Republic of Tunisia
The Ministry of Higher Education and Scientific Research

The University of Carthage

The Higher Institute of Information and Communication Technologies

END-OF-STUDY PROJECT REPORT

Submitted in Partial Fulfillment of the Requirements for the
Bachelor Degree in Computer Science

Field of Study : Software Engineering and Information Systems

---

Development of an Administrative Web Application for Human Resource Management

---

By
Khila Mohamed Aziz and Hazbri Mohamed Aziz

Conducted within Tunisia Smart Solutions - T2S

Publicly defended on May 31, 2025 in front of the jury members:

Chairman:	Mr Ouertani Mohamed Wahb, CONECT INTECH
Reporter:	Mrs Najjar Asma, Teacher, ISTIC
Examiner:	Mr Attia Moez, Teacher, ISTIC
Professional Supervisor:	Mr Berriche Karim, Project Manager, T2S
Academic Supervisor:	Mrs Belhedi Amira, Teacher, ISTIC

Academic Year: 2024-2025

Republic of Tunisia
The Ministry of Higher Education and Scientific Research

The University of Carthage

The Higher Institute of Information and Communication Technologies

END-OF-STUDY PROJECT REPORT

Submitted in Partial Fulfillment of the Requirements for the
Bachelor Degree in Computer Science

Field of Study : Software Engineering and Information Systems

---

Development of an Administrative Web Application for Human Resource Management

---

By
Khila Mohamed Aziz and Hazbri Mohamed Aziz

Conducted within Tunisia Smart Solutions - T2S

Authorization of graduation project report submission:

Professional Supervisor:
Berriche Karim

Issued on :

Signature:

Academic Supervisor:
Belhedi Amira

Issued on :

Signature:

Dedications

We dedicate this work to everyone who has supported and inspired us throughout this journey.

Thank you to our families who have always believed in us by providing endless support and staying by our side no matter what. You understand how dreamy we are, yet you push us forward as we chase our dreams.

Thank you to every teacher who supported and inspired us by sharing valuable knowledge, challenging us to achieve better, pushing us to our limits, and helping us believe in ourselves.

Thank you to our friends who were there to cheer us up whenever things get tough, you presented the real meaning of friendship.

Thank you to everyone who came across this path even with just a word, we appreciate your support.

Khila Mohamed Aziz and Hazbri Mohamed Aziz

Acknowledgements

Before presenting our work, we would like to thank all the people who contributed to the success of our internship and who helped us in the drafting of this report.

First of all, we want to express our deep gratitude to all of T2S family especially Mr. Karim Berriche for his relevant explanations and valuable tips that have the greatest impact in the success of the completed project. We would also like to thank Mrs. Meryem Ben Askri for helping us and being very kind and supportive.

We would like to thank our internship supervisor, Mrs. Amira Belhedi for her guidance and the valuable lessons we learned from her.

We would also like to thank all our professors for being very kind and caring.

Contents

Dedications i

Acknowledgements ii

General Introduction 1

1 Presentation of the project context 2
1.1 Introduction . . . 2
1.2 Host company presentation . . . 2
1.2.1 Host company . . . 2
1.2.2 Company services . . . 2
1.2.3 Company solutions . . . 3
1.3 Study of the existing . . . 3
1.3.1 Sage Paie and RH . . . 3
1.3.2 Sage Espace Employés . . . 4
1.3.3 Odoo HR . . . 4
1.3.4 Workday HCM . . . 4
1.4 Critique of the existing . . . 5
1.4.1 Sage Paie and RH . . . 5
1.4.2 Sage Espace Employés . . . 6
1.4.3 Odoo HR . . . 6
1.4.4 Workday HCM . . . 6
1.5 Proposed Solution . . . 8
1.6 Conclusion . . . 8

2 Requirements Specifications 9
2.1 Introduction . . . 9
2.2 Requirements specification . . . 9
2.2.1 Functional requirements . . . 9
2.2.2 Non-Functional requirements . . . 11
2.2.3 Technological choices . . . 11
2.3 Project Management Methodology . . . 13
2.3.1 Scrum . . . 13
2.3.2 The Scrum Team . . . 13
2.3.3 Scrum Events . . . 14
2.3.4 Scrum Artifacts . . . 14
2.4 Global use case diagram . . . 14
2.4.1 Description of the actors . . . 15
2.4.2 Presentation of the global use case diagram . . . 16

2.5 Product Backlog 16
2.6 Conclusion 18

3 Sprint 0 : System Initialization 19
3.1 Introduction 19
3.2 Identifying the sprint 0 backlog 19
3.3 Refinement of sprint 0 19
3.3.1 Refinement of the ”Authenticate” use case 19
3.3.2 Refinement of the ”Manage Users” use case 20
3.3.3 Refinement of the ”Manage Departments” use case 21
3.4 Design of sprint 0 22
3.4.1 “Authenticate” Use Case Design 22
3.4.2 “Manage Users” Use Case Design 23
3.4.3 “Manage Departments” Use Case Design 24
3.5 Implementation of sprint 0 25
3.5.1 Realization of the “Authenticate” use case 25
3.5.2 Realization of the “Manage Users” use case 27
3.5.3 Realization of the “Manage Departments” use case 29
3.6 Conclusion 32

4 Sprint 1 : Structure and Access Control 33
4.1 Introduction 33
4.2 Identifying the sprint 1 backlog 33
4.3 Refinement of sprint 1 33
4.3.1 Refinement of the ”Manage Roles” use case 33
4.3.2 Refinement of the ”View Enterprise Structure” use case 34
4.3.3 Refinement of the ”Manage Extra Validation Levels” use case . . . 35
4.4 Design of sprint 1 36
4.4.1 “Manage Roles” Use Case Design 36
4.4.2 “View Enterprise Structure” Use Case Design 37
4.4.3 “Manage Extra Validation Levels” Use Case Design 38
4.5 Implementation of sprint 1 39
4.5.1 Realization of the “Manage Roles” use case 39
4.5.2 Realization of the “View Enterprise Structure” use case 42
4.5.3 Realization of the “Manage Extra Validation Levels” use case . . . 43
4.6 Conclusion 43

5 Sprint 2 : HR Request Management 44
5.1 Introduction 44
5.2 Identifying the sprint 2 backlog 44
5.3 Refinement of sprint 2 44
5.3.1 Refinement of the ”Manage my leave request” use case 45
5.3.2 Refinement of the ”Manage my loan request” use case 45
5.3.3 Refinement of the ”Manage my assigned bonus request” use case . . 46
5.3.4 Refinement of the ”View my received bonus” use case 47
5.3.5 Refinement of the ”Manage employees leave requests” use case . . . 48
5.3.6 Refinement of the ”Manage employees loan requests” use case . . . 49
5.3.7 Refinement of the ”Manage employees bonus requests” use case . . 49
5.4 Design of sprint 2 50

5.4.1 “Manage my leave request” Use Case Design 50
5.4.2 “Manage my loan request” Use Case Design 52
5.4.3 “Manage my assigned bonus request” Use Case Design 53
5.4.4 “View my received bonus” Use Case Design 55
5.4.5 “Manage employees leave requests” Use Case Design 55
5.4.6 “Manage employees loan requests” Use Case Design 57
5.4.7 “Manage employees bonus requests” Use Case Design 59
5.5 Implementation of sprint 2 60
5.5.1 Realization of the “Manage my leave request” use case 60
5.5.2 Realization of the “Manage my loan request” use case 63
5.5.3 Realization of the “Manage my assigned bonus request” use case . . 65
5.5.4 Realization of the “View my received bonus” use case 68
5.5.5 Realization of the “Manage employees leave requests” use case . . . 69
5.5.6 Realization of the “Manage employees loan requests” use case . . . 71
5.5.7 Realization of the “Manage employees bonus requests” use case . . 73
5.6 Conclusion 75

6 Sprint 3 : Engagement Internal Policies 76
6.1 Introduction 76
6.2 Identifying the sprint 3 backlog 76
6.3 Refinement of sprint 3 76
6.3.1 Refinement of the ”Manage events” use case 76
6.3.2 Refinement of the ”View profile” use case 77
6.3.3 Refinement of the ”Chat with other users” use case 78
6.3.4 Refinement of the ”Manage company policies” use case 78
6.4 Design of sprint 3 79
6.4.1 “Manage events” Use Case Design 79
6.4.2 “View my profile” Use Case Design 81
6.4.3 “Chat with other users” Use Case Design 82
6.4.4 “Manage company policies” Use Case Design 83
6.5 Implementation of sprint 3 85
6.5.1 Realization of the “Manage events” use case 85
6.5.2 Realization of the “View my profile” use case 88
6.5.3 Realization of the “Chat with other users” use case 89
6.5.4 Realization of the “Manage company policies” use case 89
6.5.5 Realization of the “View dashboard” use case 90
6.6 Conclusion 92

General Conclusion 93

Netography 94

List of Figures

1.1 Tunisia Smart Solutions logo 2
1.2 Interfaces of existing solutions 5

2.1 Global use case diagram 16

3.1 “Authenticate” use case diagram 20
3.2 “Manage Users” use case diagram 20
3.3 “Manage Departments” use case diagram 21
3.4 ”Authenticate” Class diagram 22
3.5 ”Authenticate” Sequence diagram 23
3.6 ”Manage Users” Class diagram 23
3.7 ”Manage Users” Sequence diagram 24
3.8 ”Manage Departments” Class diagram 24
3.9 ”Manage Departments” Sequence diagram 25
3.10 ”Authenticate” Interface 26
3.11 ”Forgot Password” Interface 26
3.12 ”Forgot Password” Interface 27
3.13 ”Manage Users” Interface 27
3.14 ”View User” Interface 28
3.15 ”Add User” Interface 28
3.16 ”Delete User” Interface 29
3.17 ”Edit User” Interface 29
3.18 ”Manage Departments” Interface 30
3.19 ”View Department” Interface 30
3.20 ”Add Department” Interface 31
3.21 ”Delete Department” Interface 31
3.22 ”Edit Department” Interface 32

4.1 “Manage Roles” use case diagram 34
4.2 “View Enterprise Structure” use case diagram 35
4.3 “Manage Extra Validation Levels” use case diagram 35
4.4 ”Manage Roles” Class diagram 36
4.5 ”Manage Roles” Sequence diagram 37
4.6 ”View Enterprise Structure” Class diagram 38
4.7 ”View Enterprise Structure” Sequence diagram 38
4.8 ”Manage Extra Validation Levels” Class diagram 38
4.9 ”Manage Extra Validation Levels” Sequence diagram 39
4.10 ”Manage Roles” Interface 40
4.11 ”View role permissions” Interface 40

4.12 "Add Role" Interface 41
4.13 "Delete Role" Interface 41
4.14 "Edit Role" Interface 42
4.15 "View Enterprise Structure" Interface 42
4.16 "Manage Extra Validation Levels" Interface 43

5.1 "Manage my leave request" use case diagram 45
5.2 "Manage my loan request" use case diagram 46
5.3 "Manage my assigned bonus request" use case diagram 47
5.4 "View my received bonus" use case diagram 47
5.5 "Manage employees leave requests" use case diagram 48
5.6 "Manage employees loan requests" use case diagram 49
5.7 "Manage employees bonus requests" use case diagram 50
5.8 "Manage my leave request" Class diagram 51
5.9 "Manage my leave request" Sequence diagram 51
5.10 "Manage my loan request" Class diagram 52
5.11 "Manage my loan request" Sequence diagram 53
5.12 "Manage my assigned bonus request" Class diagram 54
5.13 "Manage my assigned bonus request" Sequence diagram 54
5.14 "View my received bonus" Class diagram 55
5.15 "View my received bonus" Sequence diagram 55
5.16 "Manage employees leave requests" Class diagram 56
5.17 "Manage employees leave requests" Sequence diagram 57
5.18 "Manage employees loan requests" Class diagram 58
5.19 "Manage employees loan requests" Sequence diagram 58
5.20 "Manage employees bonus requests" Class diagram 59
5.21 "Manage employees bonus requests" Sequence diagram 60
5.22 "Manage my leave request" Interface 61
5.23 "View leave request" Interface 61
5.24 "Add leave request" Interface 62
5.25 "Delete leave request" Interface 62
5.26 "Edit leave request" Interface 63
5.27 "Manage my loan request" Interface 63
5.28 "View loan request" Interface 64
5.29 "Add loan request" Interface 64
5.30 "Delete loan request" Interface 65
5.31 "Edit loan request" Interface 65
5.32 "Manage my assigned bonus requests" Interface 66
5.33 "View my assigned bonus request" Interface 66
5.34 "Assign bonus to employee" Interface 67
5.35 "Delete my assigned bonus request" Interface 67
5.36 "Edit my assigned bonus request" Interface 68
5.37 "View received bonuses" Interface 68
5.38 "View a received bonus" Interface 69
5.39 "Manage employees leave requests" Interface 69
5.40 "View Leave Request" Interface 70
5.41 "Approve leave request" Interface 70
5.42 "Reject leave request" Interface 71

5.43 ”Manage employees loan requests” Interface 71
5.44 ”View a loan Request” Interface 72
5.45 ”Approve loan request” Interface 72
5.46 ”Reject Loan request” Interface 73
5.47 ”Manage employees bonus requests” Interface 73
5.48 ”View a Bonus Request” Interface 74
5.49 ”Approve bonus request” Interface 74
5.50 ”Reject bonus request” Interface 75

6.1 “Manage events” use case diagram 77
6.2 “View profile” use case diagram 77
6.3 “Chat with other users” use case diagram 78
6.4 “Manage company policies” use case diagram 79
6.5 ”Manage events” Class diagram 80
6.6 ”Manage events” Sequence diagram 81
6.7 ”View my profile” Class diagram 82
6.8 ”View my profile” Sequence diagram 82
6.9 ”Chat with other users” Class diagram 82
6.10 ”Chat with other users” Sequence diagram 83
6.11 ”Manage company policies” Class diagram 83
6.12 ”Manage company policies” Sequence diagram 84
6.13 Global Class Diagram 85
6.14 ”Manage events” Interface 86
6.15 ”View event” Interface 86
6.16 ”Add event” Interface 87
6.17 ”Delete event” Interface 87
6.18 ”Edit event” Interface 88
6.19 ”View my profile” Interface 88
6.20 ”Change password” Interface 89
6.21 ”Chat with other users” Interface 89
6.22 ”View company policies” Interface 90
6.23 ”Manage company settings” Interface 90
6.24 ”View dashboard” Interface 91
6.25 ”Share feedback” Interface 91

List of Tables

1.1 Sage Solutions by T2S . . . 3
1.2 Existing solutions Comparison . . . 7

2.2 Dynamic Roles Table . . . 15
2.3 Product Backlog . . . 17

3.1 Product backlog sprint 0 . . . 19
3.2 Detailed description of the "Authentication" use case . . . 20
3.3 Detailed description of the "Manage Users" use case . . . 21
3.4 Detailed description of the "Manage Departments" use case . . . 22

4.1 Product backlog sprint 1 . . . 33
4.2 Detailed description of the "Manage Roles" use case . . . 34
4.3 Detailed description of the "View Enterprise Structure" use case . . . 35
4.4 Detailed description of the "Manage Extra Validation Levels" use case . . . 36

5.1 Product backlog sprint 2 . . . 44
5.2 Detailed description of the "Manage my leave request" use case . . . 45
5.3 Detailed description of the "Manage my loan request" use case . . . 46
5.4 Detailed description of the "Manage my assigned bonus request" use case 47
5.5 Detailed description of the "View my received bonus" use case . . . 48
5.6 Detailed description of the "Manage employees leave requests" use case . . . 48
5.7 Detailed description of the "Manage employees loan requests" use case . . . 49
5.8 Detailed description of the "Manage employees bonus requests" use case . 50

6.1 Product backlog sprint 3 . . . 76
6.2 Detailed description of the "Manage events" use case . . . 77
6.3 Detailed description of the "View profile" use case . . . 78
6.4 Detailed description of the "Chat with other users" use case . . . 78
6.5 Detailed description of the "Manage company policies" use case . . . 79

General Introduction

In today's fast-paced business environment, efficient human resource management is one of the key factors that determine a firm's success. To remain competitive and productive, an organization needs an integrated system that enables the smooth administration of various HR activities such as salary bonuses, leave, loans, and promotions. Tunisia Smart Solutions - T2S, a sector leader, recognizes the importance of automating these processes to enhance overall operational effectiveness.

It is in this context that our project entitled "Development of an Administrative Web Application for Human Resource Management" is taking place. It aims to address the challenges of managing HR-related activities through a centralized, user-friendly web application. The application includes key functionalities such as user management, department management, salary bonus management, event management (leaves and absences), loan management; all essential components of the company's HR operations come with a customizable validation circuit, we use a hierarchical one, which means that every request starts with the immediate manager of the requester, once approved it moves up to the next manager until it reaches the top level in the hierarchy, finally the request reaches extra validation levels (users) only after being approved by all the hierarchy levels. Also, we have dynamic roles so that the enterprise can manage them as they want to make the usage more flexible, these two points are the weak spots of the existing solution, Sage Paie & RH, which T2S is currently using.Their absence makes it harder for T2S to function, especially with large businesses and complex organizational structures. It is this limitation that made T2S to develop a more flexible and scalable platform; our solution, where these two aspects are one of its strongest features.

The development stages of this project are detailed in this report, which consists of four chapters.
- The first chapter introduces the host organization. Additionally, it discusses the study of the existing system, its critique, and the proposed solution.
- The second chapter focuses on the requirements specification, covering functional and non-functional requirements, actors, the global use case diagram, backlog, and work environments.
- The third chapter explores Release 1: Core System Setup, detailing the study and realization of Sprint 0 and Sprint 1, along with their respective use case diagrams, refinements, designs, and implementations.
- The fourth chapter introduces Release 2: Employee Services and Communication, presenting the study and realization of Sprint 2 and Sprint 3, followed by their respective use case diagrams, refinements, designs, and implementations.
Eventually, we finalize our report with a general conclusion and a viewpoint of the project and beyond.

Chapter 1

Presentation of the project context

1.1 Introduction

This chapter begins by presenting the host company Tunisia Smart Solutions - T2S, followed by a description of the existing study and its critique leading us finally to the proposed solution.

1.2 Host company presentation

In this section, the host company and its services are presented generally.

1.2.1 Host company

Figure 1.1 shows Tunisia Smart Solutions [1] logo which is a SAGE Consulting partner created in March 2018 and specializing in supporting small and medium-sized businesses in the integration and deployment of accessible and adapted management solutions.
Organized into three divisions, T2S offers, in addition to integration and consulting, development services to provide its clients with scalable solutions that meet their most specific needs.



Figure 1.1: Tunisia Smart Solutions logo

1.2.2 Company services

Tunisia Smart Solutions – T2S provides a variety of services related to enterprise management solutions, specifically Sage software. These include ERP and financial software

integration, business process consulting, custom software development to expand Sage functionality, and technical support and training. T2S provides these services to help businesses optimize operations, improve efficiency, and scale effectively.

1.2.3 Company solutions

Table 1.1 illustrates SAGE solutions offered by Tunisia Smart Solutions

Table 1.1: Sage Solutions by T2S

Solution	Description
Sage 100 Paie & RH	A payroll and human resources solution that simplifies employee data management, payroll processing, and compliance with labor regulations.
Sage 100 Comptabilité	An accounting solution for SMEs, providing tools for financial management, compliance, and reporting.
Sage 100 CRM	A Customer Relationship Management system that helps businesses manage interactions with current and potential customers, enhancing sales and customer service.
Sage 100 Entreprise	An Enterprise Resource Planning (ERP) solution integrating various business processes, including finance, operations, and human resources, to streamline operations.
Sage 100 Gestion Commerciale	A commercial management solution offering tools for sales, inventory, and order processing, aimed at improving commercial operations.
Sage BI Reporting	A business intelligence tool that provides data analysis and reporting capabilities, helping businesses make informed decisions based on real-time data.
Sage Espace Employés	An employee self-service portal that allows staff to access personal information, payslips, and other HR-related documents.
Sage Génération Experts	A solution designed for accounting professionals, offering tools for client management, reporting, and compliance.

1.3 Study of the existing

In this section, we present the most well-known HR management solutions. The first two are Sage Paie & RH and Sage espace employés, which are the two applications currently used at T2S. We also introduce two other applications, Odoo HR and Workday HCM. The respective interfaces of each of these applications are illustrated in Figure 1.2.

1.3.1 Sage Paie and RH

Sage Paie and RH, whose main interface is shown in Figure 1.2(a), is an extremely popular payroll and human resource activity management software. Sage Paie is primarily a payroll management software that offers features such as automatic salary calculation,

bonus management, tax deduction, and generation of pay slips. Sage RH manages human resource activities such as employee performance tracking, leave management, and promotions. These two systems support one another to simplify payroll and HR operations by providing end-to-end employee compensation and management solutions.
The system assists HR administrators in automating most of the work related to payroll, reducing manual intervention and ensuring quick and precise salary payouts. It also enables effective management of employees' leaves and absences while monitoring employee performance and promotions. The integration of Sage Paie with RH provides HR employees with a smooth experience and minimizes data duplication across various systems as much as possible. Both software have some loopholes, which are analyzed later.

1.3.2 Sage Espace Employés

Sage Espace Employés (see figure 1.2 (b)) is a self-service portal that allows employees to access and manage their personal data, payroll information, leave balances, and HR communications. This platform enables employees to view their pay stubs, track their leave requests, and update their personal information, resulting in a more autonomous experience. The integration with Sage Paie provides employees with real-time access to their payroll information and leave balances.
This self-service approach helps HR teams cut administrative costs by allowing employees to handle routine HR tasks on their own. Despite its benefits, Sage Espace Employés lacks more advanced features like performance evaluation tracking, promotion management, and other customized HR services. These functional gaps limit its effectiveness in promoting employee engagement and overall human resource management.

