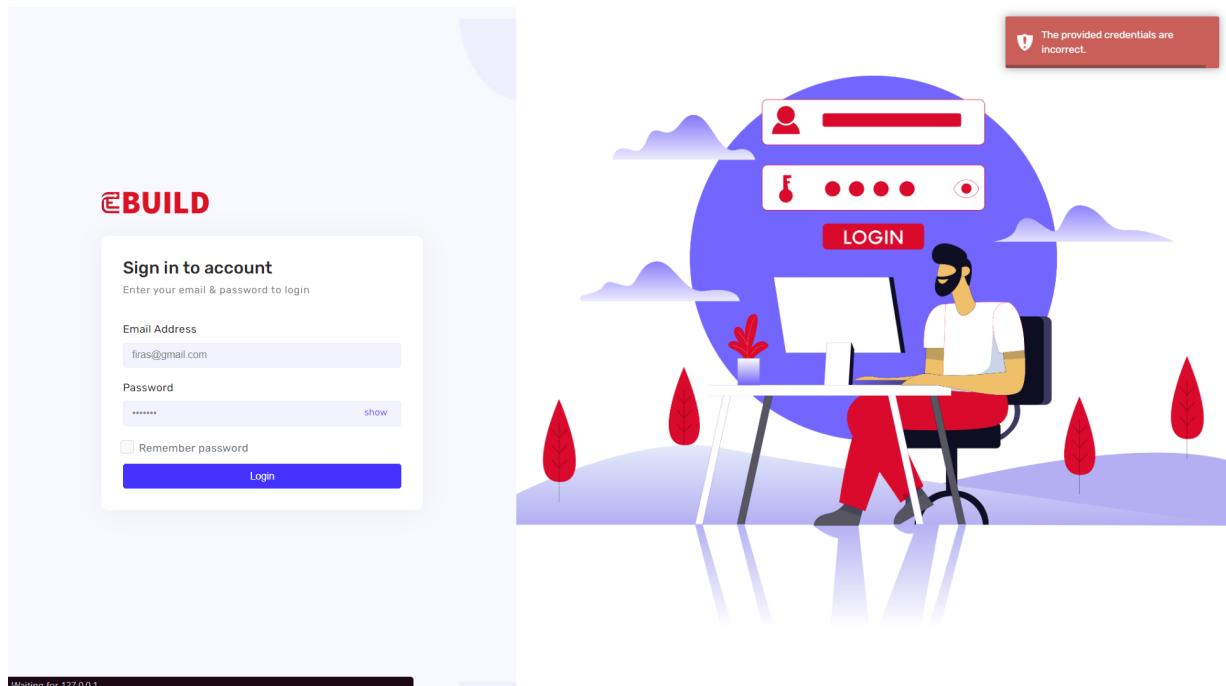


Figure 3.17: Authentication Error Notification

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Realization of the administrator story “manage users”

Manage staff

The following figure shows the manage users interface in dashboard admin through which the administrator can consult the list of the added staff members.

The screenshot shows the 'Staff' section of the Admin E-build dashboard. On the left, there's a sidebar with links: Customers, Staff (which is selected and highlighted in purple), Teams, Tickets, Documents, Projects, Tasks, and Subtasks. The main area has a title 'Add staff' with a red button. Below it is a search bar with 'Type to Search' and a 'Clear' button. To the right is a 'Per page' dropdown set to 5. A table lists two staff members:

name	email	num_tel	role	Actions
Firas Ayari	spy.firas16@gmail.com	22776572	admin	
Ala gtari	agtari957@gmail.com	28734227	dev fullstack	

At the bottom, there are navigation buttons for pages 0, <, 1 (highlighted in blue), >, and >>.

Figure 3.18: Manage Staff Interface

The administrator can add a new staff member by clicking the add staff button. A modal form is displayed with required fields to fill. Upon submitting the form, a notification pops up confirming the addition of the new staff member with the administrator being redirected to the same interface again with the updated list of staff.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

The screenshot shows the 'Add Staff' form in the E-BUILD application. The left sidebar has a 'Staff' button highlighted. The main area shows fields for Name (Ahmed chebbi), password (1548xdfcgnb51), num_tel (66666666), email (ahmed@gmail.com), CV (cv.pdf), image (image.png), and role (dev mobile). There are 'Submit' and 'Reset' buttons at the bottom. The background shows a list of staff members with edit and delete icons.

Figure 3.19: Add Staff Interface

The administrator can either update or delete a staff member by clicking the update or delete button next to their shown information.

The screenshot shows the 'Update Staff' form in the E-BUILD application. The left sidebar has a 'Staff' button highlighted. The main area shows fields for Name (Ahmed chebbi), num_tel (66666666), email (ahmed.chebbi@gmail.com), CV (cv.pdf), image (image.png), and role (dev mobile). The 'email' field has a green checkmark. There are 'Submit' and 'Reset' buttons at the bottom. The background shows a list of staff members with edit and delete icons.

Figure 3.20: Update Staff Interface

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

The screenshot shows the 'Staff' section of the E-BUILD application. On the left sidebar, 'Staff' is highlighted. The main area displays a table of staff members with columns for name, email, num_tel, and role. Each row includes a trash icon with a checkmark for deletion. The table shows:

name	email	num_tel	role
Firas Ayari	spy.firas16@gmail.com	22776572	admin
Ala gtari	agtari957@gmail.com	28734227	dev fullstack
Ahmed chebbi	ahmed.chebbi@gmail.com	66666666	dev mobile

At the bottom, there is a navigation bar with icons for back, forward, and search.

Figure 3.21: Updated Staff Interface

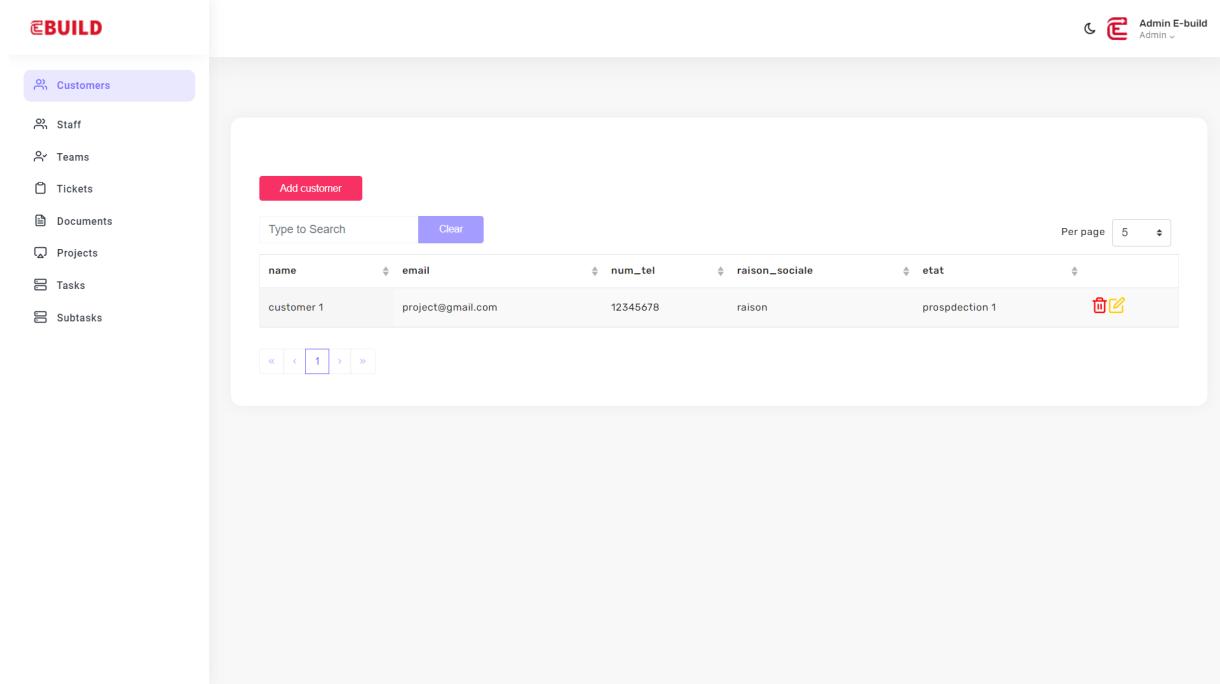
This screenshot shows the same Staff interface as Figure 3.21, but with the staff members Firas Ayari, Ala gtari, and Ahmed chebbi listed as deleted. The trash icons with checkmarks are now displayed next to each row, and the rows are shaded grey to indicate they are no longer active.

Figure 3.22: Deleted Staff Interface

Manage customers

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

The following figure shows the manage users interface in dashboard admin through which the administrator can consult the list of the added customers.



The screenshot shows the 'Customers' section of the Admin E-build dashboard. On the left, there's a sidebar with icons for Staff, Teams, Tickets, Documents, Projects, Tasks, and Subtasks. The 'Customers' icon is highlighted with a purple background. The main area has a header with the Admin E-build logo and a dropdown for 'Admin'. Below the header is a search bar with 'Type to Search' and a 'Clear' button. To the right of the search bar is a 'Per page' dropdown set to 5. A table lists customer data with columns: name, email, num_tel, raison_sociale, etat, and actions. The first row shows 'customer 1', 'project@gmail.com', '12345678', 'raison', 'prospdction 1', and two small icons for delete and edit. At the bottom of the table is a navigation bar with icons for back, forward, and search.

Figure 3.23: Manage Customers Interface

The administrator can add a new customer by clicking the add customer button. A modal form is displayed with required fields to fill. Upon submitting the form, a notification pops up confirming the addition of the new customer with the administrator being redirected to the same interface again with the updated list of customers.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

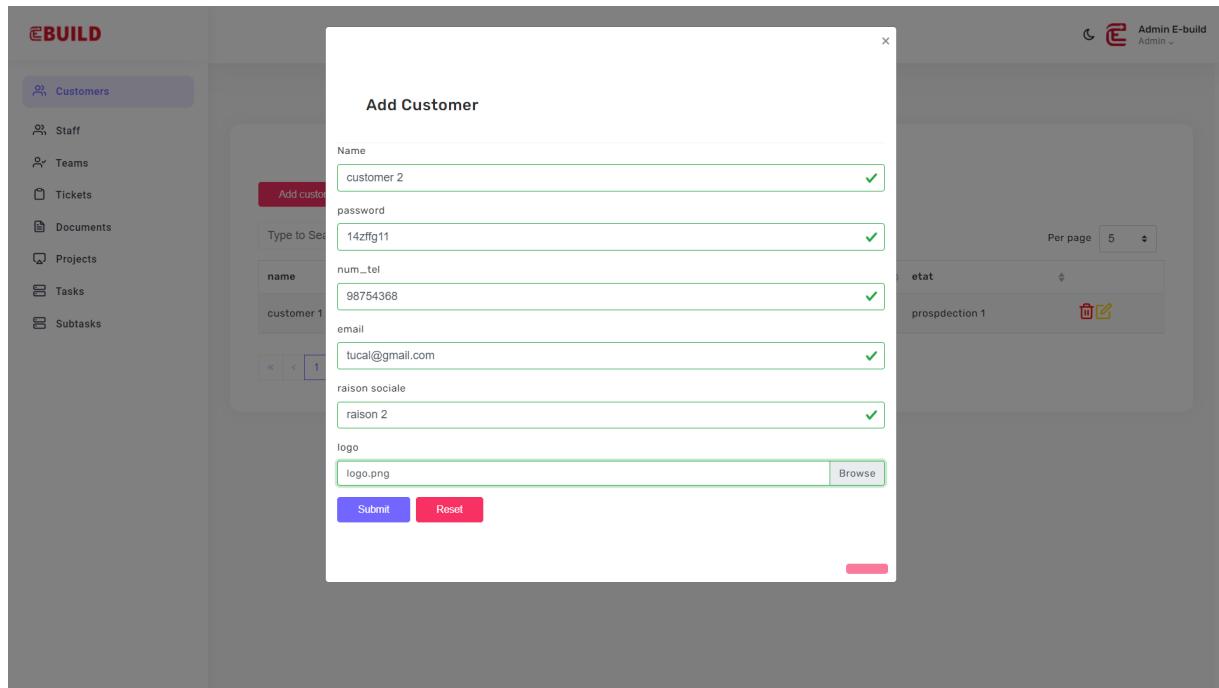


Figure 3.24: Add Customer Interface

The administrator can either update or delete a customer by clicking the update or delete button next to their shown information.

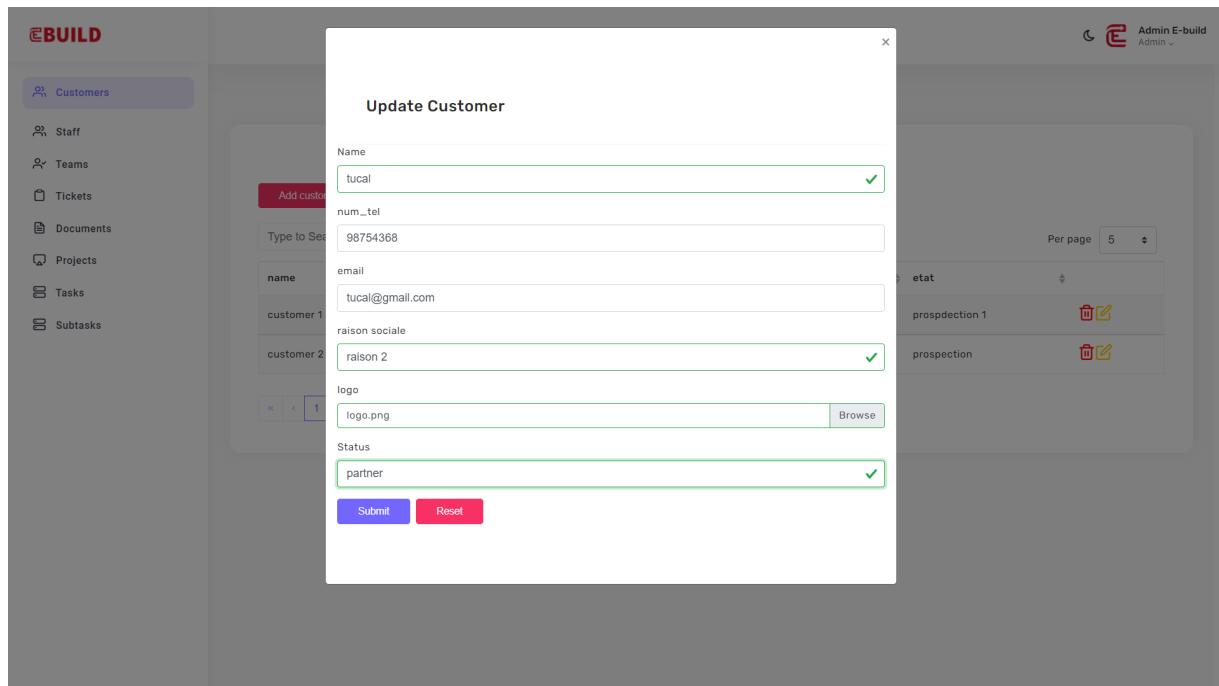
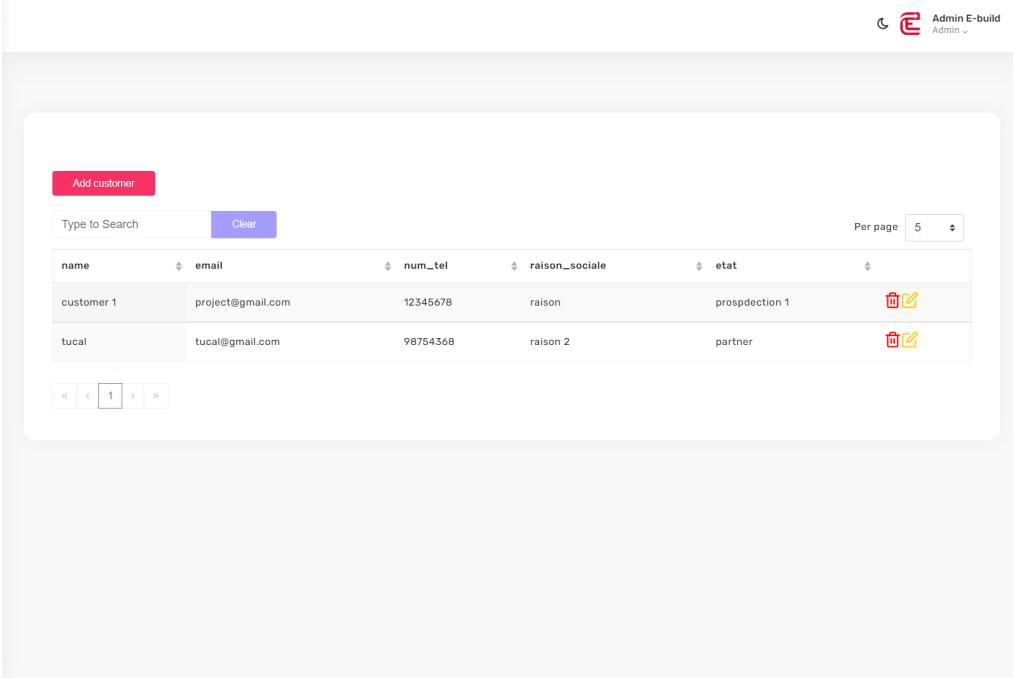


Figure 3.25: Update Customer Interface

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

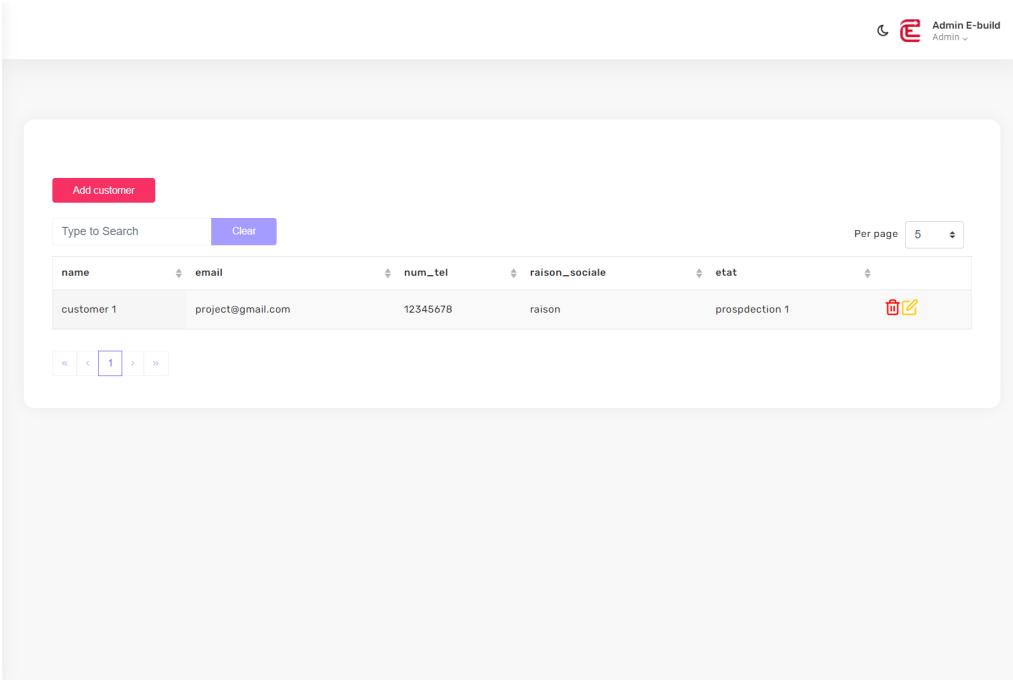


The screenshot shows a user interface for managing customers. On the left is a sidebar with icons for Customers (highlighted in purple), Staff, Teams, Tickets, Documents, Projects, Tasks, and Subtasks. The main area has a title 'Add customer' and a search bar. A table lists two entries:

name	email	num_tel	raison_sociale	etat	
customer 1	project@gmail.com	12345678	raison	prospdection 1	 
tucal	tucal@gmail.com	98754368	raison 2	partner	 

Pagination at the bottom shows pages 1, 2, and 30.

Figure 3.26: Updated Customer Interface



This screenshot shows the same customer management interface after one entry has been deleted. Only the entry 'customer 1' remains in the list.

name	email	num_tel	raison_sociale	etat	
customer 1	project@gmail.com	12345678	raison	prospdection 1	 

Figure 3.27: Deleted Customer Interface

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Realization of the administrator story “manage documents”

The following figure shows the manage documents interface in dashboard admin through which the admin can consult the list of the available documents.

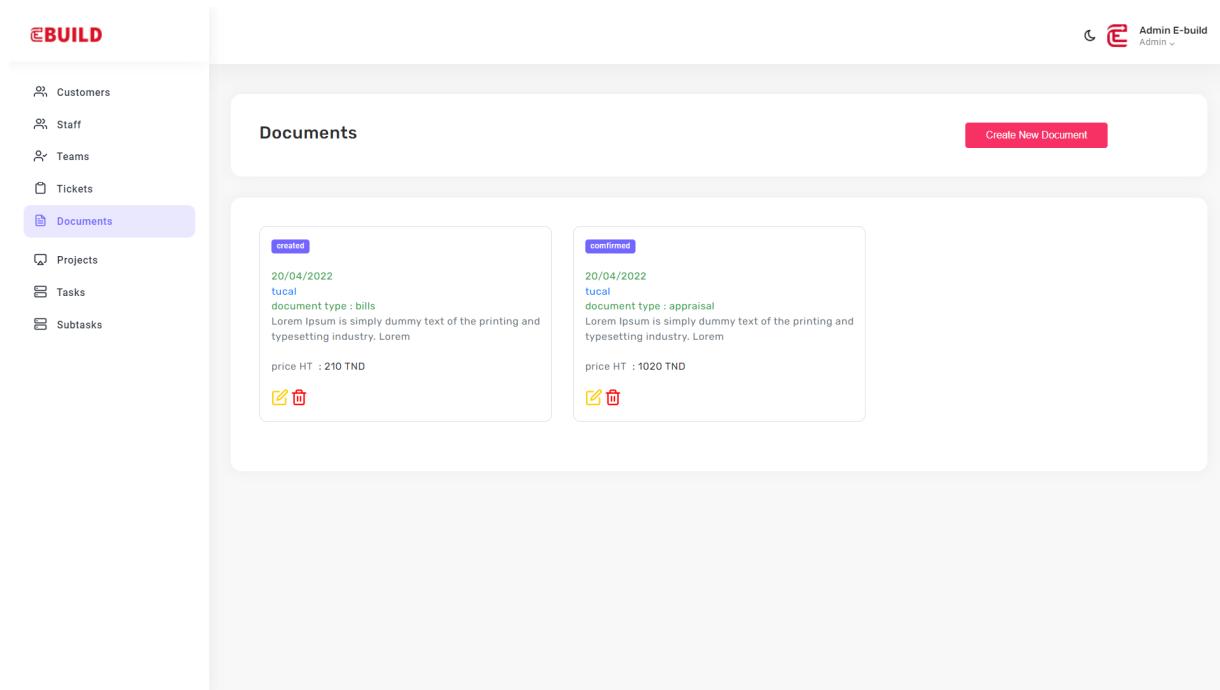


Figure 3.28: Manage Documents Interface

Upon clicking create new document button, a form pops up showing the required fields to fill. The administrator can choose between generating a pdf file and sending it to the client via email or just downloading it. When submitting the form, a notification pops up confirming the addition of the new document and its specific operations then the administrator is redirected to the documents interface.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

EBUILD

Customers
Staff
Teams
Tickets
Documents
Projects
Tasks
Subtasks

Documents

created
20/04/2022
tocal
document type : bills
Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scattered it across the page to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum

price HT : 210 TND

Create Document

Type doc: appraisal

Amount HT: 2000

Customer: customer 1

Estat doc: created

Description: Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scattered it across the page to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum

Info supp: Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scattered it across the page to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum

Operations

Nature	Amount HT	Amount TVA
nature 1	500	50
nature 2	1500	150

Add operation

Send PDF **Download PDF**

Create New Document

Admin E-build

Figure 3.29: Add Document Interface



E-BUILD

 Admin E-build
Admin ▾

-  Customers
-  Staff
-  Teams
-  Tickets
-  **Documents**
-  Projects
-  Tasks
-  Subtasks

Documents

[Create New Document](#)

<p>created</p> <p>20/04/2022 test document type : bills Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem price HT : 210 TND</p> <p> </p>	<p>confirmed</p> <p>20/04/2022 test document type : appraisal Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem price HT : 1020 TND</p> <p> </p>	<p>created</p> <p>20/04/2022 customer 1 document type : appraisal Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem price HT : 2000 TND</p> <p> </p>
--	---	---

Figure 3.30: Added Document Interface

Each document type has its own generated pdf format.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

eBUILD		Devis N°16
Tél :	25892403	HYGETEC
Date du devis	06/05/2022	
Description		Prix total HT
Conception Hiérarchique du site • Etude personnalisée de votre projet		
Webdesign du site Internet - Layout général • Identification de l'architecture des pages (menu/fenêtre) [Accueil à propos, Produits, Actualité, Contact] • Ajustements selon les désirs du client		
Intégration du Design Création / redésgin de Template • Validité du site W3C • Compatible tous navigateurs • Idénes de site • Gallerie Photos • Formulaire de contact		
Site mobile (responsive) Le web d'aujourd'hui est mobile, vos clients le sont également. Iphone, smartphone et tablette. Nous pouvons concevoir votre site en anticipant la manière dont il sera visible sur les supports mobiles (tablettes et smartphones) • Design de site adapté aux spécificités techniques des tablettes et smartphones • Augmentez votre SEO (référencement naturel)		
LIVRAISON / DEPLOIEMENT • Mise en production du site • Conguration Serveur		
Interface Backoffice • Interface Dashboard • Interface Gestion Actualités • Interface Gestion Produit • Interface Gestion Blog		

Figure 3.31: Appraisal PDF Format Example



CLIENT
Tucal
RSF9+RAV, Oued Eill

FACTURE N° 00251

Date 14-04-2022

Référence	Description	PU HT	TVA	Total HT
00251	Conception Hiérarchique du site	19 %		
00252	Webdesign de site Internet - Layout général	19 %		
00253	Intégration du Design	19 %		
00254	Site mobile (responsive)	19 %		
00255	LIVRAISON / DÉPLOIEMENT	19 %		
00256	Interface backoffice	19 %		
00257	Serveur & hébergement	19 %		

TVA	Base	Montant
19 %		

TOTAL HT
BASE TVA
TOTAL TVA
Timbre fiscal

NET À PAYER

Règlement de Paiement :

- Avance de 20 % du Montant Total TTC à la Commande
- Reste de la facture à la livraison

Offre gratuite :

- Maintenance site 12 mois
- Formation back office
- maintenance Serveur et déploiement

INFORMATIONS ENTREPRISE
MT-1751386/TAIM
CONTACT: 25882403

INFORMATIONS BANCAIRE
Banque Attijar
IBAN 00004 00120 1200077036879 9

Généré par Swiver.io

Figure 3.32: Bills PDF Format Example

The administrator can update the consulted document by clicking the update button and altering the input of the form.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

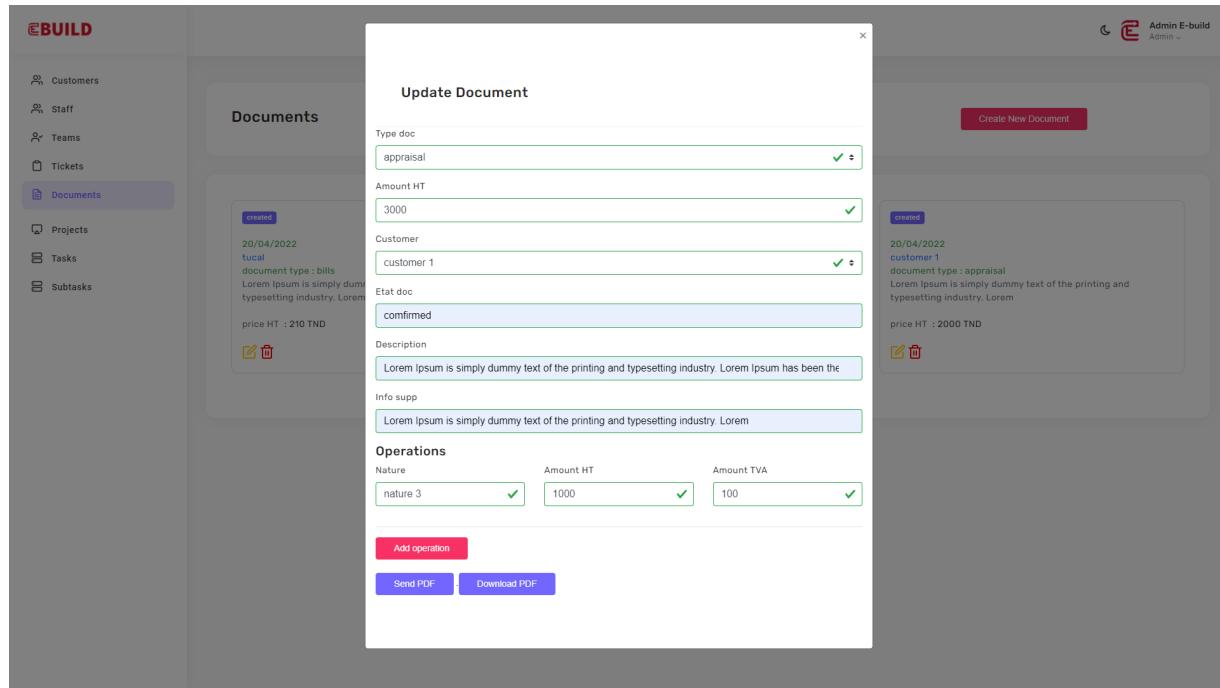


Figure 3.33: Update Document Interface

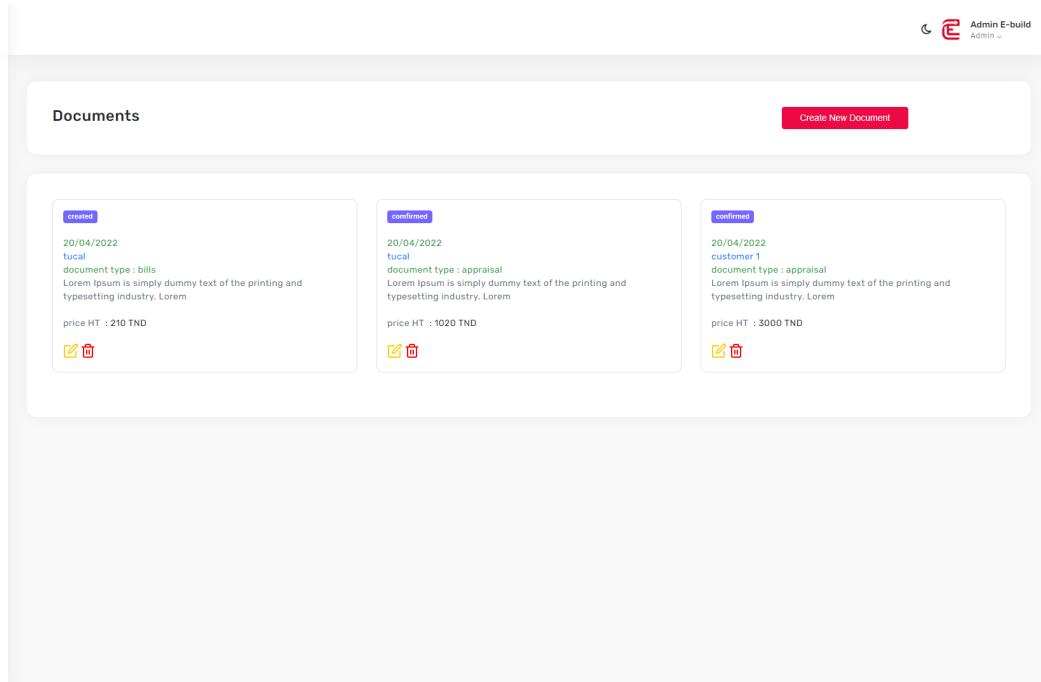


Figure 3.34: Updated Document Interface

The administrator can also delete a document and it is operations when they click the delete button.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

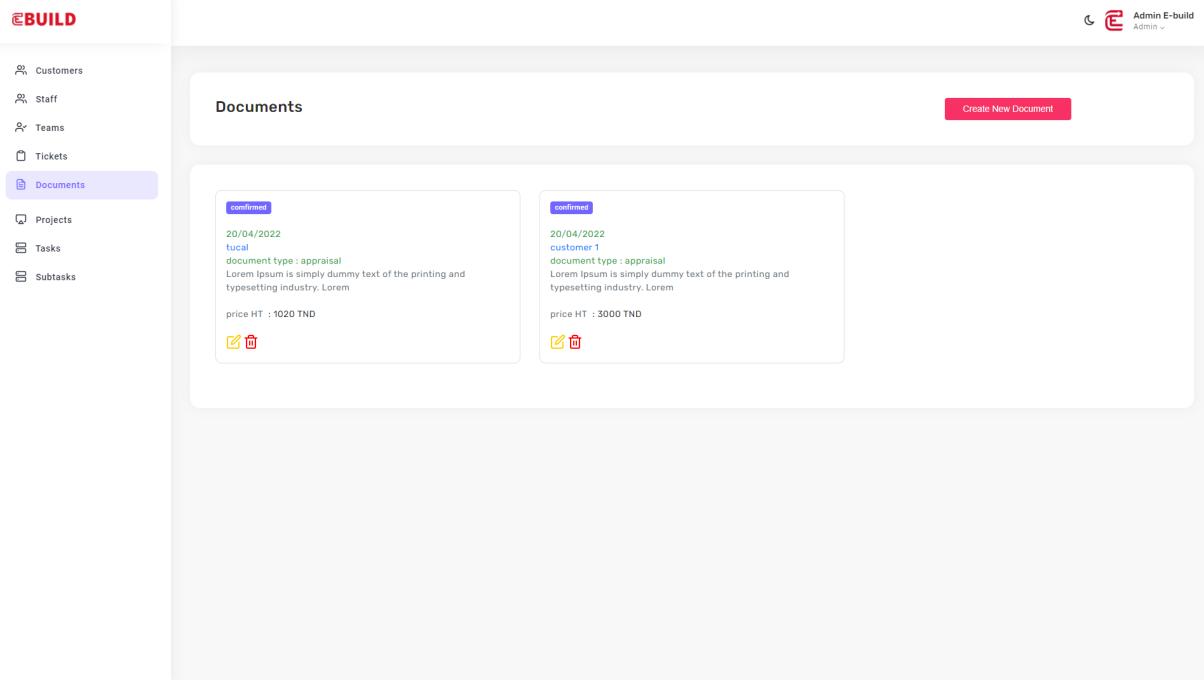


Figure 3.35: Deleted Document Interface

Realization of the administrator story “manage teams”

The following figure shows the manage teams interface in dashboard admin through which the admin can consult the list of the available teams, the administrator can either consult the members of a team when clicking on its name or add a new team by clicking on the add new team button.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

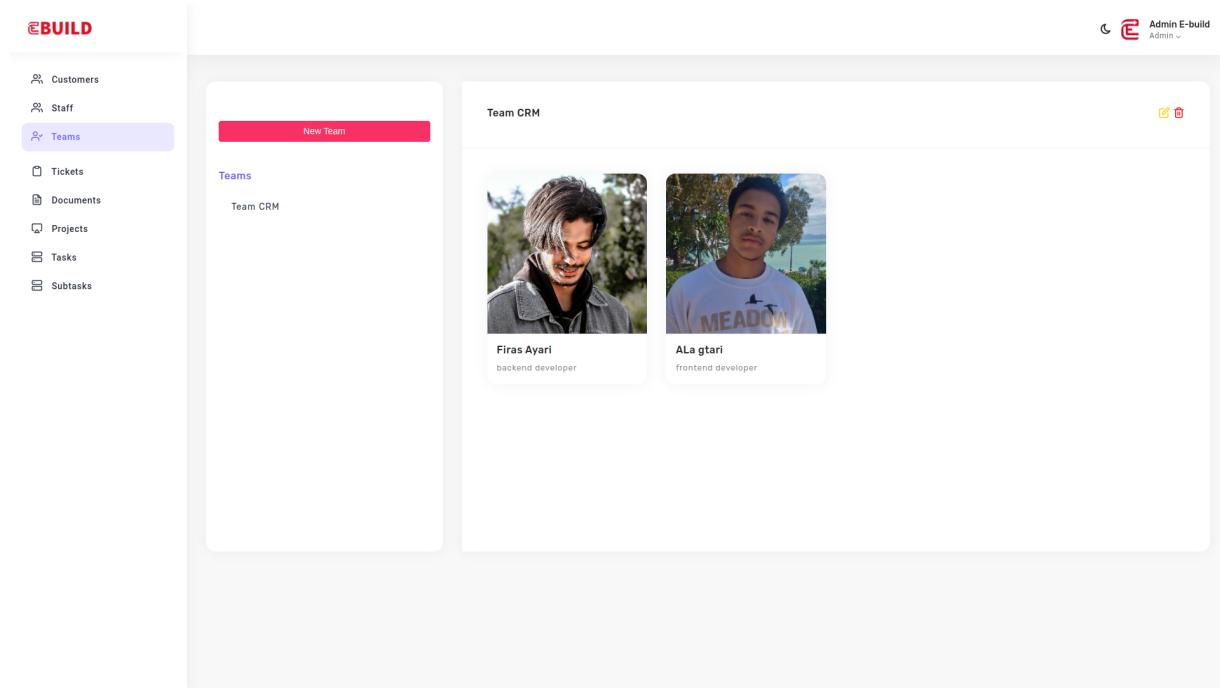


Figure 3.36: Manage Teams Interface

Upon clicking new team button, a form pops up showing the required fields to fill. When submitting the form, a notification pops up confirming the addition of the new team and then the administrator is redirected to the teams interface.

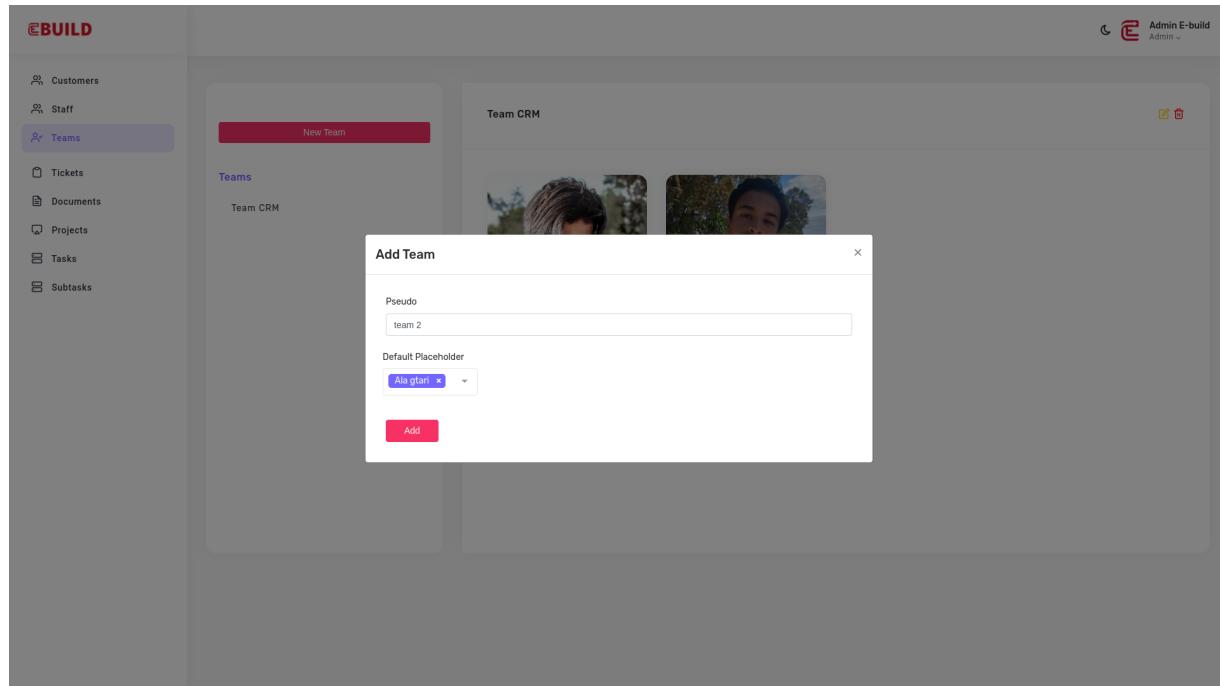


Figure 3.37: Add Team Interface

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

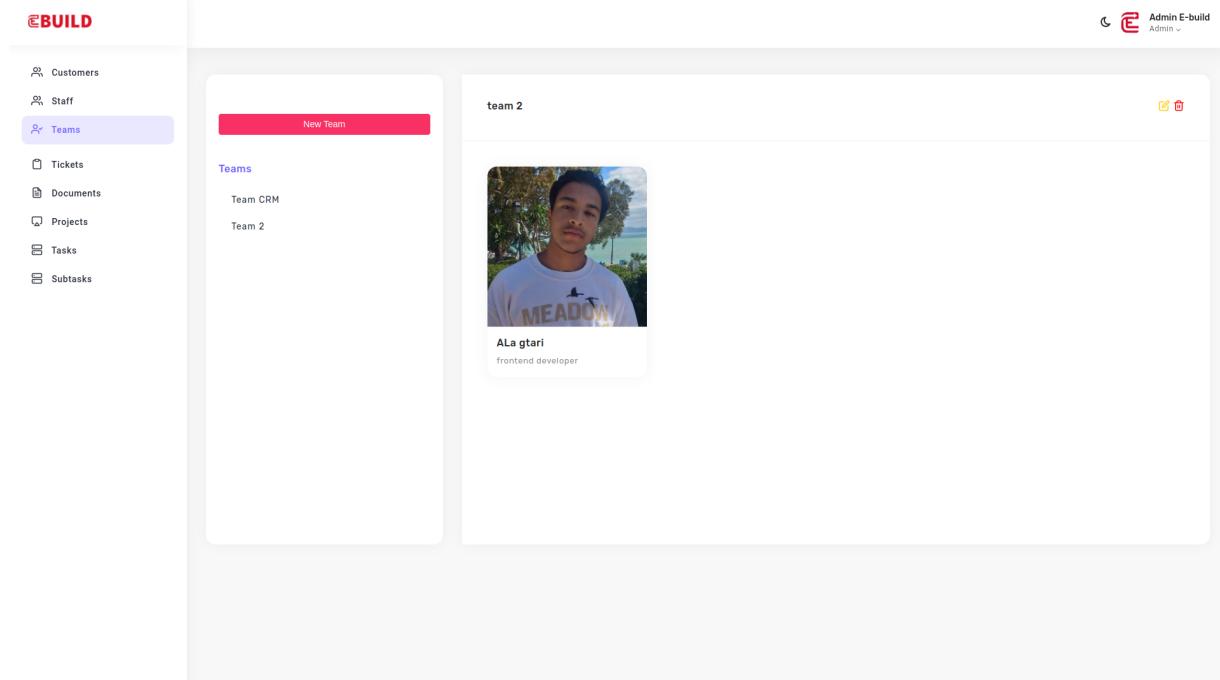


Figure 3.38: Added Team Interface

The administrator can update the consulted team by clicking the update button and altering the input of the form. While updating a team, the administrator can add other staff members to it.

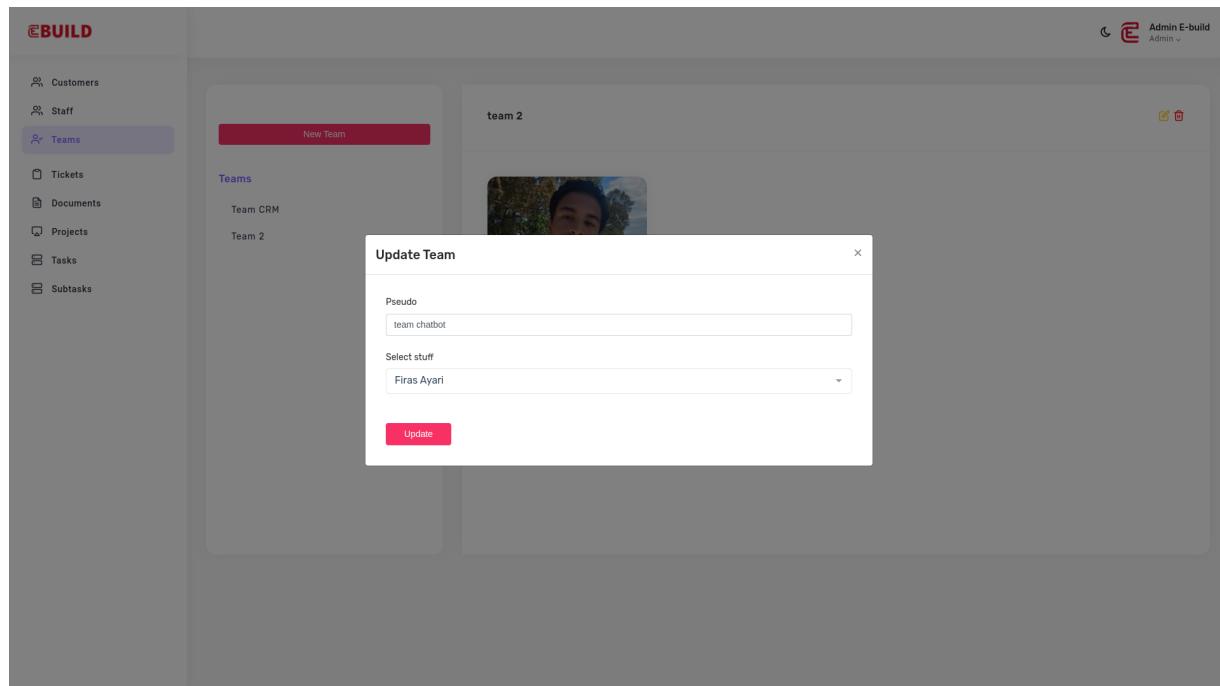


Figure 3.39: Update Team Interface

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

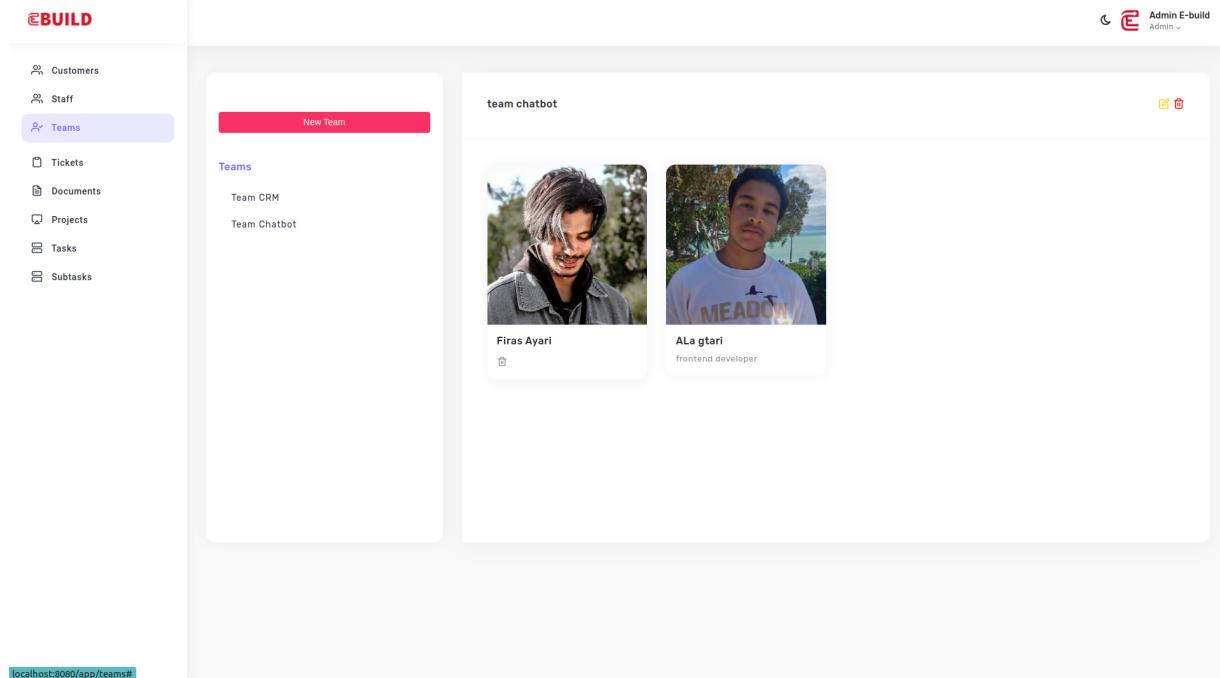


Figure 3.40: Updated Team Interface

The administrator can also delete a team or a staff member from it when they click the delete button.

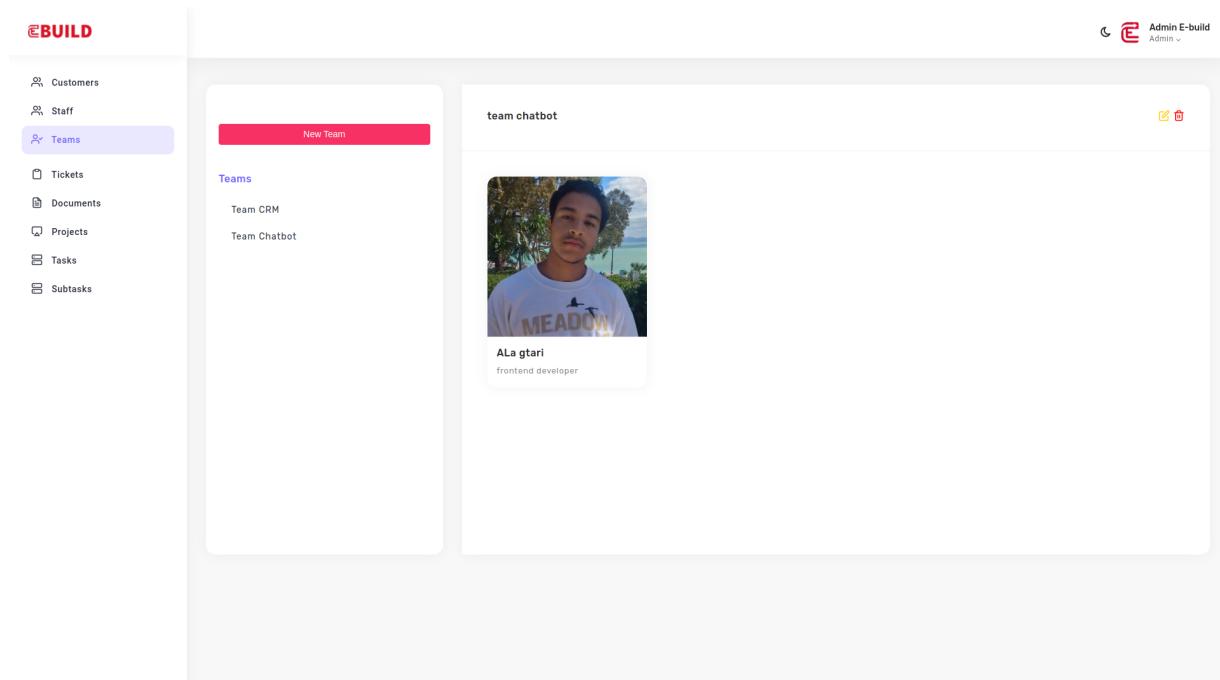


Figure 3.41: Deleted Staff Member from Team Interface

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

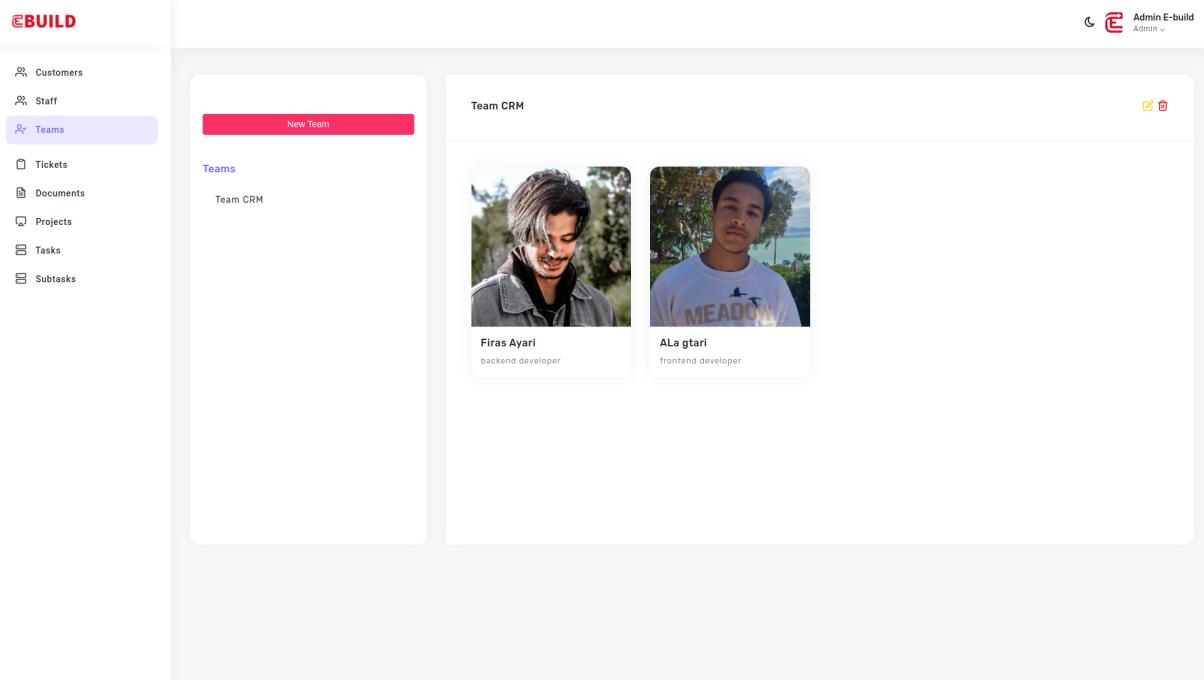


Figure 3.42: Deleted Team Interface

3.3 Sprint 1

We will be presenting the following user stories in the first sprint:

- Manage projects
- Manage tasks
- Manage subtasks
- Manage tickets

3.3.1 Identifying the Sprint1 backlog

The following table contains the backlog elements that are realised during the sprint 1:

Table 3.6: Product backlog Release 1 sprint 1

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Priority	User story	Sprint	Estimation	Release
2	As an administrator, I can manage projects	Sprint 1	High	1
2	As an administrator, I can manage tasks	Sprint 1	High	1
2	As an administrator, I can manage subtasks	Sprint 1	Medium	1
2	As a customer, I can manage tickets	Sprint 1	High	1

3.3.2 Refinement of sprint 1

In this section, we analyze the different use-case scenarios of the second sprint.