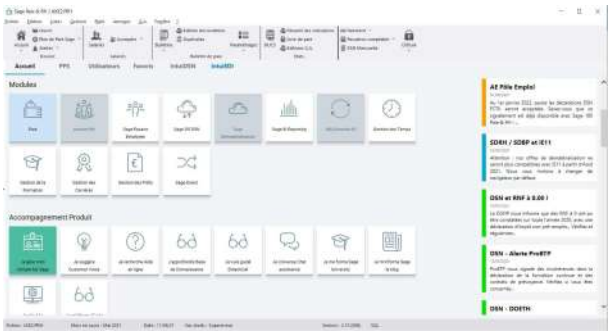
1.3.3 Odoo HR

Odoo HR [2] (see figure 1.2 (c)) is a well-known national human resource management software, delivering an all-in-one toolset for leave management, attendance, payroll, and employee records. It automatically calculates salaries with consideration for Tunisian-specific tax laws, deductions, and bonuses. The software also supports contract management, recruitment, and employee performance appraisal. Odoo's modular design enables organizations to extend and enhance their human resource functions by adding other applications, like accounting, project management, and document management.
Odoo comes in both cloud and on-premise versions, thereby catering to various business requirements. Although Odoo HR is affordable and flexible as a solution for small and medium enterprises, it needs proper configuration to deploy according to Tunisian labor law, and may require extra technical skill for some organizations to reach its full potential.

1.3.4 Workday HCM

Workday Human Capital Management (HCM) [3] (shown in figure 1.2 (d)) is a cloud-based HR software particularly made for big businesses with employees spread across the globe. It offers numerous functionalities like payroll, benefits management, recruiting employees, workforce planning, and compliance verification. Workday is equipped with one important benefit, which is AI-driven analytics to enable HR executives to make

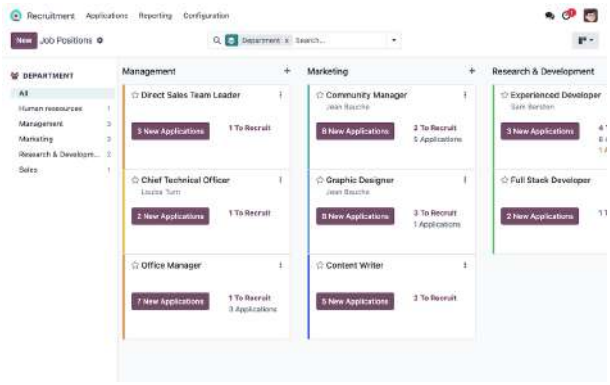
data-driven decisions on aspects such as workforce optimization and keeping employees engaged. The system enables employees to maintain their own profiles, request time off, and view payroll information on any device. Workday also accommodates many global labor laws, which is one of the reasons that most global companies like using it.
Nevertheless, even with its extensive capabilities, Workday is costly and needs a lot of training to be able to use it to its full potential. It is difficult to tailor without the help of developers, thus making it more complex for small and medium-sized businesses that need special HR processes.



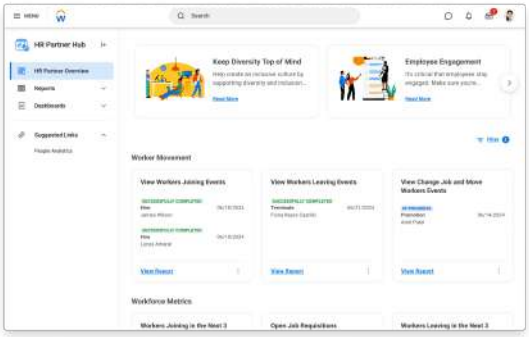
(a) Sage Paie & RH



(b) Sage Espace Employés



(c) Odoo HR



(d) Workday HCM

Figure 1.2: Interfaces of existing solutions

1.4 Critique of the existing

In this section, the limitations of the management systems listed previously are highlighted in the drawbacks column of Table 1.2, leading to the solution proposed in the next section.

1.4.1 Sage Paie and RH

Though Sage Paie and RH are robust solutions for administering payroll and human resource functions, they suffer from usability and functionality issues. To begin with, the interfaces of both systems are not user-friendly, which can lead to difficulties for employees

lacking technical skills. Secondly, complicated workflows associated with processing payroll, tracking leaves, and tracking promotions have the potential to make HR procedures less efficient, particularly when managing high levels of employee data. Although functional, the systems require significant manual intervention and modifications, especially in areas such as bonus administration and leave approvals.
A further limitation relates to insufficient levels of customization. Although Sage Paie and RH provide some level of setup, they lack flexibility to address unique business requirements. This lack of flexibility becomes an issue for organizations that demand greater levels of personalized processes. Furthermore, the reporting functionality is very limited since the system merely provides generic reports with no ability to include enhanced reporting tools that enable human resources to obtain greater insight or create more tailored reports.

1.4.2 Sage Espace Employés

Sage Espace Employés, while improving on traditional HR management systems by providing employee self-service, has a limited scope. Employees can view their payroll information and request leave, but it does not support advanced features such as performance evaluations, career development tracking, or promotion management. This lack of functionality makes it difficult for employees to manage their career advancement or remain fully engaged with the HR process without the constant involvement of HR personnel.
Additionally, user experience can improve when designing the system. The system may be difficult for employees to utilize, especially when trying to access certain reports or check their performance. Apart from this, while the self-service functions are useful, they need to expand to cater to other HR-related tasks, such as keeping goals or requesting career guidance. Security and compliance features are adequate but can be enhanced to ensure more secure protection of sensitive employee information.

1.4.3 Odoo HR

Odoo HR has a very wide range of features that are all aimed at the management of human resources, but it is not exempt from certain limitations that may affect its usefulness for growing businesses. Firstly, there is a significant issue caused by the need to manually tailor the system to be compatible with Tunisian labor laws, which may take considerable development. The system has beneficial payroll and leave management functionality, yet its automation and analytic functionality is quite weaker in nature as compared to big international players. Furthermore, Odoo's modular design allows companies to incorporate additional human resource functionality; however, the utilization of various modules results in higher costs and complexity. System updates do not please users because they frequently introduce unwanted changes, and companies have to continuously adjust their processes. In spite of these limitations, Odoo remains a suitable option for companies seeking an affordable and adaptable human resources solution that complies with local legal requirements.

1.4.4 Workday HCM

Workday HCM, although reputed to be a market leader in the global human resource management domain, has some limiting factors that might deter small organizations from

adopting it. The nature of the platform itself warrants firms to invest a lot of time and resources in training their HR personnel to effectively utilize the system. Despite the extensive set of features provided by Workday, its premium subscription price which is in the range of $100-200K annually makes it out of reach for small and medium-sized enterprises. Also, the customization in the system is a bit limited, and technical skills are needed for more specific solutions. Another difficulty is its global compliance capabilities, although Workday is dedicated to different labor laws, businesses based in Tunisia might require extra configuration to be fully compliant with local law. In general, Workday is a powerful and innovative HR solution, yet because of its cost and implementation complexity, it is more directed for large-scale businesses with dedicated HR and IT departments. Table 1.2 presents a summary of the advantages and limitations of each of the existing solutions we have just presented.

Table 1.2: Existing solutions Comparison

Software	Advantages	Drawbacks
Sage Paie & RH	• Automates salary and payroll. • Manages leave, bonuses, promotions. • Reduces manual work.	• Complex for non-technical HR. • Lacks flexibility for custom needs. • Still needs manual actions.
Sage Espace Employés	• Self-service for data and payroll. • Real-time sync with Sage Paie. • Web and mobile access.	• No performance/promotion tools. • Limited engagement features. • Interface needs better accessibility.
Odoo HR	• Tunisian law support. • Affordable for SMEs. • Cloud and on-premise.	• Weaker automation/analytics. • Updates can cause instability. • Multi-module complexity.
Workday HCM	• AI-driven analytics. • Cloud access on any device. • Global compliance.	• High cost for SMEs. • Needs HR training. • Devs needed for customization.

1.5 Proposed Solution

The proposed solution involves creating a modern administrative web application that will address the shortcomings of the current human resource management systems in use at Tunisia Smart Solutions. The application enhances and automates key HR workflows, including bonus management, leave tracking, user and role assignments, department administration, approval workflow, loan management, and promotion tracking. Additionally, it empowers employees by enabling them to apply for loans and leaves, and conduct proper interactions with colleagues via an in-built chat functionality.
The system streamlines payroll and bonus administration through automated salary calculations, thus ensuring accurate calculations of bonuses, taxes, and deductions. The system provides employees with timely access to their payment histories, pay structures, and bonuses, thus improving transparency and efficiency.
The leave request module enables employees to directly request off-time through the system, where the power to grant or reject the request is vested in the human resources and the management. If approved, the system automatically adjusts leave balances, thus ensuring accurate recording and less administrative burden.
The user and role functionality enables the simple administration of access control, which presents the assignment of specific role to each user, and based on those roles the features that different users can do vary, which is a weak point of the existing solutions. Department-level management helps create a systematic system within the organization, making human resource operations and administrative oversight more efficient.
Profile management helps employees make changes to their personal and professional information, creating a feeling of responsibility towards their professional growth. Validation helps maintain the system approval process in place, ensuring that leave as well as loan applications comply with the set protocol.
The loan module allows employees to apply for internal loans right within the system, with control and monitoring via the HR department. The promotion management feature covers employee performance evaluation, thus ensuring that opportunities for professional growth are administered systematized based on set criteria. To improve interactions as well as collaboration between staff, the site has an integrated real time chatting system that enables interaction between employees.
With effortless integration into popular packages like Sage Paie and RH, the groundbreaking platform minimizes process fragmentation, improves data accuracy, and increases overall human resource operation efficiency. Defined by its stringent scalability and ease of use, this solution transforms HR management at T2S, offering a more integrated and independent experience for employees and HR personnel alike.

1.6 Conclusion

In this first chapter, we present the host company, an overview of the project composed of a study and critique of the existing, followed by the proposed solution. The next chapter is dedicated to the requirements specifications.

Chapter 2

Requirements Specifications

2.1 Introduction

This chapter begins by identifying the functional and non-functional requirements of the system. Then it presents the technical requirements. Furthermore, the methodology used for the design is discussed, followed by a global use case diagram. Finally, the product backlog is introduced.

2.2 Requirements specification

The requirements specification is used to analyze the functionalities of our project in order to specify the functional, non-functional and technical 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 fulfill during the process of development, what helps them to keep track of their progress.
The main functions of our project are listed below.

- **Authenticate** – Users must log in to the system using their credentials to access its functionalities securely.
- **Manage users** – Authorized users can view, create, update, or delete users, as well as assign their roles and their managers.
- **Manage departments** – Authorized users can view, create, update or delete departments to organize users within the system.
- **Manage roles** – Authorized users who have the appropriate permissions can create new roles, modify existing ones, or remove roles entirely. Each role defines a set of permissions that determine what actions a user assigned to that role can perform within the system. This functionality ensures that access to sensitive features and data is restricted based on the user's role, enhancing security and maintaining proper control over the system's operations (it have nothing to do with the current job of the employees).

- **View enterprise structure** – Users can view the hierarchical enterprise structure which represents as well the validation circuit that each request (loan, leave or bonus assign) should go through to be approved or rejected.
- **Manage extra validation levels** – Users can add extra levels to the validation circuit (each level contains a user as extra validator) so that the request after going through the basic validation structure (which is the enterprise structure) should go through this extra levels before it can be approved.
- **Manage my leave request** – Users can Add (apply) for a leave request, they can as well view a pending request, edit or delete it. The leave request should go through a validation circuit so it can be either approved or rejected in the end.
- **Manage my loan request** – Users can Add (apply) for a loan request, they can as well view a pending request, edit or delete it. The loan request should go through a validation circuit so it can be either approved or rejected in the end.
- **Manage my assigned bonus request** – Authorized users can Add (assign) a salary bonus to an employee who is hierarchically subordinate to them, they can as well view a pending request, edit or delete it. The bonus request should go through a validation circuit so it can be either approved or rejected in the end.
- **View my received bonus** – Users can view the bonuses that they received from hierarchically higher users (managers).
- **Manage employees leave requests** – Authorized users can review or/and approve/reject leave requests submitted by employees. Each leave request goes through the validation circuit: after an initial review, the request is forwarded to the next authorized user in the validation circuit automatically. Each approver can either validate the request and pass it to the next user in the validation circuit, or reject it and add the reason. The request is only fully approved once all required users in the global validation circuit (extra validator included) have validated it.
- **Manage employees loan requests** – Authorized users can review or/and approve/reject loan requests submitted by employees. Same validation steps as leave requests.
- **Manage employees bonus requests** – Authorized users can review or/and approve/reject bonus requests submitted by employees. Same validation steps as leave and loan requests.
- **Manage events** – Authorized users can view or/and add or/and edit or/and delete events in the calendar.
- **View my profile** – Users can view their personal information, such as contact details, profile picture etc.
- **Chat with other users** – Users can communicate with colleagues through an integrated messaging system within the platform.
- **Manage company policies** – Users can view the company policies, create custom policies as well as editing them in addition to the default attendance and leave settings and delete those settings.

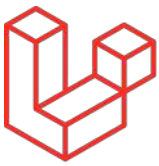
2.2.2 Non-Functional requirements

Non-functional requirements represent quality criteria that indirectly influence the effectiveness of functional requirements. Although they do not directly affect the functionalities of the website, they are essential to ensuring an optimal user experience and satisfactory performance. To achieve this, the following requirements must be met.

- **Reliability** The application must operate continuously without failure and provide unwavering stability, even during high traffic periods.
- **Errors management** Clear and informative error messages must be displayed in case of a problem, allowing users to understand and quickly correct errors without frustration.
- **Ergonomics and User Interface** The interface must be intuitive, making navigation between different sections of the site effortless. Visual elements, such as colors and layout, should be pleasant and optimized for a smooth user experience.
- **Security** The confidentiality of users' personal data must be strictly protected. The security of transactions and sensitive information, such as banking details, is a top priority.
- **Maintenance and Reusability** The application's architecture must be clear, well-documented, and compliant with standards to facilitate future maintenance and scalability.
- **Compatibility and Portability** The application must be compatible with various hardware and software environments, allowing its use across different devices (computers, tablets, smartphones).

2.2.3 Technological choices

For this part, we choose our technological choices during the process of developing the system.



Laravel [4] is an open-source PHP web application framework designed to make developing web applications easier and more efficient. It follows the MVC (Model-View-Controller) architectural pattern and provides a clean and elegant syntax for developers. Laravel offers a wide range of built-in tools, such as routing, middleware, authentication, and an ORM (Object-Relational Mapping) system called Eloquent, which simplifies database interactions.



HTML [5] (HyperText Markup Language) is the standard language used to create and structure content on the web. It provides the basic building blocks for web pages by defining elements like headings, paragraphs, links, images, and forms. HTML uses a system of tags to mark up content, allowing browsers to render text, media, and interactive elements on the screen.



CSS [6] (Cascading Style Sheets) is a stylesheet language used to describe the presentation of a web page's HTML or XML structure. It controls the layout, design, and overall visual appearance of elements on a webpage, such as colors, fonts, spacing, and positioning.



Javascript [7] (JS) is a high-level, dynamic programming language commonly used to create interactive effects within web browsers. It is one of the core technologies of web development, alongside HTML and CSS. JavaScript allows for the manipulation of the Document Object Model (DOM), enabling the creation of responsive, user-friendly websites and applications.



Git [8] is a distributed version control system (VCS) that allows developers to track changes in their codebase and collaborate with others efficiently. Unlike centralized version control systems, Git enables each user to have a full copy of the entire repository, including its history, on their local machine.



Github [9] is a web-based platform for version control and collaborative software development. It allows developers to store and manage their code repositories, track changes, and collaborate on projects with others. GitHub uses Git, a distributed version control system that enables developers to manage and track their code history efficiently.



Visual Studio Code [10] (VS Code) is a free, open-source code editor developed by Microsoft. It is highly customizable and provides support for a wide range of programming languages and frameworks. VS Code includes essential features like syntax highlighting, code completion (IntelliSense), debugging tools, and version control integration through Git.



Postman [11] is a popular collaboration platform for API development that allows developers to build, test, and document APIs. It provides a user-friendly interface for sending HTTP requests, inspecting responses, and analyzing API performance. Postman supports all HTTP methods (GET, POST, PUT, DELETE, etc.) and allows users to set parameters, headers, authentication, and request bodies.



SQL Server Management Studio [12] (SSMS) is an integrated environment developed by Microsoft for managing SQL Server databases. It provides a comprehensive set of tools for database administrators and developers to configure, manage, and administer SQL Server instances. SSMS allows users to execute queries, manage database security, perform backups and restores, and analyze performance.



StarUml [13] is a software modeling tool that supports multiple UML (Unified Modeling Language) diagrams for designing and visualizing software systems. It allows developers and architects to create models that represent the structure, behavior, and interactions within a software application.

2.3 Project Management Methodology

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

2.3.1 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. Here 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.

2.3.2 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.
Product Owner : The Product Owner is the member who is responsible for setting up

the Product Backlog, and establishing that it is translucent, clear, and understood. This responsibility also includes representing the needs of stakeholders.
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.

2.3.3 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.
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?
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.
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.
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.

2.3.4 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.
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.
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.

2.4 Global use case diagram

This section presents the overall use case diagram, which illustrates the main interactions between the system and its users. It identifies the key actors and their corresponding actions within the system.

2.4.1 Description of the actors

In this subsection, we describe how users interact with the system, outlining their roles and responsibilities, which are dynamically managed based on assigned permissions as illustrated in Table 2.2
In our system, user roles are dynamic, meaning they are not predefined but can be created and managed based on specific needs. The system allows administrators or users with the appropriate permissions to define new roles and assign them specific access rights. Each user is then assigned a role, which determines their level of access to various functionalities within the platform. This flexible approach makes it easier for any enterprise to tailor role management to its specific needs, ensuring that access control aligns with its internal structure and operational requirements.

Table 2.2: Dynamic Roles Table

Actor	**Roles**
User (Dynamic Role)	• Authenticate • Manage users (if role allows) • Manage departments (if role allows) • Manage roles (if role allows) • View enterprise structure • Manage extra validation levels (if role allows) • Manage my leave request • Manage my loan request • Manage my assigned bonus request • View my received bonus • Manage employees leave requests • Manage employees loan requests • Manage employees bonus requests • Manage events (if role allows) • View my profile • Chat with other users • Manage company policies (if role allows)

2.4.2 Presentation of the global use case diagram

The figure 2.1 provides a visual representation of the system's global use case, highlighting the relationships between actors and their functionalities.

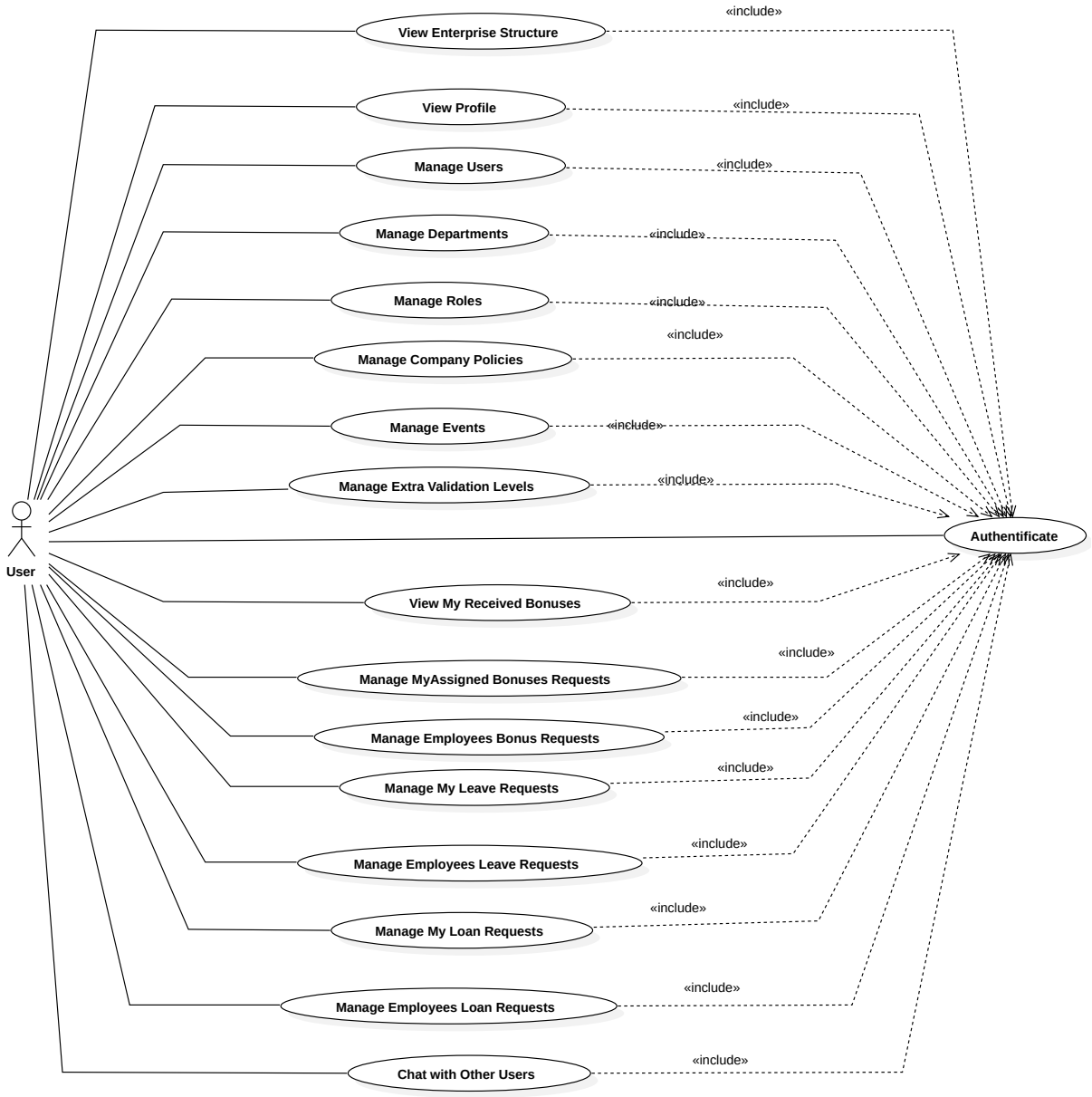


Figure 2.1: Global use case diagram

2.5 Product Backlog

The table 2.3 presents the list of user stories with their priorities, estimations, schedules, and associated releases, as part of product development.