Refinement of the administrator story “manage projects”

The following figure showcases the use case .

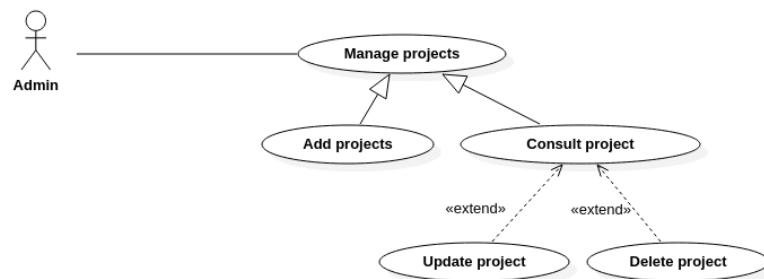


Figure 3.43: use case diagram “manage projects”

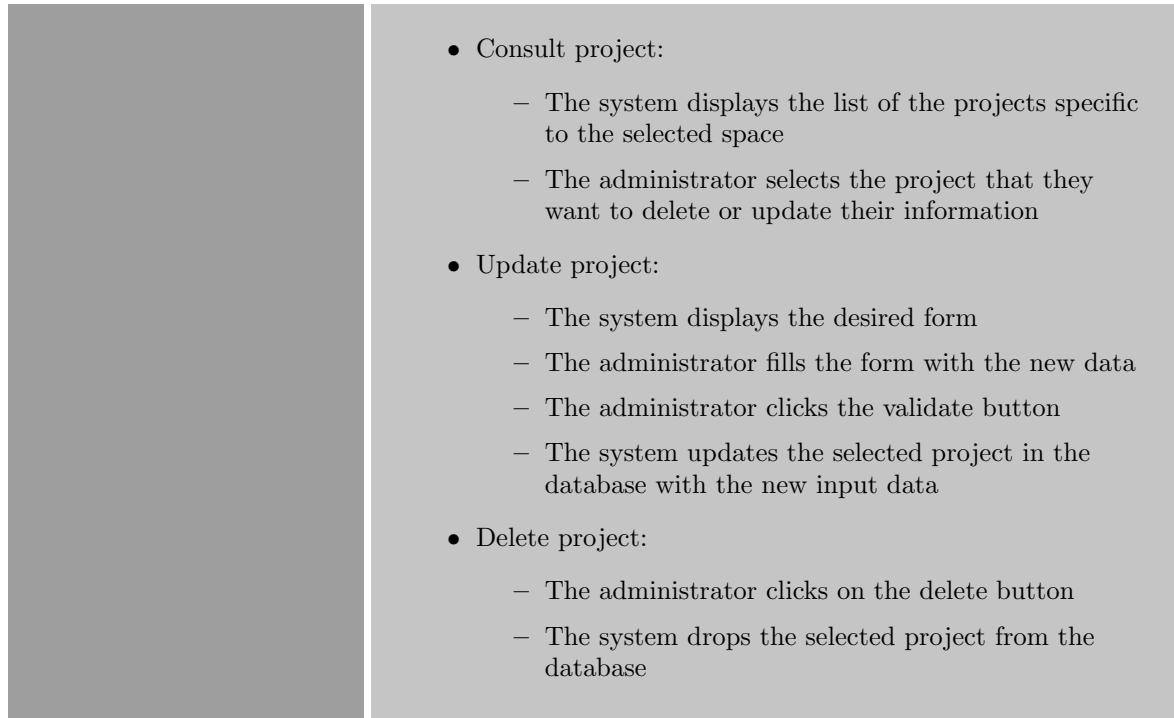
The following table elaborates on this user story with textual description.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Table 3.7: Detailed description of the "manage projects" use case

Use Case Scenario	As an administrator, I can manage projects
Actors	Administrator
Pre-Conditions	The administrator must be connected
Post-Conditions	projects managed
Extensions	<ul style="list-style-type: none"> • Add • Consult • Update • Delete
Main Scenario	<ul style="list-style-type: none"> • The administrator clicks on the projects space button • The system displays the list of projects
Secondary Scenario	<ul style="list-style-type: none"> • Add project: <ul style="list-style-type: none"> – The administrator clicks the add button – The system displays the desired form – The administrator types the required information about the project – The administrator clicks the validate button – The system adds a new project to the database

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT



Refinement of the administrator story “manage tasks”

The following figure showcases the use case .

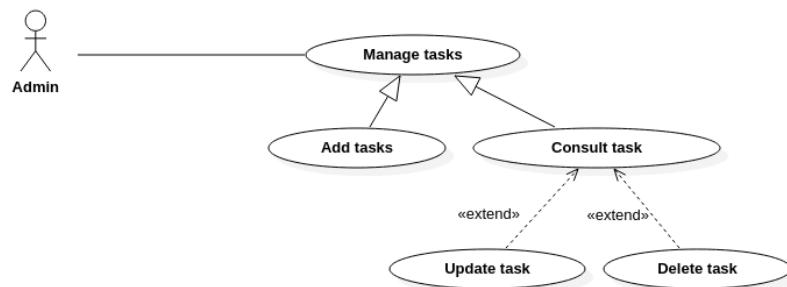


Figure 3.44: use case diagram “manage tasks”

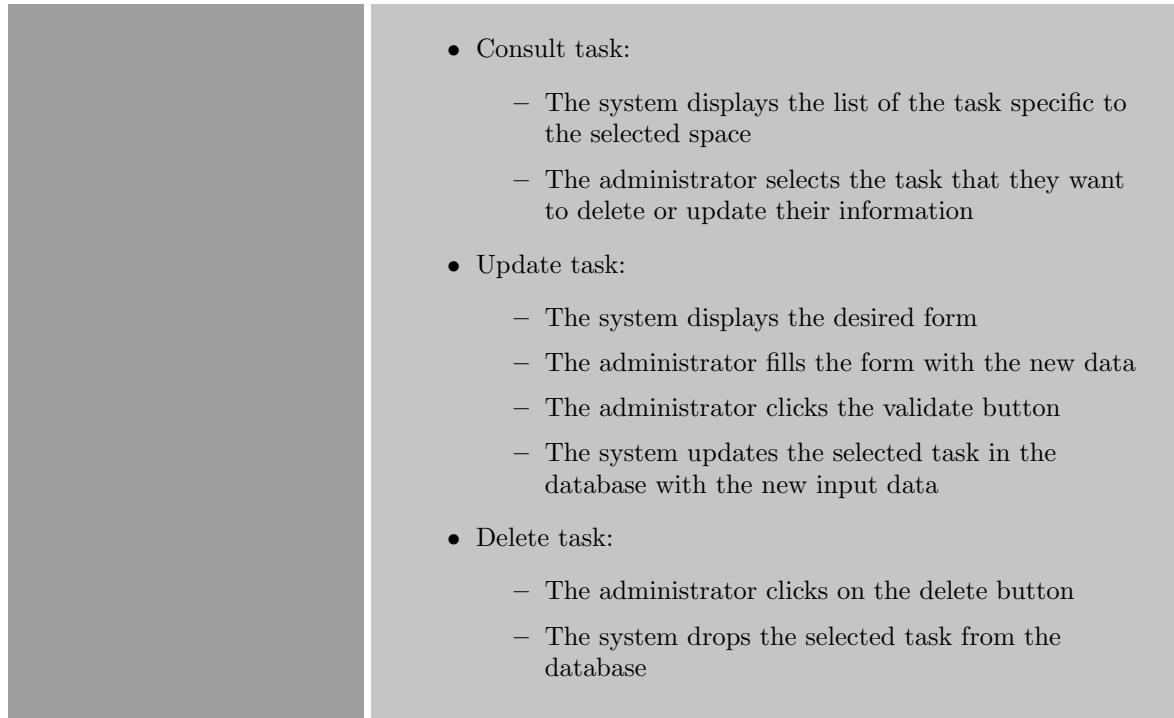
The following table elaborates on this user story with textual description.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Table 3.8: Detailed description of the "manage tasks" use case

Use Case Scenario	As an administrator, I can manage tasks
Actors	Administrator
Pre-Conditions	The administrator must be connected
Post-Conditions	tasks managed
Extensions	<ul style="list-style-type: none">• Add• Consult• Update• Delete
Main Scenario	<ul style="list-style-type: none">• The administrator clicks on the tasks space button• The system displays the list of tasks
Secondary Scenario	<ul style="list-style-type: none">• Add task:<ul style="list-style-type: none">– The administrator clicks the add button– The system displays the desired form– The administrator types the required information about the task– The administrator clicks the validate button– The system adds a new task to the database

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT



Refinement of the administrator story “manage subtasks”

The following figure showcases the use case .

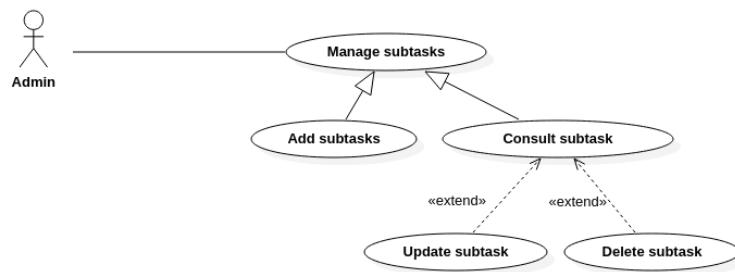


Figure 3.45: use case diagram “manage subtasks”

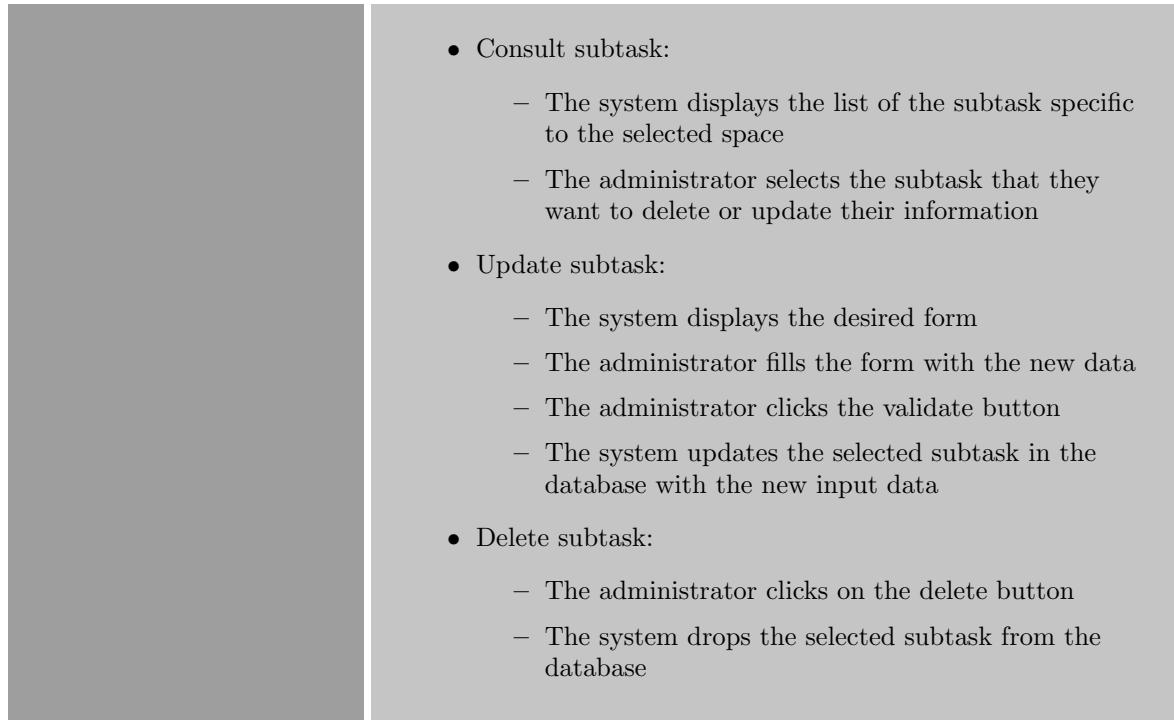
The following table elaborates on this user story with textual description.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Table 3.9: Detailed description of the "manage subtasks" use case

Use Case Scenario	As an administrator, I can manage subtasks
Actors	Administrator
Pre-Conditions	The administrator must be connected
Post-Conditions	subtasks managed
Extensions	<ul style="list-style-type: none"> • Add • Consult • Update • Delete
Main Scenario	<ul style="list-style-type: none"> • The administrator clicks on the subtasks space button • The system displays the list of subtasks
Secondary Scenario	<ul style="list-style-type: none"> • Add subtasks: <ul style="list-style-type: none"> – The administrator clicks the add button – The system displays the desired form – The administrator types the required information about the subtask – The administrator clicks the validate button – The system adds a new subtask to the database

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT



Refinement of the customer story “manage tickets”

The following figure showcases the use case .

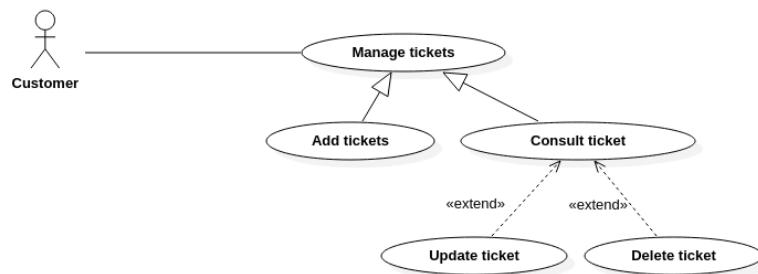


Figure 3.46: use case diagram “manage tickets”

The following table elaborates on this user story with textual description.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Table 3.10: Detailed description of the "manage tickets" use case

Use Case Scenario	As a customer, I can manage tickets
Actors	customer
Pre-Conditions	The customer must be connected
Post-Conditions	tickets managed
Extensions	<ul style="list-style-type: none">• Add• Consult• Update• Delete
Main Scenario	<ul style="list-style-type: none">• The customer clicks on the tickets' space button• The system displays the list of tickets
Secondary Scenario	<ul style="list-style-type: none">• Add ticket:<ul style="list-style-type: none">– The customer clicks the add button– The system displays the desired form– The customer types the required information about the ticket– The customer clicks the validate button– The system adds a new ticket to the database

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

- Consult ticket:
 - The system displays the list of the tickets specific to the selected space
 - The customer selects the ticket that they want to delete or update their information
- Update ticket:
 - The system displays the desired form
 - The customer fills the form with the new data
 - The customer clicks the validate button
 - The system updates the selected ticket in the database with the new input data
- Delete ticket:
 - The customer clicks on the delete button
 - The system drops the selected ticket from the database

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

3.3.3 Design of sprint 1

We will present the class and sequence diagrams of the different use-case scenarios that we implemented in the last section of their refinements.

Design of the administrator story “manage projects”

Class diagram

The following figure represents the class diagram of this use-case.

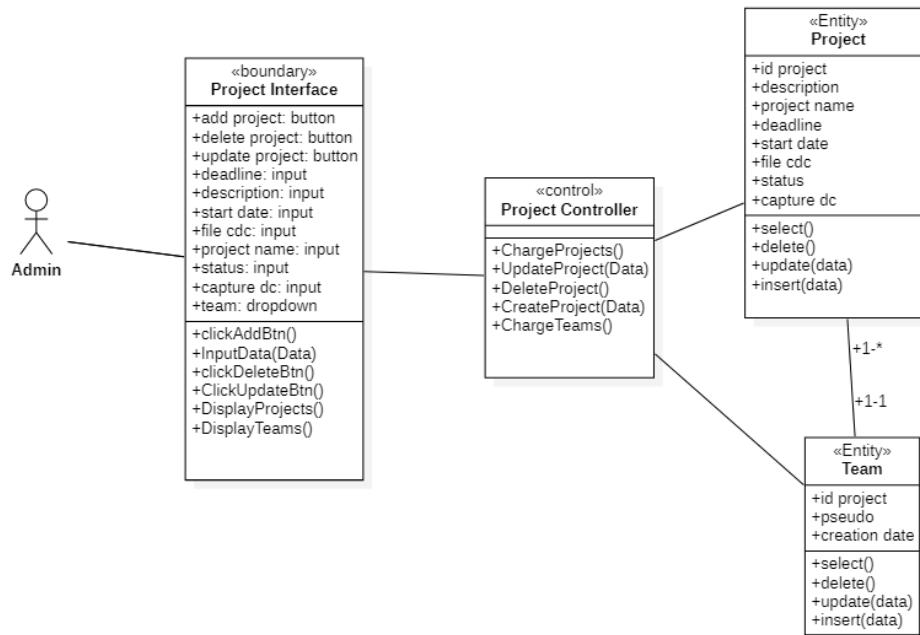


Figure 3.47: class diagram of the administrator story “manage projects”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Sequence diagram

The following figure represents the sequence diagram of this use-case.

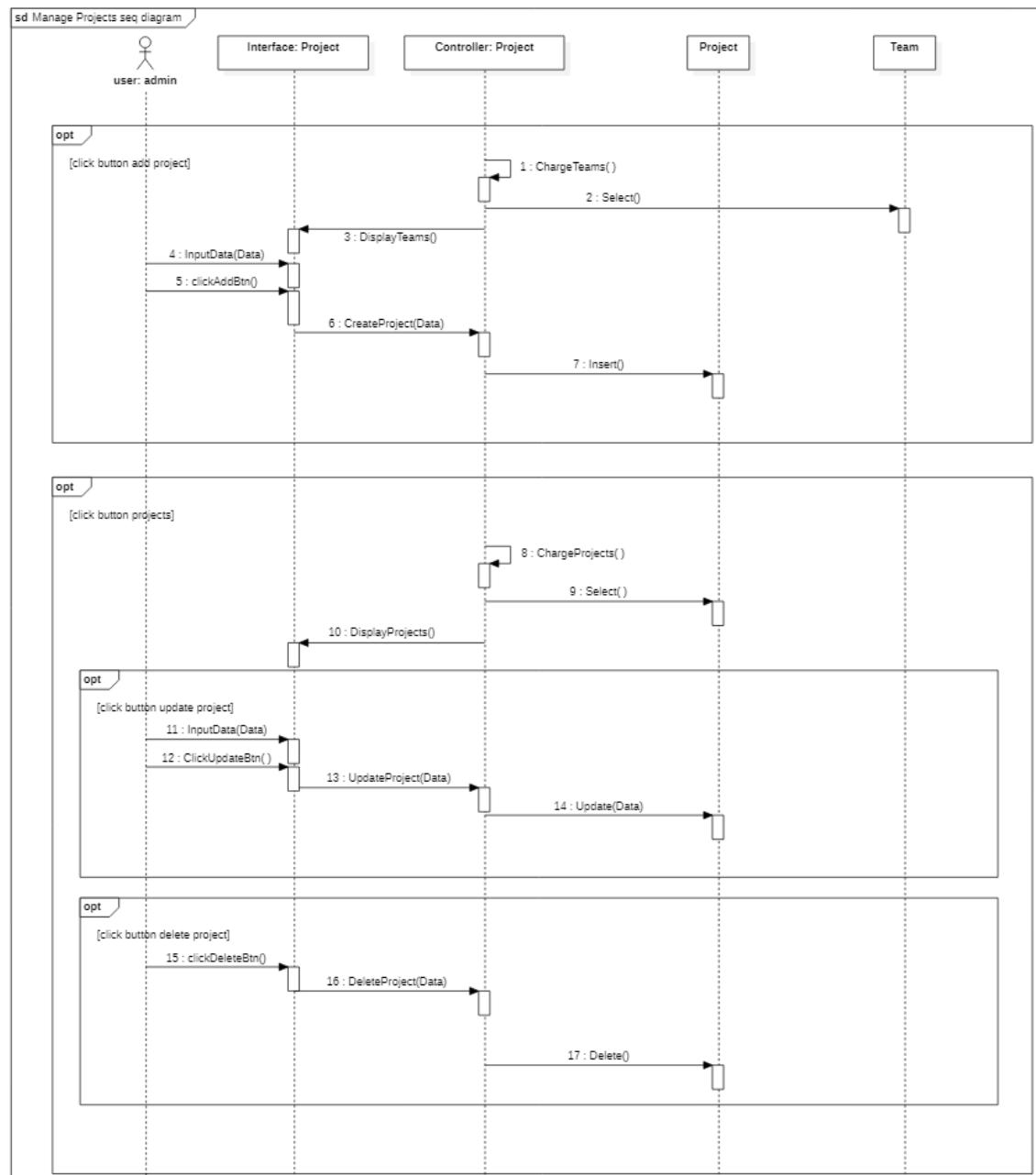


Figure 3.48: sequence diagram of the administrator story “manage projects”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Design of the customer story “manage tasks”

Class diagram

The following figure represents the class diagram of this use-case.

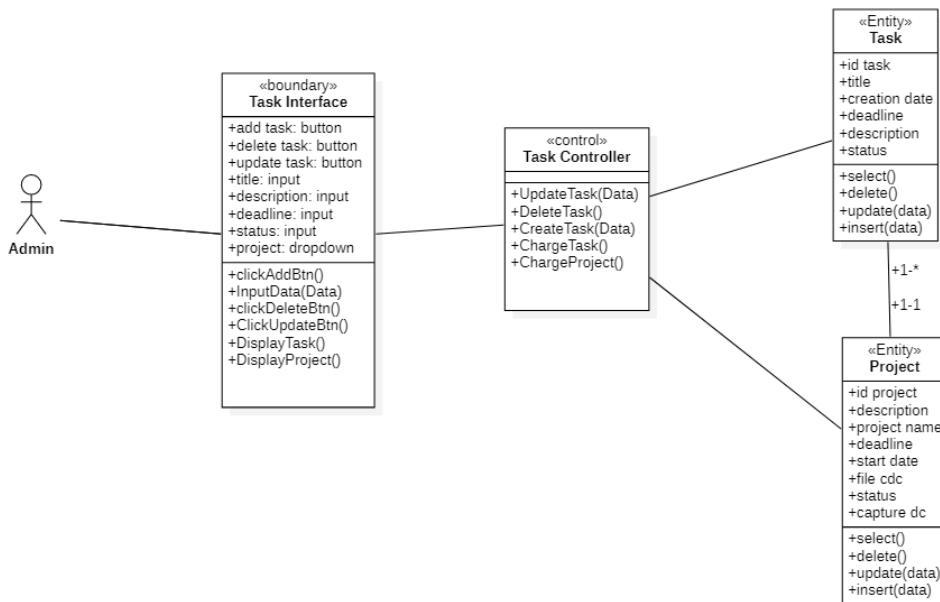


Figure 3.49: class diagram of the customer story “manage tasks”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Sequence diagram

The following figure represents the sequence diagram of this use-case.