Table 2.3: Product Backlog

User Story	Priority	Estimation	Sprint
As a user, I can authenticate myself.	1	Average	Sprint 0
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete users.	1	Average	Sprint 0
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete departments.	1	Average	Sprint 0
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete roles.	2	High	Sprint 1
As a user, I can view the enterprise hierarchical structure.	2	Average	Sprint 1
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete extra validation levels.	2	High	Sprint 1
As a user, I can create (apply) or/and view or/and edit or/and delete my leave request.	3	High	Sprint 2
As a user, I can create (apply) or/and view or/and edit or/and delete my loan request.	3	High	Sprint 2
As a user, I can create (assign) or/and view or/and edit or/and delete a salary bonus to my subordinate employees.	3	High	Sprint 2
As a user, I can view the bonuses that I received.	3	High	Sprint 2
As a user, I can manage (view or/and approve/reject) leave requests of my subordinates employees.	3	High	Sprint 2
As a user, I can manage (view or/and approve/reject) loan requests of my subordinates employees.	3	High	Sprint 2
As a user, I can manage (view or/and approve/reject) assigned bonus requests to my subordinates employees.	3	High	Sprint 2
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete events in the calendar.	4	Average	Sprint 3
As a user, I can view my profile.	4	Low	Sprint 3
As a user, I can chat with other users.	4	Average	Sprint 3
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete company policies.	4	Average	Sprint 3

2.6 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 Sprint 0.

Chapter 3

Sprint 0 : System Initialization

3.1 Introduction

In this chapter, we present the use cases for Sprint 0. The study of this sprint includes refinement, design, and implementation.

3.2 Identifying the sprint 0 backlog

The table 3.1 contains the backlog elements that are executed during sprint 0:

Table 3.1: Product backlog sprint 0

User Story	Priority	Estimation	Sprint
As a user, I can authenticate myself.	1	Average	Sprint 0
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete users.	1	Average	Sprint 0
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete departments.	1	Average	Sprint 0

3.3 Refinement of sprint 0

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

3.3.1 Refinement of the ”Authenticate” use case

The figure 3.1 showcases the use case.

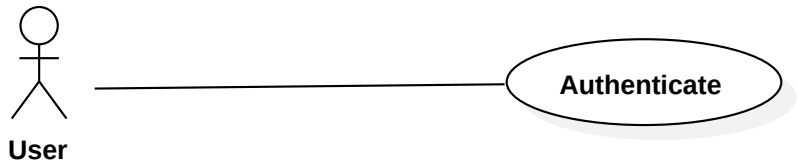


Figure 3.1: “Authenticate” use case diagram

The table 3.2 elaborates on this user story with textual description.

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

Use Case	« Authenticate »
Actor(s)	User
Pre-condition	System running
Post-condition	Actor authenticated and can access the platform.
Principal scenario	- System displays authentification interface. - Actor types his email and password. - Actor clicks on « Login » button. - The system verifies the login and password combination. - The system displays the home page according to the user profile.
Exception	The system displays an error message if the credentials are incorrect.

3.3.2 Refinement of the ”Manage Users” use case

The figure 3.2 showcases the use case.

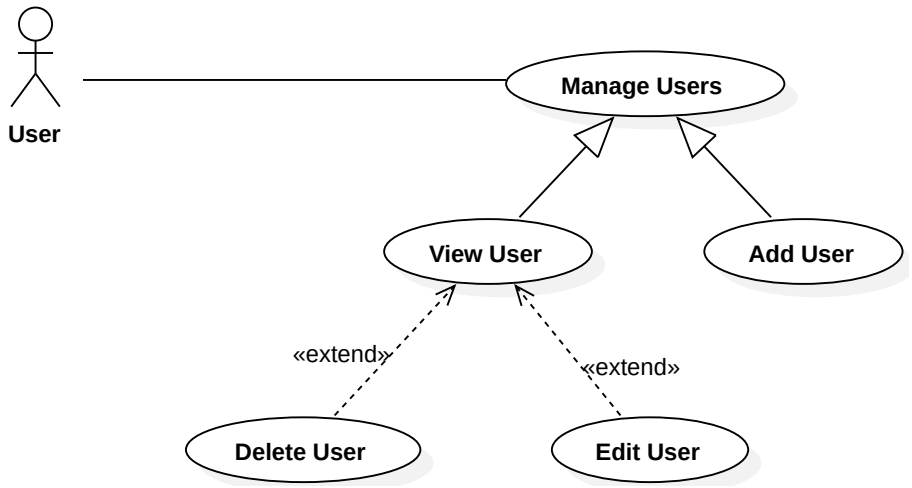


Figure 3.2: “Manage Users” use case diagram

The table 3.3 elaborates on this user story with textual description.

Table 3.3: Detailed description of the ”Manage Users” use case

Use Case	« Manage Users »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor have permissions to manage users.
Post-condition	Managed Users.
Principal scenario	- System displays user management interface. - Actor enters information for a new user then clicks on « submit » button. - Actor clicks on « Edit » button and makes needed changes then clicks on « save changes » button . - Actor clicks on « Delete » button and confirms deletion in the popped up window .
Exception	The system displays an error message if the changed information is incorrect.

3.3.3 Refinement of the ”Manage Departments” use case

The figure 3.3 showcases the use case.

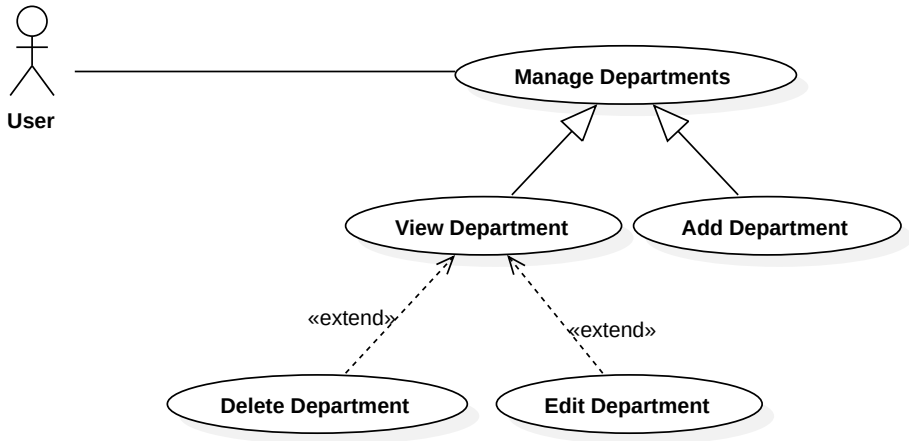


Figure 3.3: “Manage Departments” use case diagram

The table 3.4 elaborates on this user story with textual description.

Table 3.4: Detailed description of the "Manage Departments" use case

Use Case	« Manage Departments »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor have permissions to manage departments.
Post-condition	Managed Departments.
Principal scenario	- System displays department management interface. - Actor enters information for a new department then clicks on « submit » button. - Actor clicks on « Edit » button and makes needed changes then clicks on « save changes » button . - Actor clicks on « Delete » button and confirms deletion in the popped up window .
Exception	The system displays an error message if the changed information is incorrect.

3.4 Design of sprint 0

Design is a significant activity for understanding the development of a system to make it more reliable to the customer's needs. Therefore, we specify our system before implementing it.

3.4.1 "Authenticate" Use Case Design

Class diagram for the "Authenticate" use case is shown in Figure 3.4.

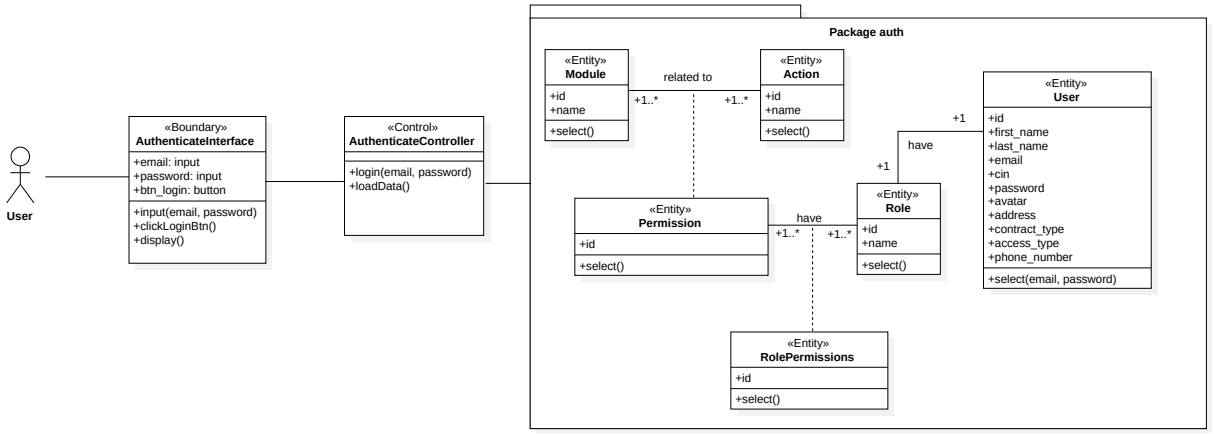


Figure 3.4: "Authenticate" Class diagram

Sequence diagram for the "Authenticate" use case is shown in Figure 3.5.

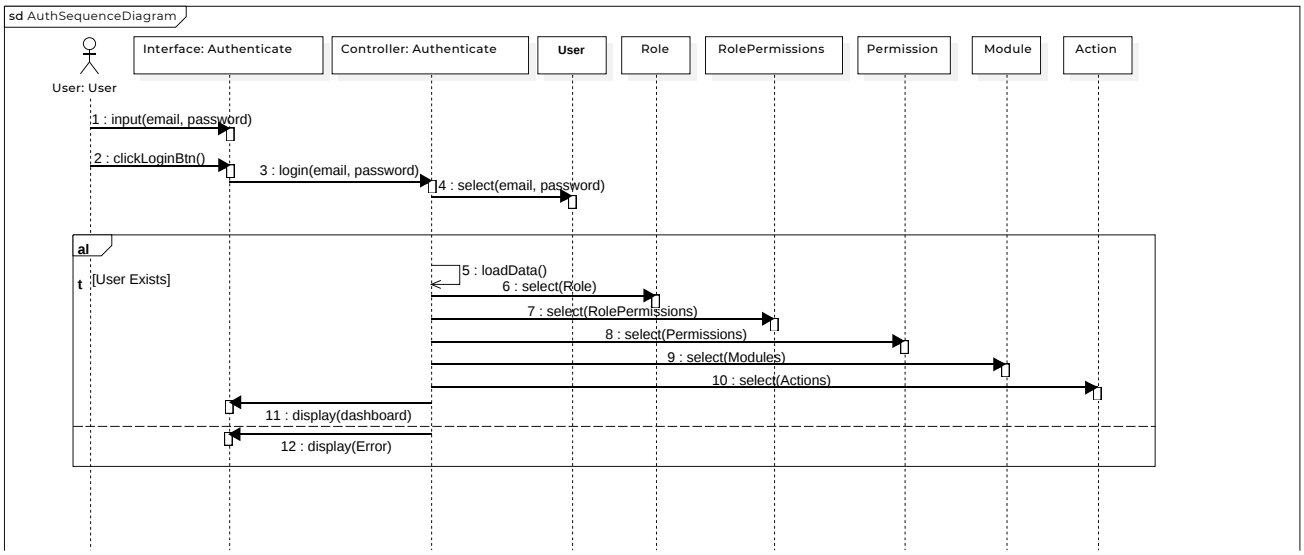


Figure 3.5: "Authenticate" Sequence diagram

3.4.2 "Manage Users" Use Case Design

Class diagram for the "Manage Users" use case is shown in Figure 3.6.

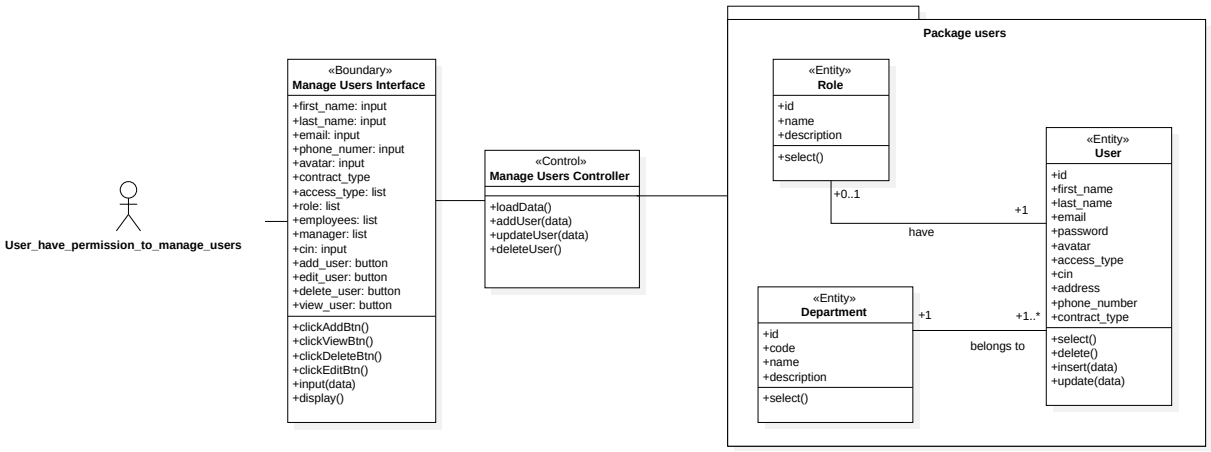


Figure 3.6: "Manage Users" Class diagram

Sequence diagram for the "Manage Users" use case is shown in Figure 3.7.

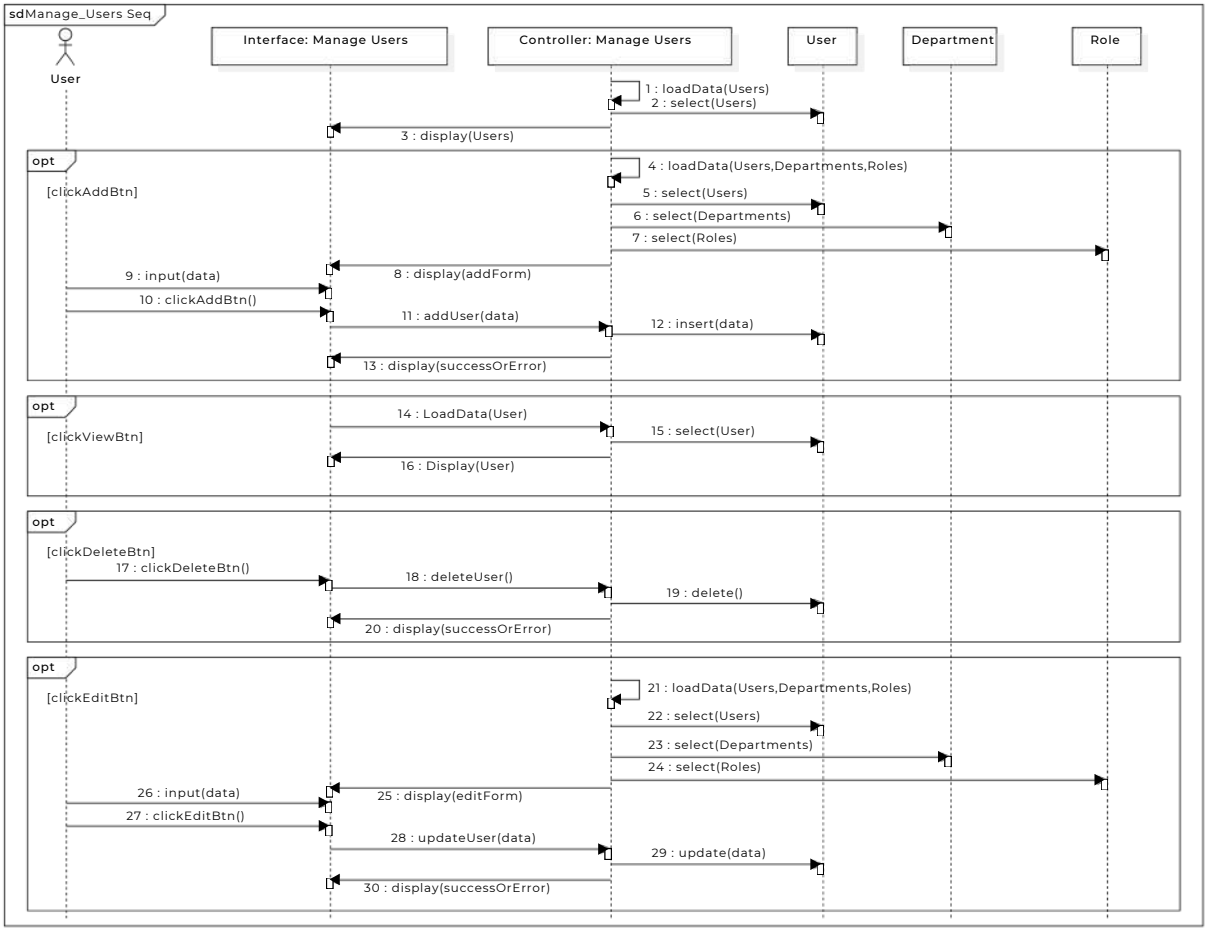


Figure 3.7: ”Manage Users” Sequence diagram

3.4.3 “Manage Departments” Use Case Design

Class diagram for the ”Manage Departments” use case is shown in Figure 3.8.

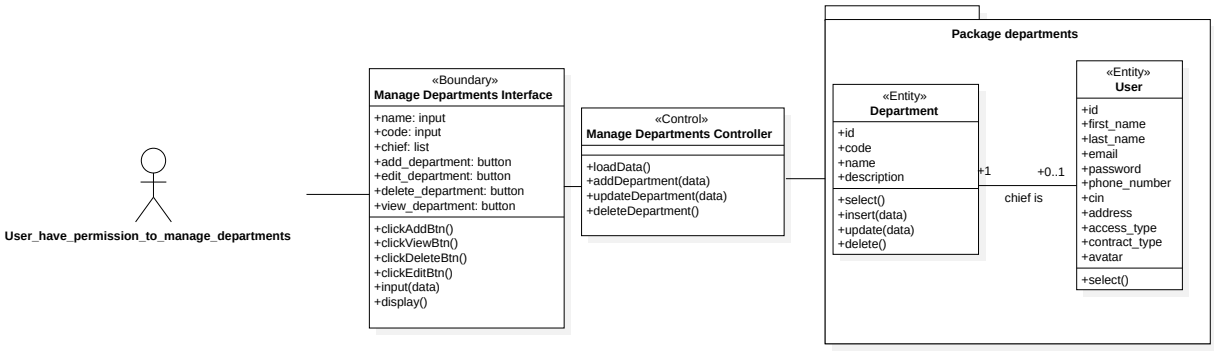


Figure 3.8: ”Manage Departments” Class diagram

Sequence diagram for the ”Manage Departments” use case is shown in Figure 3.9.

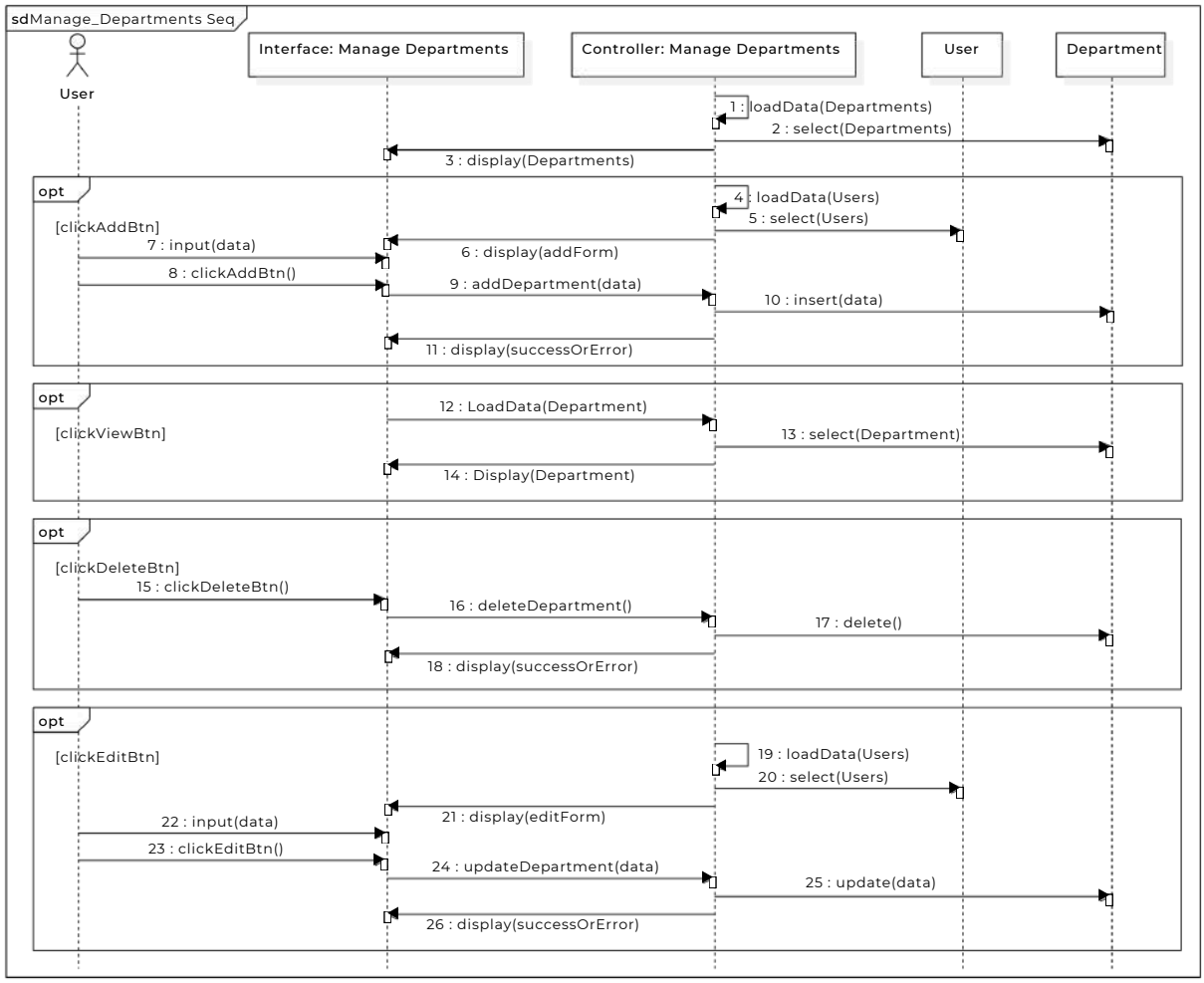


Figure 3.9: ”Manage Departments” Sequence diagram

3.5 Implementation of sprint 0

In this section, we present the interfaces of each use case accompanied by a short description to explain how users interact with it.