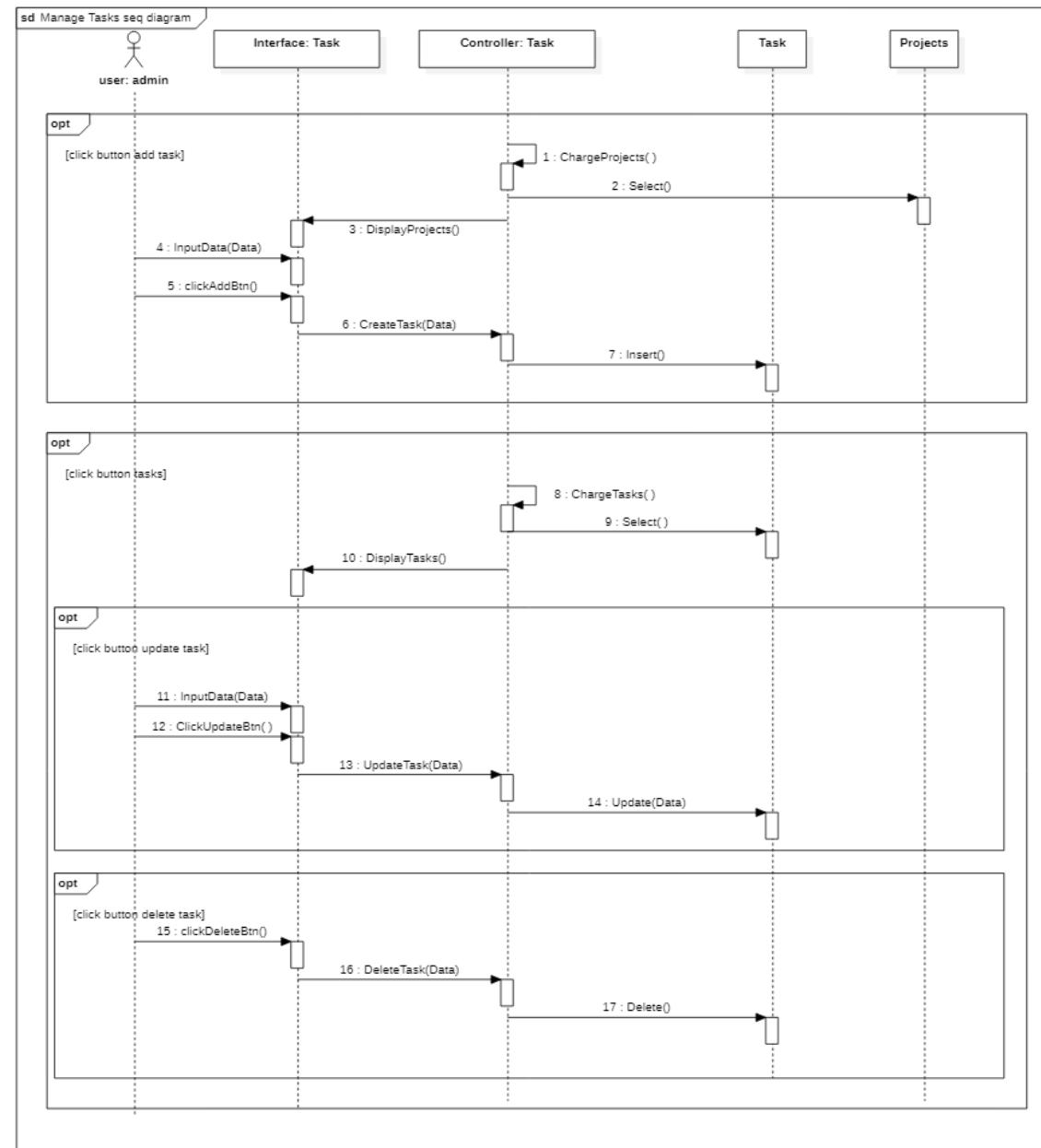


Figure 3.50: sequence diagram of the customer story “manage tasks”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Design of the administrator story “manage subtasks”

Class diagram

The following figure represents the class diagram of this use-case.

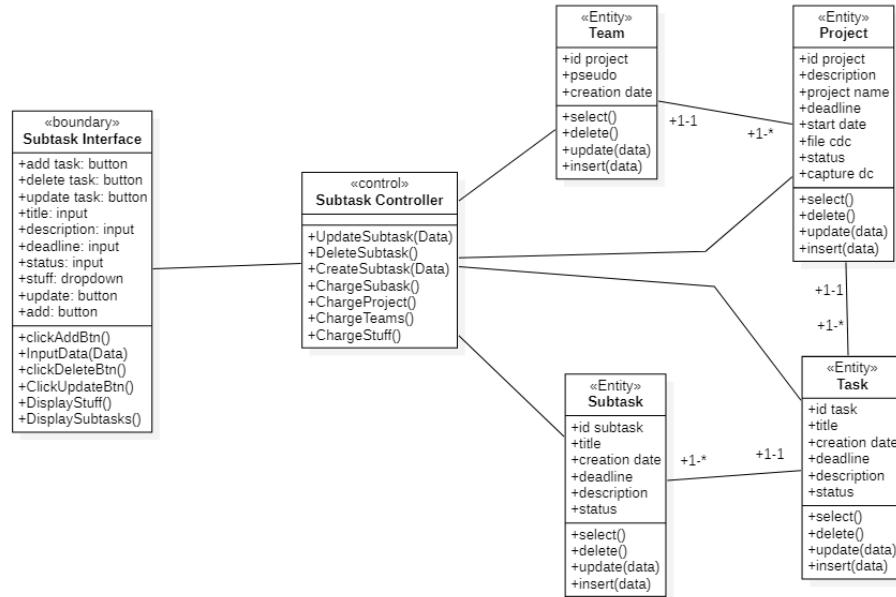


Figure 3.51: class diagram of the administrator story “manage subtasks”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Sequence diagram

The following figure represents the sequence diagram of this use-case.

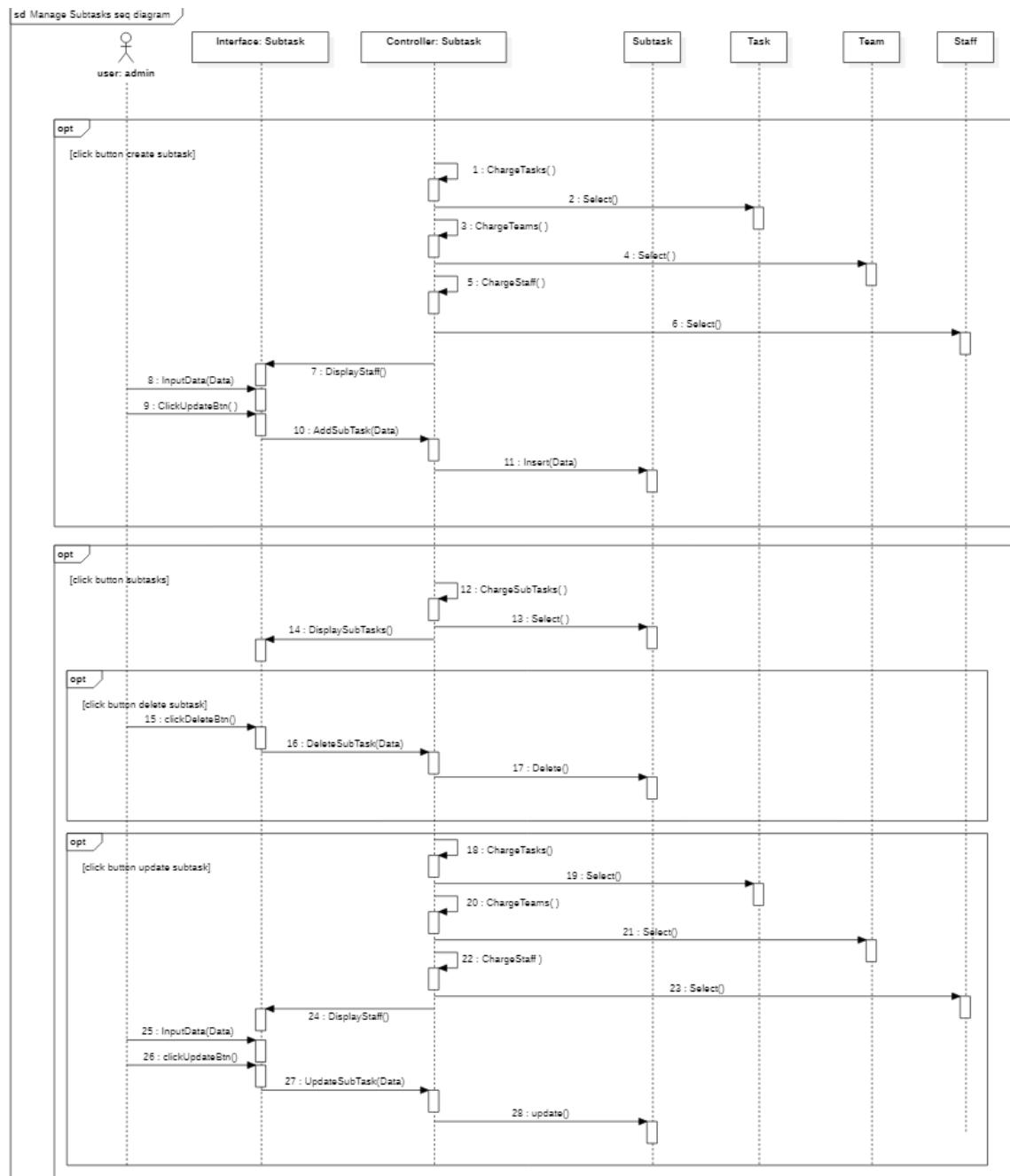


Figure 3.52: sequence diagram of the administrator story “manage subtasks”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Design of the customer story “manage tickets”

Class diagram

+The following figure represents the class diagram of this use-case.

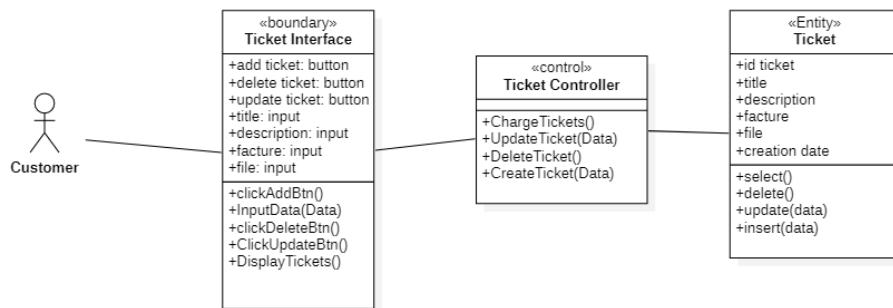


Figure 3.53: class diagram of the customer story “manage tickets”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Sequence diagram

The following figure represents the sequence diagram of this use-case.

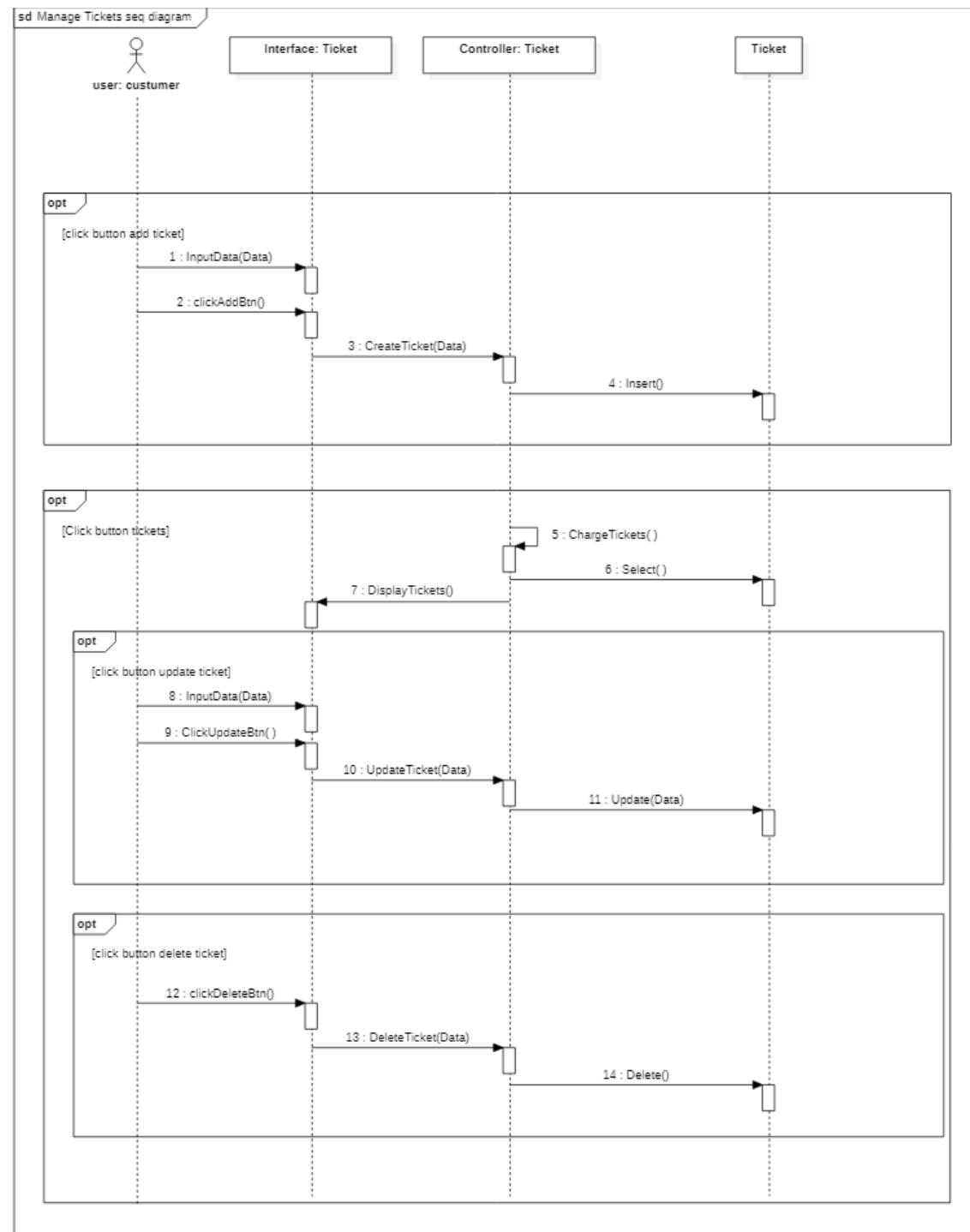


Figure 3.54: sequence diagram of the customer story “manage tickets”

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

3.4 Implementation of sprint 1

Realization of the administrator story “manage projects”

The following figure shows the manage projects interface in dashboard admin through which the admin can consult the list of the created projects, or add a new project by clicking on the create new project.

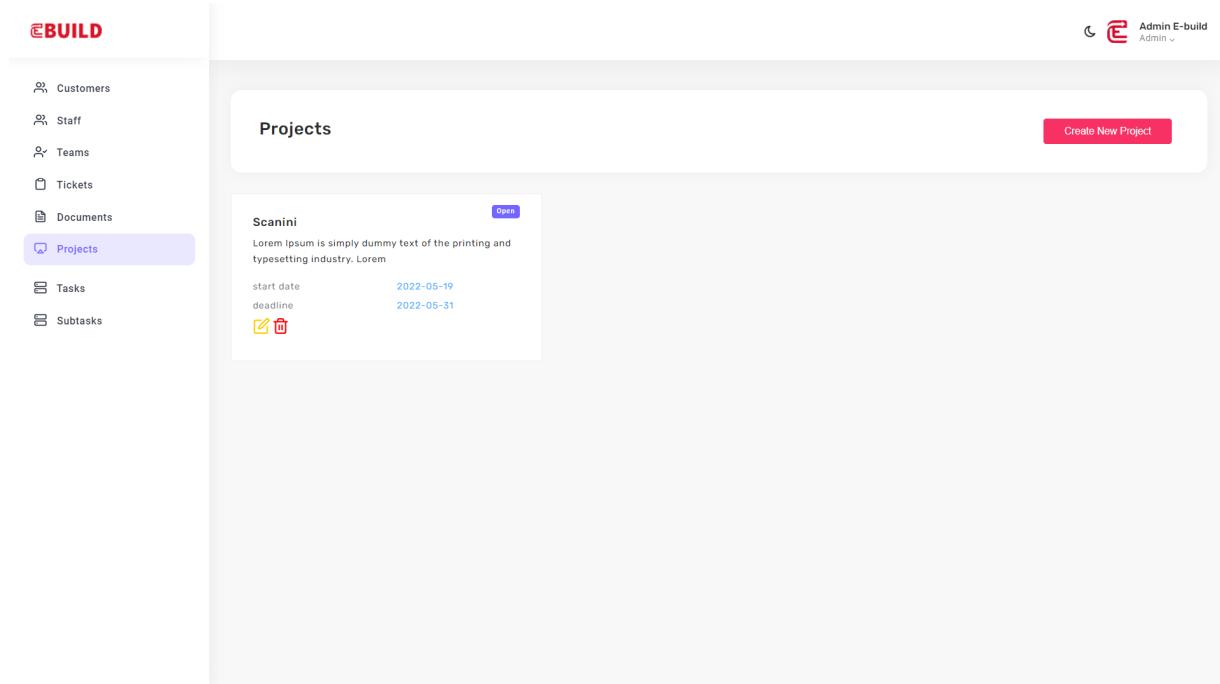


Figure 3.55: Manage Projects Interface

Upon clicking create new project button, a form pops up showing the required fields to fill. When submitting the form, a notification pops up confirming the addition of the new project and then the administrator is redirected to the projects interface with the updated list of projects.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

The screenshot shows the 'Add Project' form in the E-BUILD application. The form includes fields for Project name ('tucal'), Description ('Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem'), Team ('team CRM'), Start date ('Thursday, 30 June 2022') and Deadline ('Friday, 20 May 2022'), File CCC ('image.png'), Capture DC ('image.png'), and Status ('Open'). There are 'Submit' and 'Reset' buttons at the bottom.

Figure 3.56: Add Projects Interface

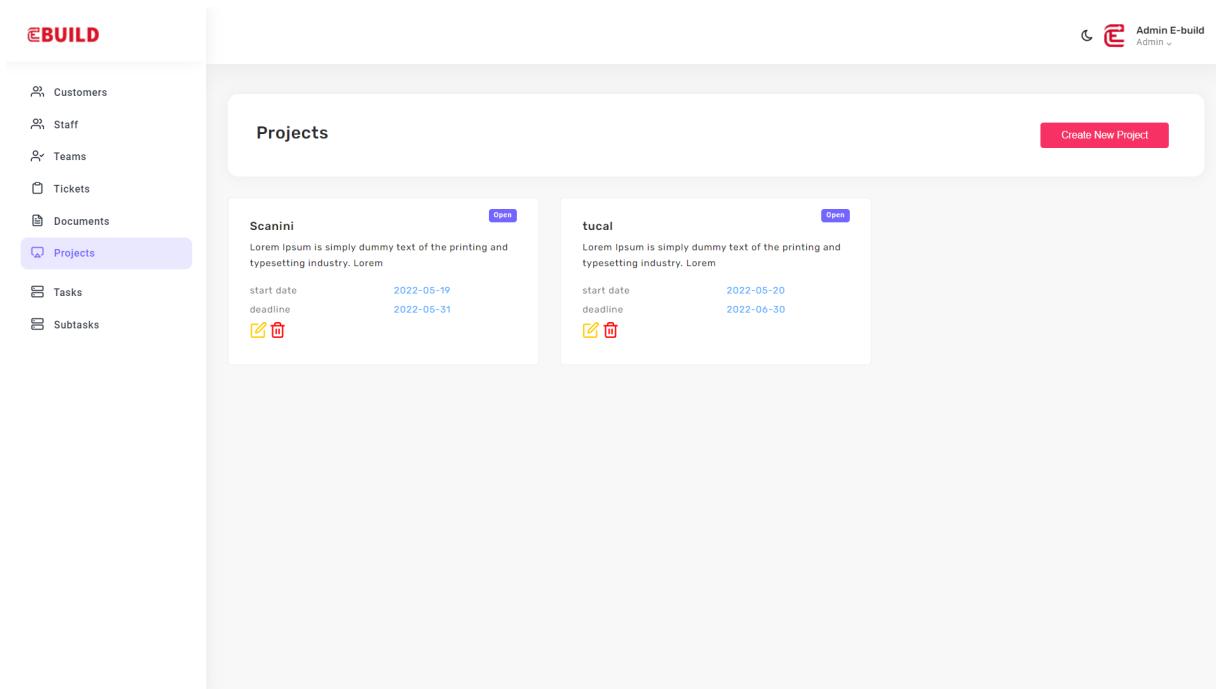


Figure 3.57: Added Project Interface

The administrator can update the consulted project by clicking the update button and altering the input of the form. The administrator can also delete a project when they click the delete button.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

Realization of the administrator story “manage tasks”

The following figure shows the manage tasks interface in dashboard admin through which the admin can consult the list of the added tasks to a selected project, or add a new task by clicking on the add task.

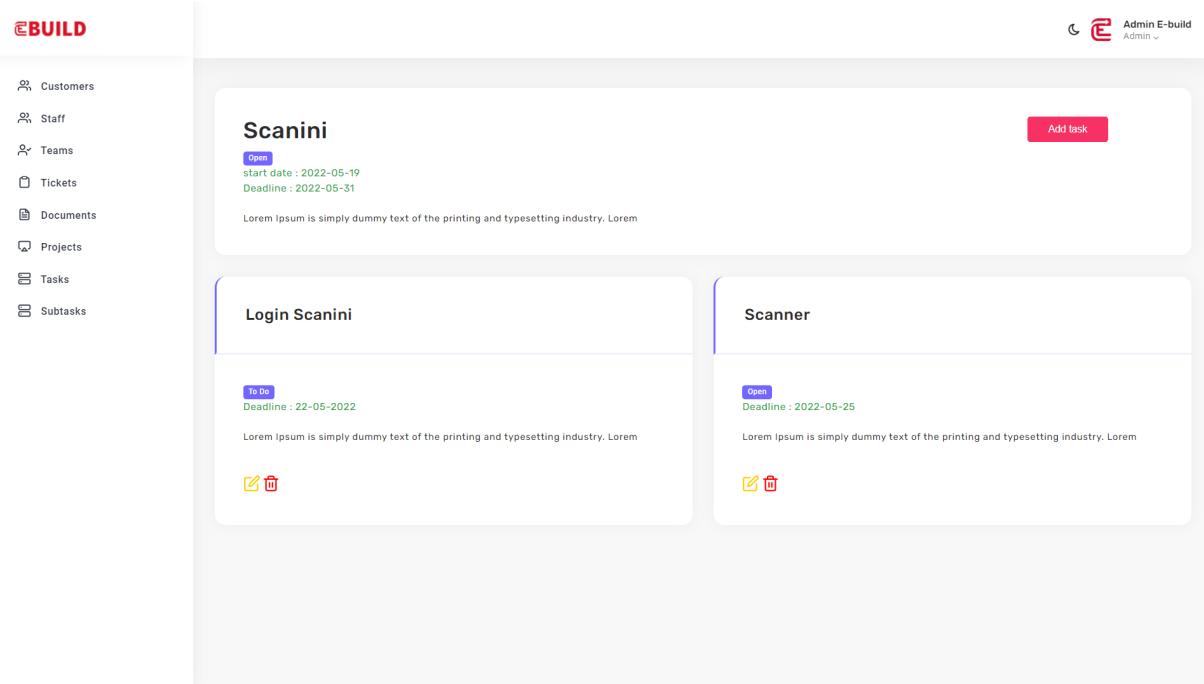


Figure 3.58: Manage Tasks Interface

Upon clicking add task button, a form pops up showing the required fields to fill. When submitting the form, a notification pops up confirming the addition of the new task and then the administrator is redirected to the projects' tasks interface with the updated list of tasks.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

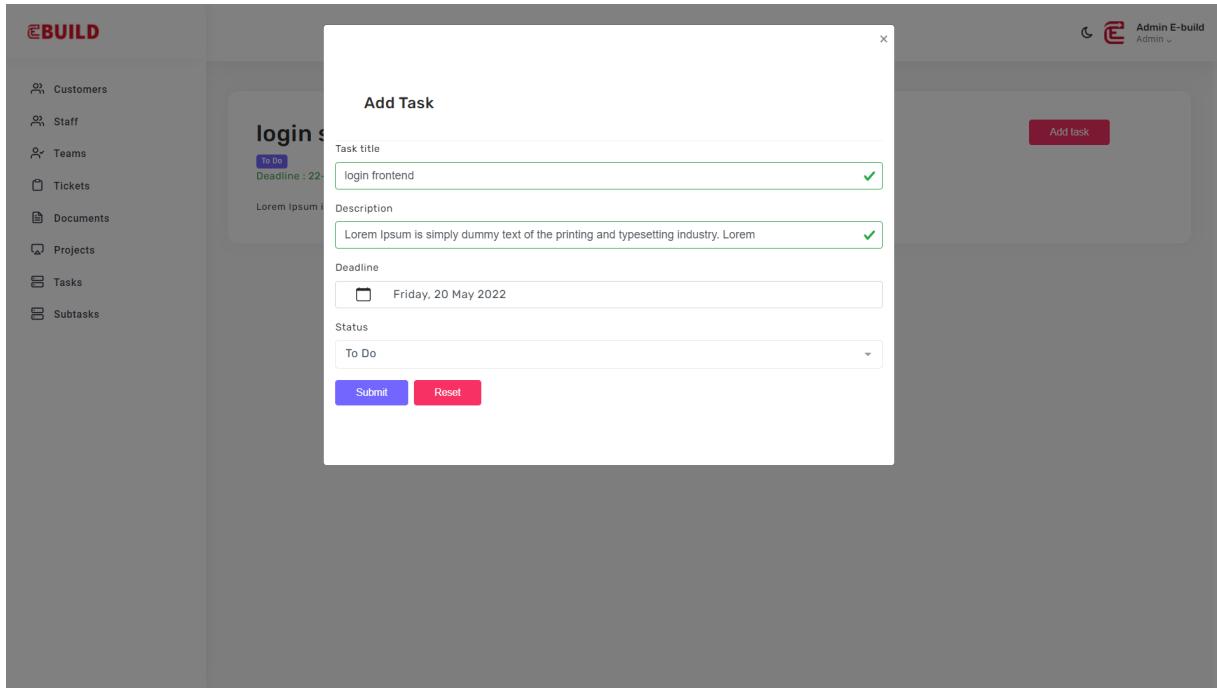


Figure 3.59: Add Task Interface

The administrator can update the consulted task by clicking the update button and altering the input of the form. The administrator can also delete a task when they click the delete button.