3.5.1 Realization of the “Authenticate” use case

In the authentication interface, the user types his email address and password, then clicks the ”login” button to access his account, as shown in Figure 3.10.

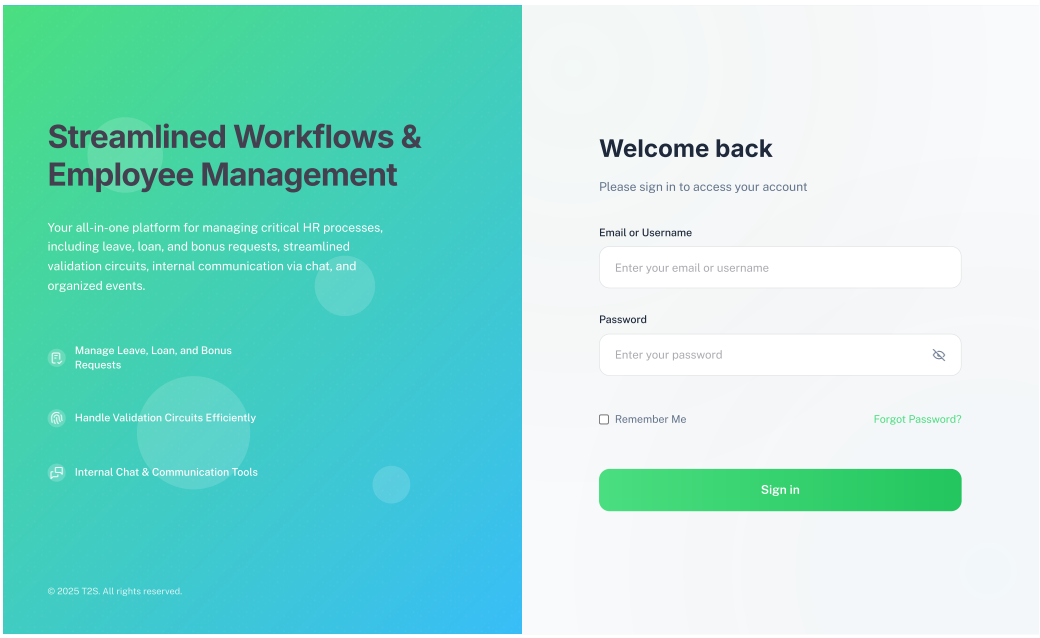


Figure 3.10: "Authenticate" Interface

In the forgot password interface, the user types his email address so a password reset link will be sent to him so he could securely reset his password in case he have forgotten it, as shown in Figure 3.11.

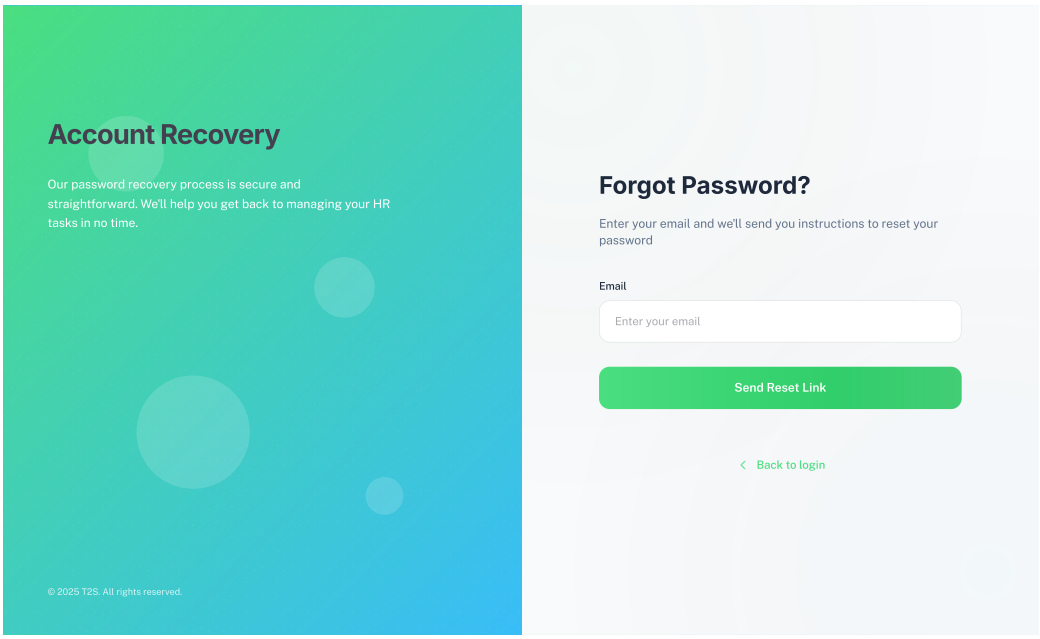


Figure 3.11: "Forgot Password" Interface

In the reset password interface which the user can access via the link sent to his email, the user types his new password and confirms it then clicks on "set new password" button, as shown in Figure 3.12.

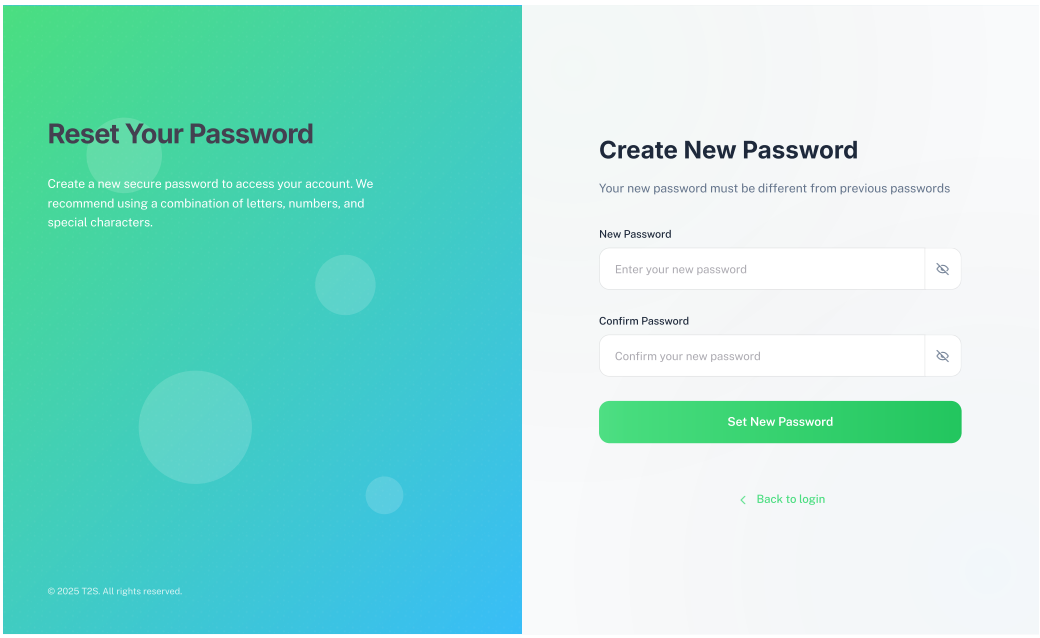


Figure 3.12: ”Forgot Password” Interface

3.5.2 Realization of the “Manage Users” use case

In this interface, the user can view the list of users that exist, as shown in Figure 3.13.

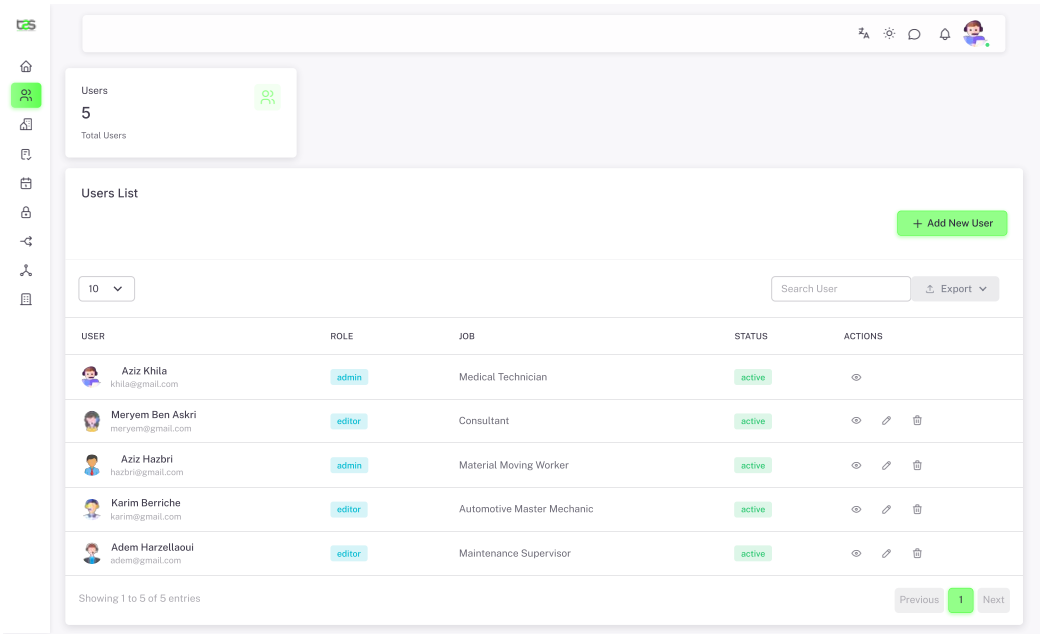


Figure 3.13: ”Manage Users” Interface

In this interface, the user can view the data related to a specific user, as shown in Figure 3.14.

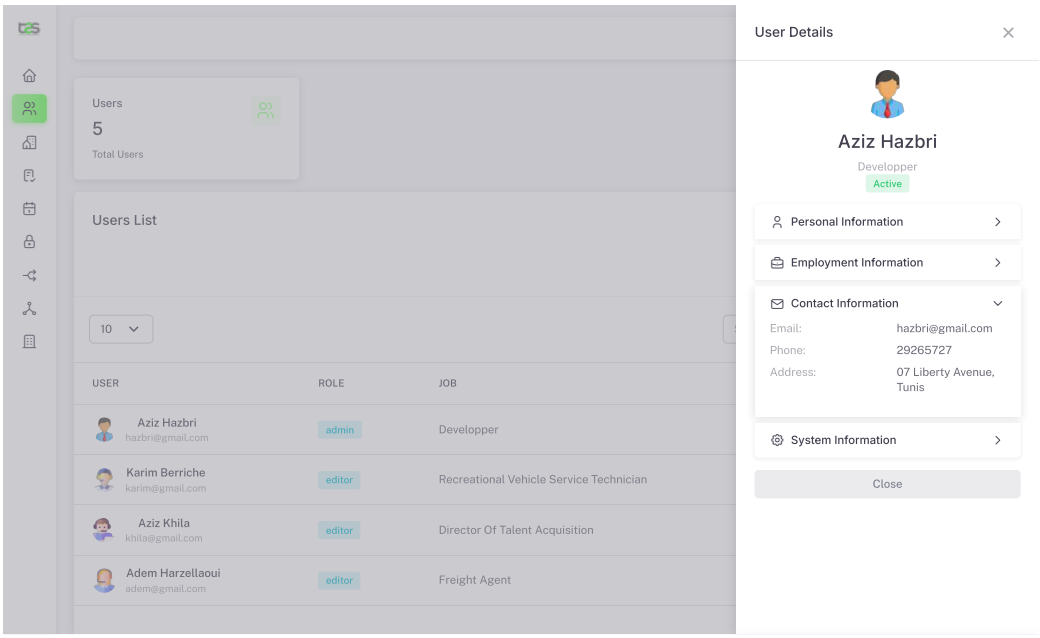


Figure 3.14: "View User" Interface

In this interface, the user enters the new user's data (first name, last name ...) then clicks on "Submit" so the new user is created, as shown in Figure 3.15.

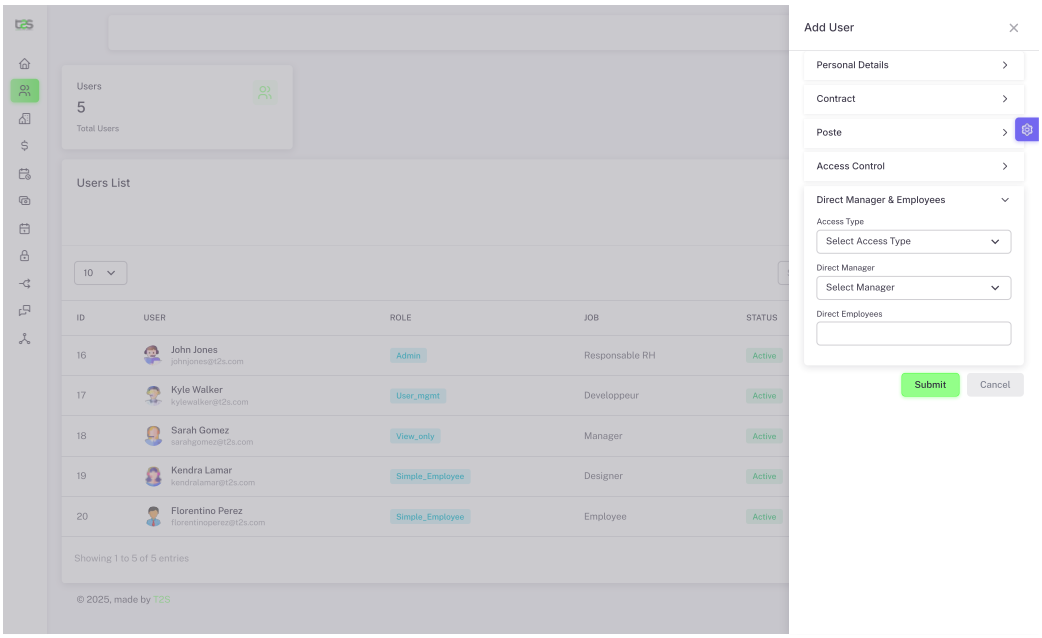


Figure 3.15: "Add User" Interface

The user clicks on the delete button, a confirmation model appears, then the user clicks on the "Yes, Delete" button so the targeted user is deleted, as shown in Figure 3.16.

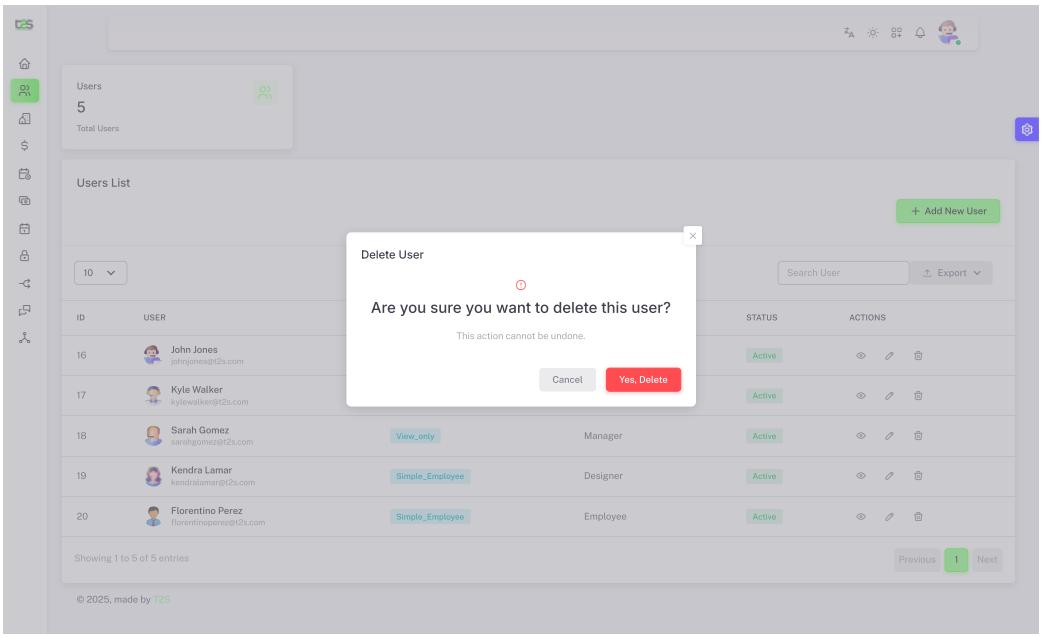


Figure 3.16: "Delete User" Interface

In this interface, the user updates the desired data, then clicks on the "Save Changes" button so the targeted user's data is updated, as shown in Figure 3.17.

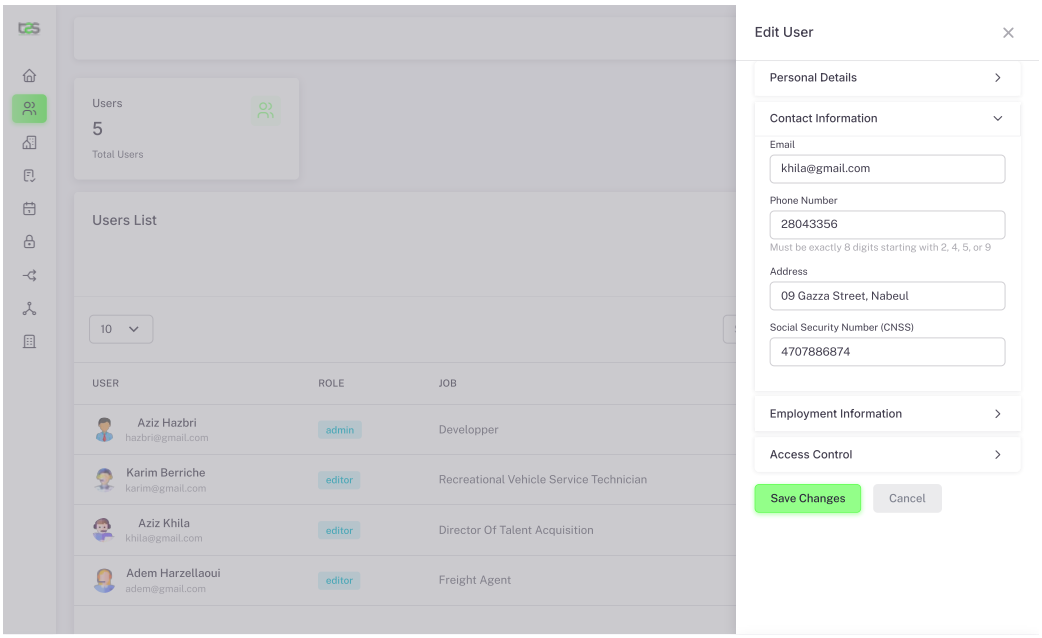


Figure 3.17: "Edit User" Interface

3.5.3 Realization of the "Manage Departments" use case

In this interface, the user can view the list of departments that exist, as shown in Figure 3.18.

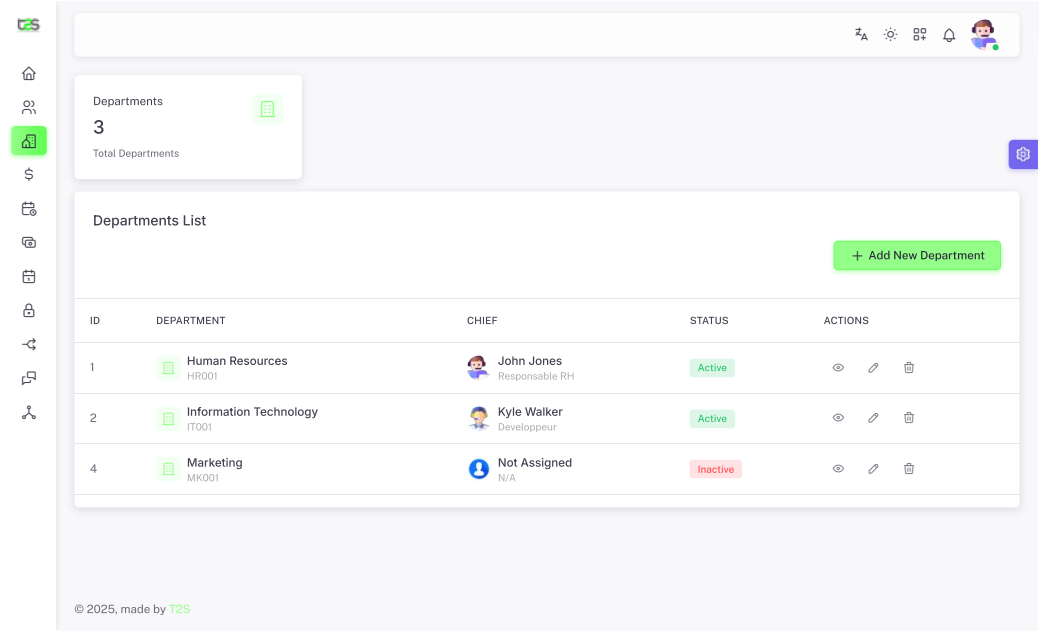


Figure 3.18: "Manage Departments" Interface

In this interface, the user can view the data related to a specific department, as shown in Figure 3.19.

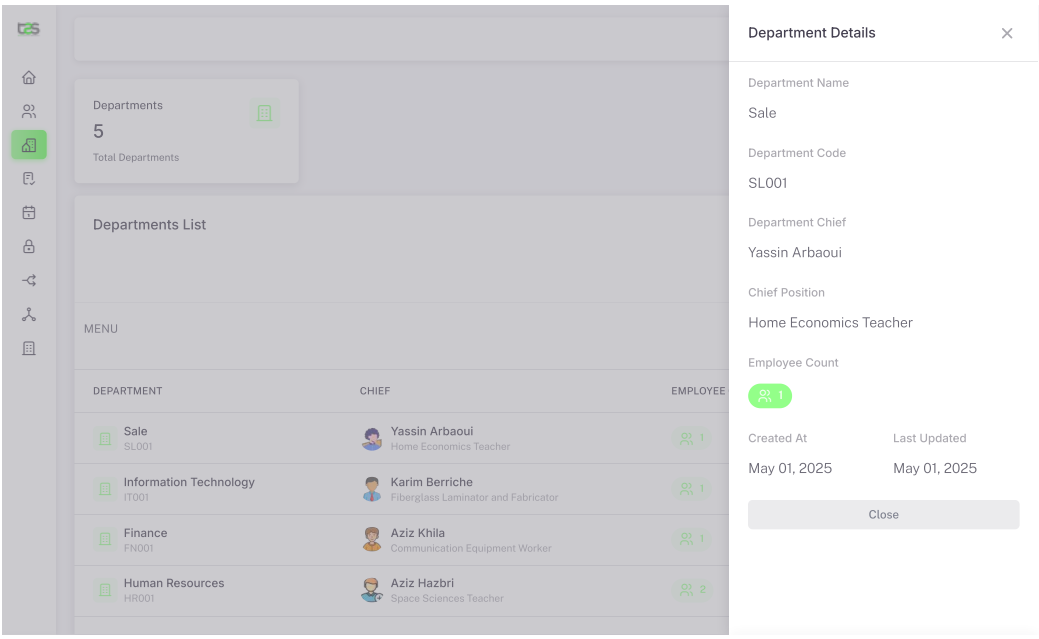


Figure 3.19: "View Department" Interface

In this interface, the user enters the desired data, then clicks on the "Submit" button so that the new department is created, as shown in Figure 3.20.

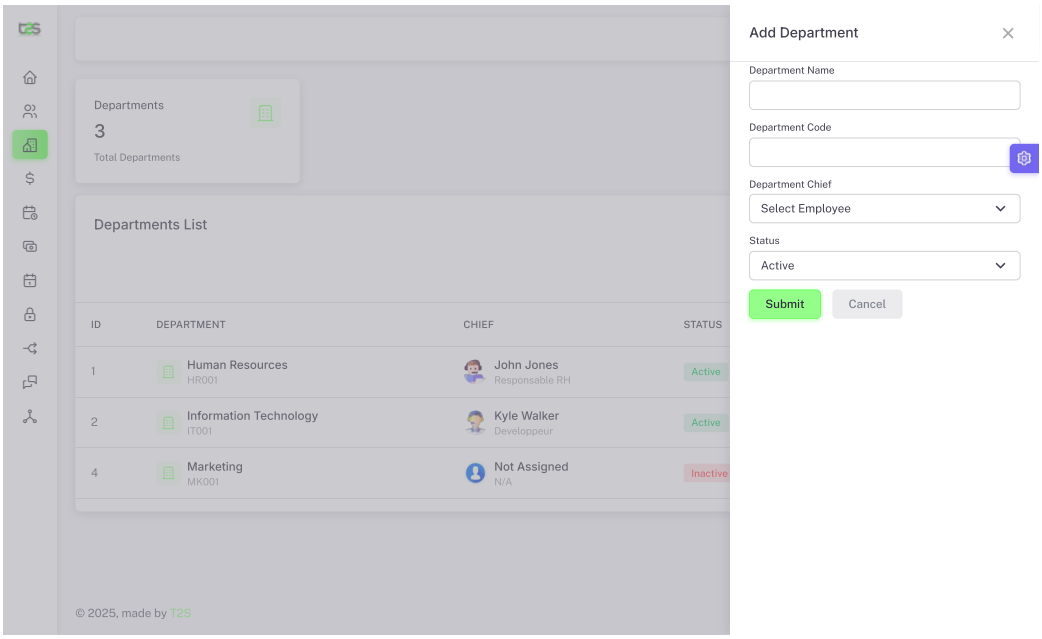


Figure 3.20: ”Add Department” Interface

In this interface, after clicking on the delete button of the department, confirmation modal will show up to confirm the deletion of the department, as shown in Figure 3.21.

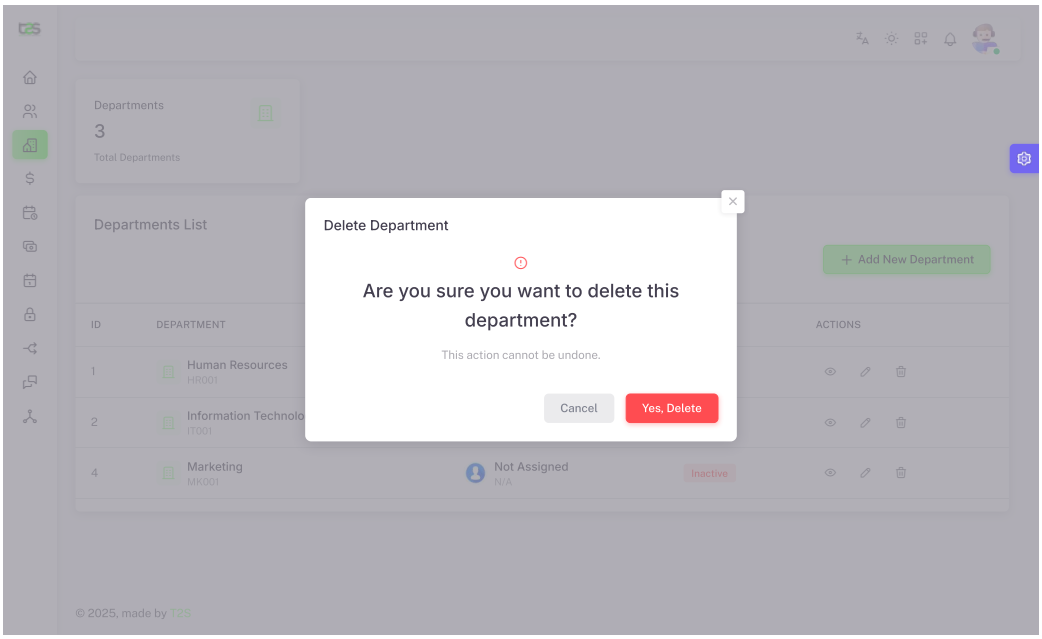


Figure 3.21: ”Delete Department” Interface

In this interface, the user updates the desired data, then clicks on the ”Save Changes” button so the targeted department data is updated, as shown in Figure 3.22.

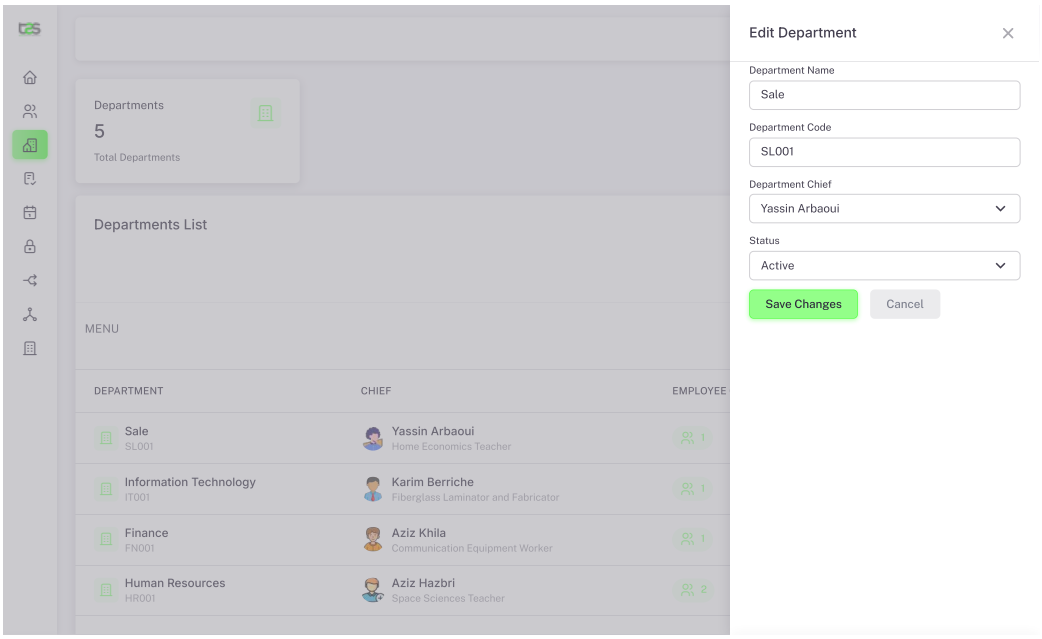


Figure 3.22: "Edit Department" Interface

3.6 Conclusion

In this chapter, we present the refinement, the design, and the interfaces of use cases of sprint 0.

Chapter 4

Sprint 1 : Structure and Access Control

4.1 Introduction

In this chapter, we present the use cases for Sprint 1. The study of this sprint includes refinement, design, and implementation.

4.2 Identifying the sprint 1 backlog

The table 4.1 contains the backlog elements that are executed during sprint 1:

Table 4.1: Product backlog sprint 1

User Story	Priority	Estimation	Sprint
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete roles.	2	High	Sprint 1
As a user, I can view the enterprise hierarchical structure.	2	Average	Sprint 1
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete extra validation levels.	2	High	Sprint 1

4.3 Refinement of sprint 1

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

4.3.1 Refinement of the ”Manage Roles” use case

The figure 4.1 showcases the use case.

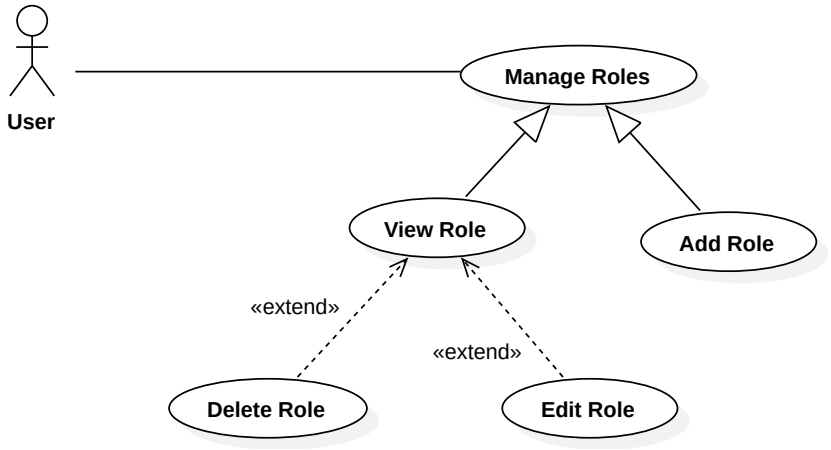


Figure 4.1: “Manage Roles” use case diagram

The table 4.2 elaborates on this user story with textual description.

Table 4.2: Detailed description of the ”Manage Roles” use case

Use Case	« Manage Roles »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor have permissions to manage roles.
Post-condition	Managed roles.
Principal scenario	- System displays role management interface. - Actor clicks on « add new role » button then chooses the desired permissions for that role then clicks on « create role » button. - Actor clicks on « Edit » button and makes needed changes then clicks on « update role » button . - Actor clicks on « Delete » button and confirms deletion in the popped up window .
Exception	The system doesn’t create the role if a similar role exists.

4.3.2 Refinement of the ”View Enterprise Structure” use case

Each company have its own structure, which makes the validation circuit fully configurable and adaptable to each company, whatever its structure.

The figure 4.2 showcases the use case.

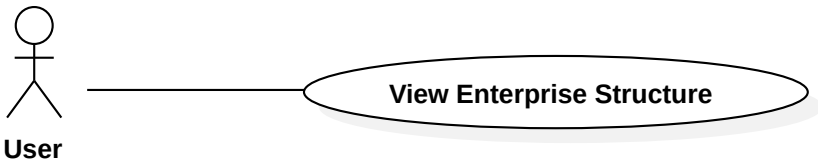


Figure 4.2: “View Enterprise Structure” use case diagram

The table 4.3 elaborates on this user story with textual description.

Table 4.3: Detailed description of the ”View Enterprise Structure” use case

Use Case	« View Enterprise Structure »
Actor(s)	User
Pre-condition	Actor authenticated
Post-condition	Enterprise structure observed.
Principal scenario	- System displays enterprise structure interface. - Actor observe the hierarchy of the enterprise which represents the base of the validation circuit.
Exception	No exception

4.3.3 Refinement of the ”Manage Extra Validation Levels” use case

Important to note that every request in the system passes through the validation circuit, we use a hierarchical validation circuit which means that every request starts with the immediate manager of the requester. Once approved, it moves up to the next manager until it reaches the top level in the hierarchy. Finally, the request reaches the extra validation levels only after being approved by all the hierarchy levels.

The figure 4.3 showcases the use case.

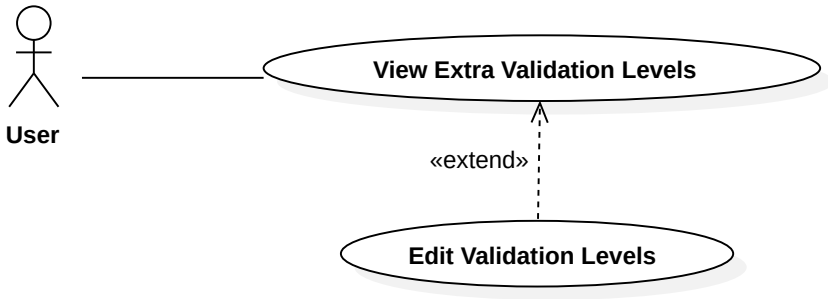


Figure 4.3: “Manage Extra Validation Levels” use case diagram

The table 4.4 elaborates on this user story with textual description.

Table 4.4: Detailed description of the ”Manage Extra Validation Levels” use case

Use Case	« Manage Extra Validation Levels »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor have permissions to manage extra validation levels.
Post-condition	Managed extra validation levels.
Principal scenario	- System displays extra validation management interface. - Actor clicks on « add level » button and chooses the user to be a validator at that level then clicks on « save changes » button. - Actor updates the levels that he wants to be updated then clicks on « save changes » button . - Actor clicks on « Delete » button to get rid of a level .
Exception	You cannot have duplicate users in different levels.

4.4 Design of sprint 1

We present now the design of the sprint 1 use cases.

4.4.1 “Manage Roles” Use Case Design

Class diagram for the ”Manage Roles” use case is shown in Figure 4.4.

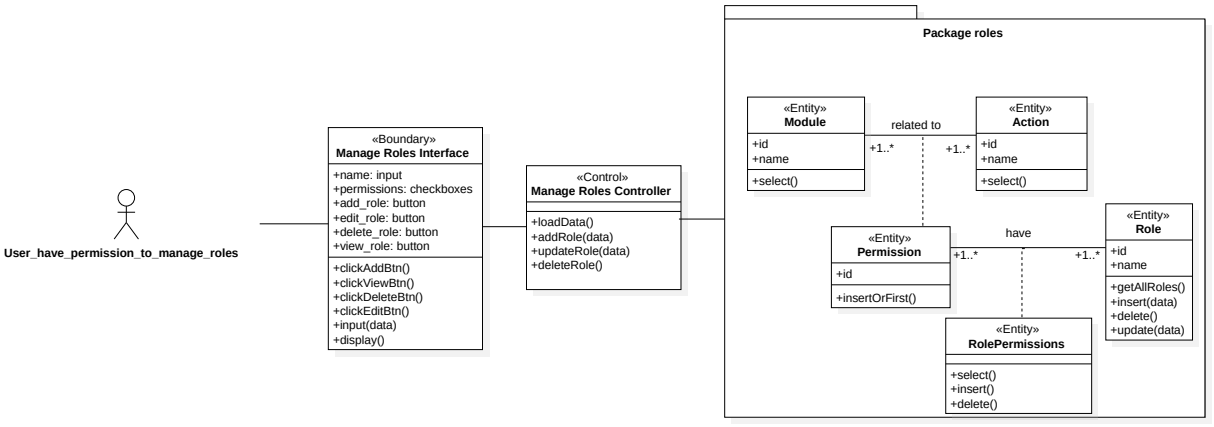


Figure 4.4: ”Manage Roles” Class diagram

Sequence diagram for the ”Manage Roles” use case is shown in Figure 4.5.

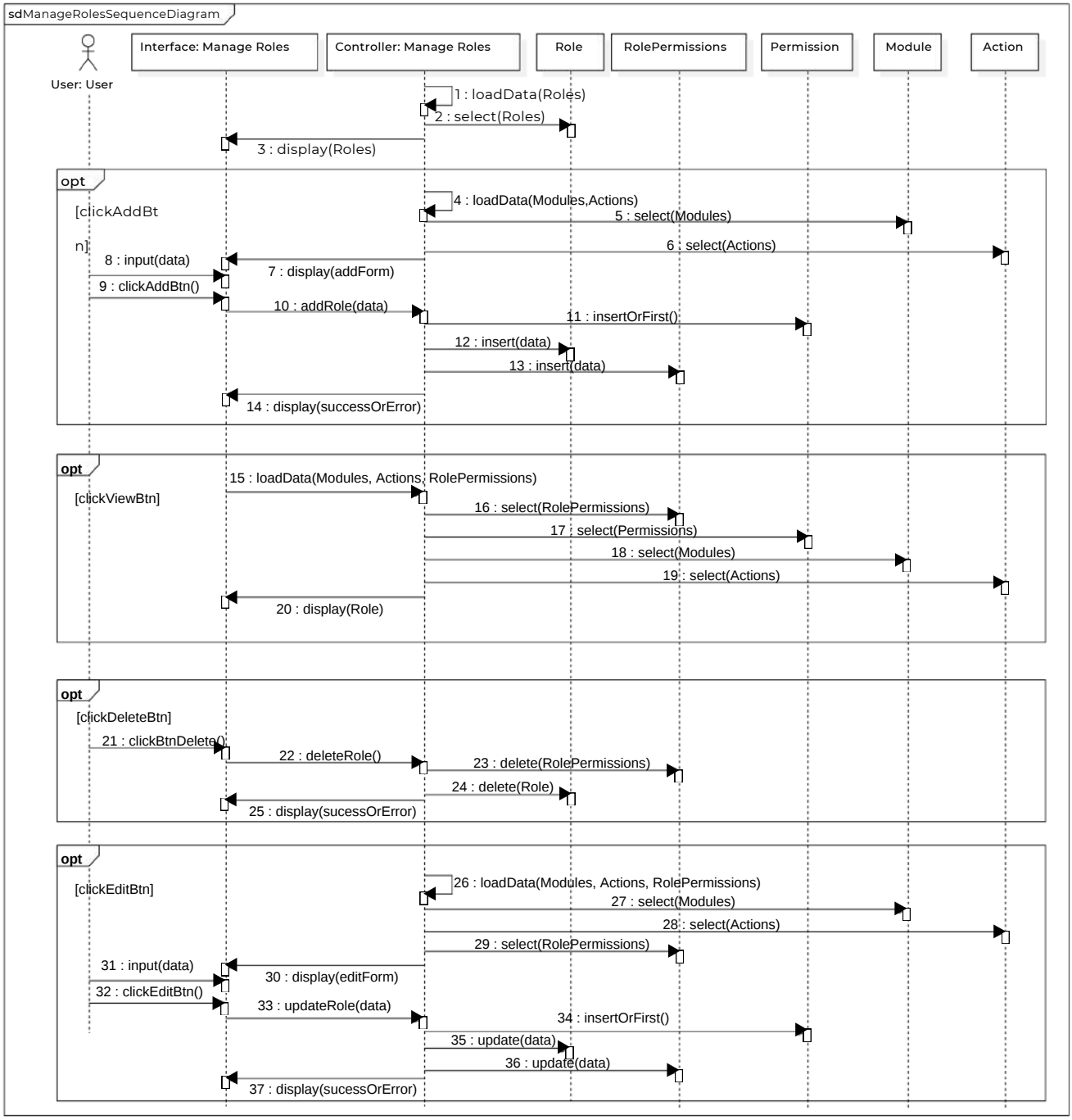


Figure 4.5: ”Manage Roles” Sequence diagram

4.4.2 “View Enterprise Structure” Use Case Design

Class diagram for the “View Enterprise Structure” use case is shown in Figure 4.6.

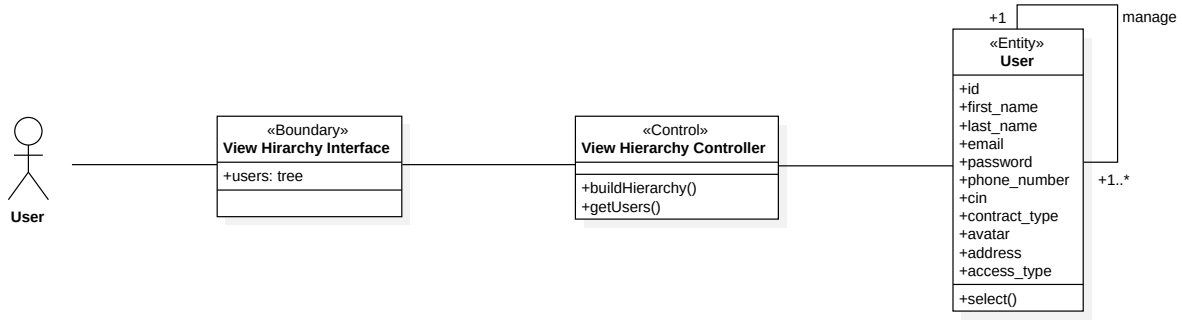


Figure 4.6: "View Enterprise Structure" Class diagram

Sequence diagram for the "View Enterprise Structure" use case is shown in Figure 4.7.