Realization of the administrator story “manage subtasks”

The following figure shows the manage subtasks interface in dashboard admin through which the admin can consult the list of the added subtasks to a selected task, or add a new subtask by clicking on the add subtask. Upon clicking add subtask button, a form pops up showing the required fields to fill. When submitting the form, a notification pops up confirming the addition of the new subtask and then the administrator is redirected to the projects’ tasks interface with the updated list of subtasks.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

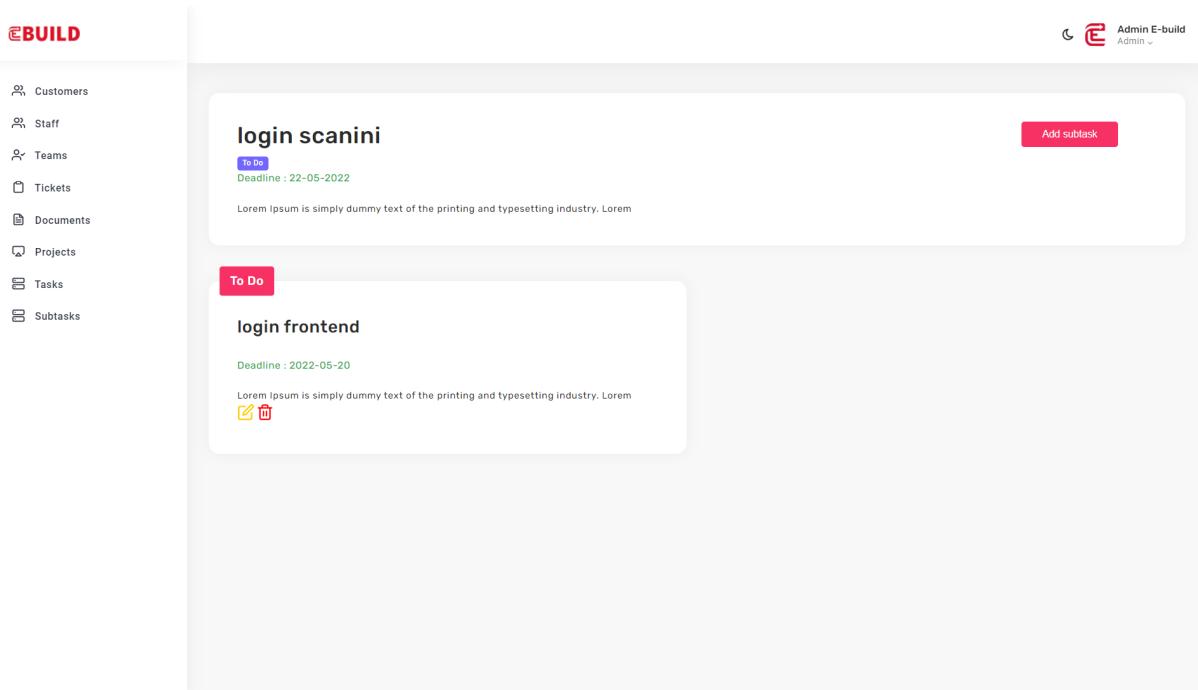


Figure 3.60: Added Subtask Interface

The administrator can update the consulted subtask by clicking the update button and altering the input of the form. The administrator can also delete a subtask when they click the delete button.

Realization of the customer story “manage tickets”

The following figure shows the manage tickets interface in dashboard customer through which the customer can consult the list of the created Tickets, or add a new ticket by clicking on the add ticket.

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

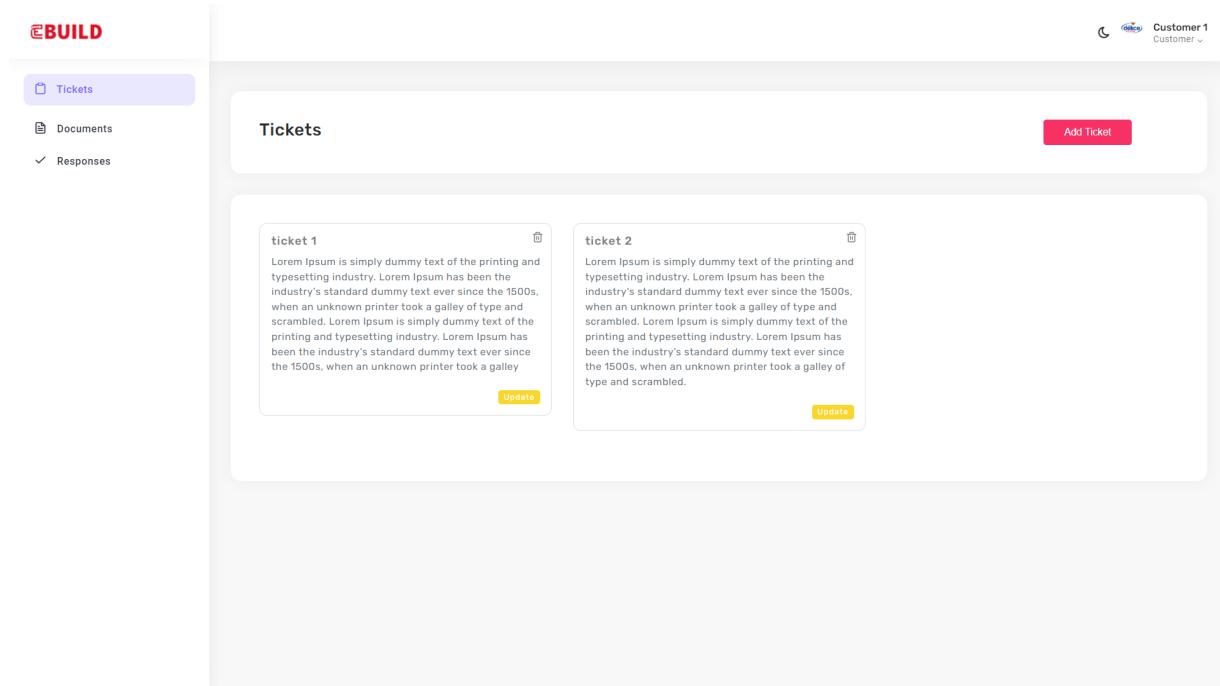


Figure 3.61: Manage Tickets Interface

Upon clicking add ticket button, a form pops up showing the required fields to fill. When submitting the form, a notification pops up confirming the addition of the new ticket and then the customer is redirected to the projects interface with the updated list of projects.

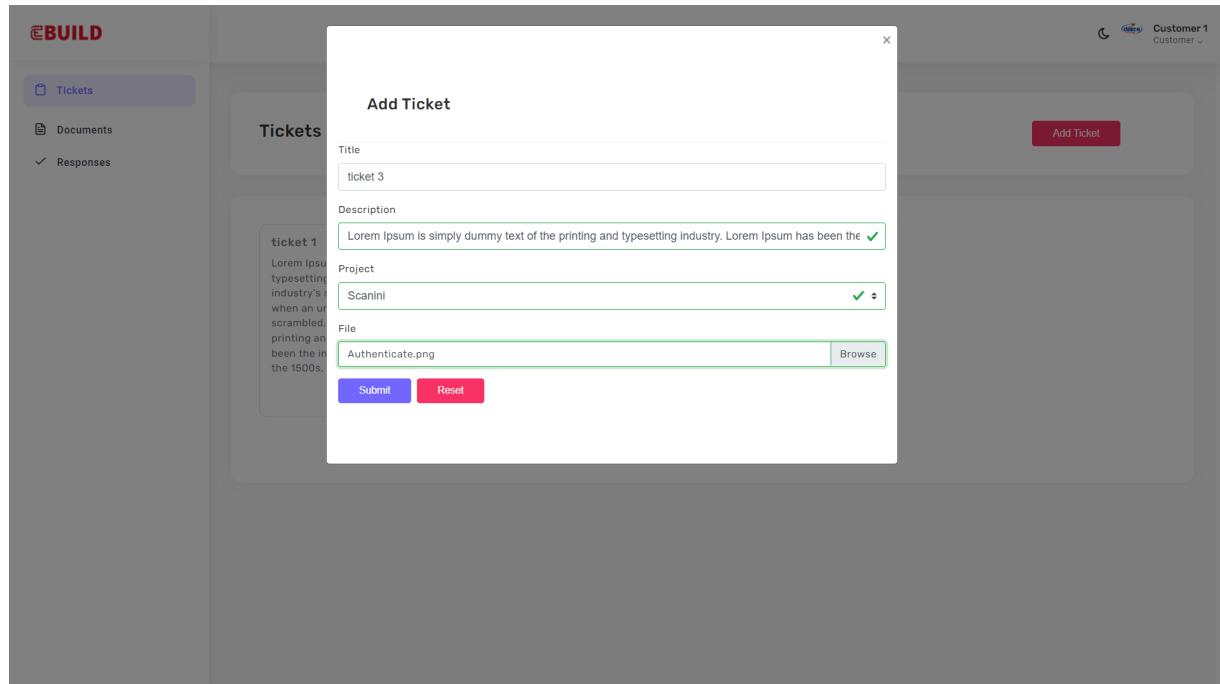


Figure 3.62: Add Ticket Interface

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

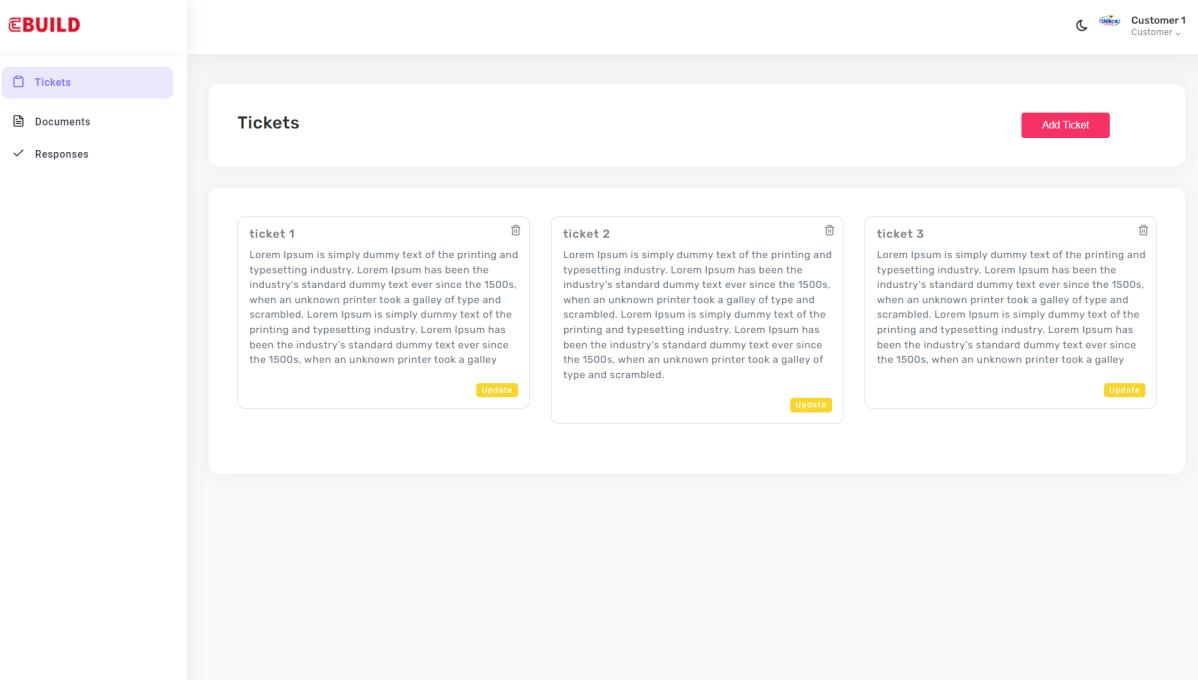


Figure 3.63: Added Ticket Interface

The customer can update the consulted project by clicking the update button and altering the input of the form.

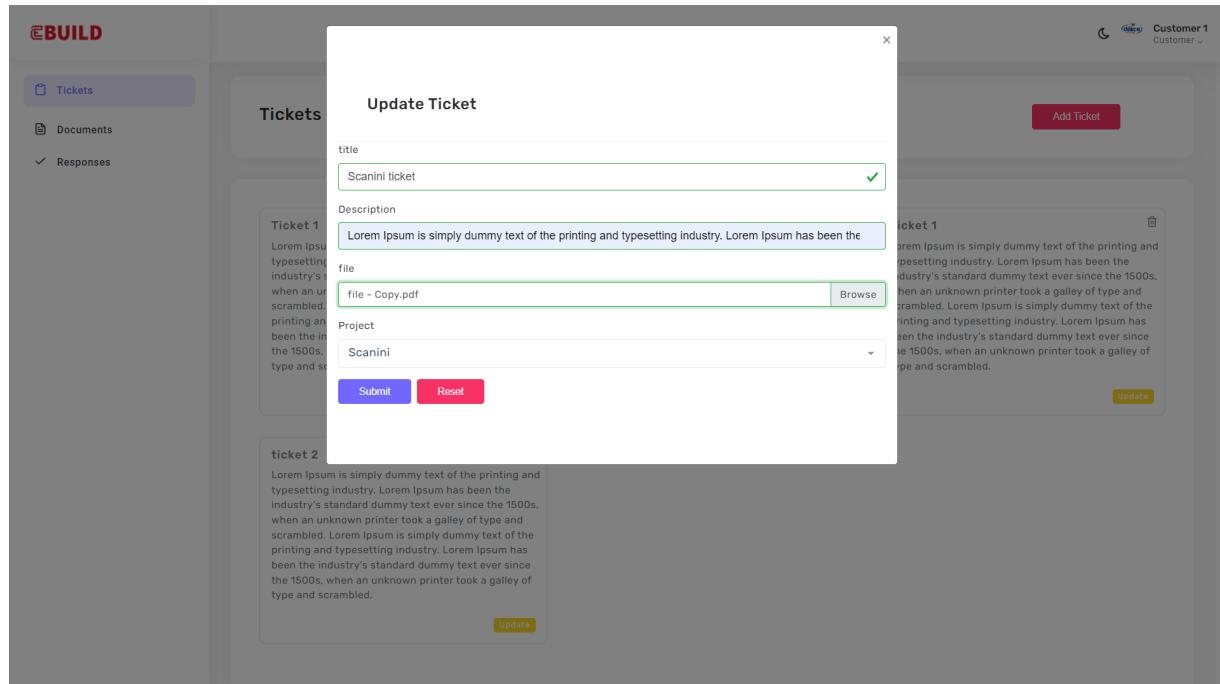


Figure 3.64: Update Ticket Interface

CHAPTER 3. RELEASE 1 : IMPLEMENTATION OF THE PROJECT MANAGEMENT

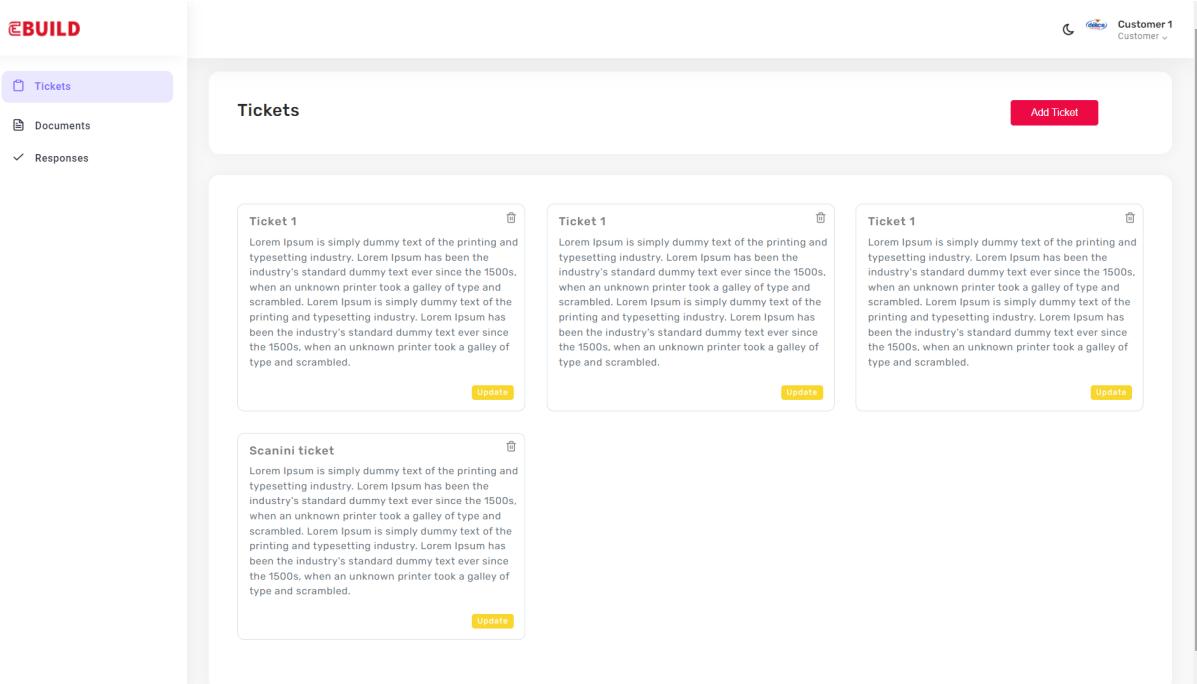


Figure 3.65: Updated Ticket Interface

The administrator can also delete a ticket when they click the delete button.

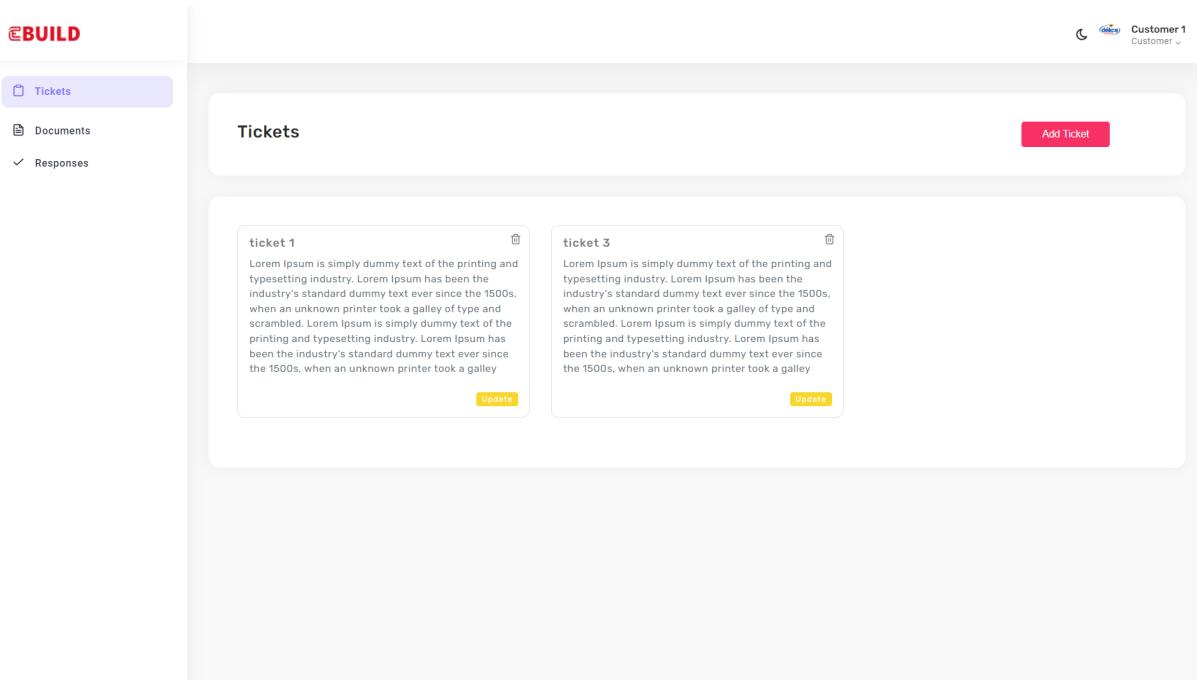


Figure 3.66: Deleted Ticket Interface

3.5 Conclusion

All along this chapter, we have gone through the implementation of the first part of our project. Release 1 is about the collection of functions that allows the admin to supervise users and initialize projects while inspecting the teams and generating the required documents. This first step of development combines the first two sprints. Pursuing Scrum methodology, we started with refining every use case showcasing the admin privileges over the other staff members. Then we elaborated further on the description by visualizing their sequence and class diagrams. Eventually, we have included some screenshots of the developed implementation based on the analysis and conception for this release.

Chapter 4

Release 2 : Implementation Of The Project Follow-Up

4.1 Introduction

Now that we set up a perfect understanding of our system by analyzing its use cases, classifying them by priorities into 4 sprints of 2 releases, and identifying their actors relatively, we proceed to the next chapter and the first phase of development. We will be going through combined uses cases of our system, which have the highest priority, by explaining their backlogs, elaborating their user stories with the refinements, then eventually showcasing them by implementing the class and sequence diagrams in addition to the final reveal of the developed interfaces.

4.2 Sprint 2

We will be presenting the following user stories in the first sprint:

- Consult tickets
- Consult projects
- Consult tasks
- Consult subtasks

4.2.1 Identifying the release2 backlog

The following table contains the backlog elements that are realised during the sprint 2:

Table 4.1: Product backlog Release 2 sprint 2

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Priority	User story	Sprint	Estimation	Release
3	As a staff member, I can consult tickets	Sprint 2	Medium	2
3	As a staff member, I can consult projects	Sprint 2	Medium	2
3	As a staff member, I can consult tasks	Sprint 2	Medium	2
3	As a staff member, I can consult subtasks	Sprint 2	Medium	2

4.2.2 Refinement of sprint 2

Refinement of the staff story “consult tickets”

The following figure showcases the use case .



Figure 4.1: use case diagram “consult tickets”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

The following table elaborates on this user story with textual description.

Table 4.2: Detailed description of the "consult tickets" use case

Use Case Scenario	As a staff, I can consult tickets
Actors	Staff
Pre-Conditions	The staff must be authenticated
Post-Conditions	ticket consulted
Main Scenario	<ul style="list-style-type: none">• The staff clicks on the tickets space button• The system displays the list of the available tickets• The staff selects a ticket to consult• The system displays the details of the selected ticket

Refinement of the administrator story “consult projects”

The following figure showcases the use case .



Figure 4.2: use case diagram “consult projects”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

The following table elaborates on this user story with textual description.

Table 4.3: Detailed description of the "consult projects" use case

Use Case Scenario	As a staff, I can consult projects
Actors	Staff
Pre-Conditions	The staff must be authenticated
Post-Conditions	project consulted
Main Scenario	<ul style="list-style-type: none">• The staff clicks on the projects space button• The system displays the list of the available projects• The staff selects a project to consult• The system displays the details of the selected project

Refinement of the administrator story “consult tasks”

The following figure showcases the use case .



Figure 4.3: use case diagram “consult tasks”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

The following table elaborates on this user story with textual description.

Table 4.4: Detailed description of the "consult tasks" use case

Use Case Scenario	As a staff, I can consult tasks
Actors	Staff
Pre-Conditions	The staff must be authenticated
Post-Conditions	task consulted
Main Scenario	<ul style="list-style-type: none">• The staff clicks on the tasks space button• The system displays the list of the available tasks• The staff selects a task to consult• The system displays the details of the selected task

Refinement of the administrator story “consult subtasks”

The following figure showcases the use case .



Figure 4.4: use case diagram “consult subtasks”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

The following table elaborates on this user story with textual description.

Table 4.5: Detailed description of the "consult subtasks" use case

Use Case Scenario	As a staff, I can consult subtasks
Actors	Staff
Pre-Conditions	The staff must be authenticated
Post-Conditions	subtask consulted
Main Scenario	<ul style="list-style-type: none"> • The staff clicks on the subtasks space button • The system displays the list of the available subtasks • The staff selects a subtask to consult • The system displays the details of the selected subtask

4.2.3 Design of sprint 2

We will present the class and sequence diagrams of the different use-case scenarios that we implemented in the last section of their refinements.

Design of the staff story “consult tickets”

Class diagram

The following figure represents the class diagram of this use-case.

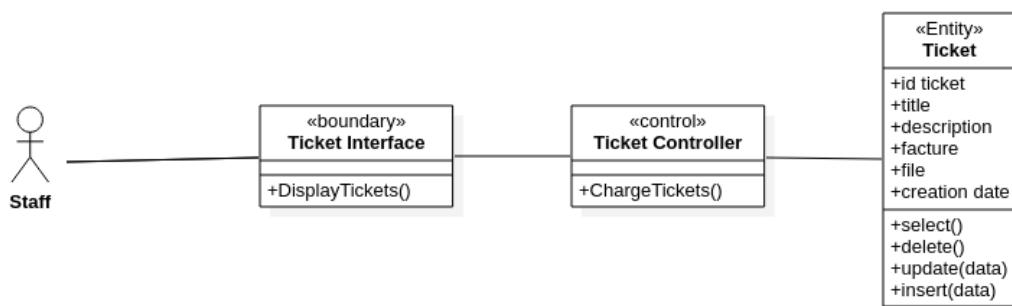


Figure 4.5: class diagram of the staff story “consult tickets”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

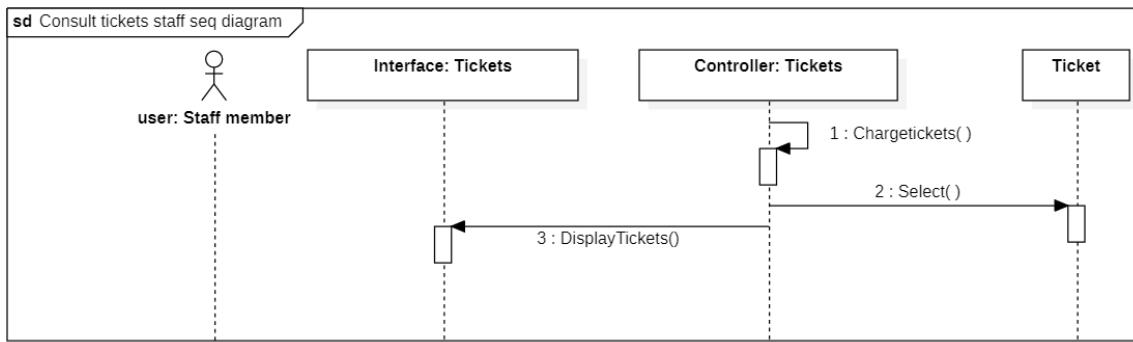


Figure 4.6: sequence diagram of the staff story “consult tickets”

Design of the staff story “consult projects”

Class diagram

The following figure represents the class diagram of this use-case.

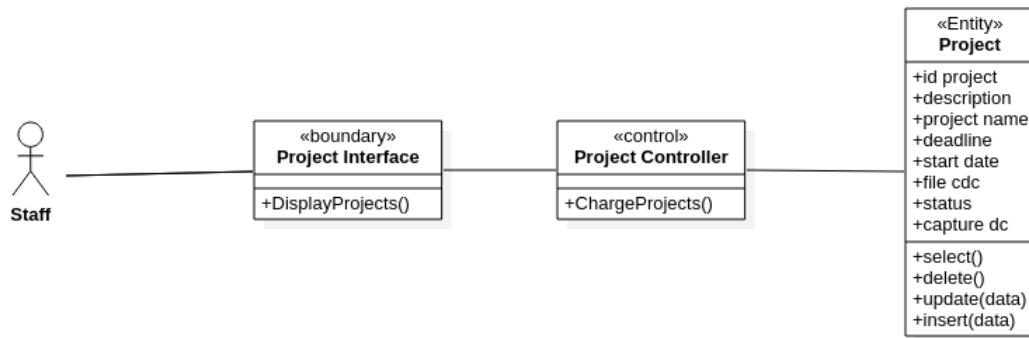


Figure 4.7: class diagram of the staff story “consult projects”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

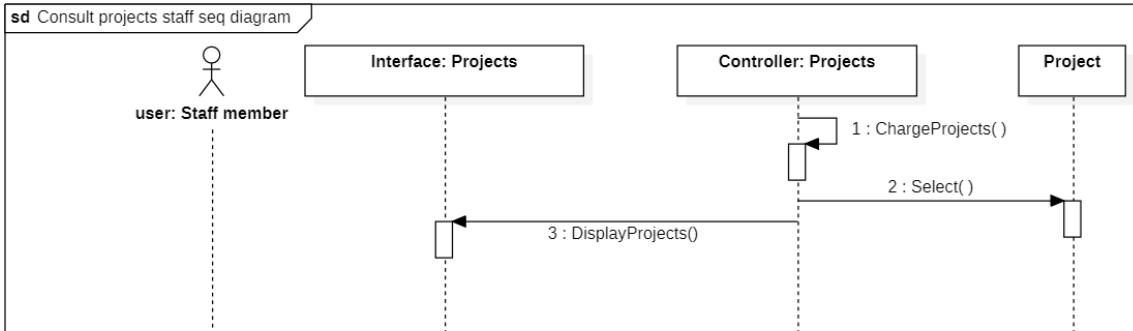


Figure 4.8: sequence diagram of the staff story “consult projects”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Design of the staff story “consult tasks”

Class diagram

The following figure represents the class diagram of this use-case.

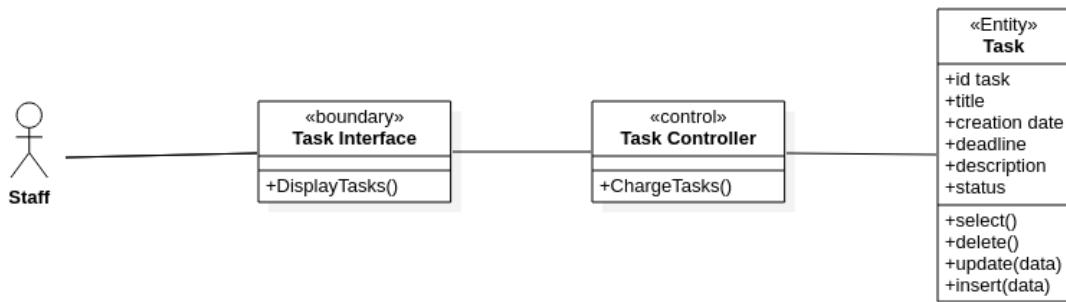


Figure 4.9: class diagram of the staff story “consult tasks”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

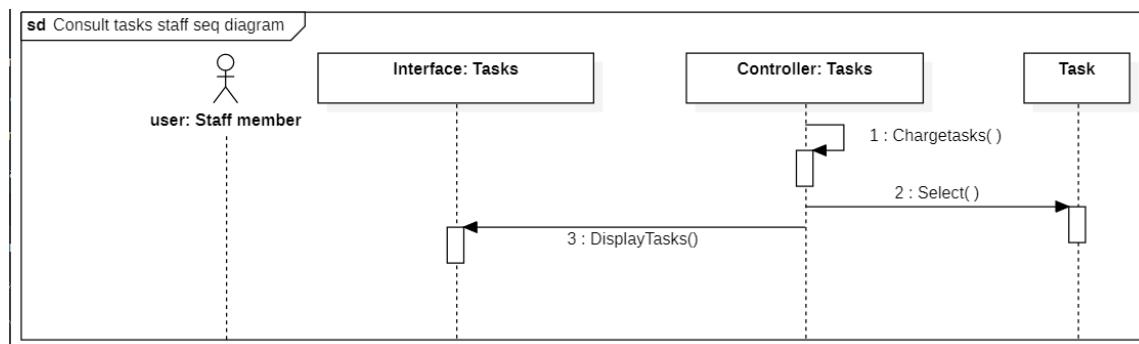
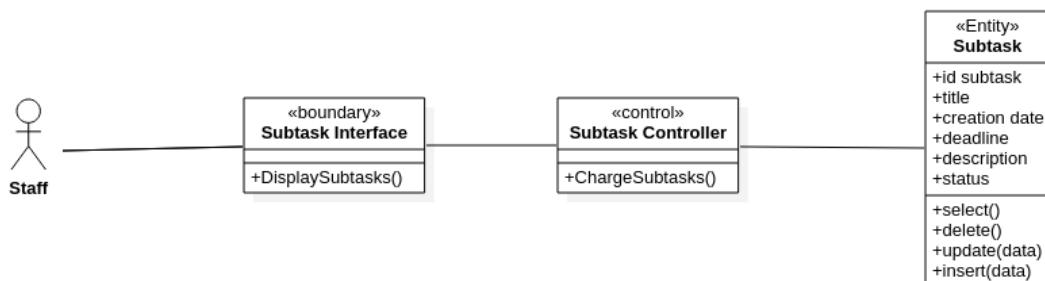


Figure 4.10: sequence diagram of the staff story “consult tasks”

Design of the staff story “consult subtasks”

Class diagram

The following figure represents the class diagram of this use-case.



CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Figure 4.11: class diagram of the staff story “consult subtasks”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

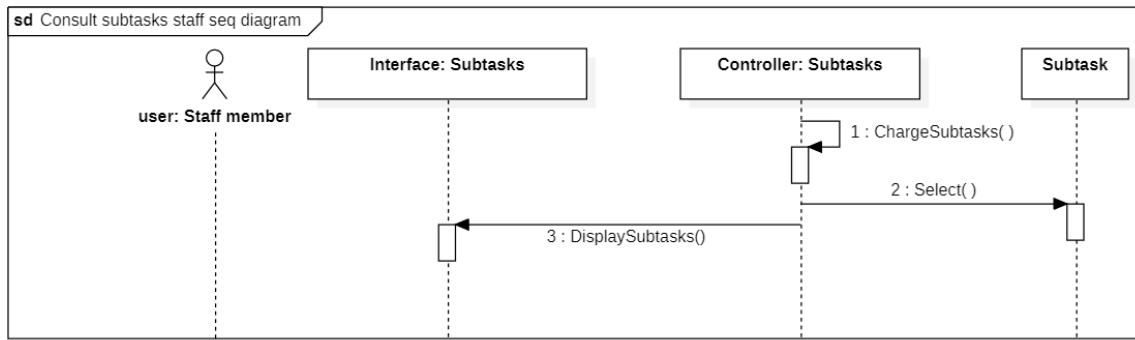


Figure 4.12: sequence diagram of the staff story “consult subtasks”

4.3 Implementation of sprint 2

Realization of the staff story “consult tickets”

The following figure shows the consult tickets interface in dashboard staff through which the staff member can consult the list of the created tickets which are associated to the projects they are working on.

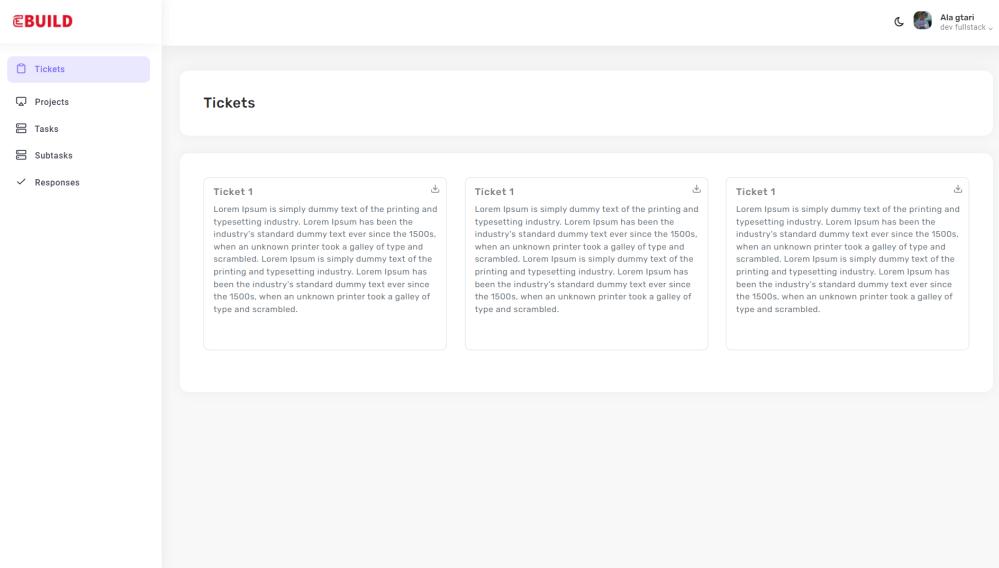


Figure 4.13: Consult Tickets Interface

Realization of the staff story “consult Projects”

The following figure shows the consult projects interface in dashboard staff through which the staff member can consult the list of the projects which are assigned to their team.

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

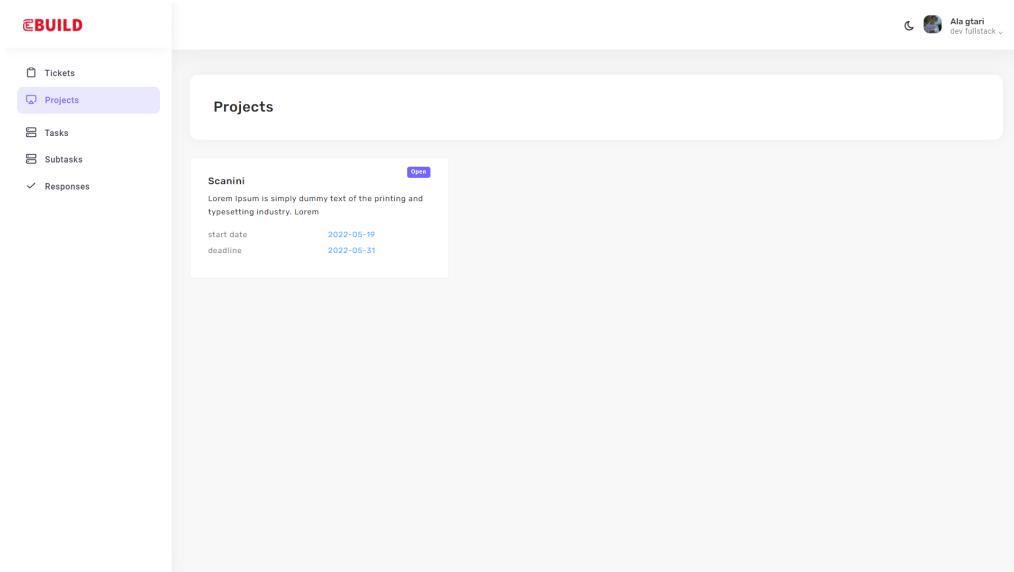


Figure 4.14: Consult Projects Interface

Realization of the staff story “consult tasks”

The following figure shows the consult tasks interface in dashboard staff through which the staff member can consult the list of the tasks which are assigned to them.

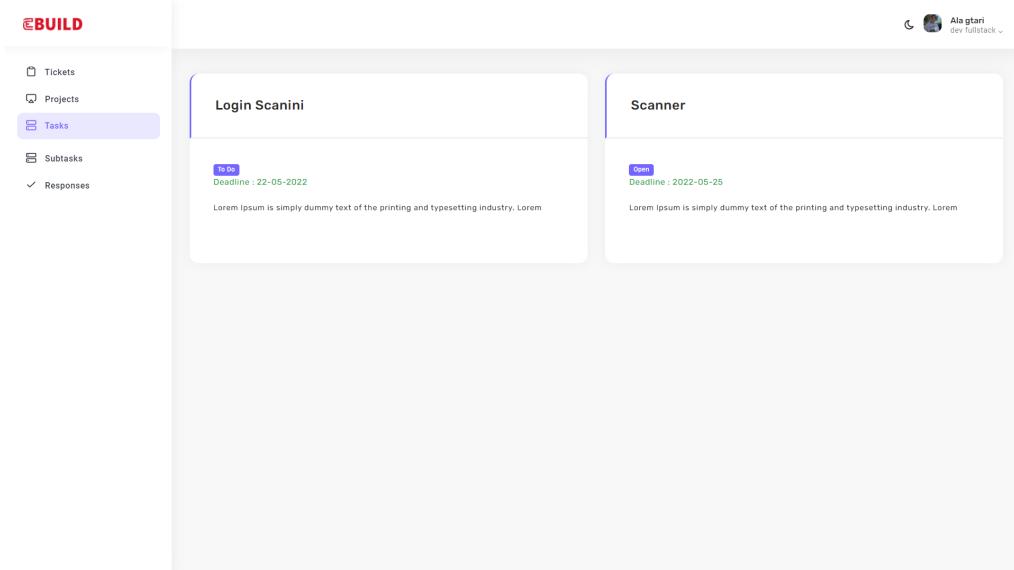


Figure 4.15: Consult Tasks Interface

Realization of the staff story “consult Subtasks”

The following figure shows the consult subtasks interface in dashboard staff through which the staff member can consult the list of the subtasks which are assigned to them.

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

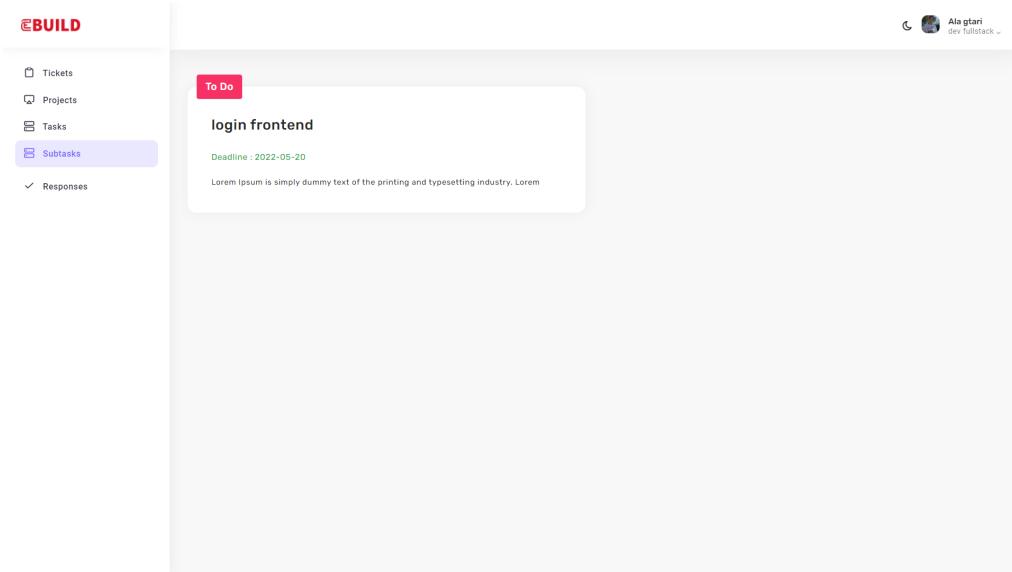


Figure 4.16: Consult Subtasks Interface

4.4 Sprint 3

We will be presenting the following user stories in the first sprint:

- Manage responses
- Manage comments
- Consult documents
- Consult responses

4.4.1 Identifying the release2 backlog

The following table contains the backlog elements that are realised during the sprint 2:

Table 4.6: Product backlog Release 2 sprint 3

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Priority	User story	Sprint	Estimation	Release
4	As a staff member, I can manage responses	Sprint 3	High	2
4	As a staff member, I can manage comments	Sprint 3	High	2
4	As a customer, I can consult documents	Sprint 3	medium	2
4	As a customer, I can consult Responses	Sprint 3	Medium	2

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

4.4.2 Refinement of sprint 2

Refinement of the staff story “manage responses”

The following figure showcases the use case .

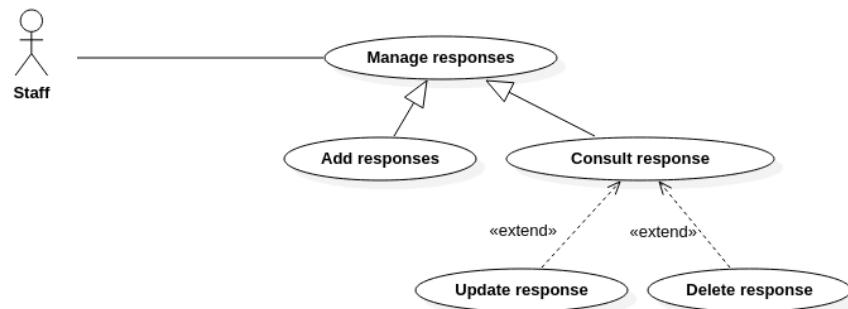


Figure 4.17: use case diagram “manage responses”

The following table elaborates on this user story with textual description.

Table 4.7: Detailed description of the ”manage responses” use case

Use Case Scenario	As a staff, I can manage responses
Actors	staff
Pre-Conditions	The staff must be connected
Post-Conditions	responses managed
Extensions	<ul style="list-style-type: none">• Add• Consult• Update• Delete

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Main Scenario	<ul style="list-style-type: none">● The staff clicks on the responses' space button● The system displays the list of responses
Secondary Scenario	<ul style="list-style-type: none">● Add response:<ul style="list-style-type: none">– The staff clicks the add button– The system displays the desired form– The staff types the required information about the response– The staff clicks the validate button– The system adds a new response to the database● Consult response:<ul style="list-style-type: none">– The system displays the list of the responses specific to the selected space– The staff selects the response that they want to delete or update their information● Update response:<ul style="list-style-type: none">– The system displays the desired form– The staff fills the form with the new data– The staff clicks the validate button– The system updates the selected response in the database with the new input data● Delete response:<ul style="list-style-type: none">– The staff clicks on the delete button– The system drops the selected response from the database

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Refinement of the staff story “manage comments”

The following figure showcases the use case .

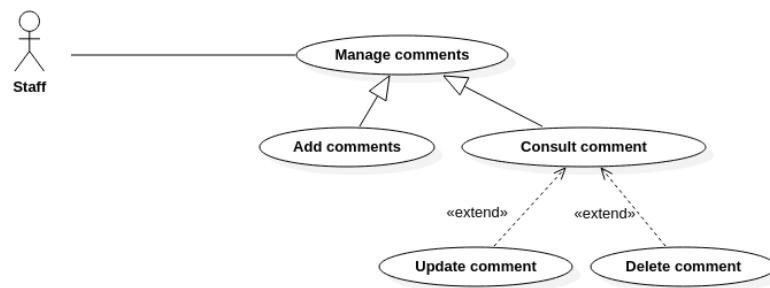


Figure 4.18: use case diagram “manage comments”

The following table elaborates on this user story with textual description.

Table 4.8: Detailed description of the ”manage comments” use case

Use Case Scenario	As a staff, I can manage comments
Actors	staff
Pre-Conditions	The staff must be connected
Post-Conditions	comments managed
Extensions	<ul style="list-style-type: none">• Add• Consult• Update• Delete

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Main Scenario	<ul style="list-style-type: none">● The staff clicks on the comments' space button● The system displays the list of comments
Secondary Scenario	<ul style="list-style-type: none">● Add comment:<ul style="list-style-type: none">– The staff clicks the add button– The system displays the desired form– The staff types the required information about the comment– The staff clicks the validate button– The system adds a new comment to the database● Consult comment:<ul style="list-style-type: none">– The system displays the list of the comments specific to the selected space– The staff selects the comment that they want to delete or update their information● Update comment:<ul style="list-style-type: none">– The system displays the desired form– The staff fills the form with the new data– The staff clicks the validate button– The system updates the selected comment in the database with the new input data● Delete comment:<ul style="list-style-type: none">– The staff clicks on the delete button– The system drops the selected comment from the database

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Refinement of the customer story “consult documents”

The following figure showcases the use case .



Figure 4.19: use case diagram “consult documents”

The following table elaborates on this user story with textual description.

Table 4.9: Detailed description of the ”consult documents” use case

Use Case Scenario	As a customer, I can consult documents
Actors	customer
Pre-Conditions	The customer must be authenticated
Post-Conditions	document consulted
Main Scenario	<ul style="list-style-type: none">• The customer clicks on the documents space button• The system displays the list of the available documents• The customer selects a document to consult• The system displays the details of the selected document

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Refinement of the customer story “consult responses”

The following figure showcases the use case .



Figure 4.20: use case diagram “consult responses”

The following table elaborates on this user story with textual description.

Table 4.10: Detailed description of the ”consult responses” use case

Use Case Scenario	As a customer, I can consult responses
Actors	customer
Pre-Conditions	The customer must be authenticated
Post-Conditions	reponse consulted
Main Scenario	<ul style="list-style-type: none">• The customer clicks on the reponses space button• The system displays the list of the available reponses• The customer selects a reponse to consult• The system displays the details of the selected reponse

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

4.4.3 Design of sprint 2

We will present the class and sequence diagrams of the different use-case scenarios that we implemented in the last section of their refinements.

Design of the staff story “manage responses”

Class diagram

The following figure represents the class diagram of this use-case.

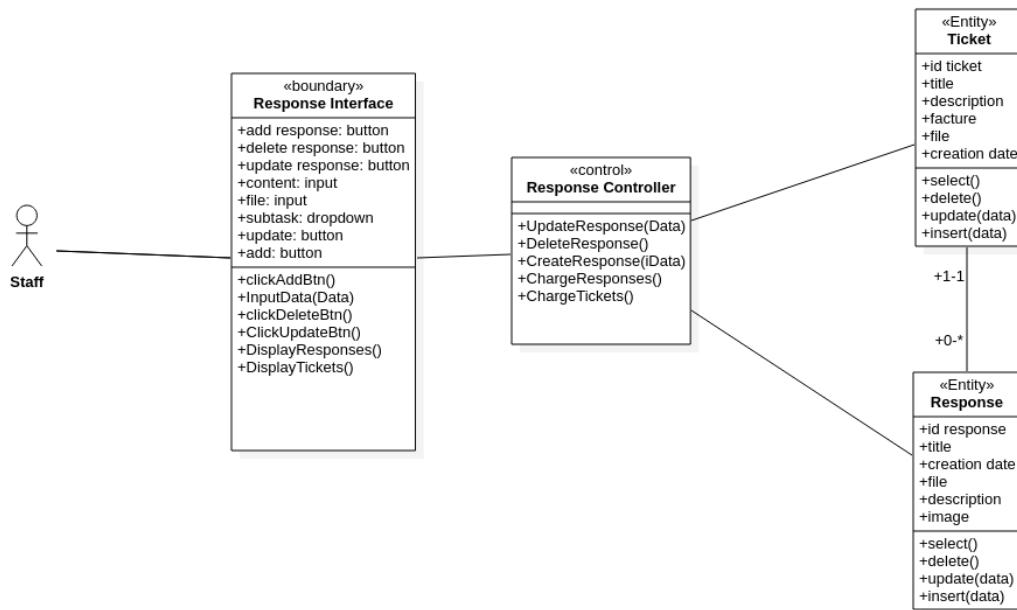


Figure 4.21: class diagram of the staff story “manage responses”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Sequence diagram

The following figure represents the sequence diagram of this use-case.

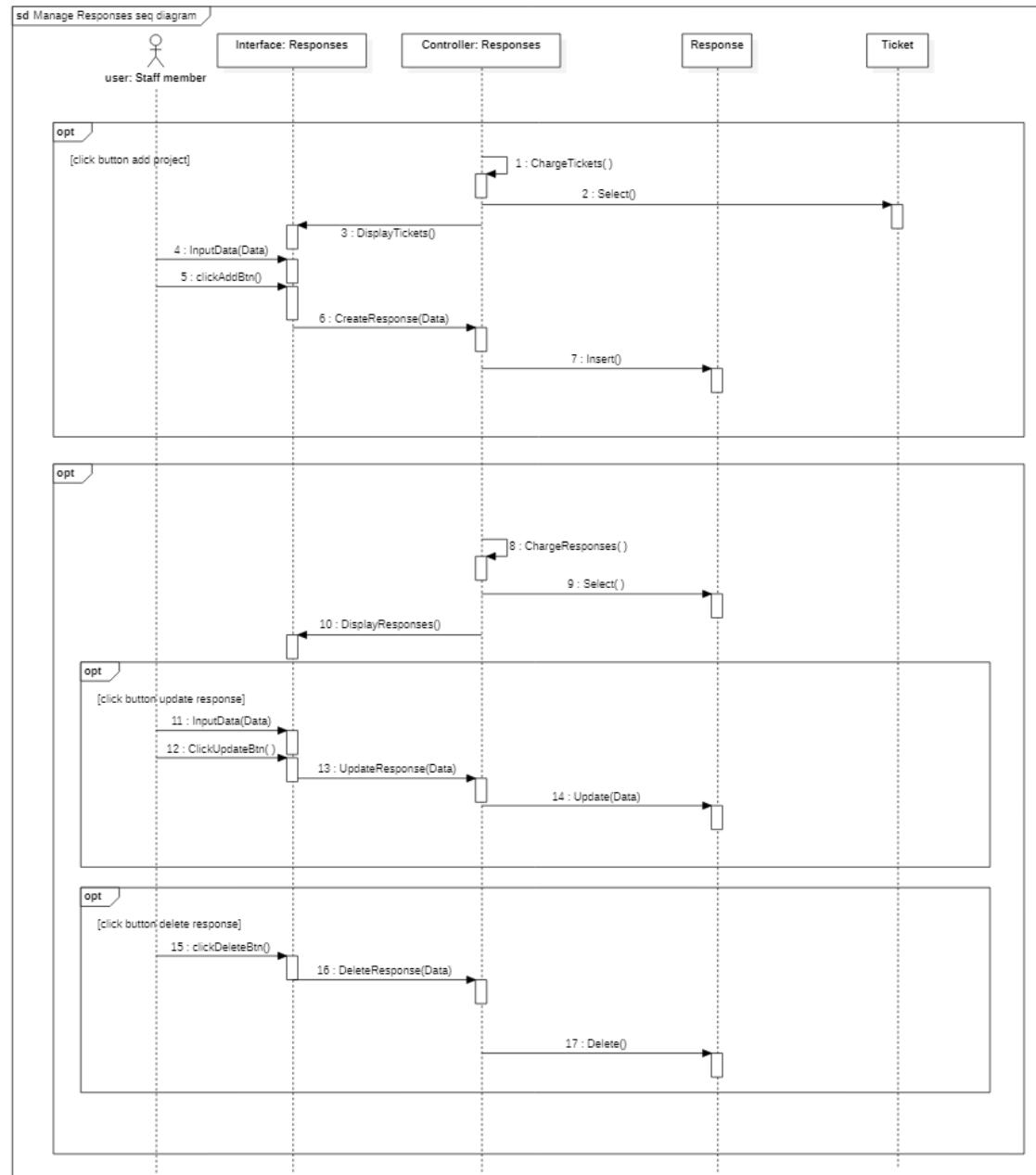


Figure 4.22: sequence diagram of the staff story “manage responses”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Design of the staff story “manage comments”

Class diagram

The following figure represents the class diagram of this use-case.