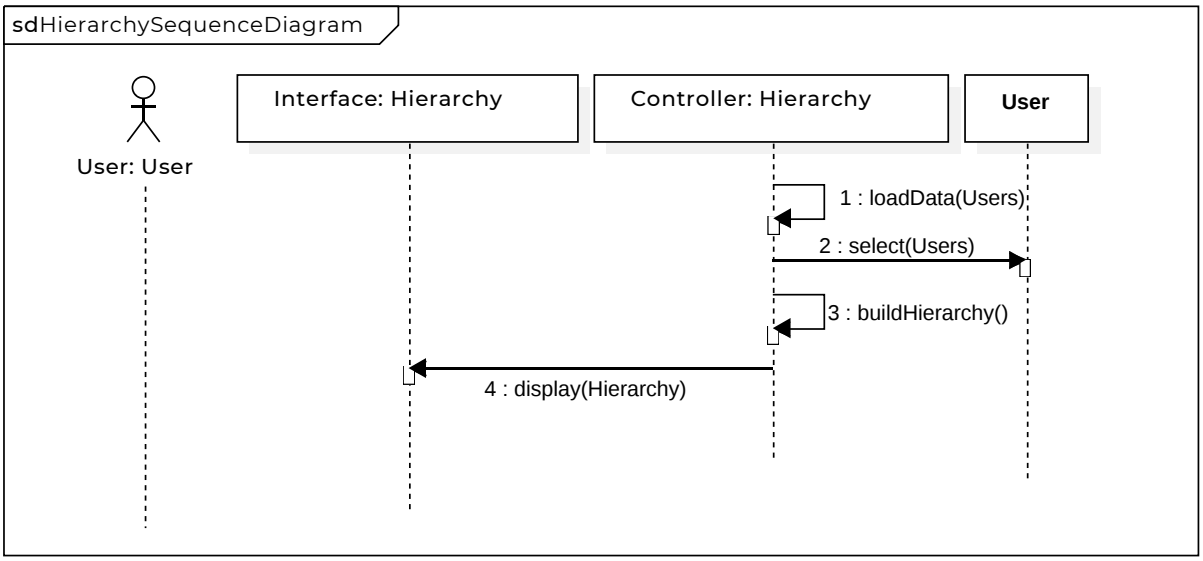


Figure 4.7: "View Enterprise Structure" Sequence diagram

4.4.3 "Manage Extra Validation Levels" Use Case Design

Class diagram for the "Manage Extra Validation Levels" use case is shown in in Figure 4.8.

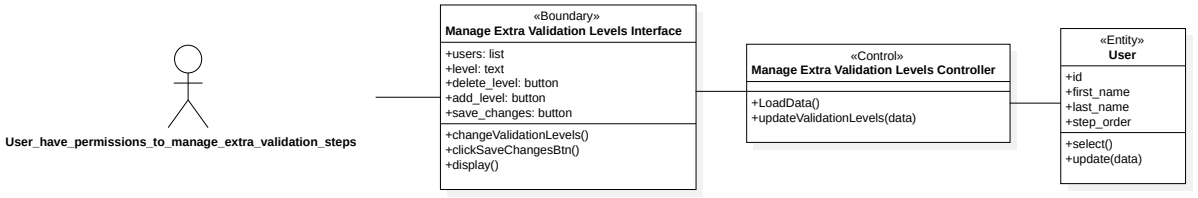


Figure 4.8: "Manage Extra Validation Levels" Class diagram

Sequence diagram for the "Manage Extra Validation Levels" use case is shown in Figure 4.9.

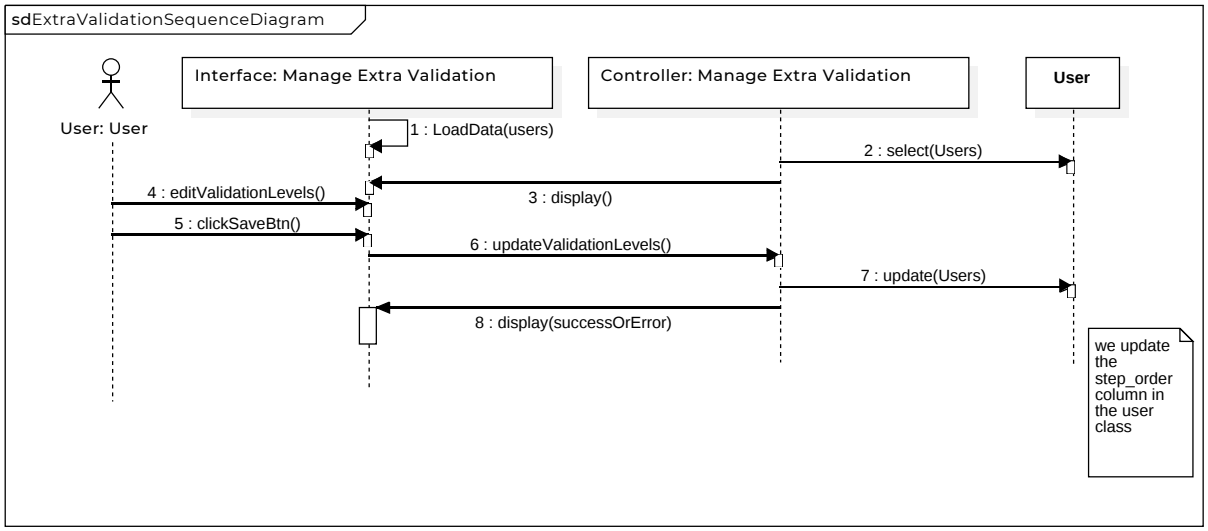


Figure 4.9: "Manage Extra Validation Levels" Sequence diagram

4.5 Implementation of sprint 1

In this section, we present the interfaces of each use case of this second sprint accompanied by a short description to explain how users interact with it.

4.5.1 Realization of the "Manage Roles" use case

In this interface, the user can view the existing roles in the system, as shown in Figure 4.10.

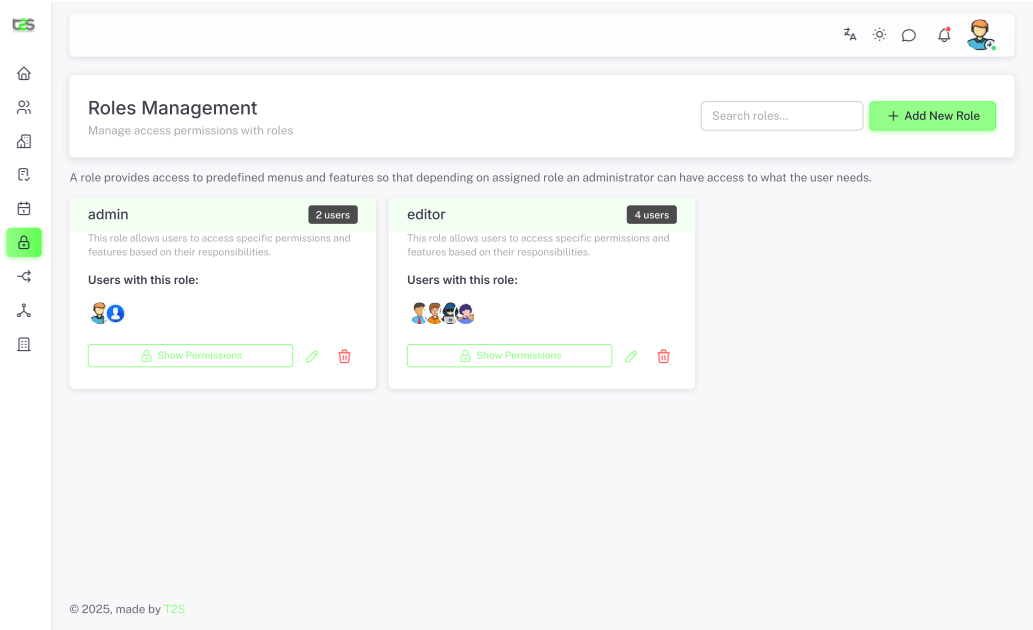


Figure 4.10: "Manage Roles" Interface

In this interface, the user can view the permissions related to a specific role, as shown in Figure 4.11.

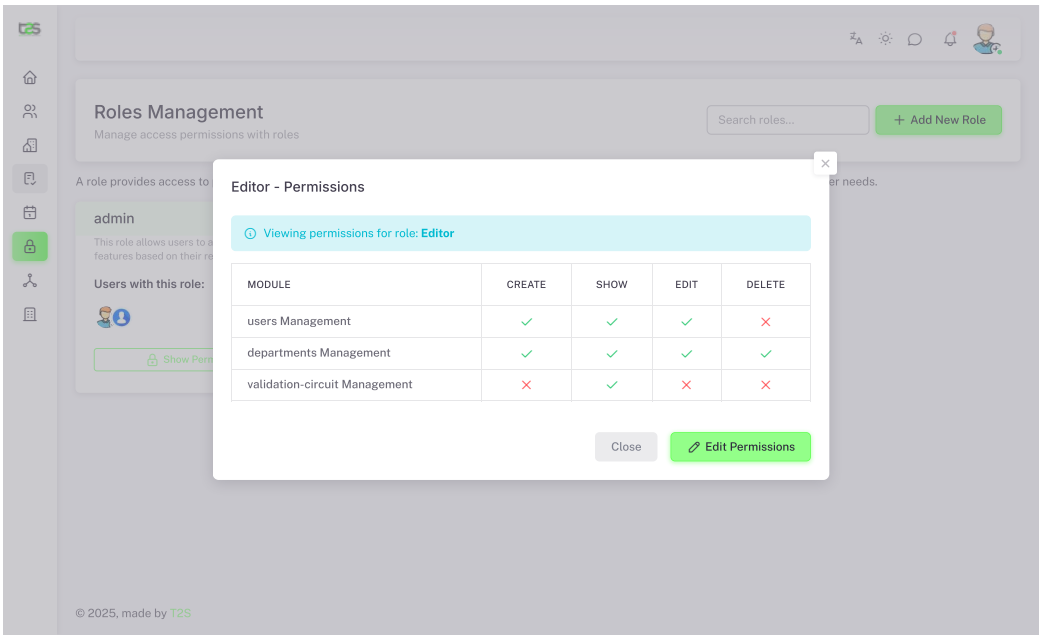


Figure 4.11: "View role permissions" Interface

In this interface, the user types the new role name and chooses the desired permissions to affect to this new role then clicks on "create role" button, as shown in Figure 4.12.

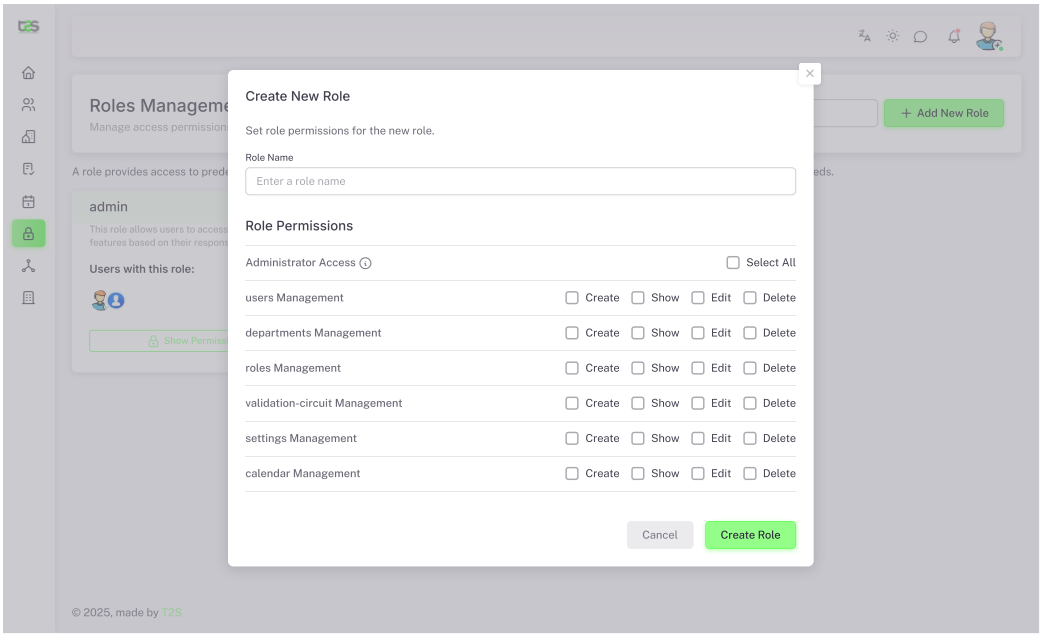


Figure 4.12: "Add Role" Interface

In this interface, after clicking on the delete button of the role, a confirmation modal will show up to confirm the deletion of that role, as shown in Figure 4.13.

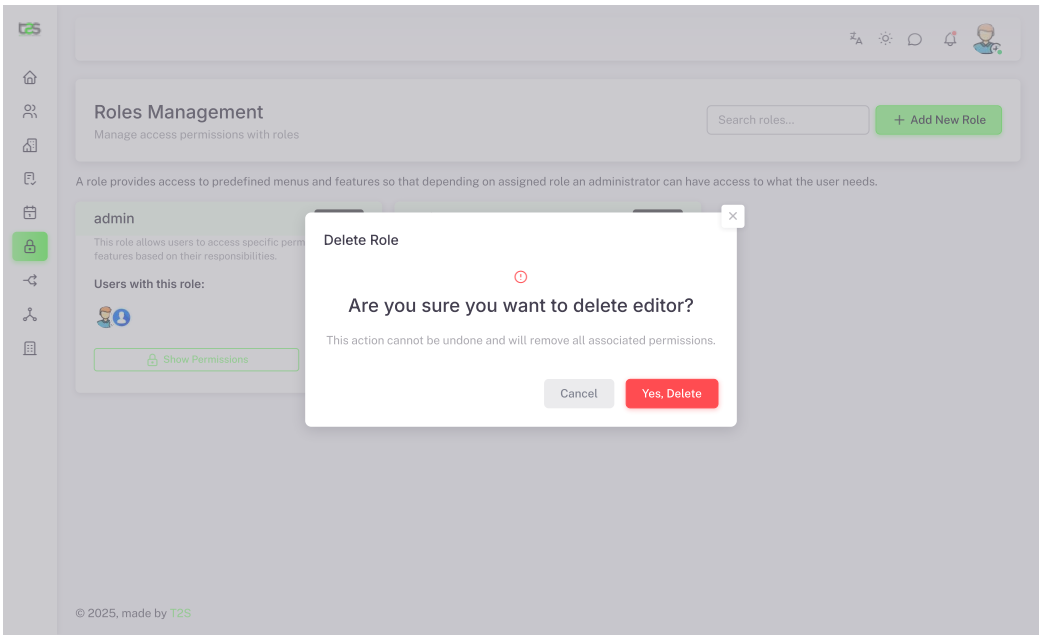


Figure 4.13: "Delete Role" Interface

In this interface, the user updates the desired role permissions, then clicks on the "Update Role" button so the targeted role is updated, as shown in Figure 4.14.

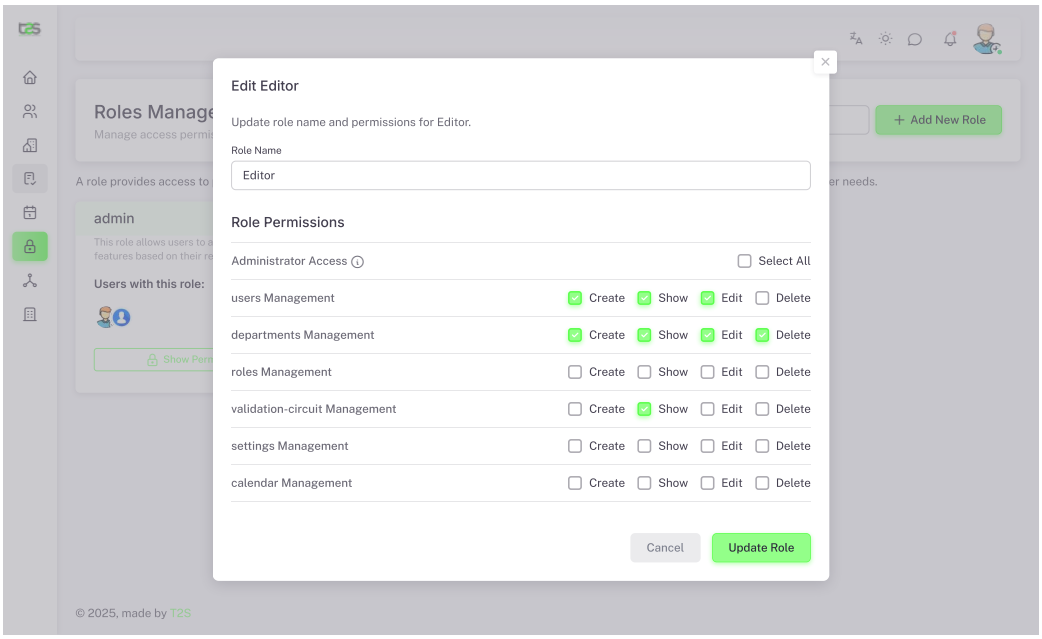


Figure 4.14: "Edit Role" Interface

4.5.2 Realization of the "View Enterprise Structure" use case

In this interface, the user can view the hierarchical enterprise structure which represents the validation circuit, as shown in Figure 4.15.

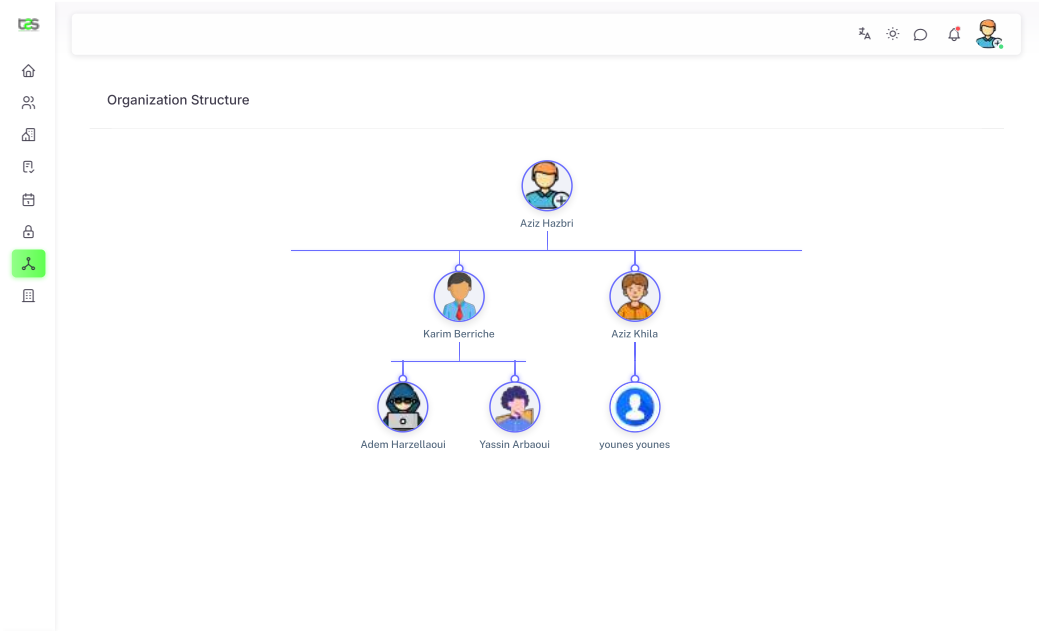


Figure 4.15: "View Enterprise Structure" Interface

4.5.3 Realization of the “Manage Extra Validation Levels” use case

In this interface, the user can set levels (each level contains a user) that will be added to the hierarchical validation circuit, as shown in Figure 4.16.

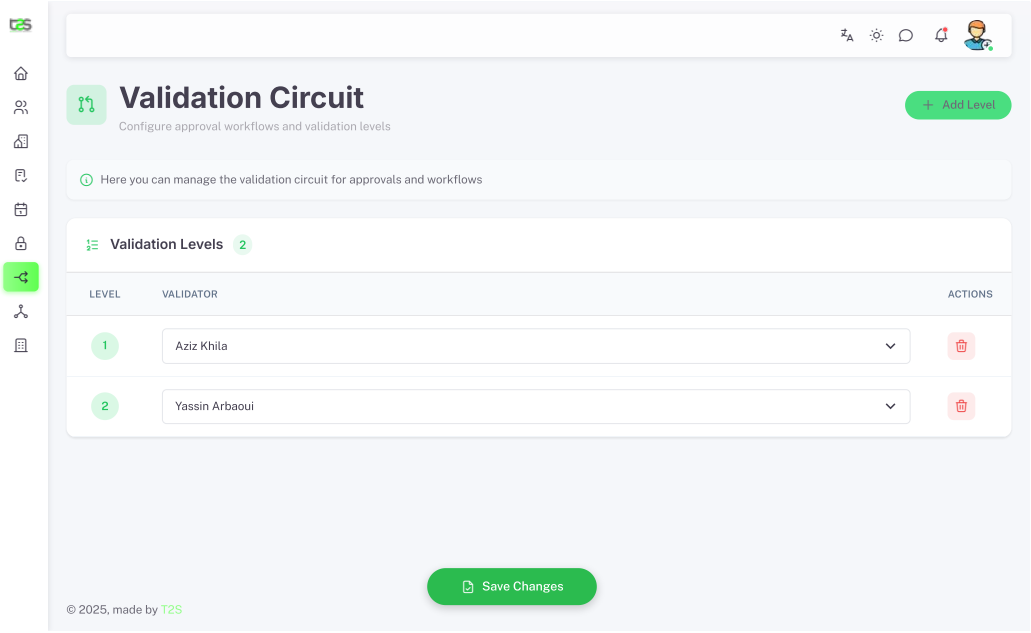


Figure 4.16: ”Manage Extra Validation Levels” Interface

4.6 Conclusion

In this chapter, we present the refinement, the design, and the interfaces of use cases of sprint 1.