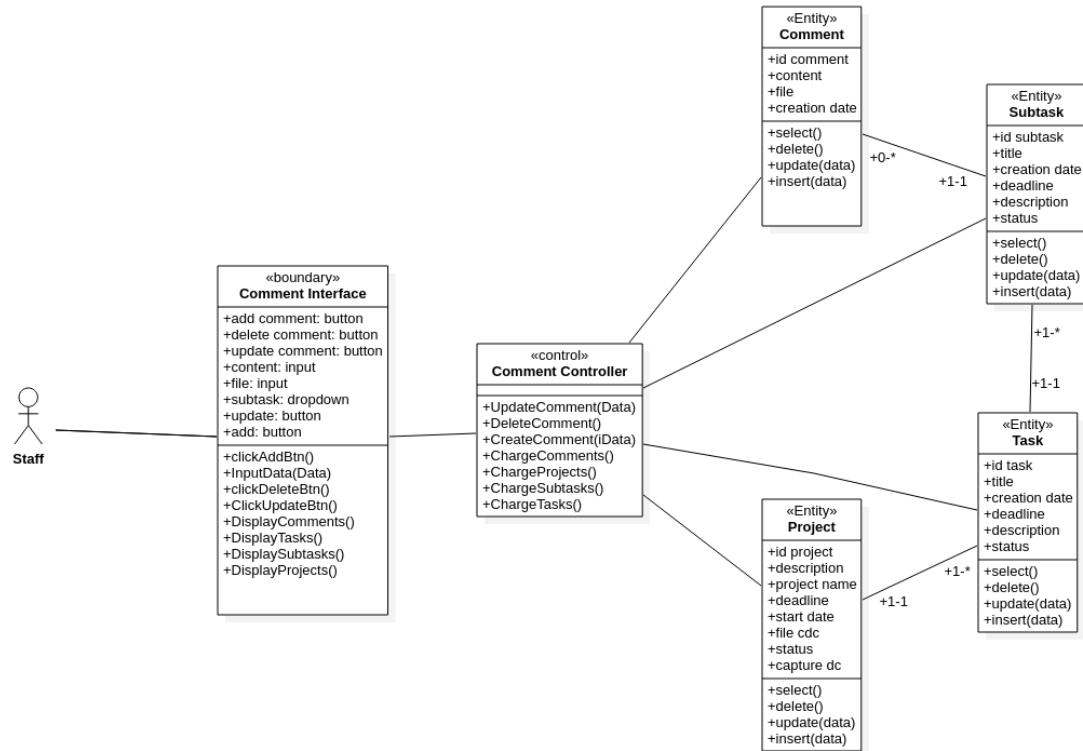


Figure 4.23: class diagram of the staff story “manage comments”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

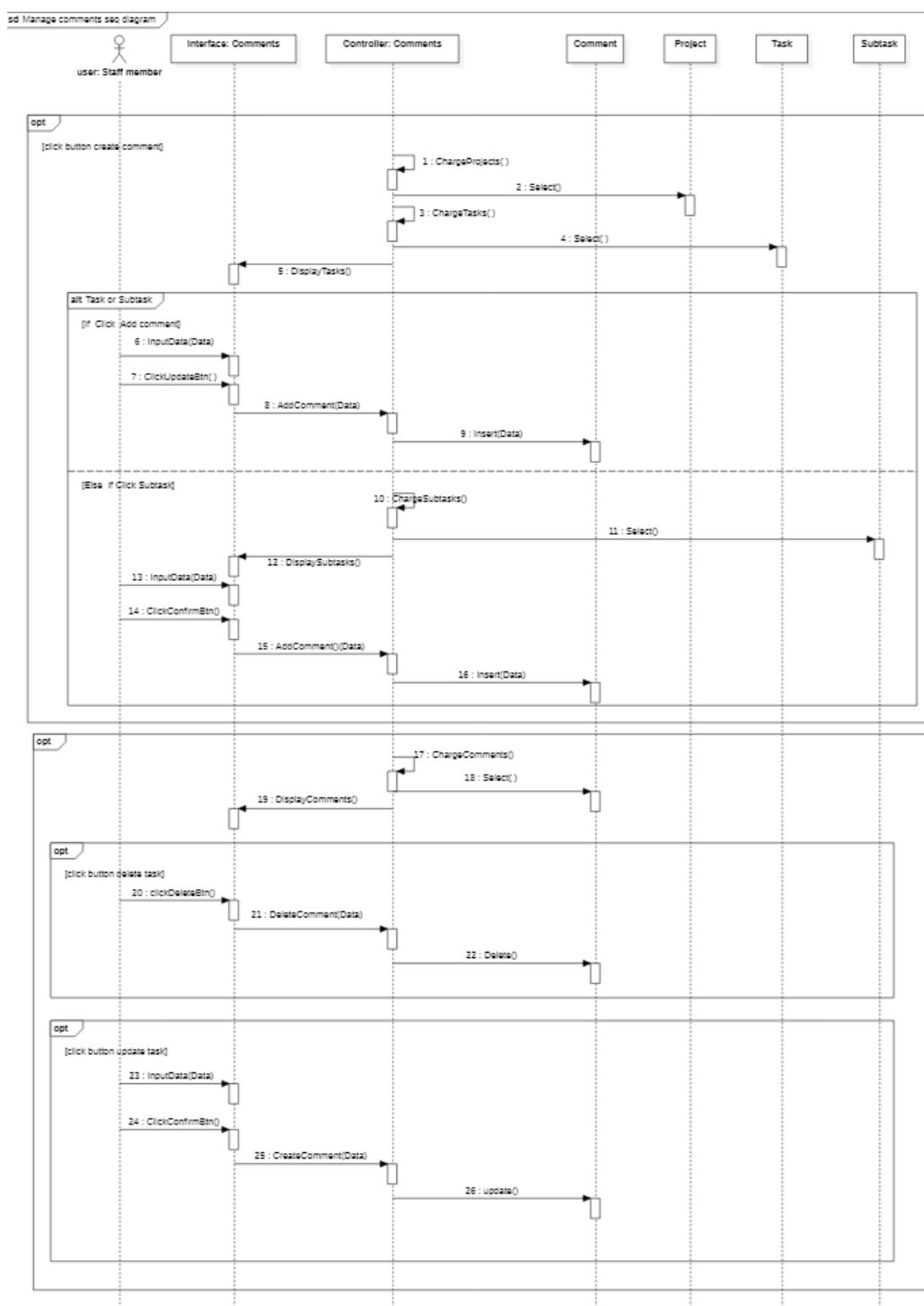


Figure 4.24: sequence diagram of the staff story “manage comments”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Design of the customer story “consult documents”

Class diagram

The following figure represents the class diagram of this use-case.

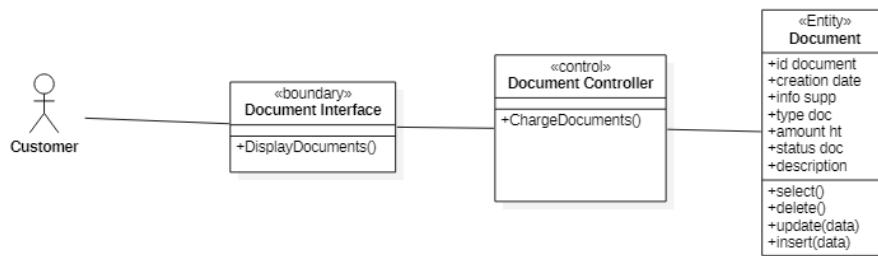


Figure 4.25: class diagram of the customer story “consult documents”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

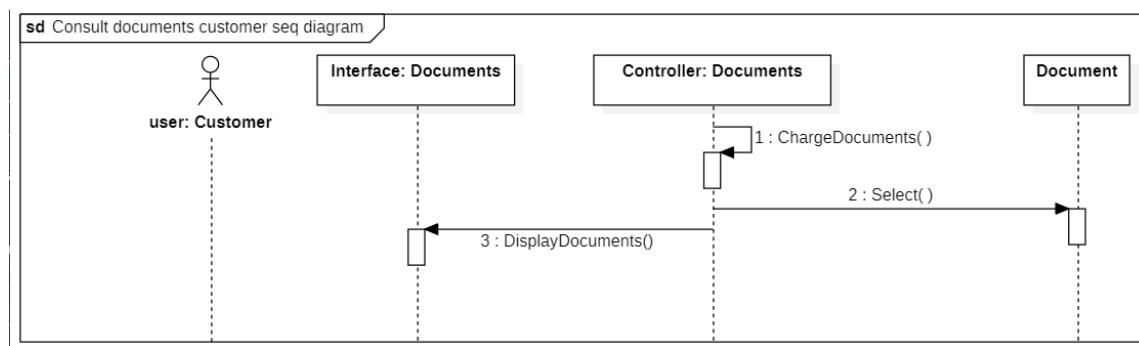


Figure 4.26: sequence diagram of the customer story “consult documents”

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

Design of the customer story “consult responses”

Class diagram

The following figure represents the class diagram of this use-case.

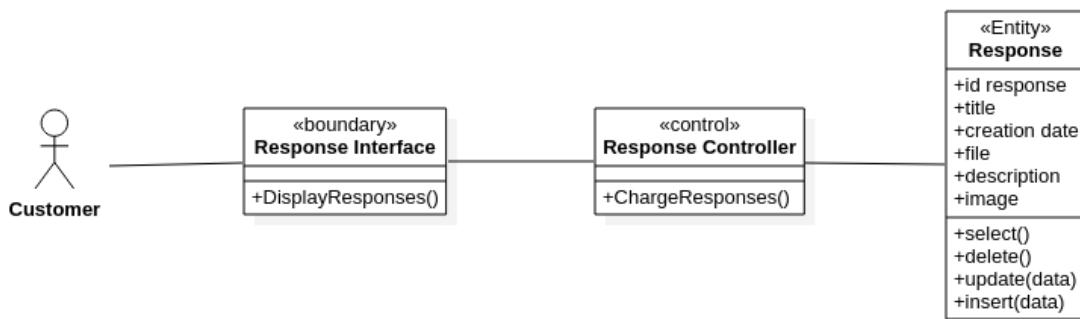


Figure 4.27: class diagram of the customer story “consult responses”

Sequence diagram

The following figure represents the sequence diagram of this use-case.

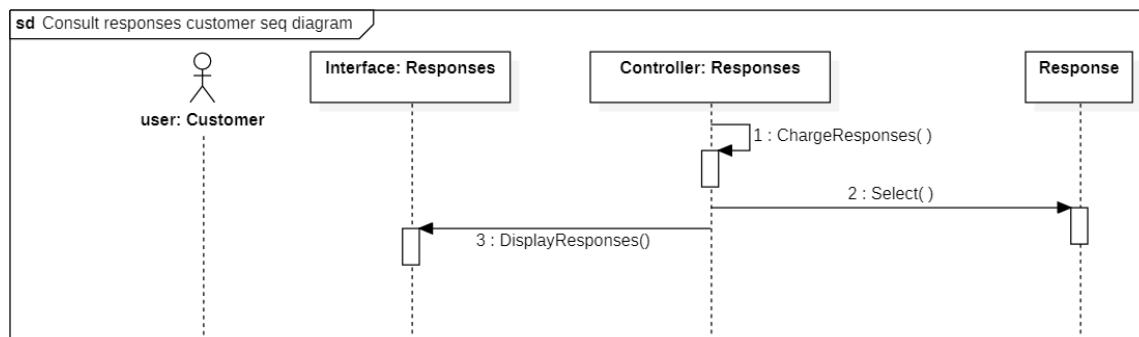


Figure 4.28: sequence diagram of the customer story “consult responses”

4.5 Global class diagram

The following figure concludes the global class diagram of our project

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

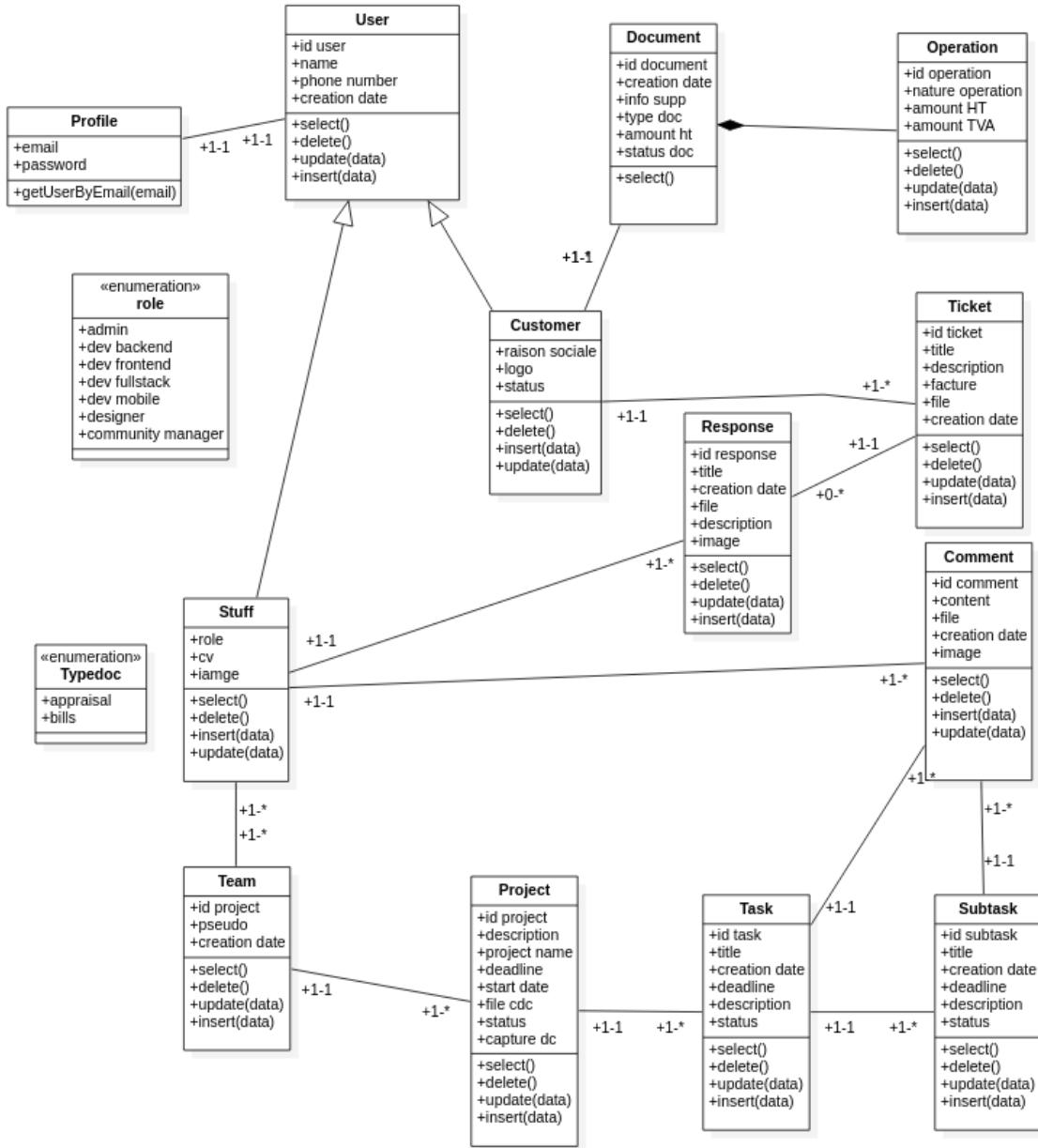


Figure 4.29: Global class diagram

4.6 Implementation of sprint 3

Realization of the staff story "Manage Responses"

The following figure shows the manage responses interface in dashboard staff through which the staff member can consult the list of the added responses, or add a new response by clicking on the add response.

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

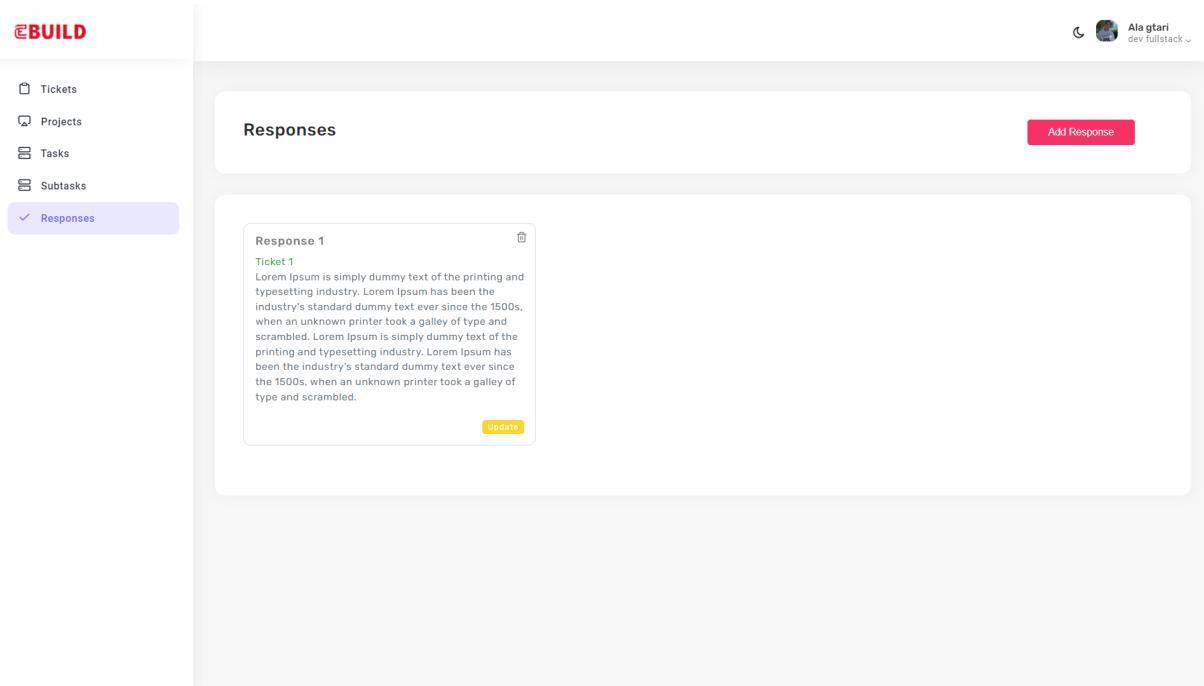


Figure 4.30: Manage Responses Interface

Upon clicking add response button, a form pops up showing the required fields to fill. When submitting the form, a notification pops up confirming the addition of the new response and then the staff is redirected to the responses interface with the updated list of responses.

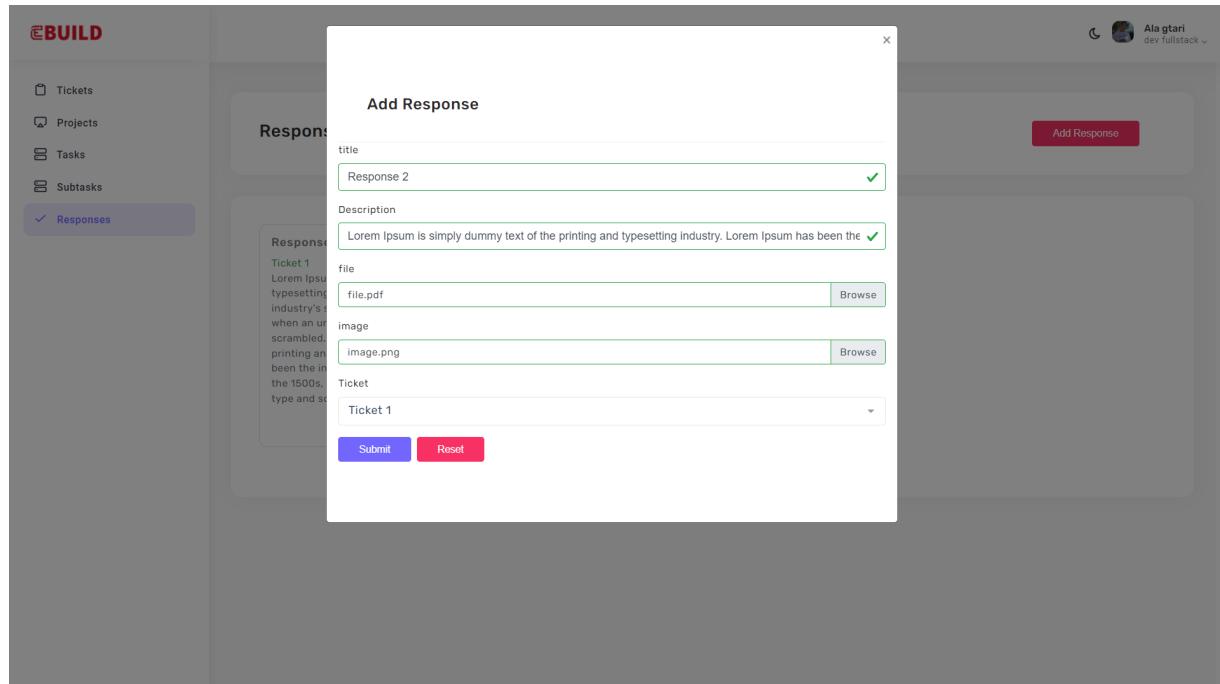


Figure 4.31: Add Response Interface

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

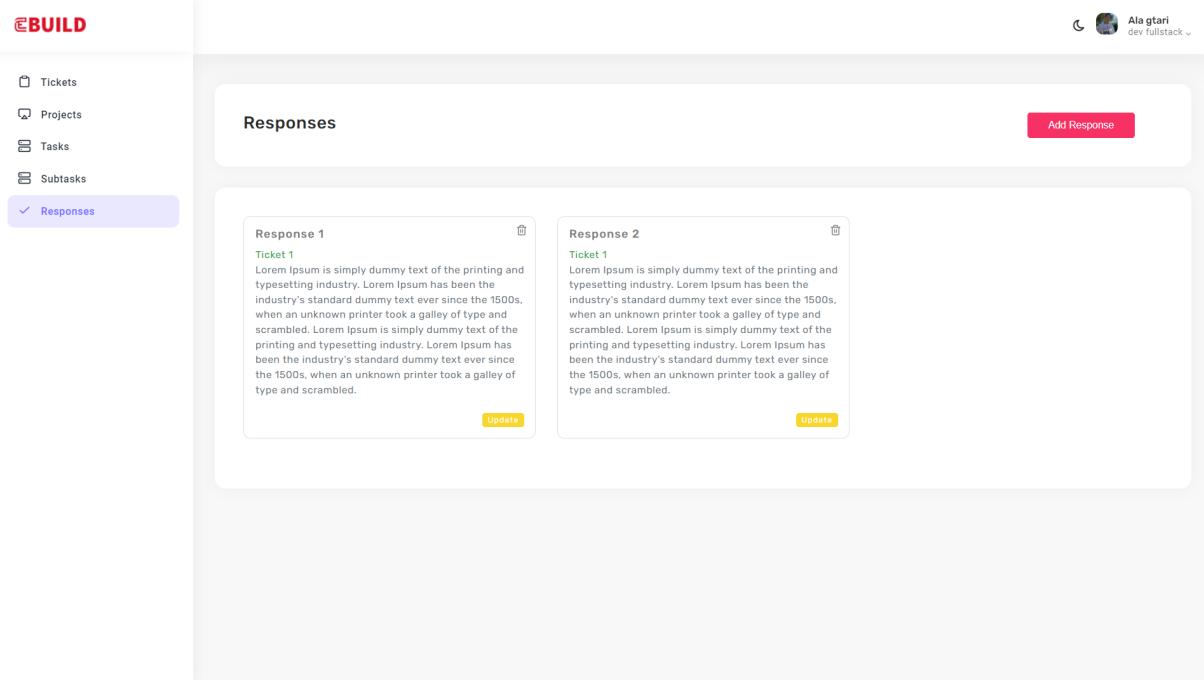


Figure 4.32: Added Response Interface

The staff member can update the consulted response by clicking the update button and altering the input of the form.

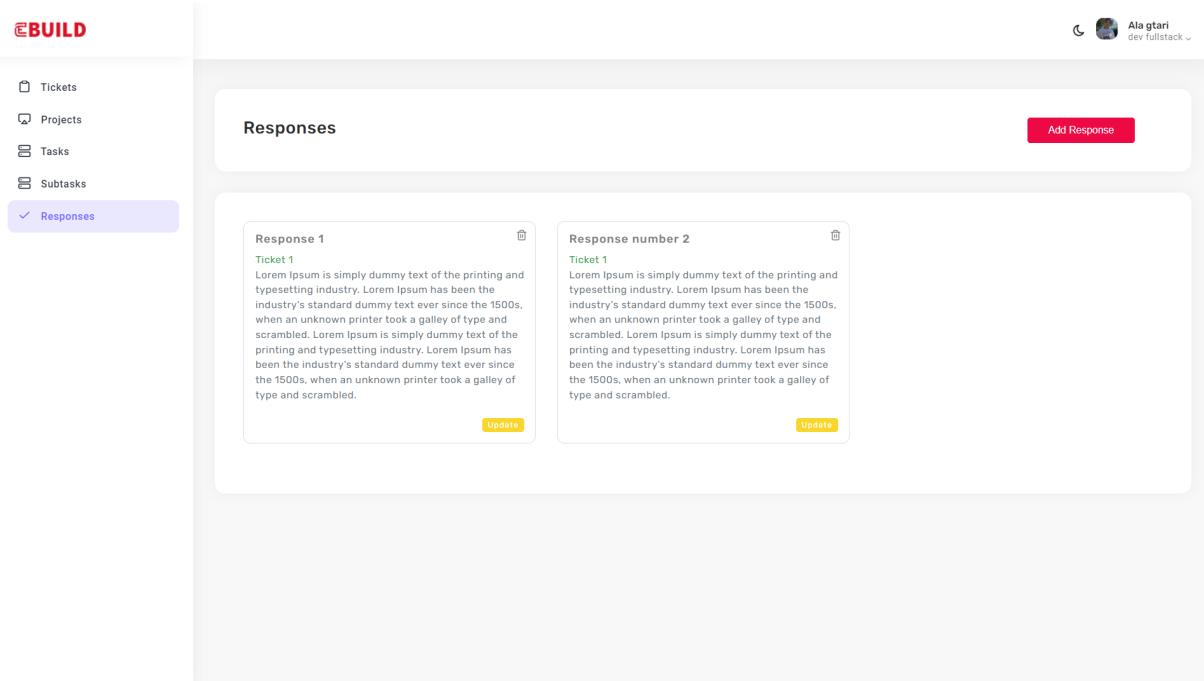


Figure 4.33: Updated Response Interface

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

The administrator can also delete a response when they click the delete button.