Chapter 5

Sprint 2 : HR Request Management

5.1 Introduction

In this chapter, we present the use cases for Sprint 2. The study of this sprint includes refinement, design, and implementation.

5.2 Identifying the sprint 2 backlog

The table 5.1 contains the backlog elements that are executed during sprint 2:

Table 5.1: Product backlog sprint 2

User Story	Priority	Estimation	Sprint
As a user, I can create (apply) or/and view or/and edit or/and delete my leave request.	3	High	Sprint 2
As a user, I can create (apply) or/and view or/and edit or/and delete my loan request.	3	High	Sprint 2
As a user, I can create (assign) or/and view or/and edit or/and delete a salary bonus to my subordinate employees.	3	High	Sprint 2
As a user, I can view the bonuses that I received.	3	High	Sprint 2
As a user, I can manage (view or/and approve/reject) leave requests of my subordinates employees.	3	High	Sprint 2
As a user, I can manage (view or/and approve/reject) loan requests of my subordinates employees.	3	High	Sprint 2
As a user, I can manage (view or/and approve/reject) assigned bonus requests to my subordinates employees.	3	High	Sprint 2

5.3 Refinement of sprint 2

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

5.3.1 Refinement of the ”Manage my leave request” use case

The figure 5.1 showcases the use case.

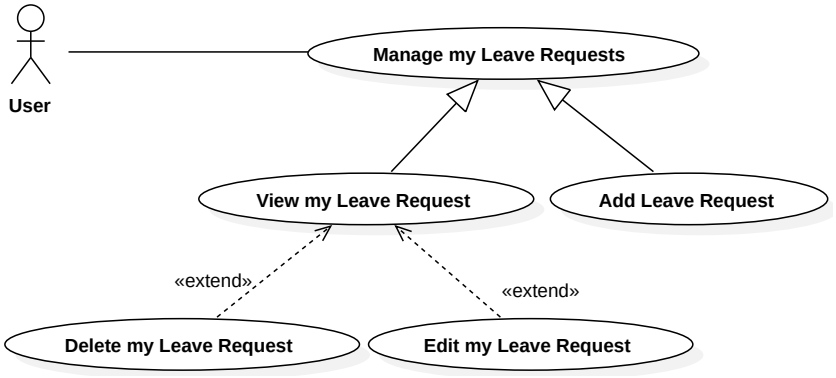


Figure 5.1: “Manage my leave request” use case diagram

The table 5.2 elaborates on this user story with textual description.

Table 5.2: Detailed description of the ”Manage my leave request” use case

Use Case	« Manage my leave request »
Actor(s)	User
Pre-condition	Actor authenticated
Post-condition	Leave request managed.
Principal scenario	- System displays leave request modal. - Actor enters information of the wanted leave request. - Actor clicks on « Submit Request » button to create new leave request. - Actor clicks on « View Request » button to see full details of an existing request. - Actor clicks on « Edit Request » button to update an existing request. - Actor clicks on « Delete Request » button to delete an existing request.
Exception	The system displays an error message if the updated data is incorrect.

5.3.2 Refinement of the ”Manage my loan request” use case

The figure 5.2 showcases the use case.

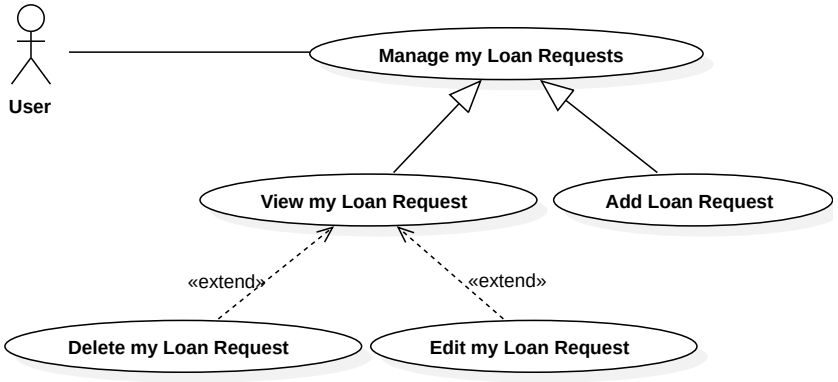


Figure 5.2: “Manage my loan request” use case diagram

The table 5.3 elaborates on this user story with textual description.

Table 5.3: Detailed description of the ”Manage my loan request” use case

Use Case	« Manage my loan request »
Actor(s)	User
Pre-condition	Actor authenticated
Post-condition	Loan request managed.
Principal scenario	- System displays loan request modal. - Actor enters information of the wanted loan request. - Actor clicks on « Submit Request » button to create new loan request. - Actor clicks on « View Request » button to see full details of an existing request. - Actor clicks on « Edit Request » button to update an existing request. - Actor clicks on « Delete Request » button to delete an existing request.
Exception	The system displays an error message if the updated data is incorrect.

5.3.3 Refinement of the ”Manage my assigned bonus request” use case

The figure 5.3 showcases the use case.

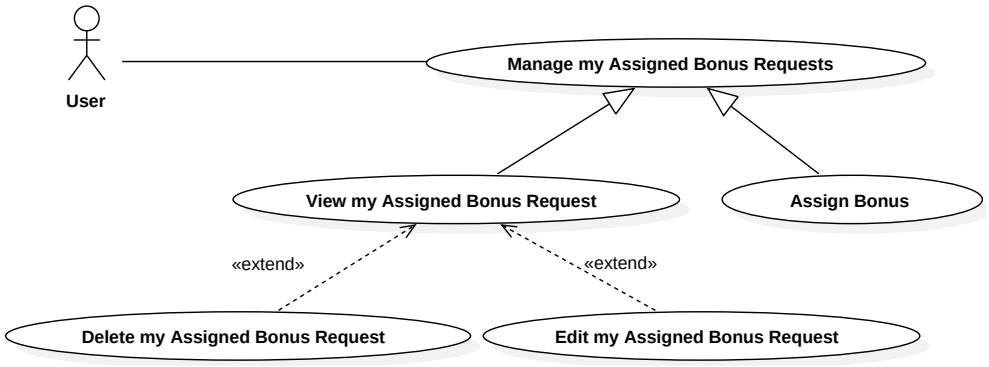


Figure 5.3: “Manage my assigned bonus request” use case diagram

The table 5.4 elaborates on this user story with textual description.

Table 5.4: Detailed description of the ”Manage my assigned bonus request” use case

Use Case	« Manage my assigned bonus request »
Actor(s)	User
Pre-condition	Actor authenticated
Post-condition	Assigned bonus request managed.
Principal scenario	- System displays bonus request modal. - Actor enters information of the wanted bonus request. - Actor clicks on « Submit Bonus » button to create new bonus request. - Actor clicks on « View Request » button to see full details of an existing request. - Actor clicks on « Edit Request » button to update an existing request. - Actor clicks on « Delete Request » button to delete an existing request.
Exception	The system displays an error message if the updated date is incorrect.

5.3.4 Refinement of the ”View my received bonus” use case

The figure 5.4 showcases the use case.

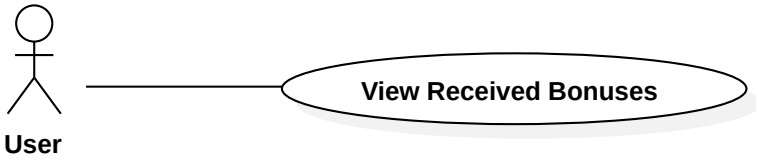


Figure 5.4: “View my received bonus” use case diagram

The table 5.5 elaborates on this user story with textual description.

Table 5.5: Detailed description of the ”View my received bonus” use case

Use Case	« View my received bonus »
Actor(s)	User
Pre-condition	Actor authenticated
Post-condition	Received bonus consulted.
Principal scenario	- System displays bonus request modal. - Actor consults the bonuses that were affected to him. - Actor clicks on « View Request » button to see full details of a received bonus.
Exception	Server error.

5.3.5 Refinement of the ”Manage employees leave requests” use case

The figure 5.5 showcases the use case.

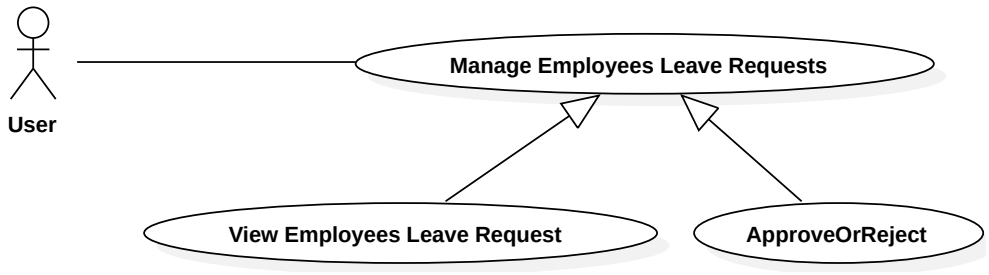


Figure 5.5: “Manage employees leave requests” use case diagram

The table 5.6 elaborates on this user story with textual description.

Table 5.6: Detailed description of the ”Manage employees leave requests” use case

Use Case	« Manage employees leave requests »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor is hierarchically superior to the requester.
Post-condition	Employee leave request managed.
Principal scenario	- System displays the employees leave request interface. - Actor clicks on « View Request » button to see full details of the pending request. - Actor clicks on « Approve » button and confirms to approve the request. - Actor clicks on « Reject » button and adds a rejection reason to reject the request.
Exception	The system displays an error message if the rejection reason area is empty.

5.3.6 Refinement of the ”Manage employees loan requests” use case

The figure 5.6 showcases the use case.

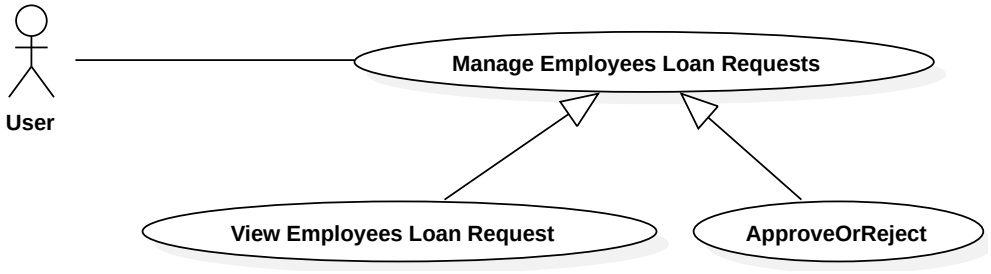


Figure 5.6: “Manage employees loan requests” use case diagram

The table 5.7 elaborates on this user story with textual description.

Table 5.7: Detailed description of the ”Manage employees loan requests” use case

Use Case	« Manage employees loan requests »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor is hierarchically superior to the requester.
Post-condition	Employee loan request managed.
Principal scenario	- System displays the employees loan request interface. - Actor clicks on « View Request » button to see full details of the pending request. - Actor clicks on « Approve » button and confirms to approve the request. - Actor clicks on « Reject » button and adds a rejection reason to reject the request.
Exception	The system displays an error message if the rejection reason area is empty.

5.3.7 Refinement of the ”Manage employees bonus requests” use case

The figure 5.7 showcases the use case.

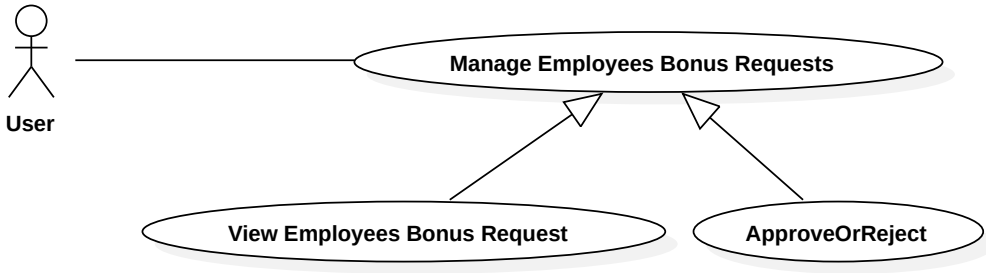


Figure 5.7: “Manage employees bonus requests” use case diagram

The table 5.8 elaborates on this user story with textual description.

Table 5.8: Detailed description of the ”Manage employees bonus requests” use case

Use Case	« Manage employees bonus requests »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor is hierarchically superior to the requester.
Post-condition	Employee bonus request managed.
Principal scenario	- System displays the employees bonus request interface. - Actor clicks on « View Request » button to see full details of the pending request. - Actor clicks on « Approve » button and confirms to approve the request. - Actor clicks on « Reject » button and adds a rejection reason to reject the request.
Exception	The system displays an error message if the rejection reason area is empty.

5.4 Design of sprint 2

We present now the design of the sprint 2 use cases.

5.4.1 “Manage my leave request” Use Case Design

Class diagram for the “Manage my leave request” use case is shown in Figure 5.8.

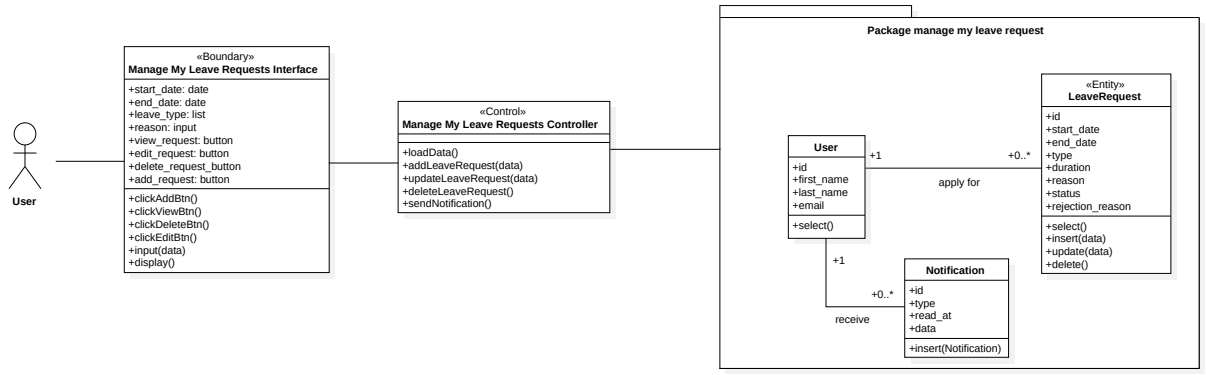


Figure 5.8: "Manage my leave request" Class diagram

Sequence diagram for the "Manage my leave request" use case is shown in Figure 5.9.

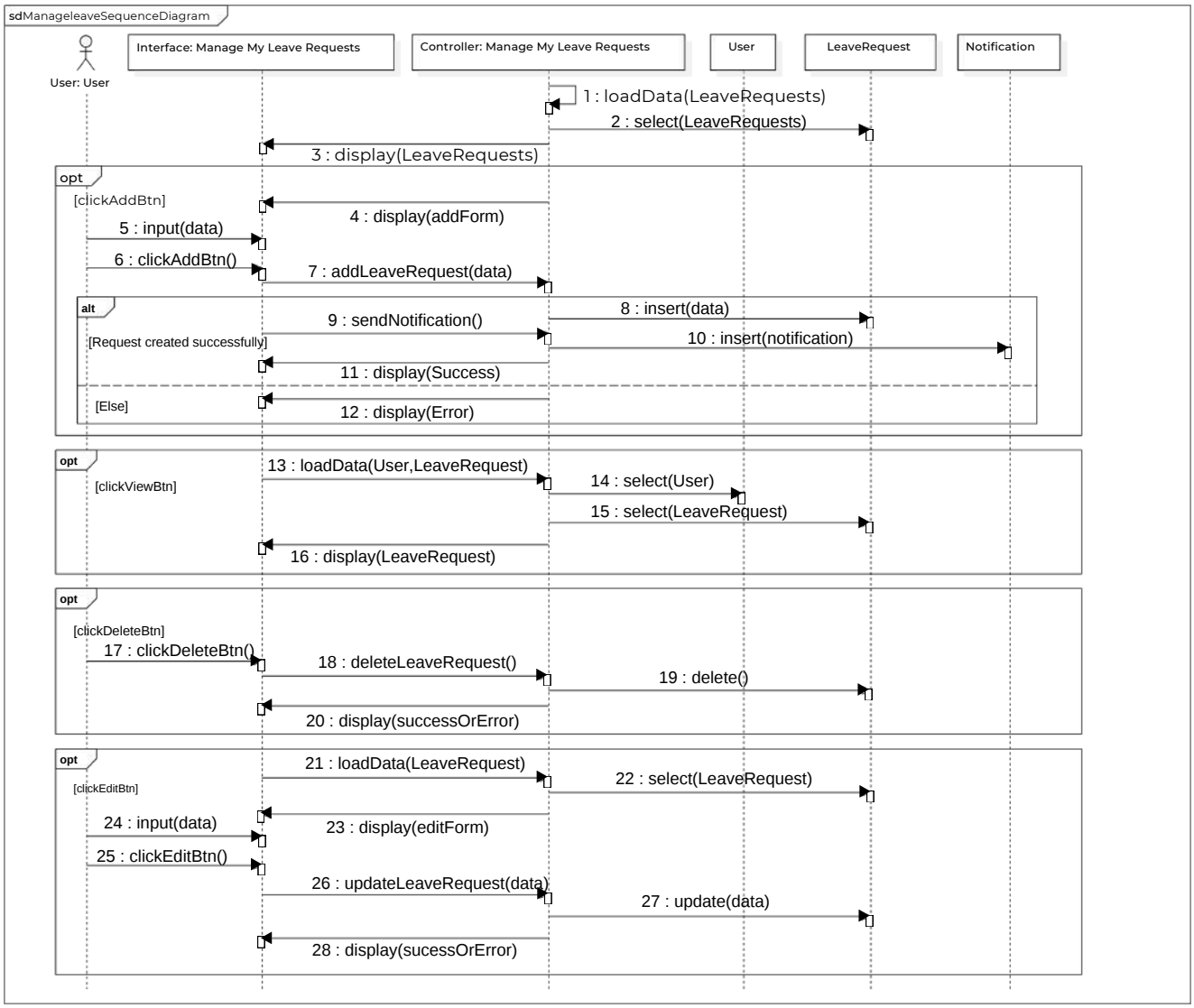


Figure 5.9: "Manage my leave request" Sequence diagram

5.4.2 "Manage my loan request" Use Case Design

Class diagram for the "Manage my loan request" use case is shown Figure 5.10.

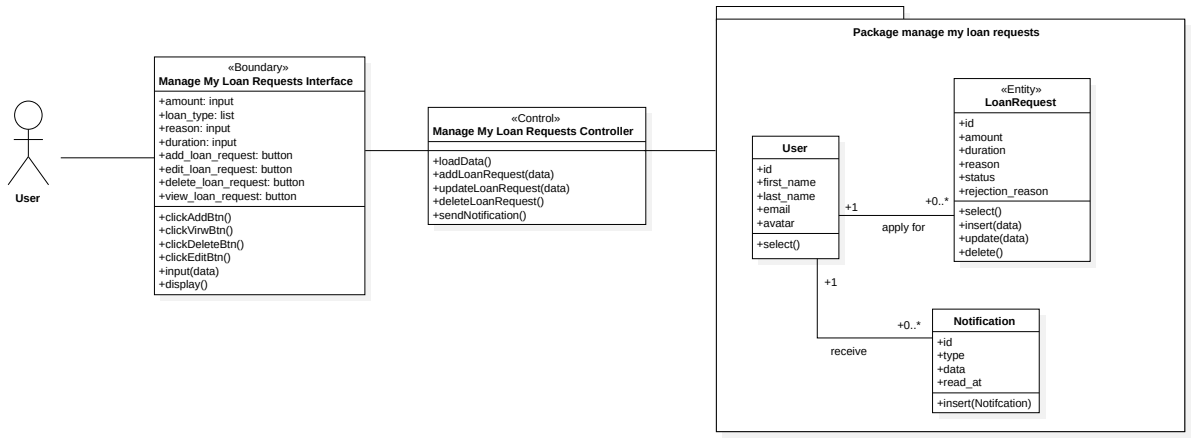


Figure 5.10: "Manage my loan request" Class diagram

Sequence diagram for the "Manage my loan request" use case is shown in Figure 5.11.