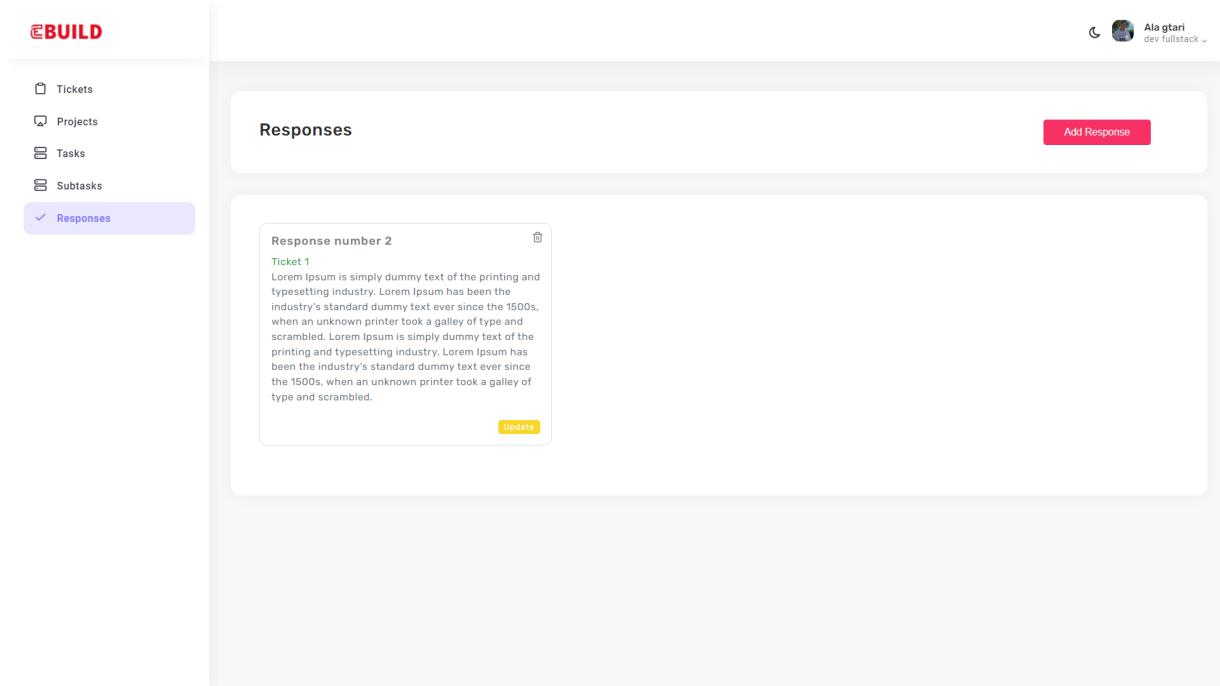


Figure 4.34: Deleted Response Interface

Realization of the staff story "Manage Responses"

The following figure shows the manage comments interface in dashboard staff through which the staff member can consult the list of the added comments to the selected task/subtask, or add a new comment.

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

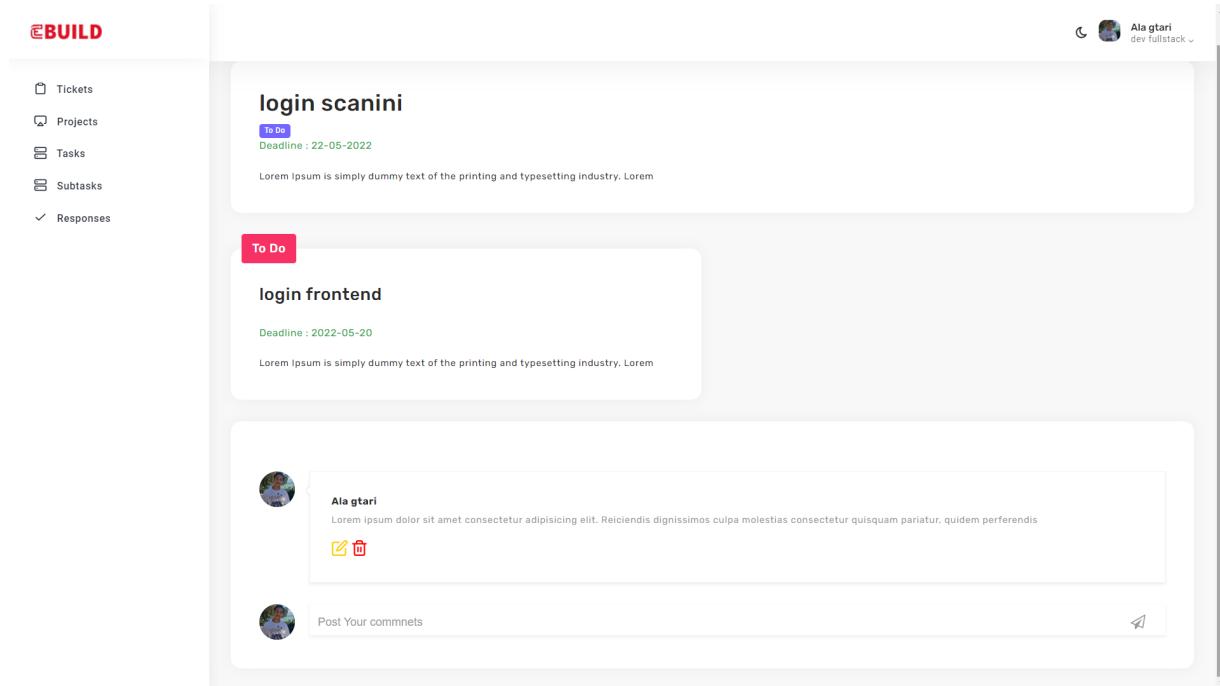


Figure 4.35: Manage Comments Interface

Upon typing in the comment section of a selected task/subtask and clicking on the send comment button, the added comment is displayed instantly. The staff member can update the consulted task or subtasks added comments by clicking the update button and altering the input of the form.

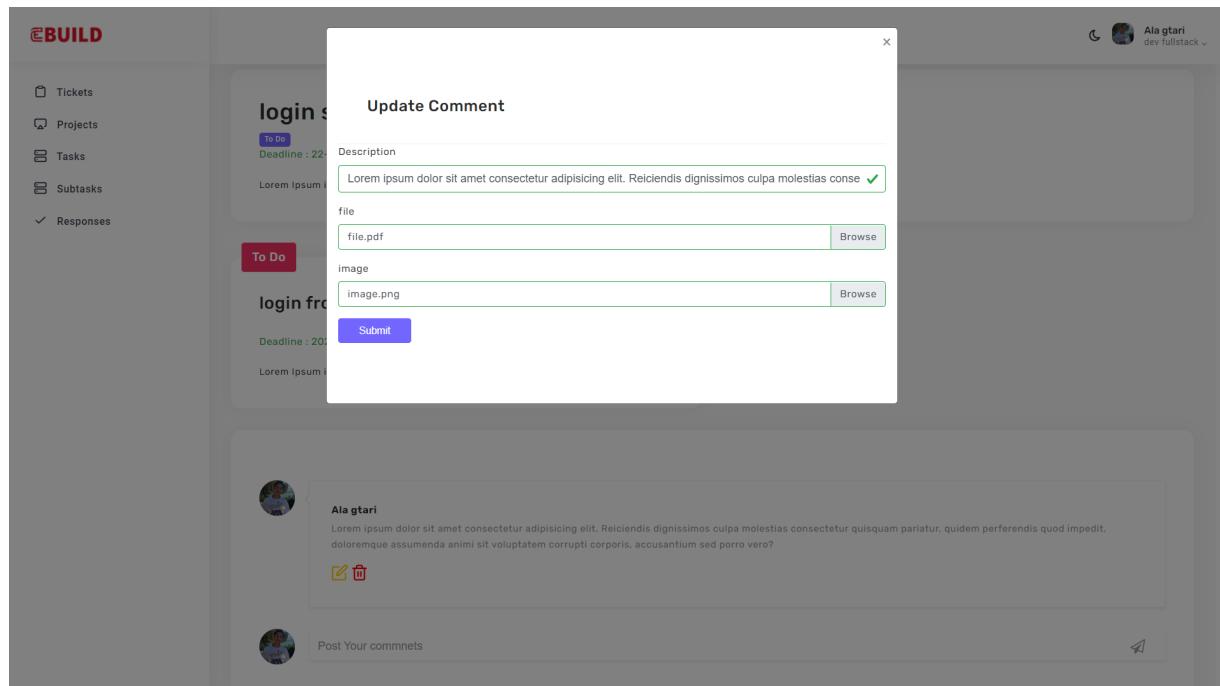


Figure 4.36: Update Task Comment Interface

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

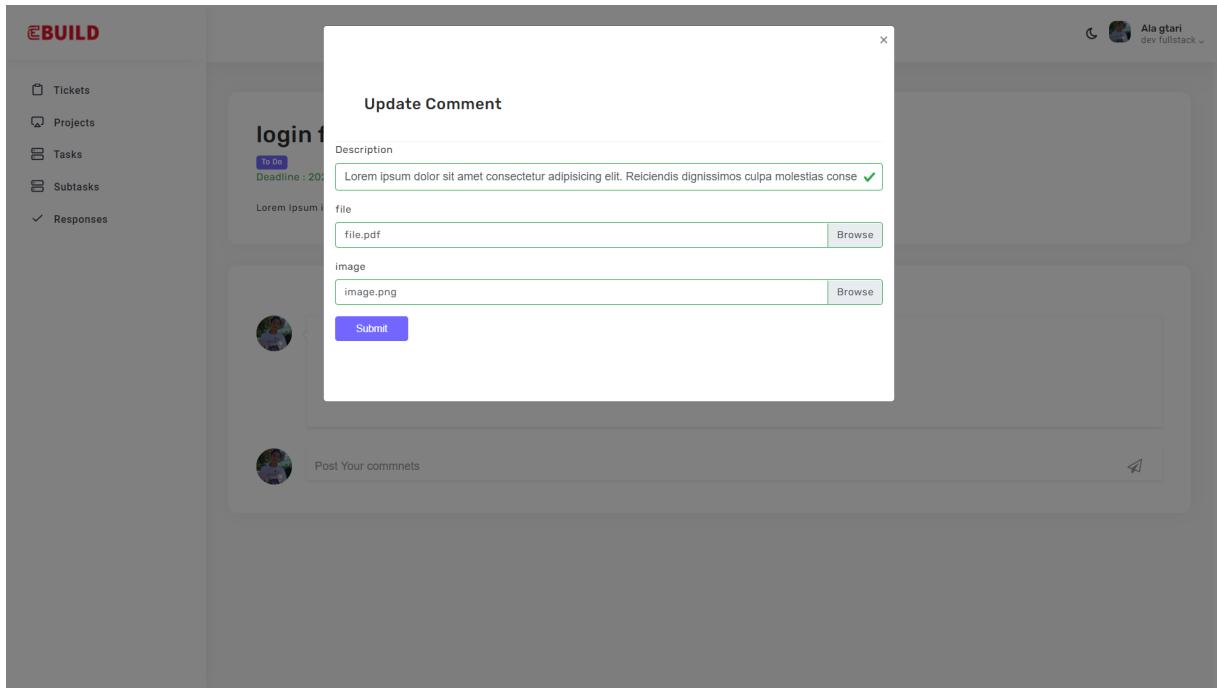


Figure 4.37: Update Subtask Comment Interface

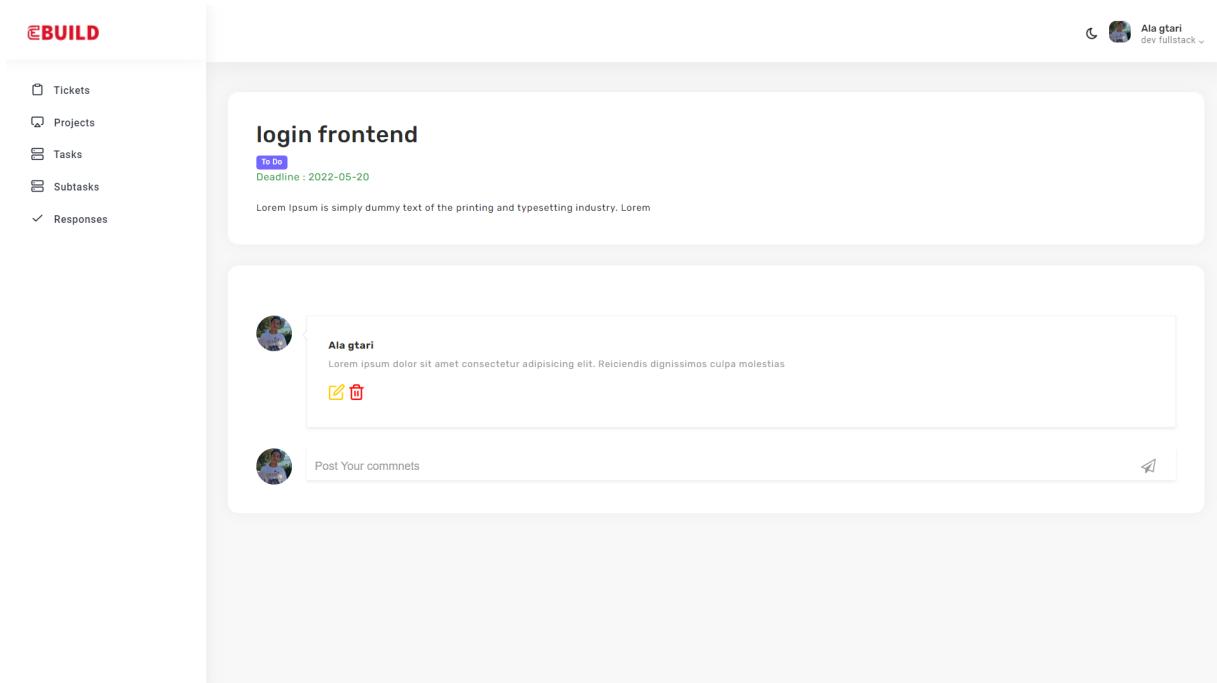


Figure 4.38: Updated Subtask Comment Interface

The administrator can also delete a response when they click the delete button.

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

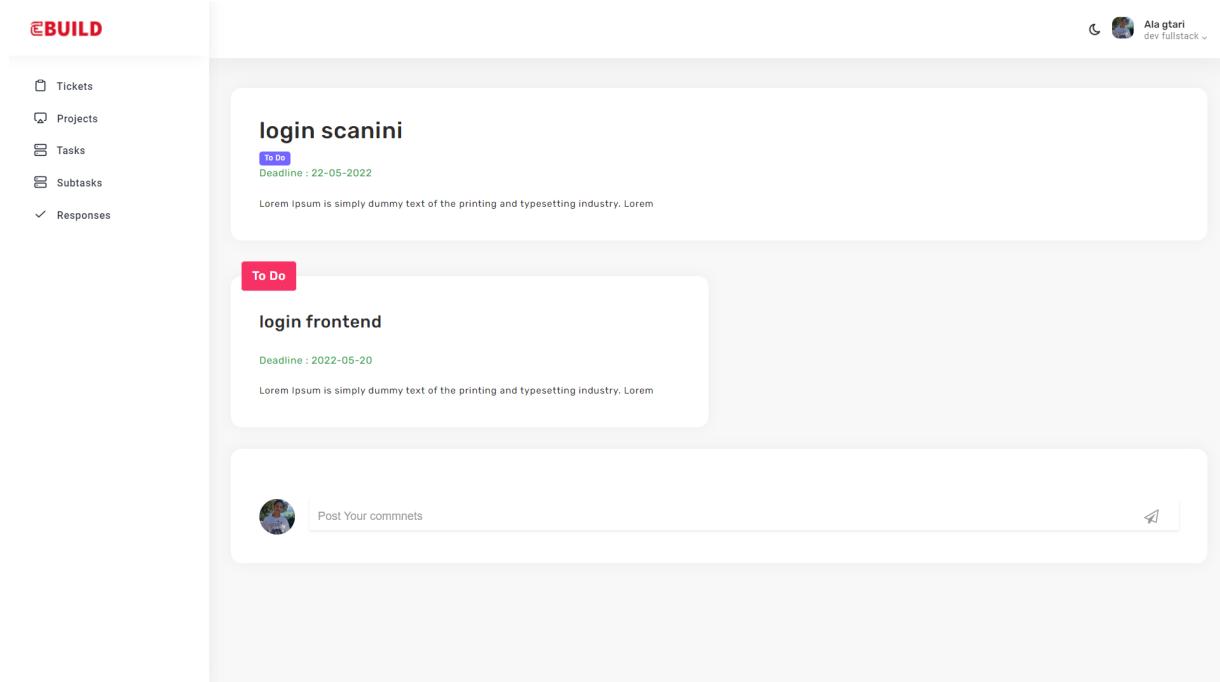


Figure 4.39: Deleted Task Comment Interface

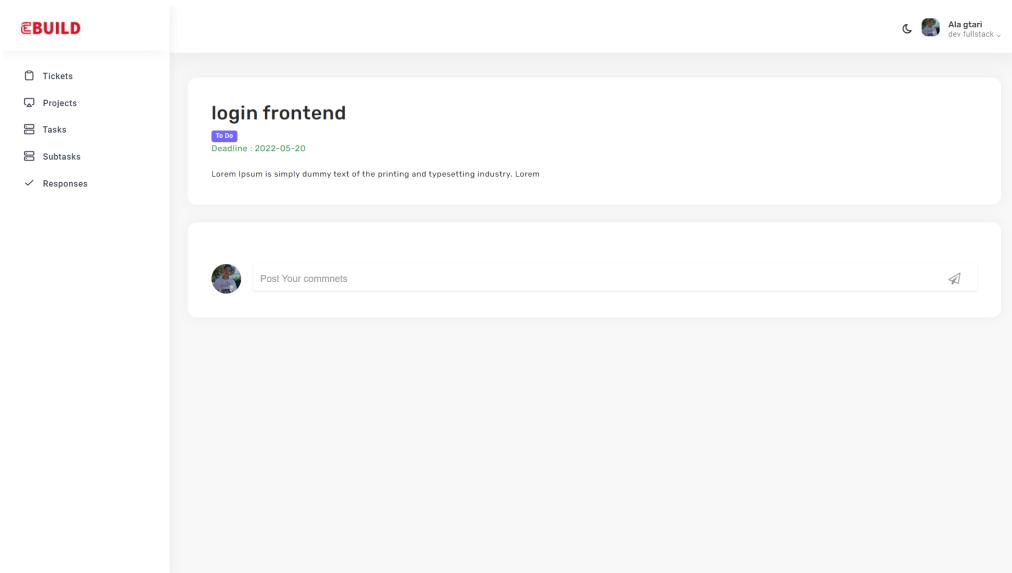


Figure 4.40: Deleted Subtask Comment Interface

Realization of the customer story “consult documents”

The following figure shows the consult documents interface in dashboard customer through which the customer can consult the list of the documents which are associated to them.

CHAPTER 4. RELEASE 2 : IMPLEMENTATION OF THE PROJECT FOLLOW-UP

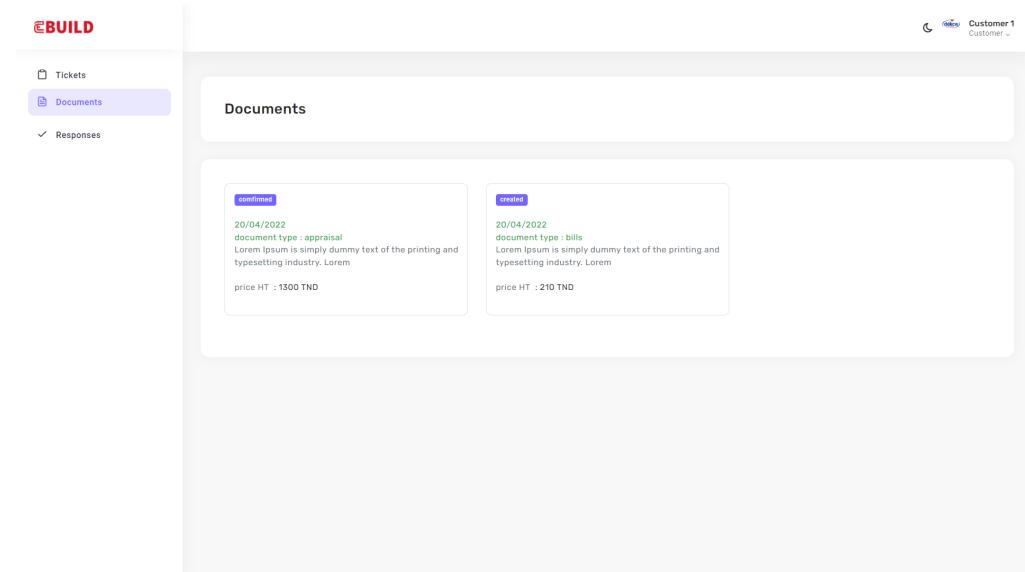


Figure 4.41: Consult Documents Interface

Realization of the customer story “consult responses”

The following figure shows the consult responses interface in dashboard customer through which the customer can consult the list of the responses which are associated to their tickets.

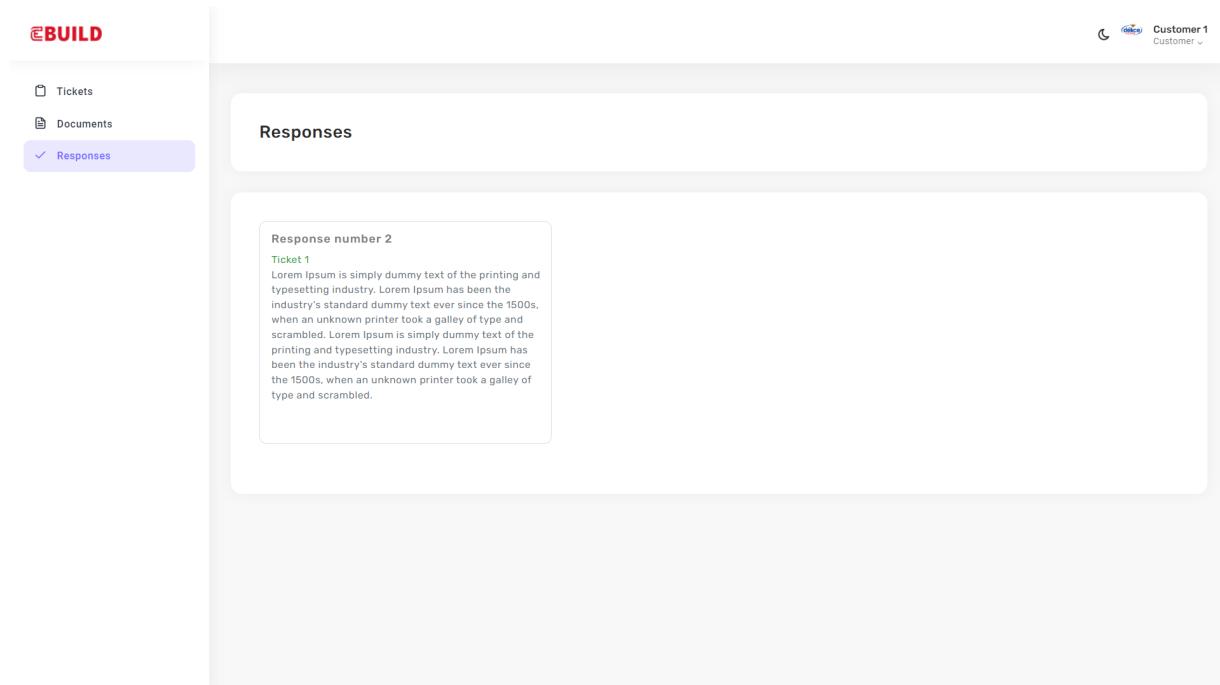


Figure 4.42: Consult Documents Interface

4.7 Conclusion

In this chapter, we have gone through the second release and the final implementation of our application. This release combines both the functions set for customers and staff members. It is composed of the last two sprints, so it is very much similar to the first release structure-wise. That is why we started with the refinement and explaining every use case story, then we carried on with our work to the conception of class and sequence diagrams attentively so that we introduce the visualization of the developed implementation eventually.

General Conclusion

Our PFE-internship CRM project was developed specifically to meet the needs of Ebuild, a new growing company dedicating its services to customers looking to build their brands. Ebuild's CRM aims to improve the Customer Lifetime value (CLV) by collecting the data and turning it into actionable insights, as well as organizing the working space all along the projects lifetime by having each staff member contributing more than ever with the new developed tasks and comments services.

We have seized this opportunity to make the best of it by broadening our computer science skills throughout learning new technologies and implementing them into the developed solution, which is more advanced than what we used to build as university projects. Moreover, we had the chance to refine our communication skills as we had our professional supervisor setting up meetings on a daily basis in addition to the teamwork experience with everyone else at the company all along the development of this project.

Eventually, we are discussing with our professional supervisor the possibility of taking E-Build's CRM to the next step by making it a cloud based CRM. This innovation would integrate many features such as marketing automation as well as extending the customer service systems. We are aiming to keep our project up to date with the new technologies by implementing new services that satisfy the growing customers needs. We are looking forward to seeing it solving the real world problem for this company, and making part of the team maintaining it in the future.

Bibliography

- [1] <https://www.scrum.org/>.
- [10] <https://www.scrum.org/>.
- [11] <https://www.scrum.org/>.
- [12] <https://vuejs.org/>.
- [13] <https://laravel.com/>.
- [14] <https://www.mysql.com/>.
- [15] <https://www.redhat.com/en/topics/api/what-is-a-rest-api>.
- [16] <https://code.visualstudio.com/>.
- [17] <https://www.postman.com/>.
- [18] <https://staruml.io/>.
- [19] <https://xamppapp.com/>.
- [2] <https://www.scrum.org/>.
- [20] <https://clickup.com/about>.
- [21] <https://github.com/>.
- [3] <https://www.scrum.org/>.
- [4] <https://www.scrum.org/>.
- [5] <https://www.scrum.org/>.
- [6] <https://www.scrum.org/>.
- [7] <https://www.scrum.org/>.
- [8] <https://www.scrum.org/>.
- [9] <https://www.scrum.org/>.