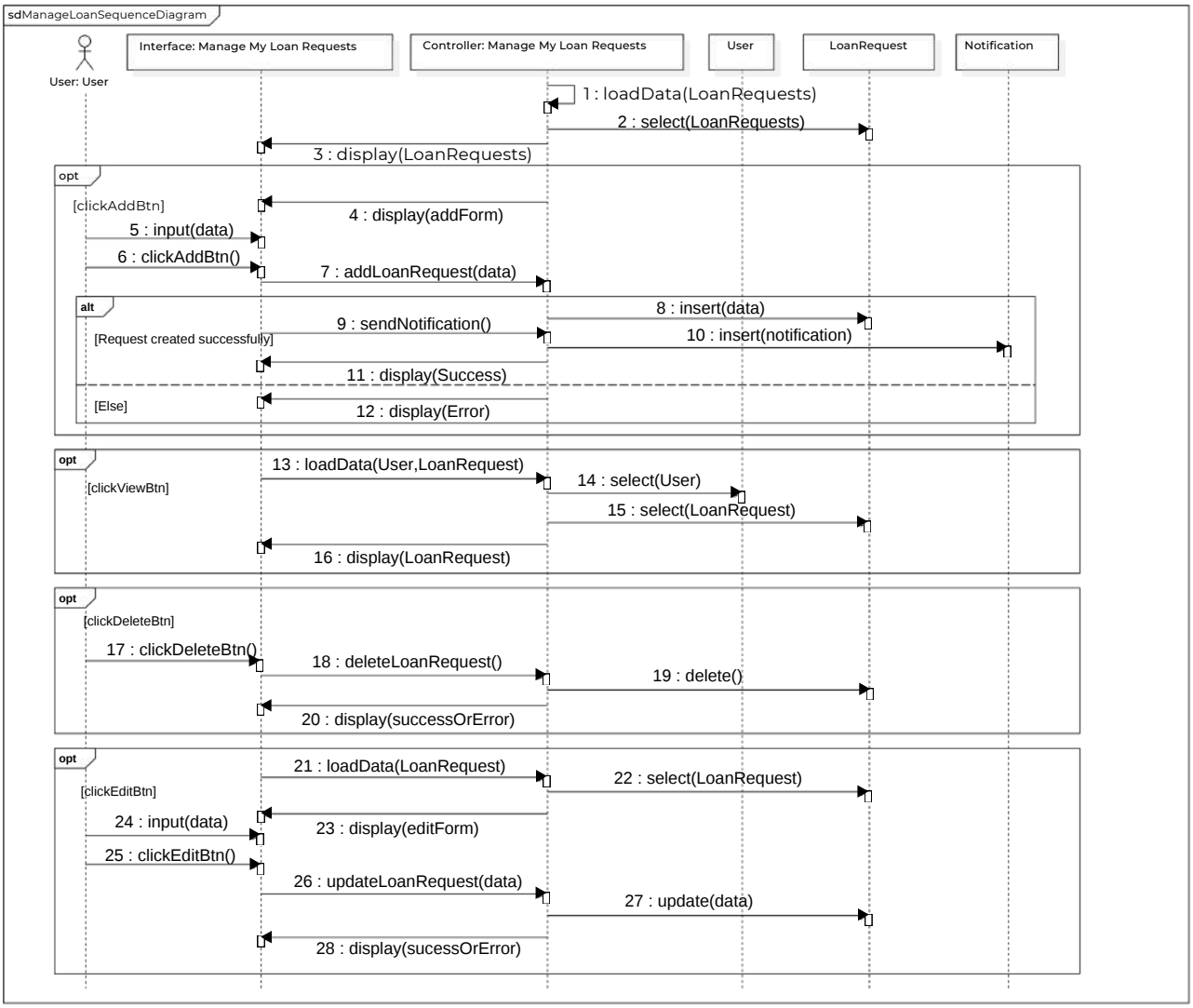


Figure 5.11: "Manage my loan request" Sequence diagram

5.4.3 "Manage my assigned bonus request" Use Case Design

Class diagram for the "Manage my assigned bonus request" use case is shown Figure 5.12.

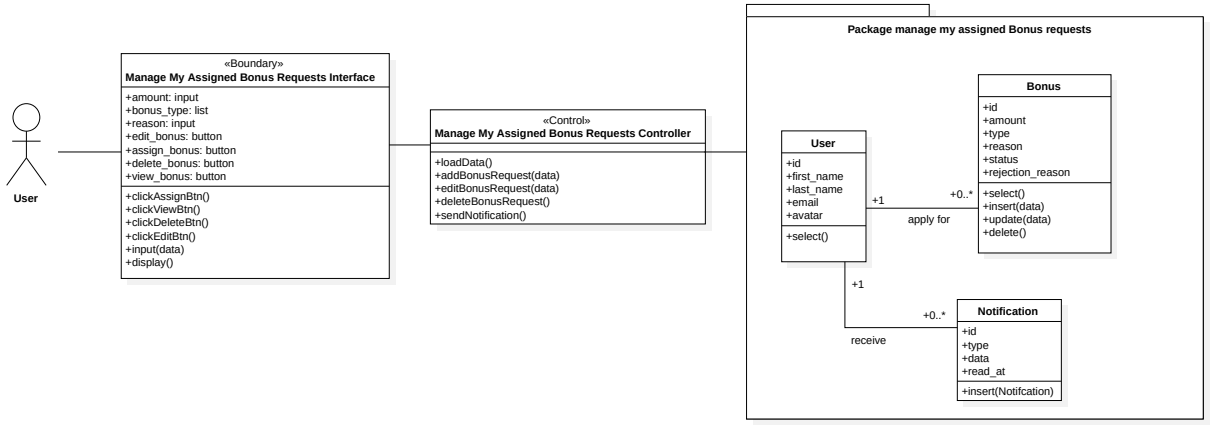


Figure 5.12: "Manage my assigned bonus request" Class diagram

Sequence diagram for the "Manage my assigned bonus request" use case is shown in Figure 5.13.

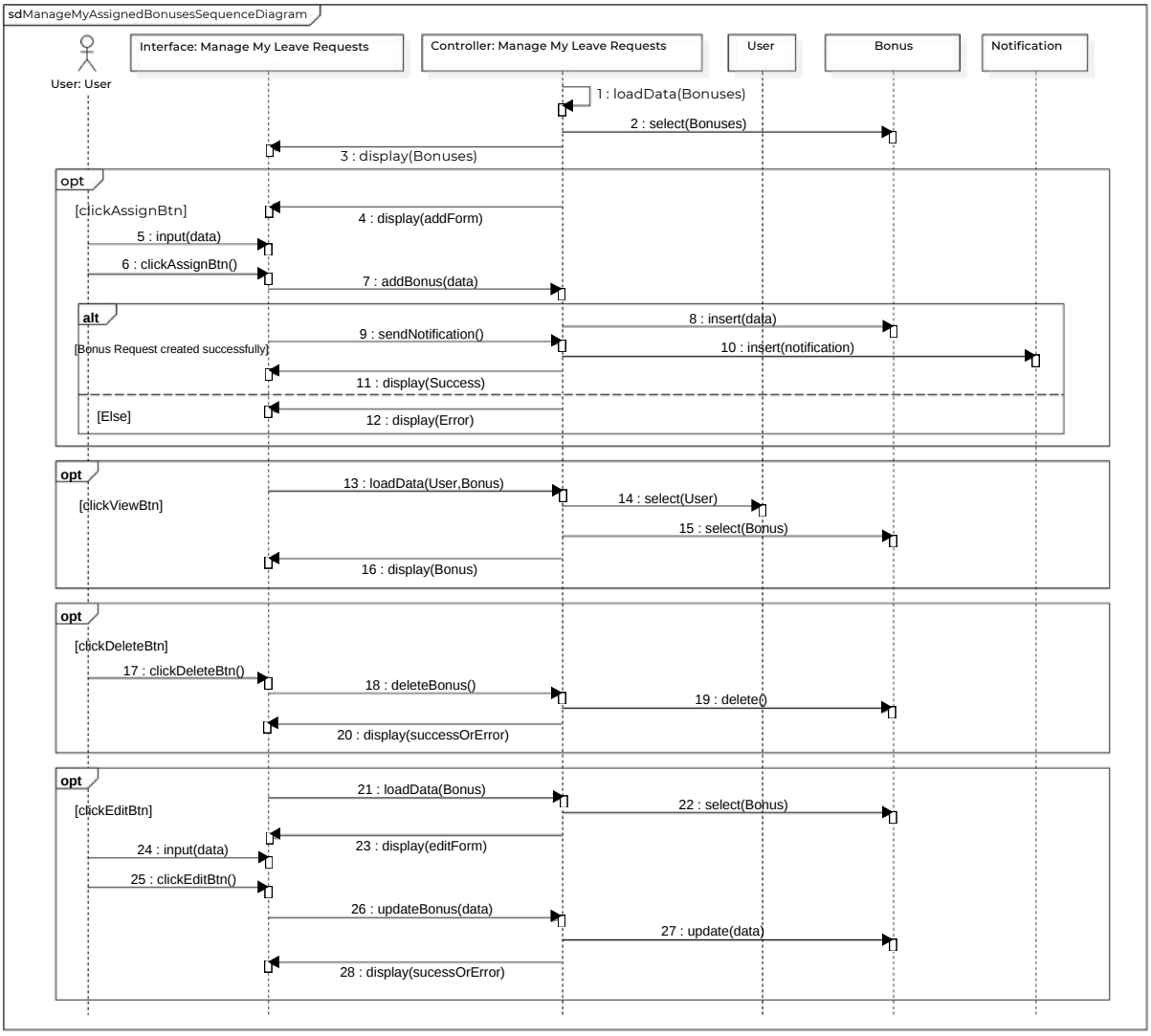


Figure 5.13: "Manage my assigned bonus request" Sequence diagram

5.4.4 "View my received bonus" Use Case Design

Class diagram for the "View my received bonus" use case is shown Figure 5.14.

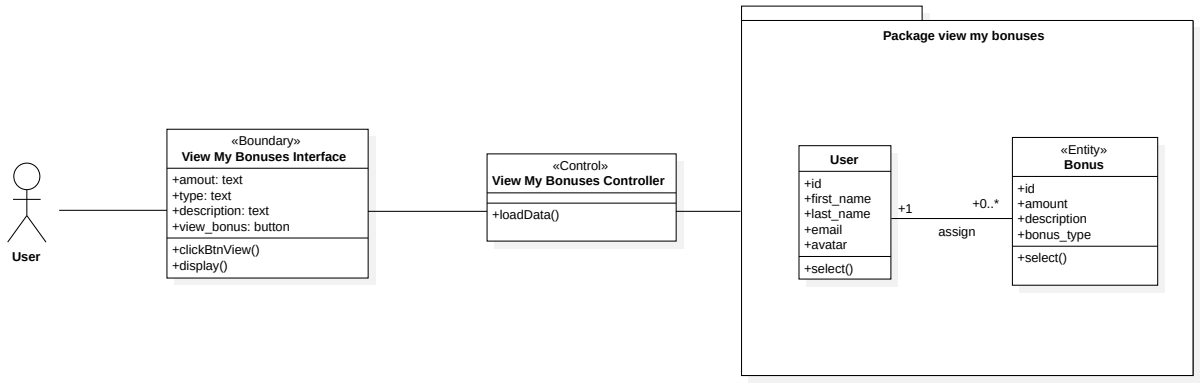


Figure 5.14: "View my received bonus" Class diagram

Sequence diagram for the "View my received bonus" use case is shown in Figure 5.15.

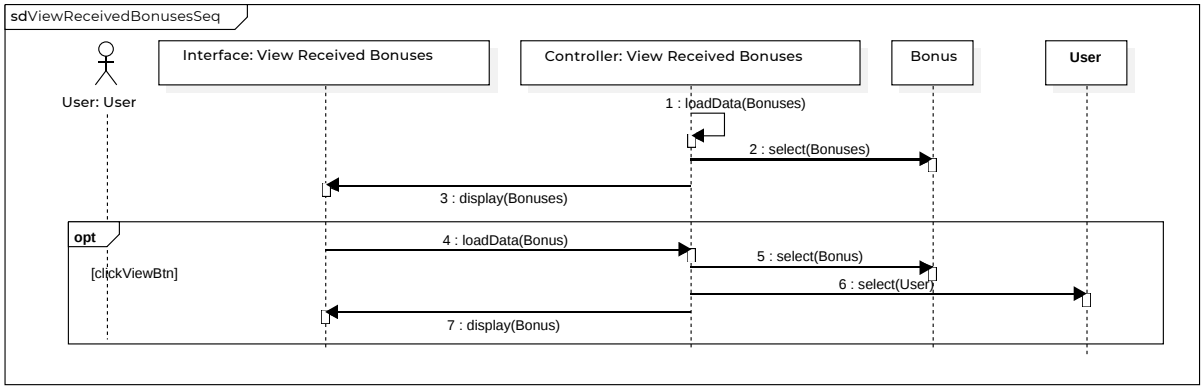


Figure 5.15: "View my received bonus" Sequence diagram

5.4.5 "Manage employees leave requests" Use Case Design

Class diagram for the "Manage employees leave requests" use case is shown Figure 5.16.

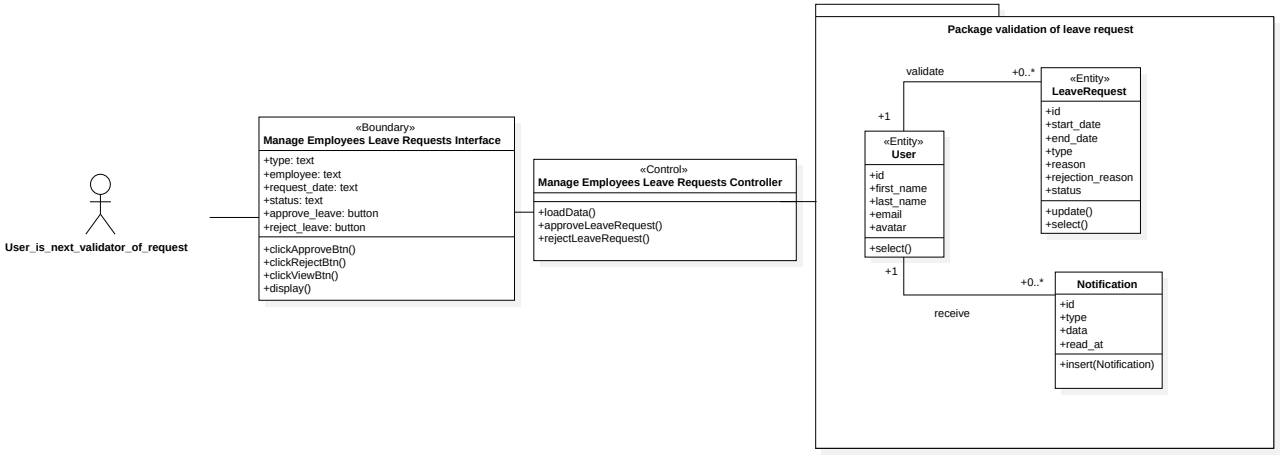


Figure 5.16: "Manage employees leave requests" Class diagram

Sequence diagram for the "Manage employees leave requests" use case is shown in Figure 5.17.



Figure 5.17: "Manage employees leave requests" Sequence diagram

5.4.6 "Manage employees loan requests" Use Case Design

Class diagram for the "Manage employees loan requests" use case is shown Figure 5.18.

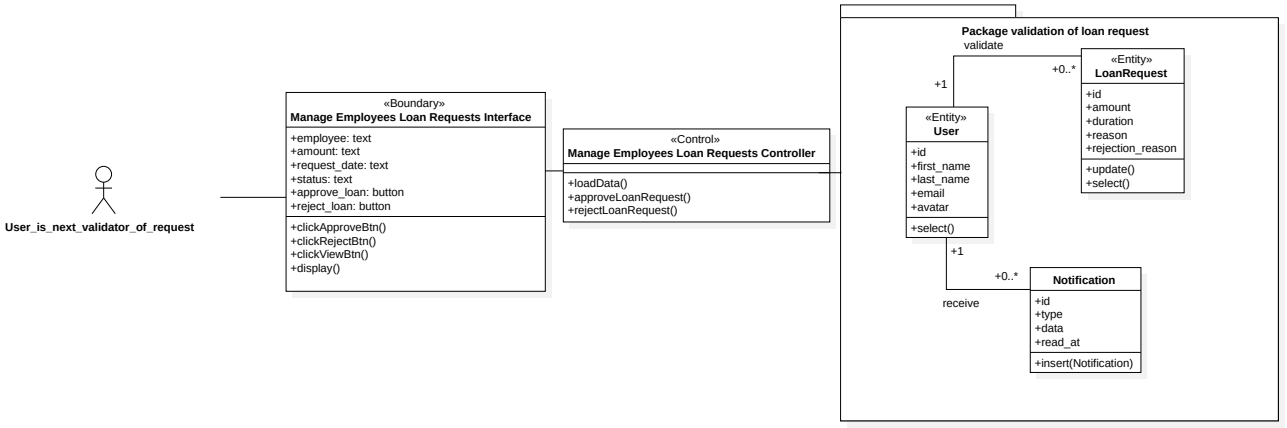


Figure 5.18: "Manage employees loan requests" Class diagram

Sequence diagram for the "Manage employees loan requests" use case is shown in Figure 5.19.

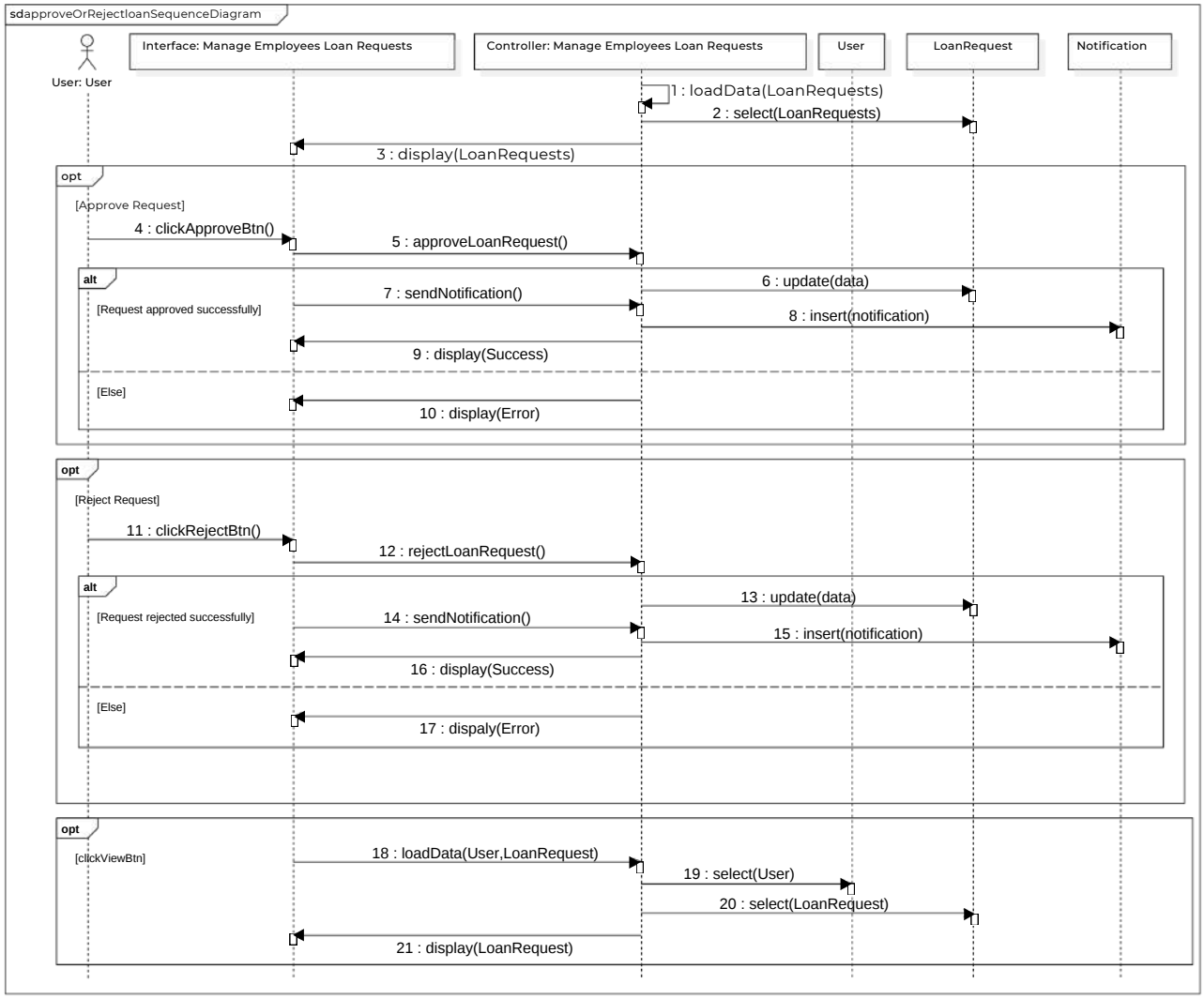


Figure 5.19: "Manage employees loan requests" Sequence diagram

5.4.7 "Manage employees bonus requests" Use Case Design

Class diagram for the "Manage employees bonus requests" use case is shown Figure 5.20.

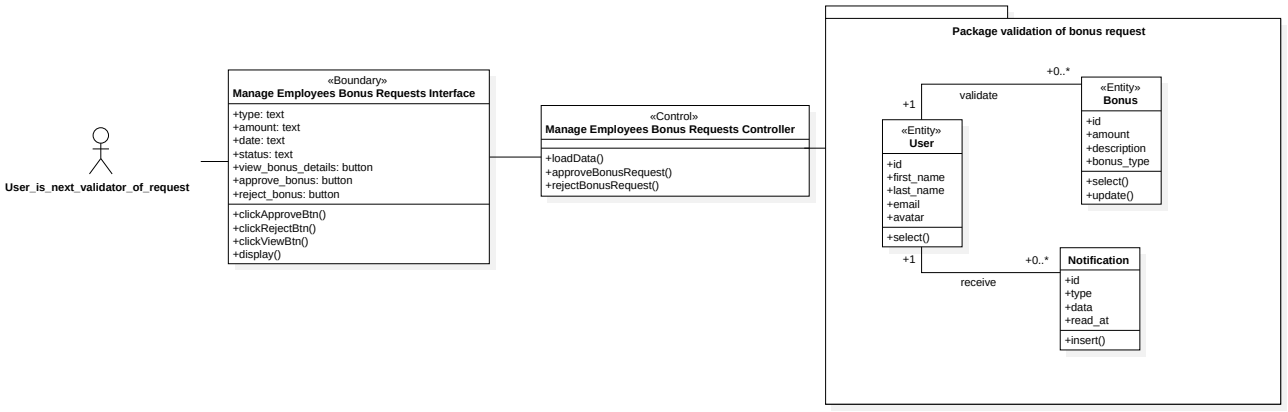


Figure 5.20: "Manage employees bonus requests" Class diagram

Sequence diagram for the "Manage employees bonus requests" use case is shown in Figure 5.21.

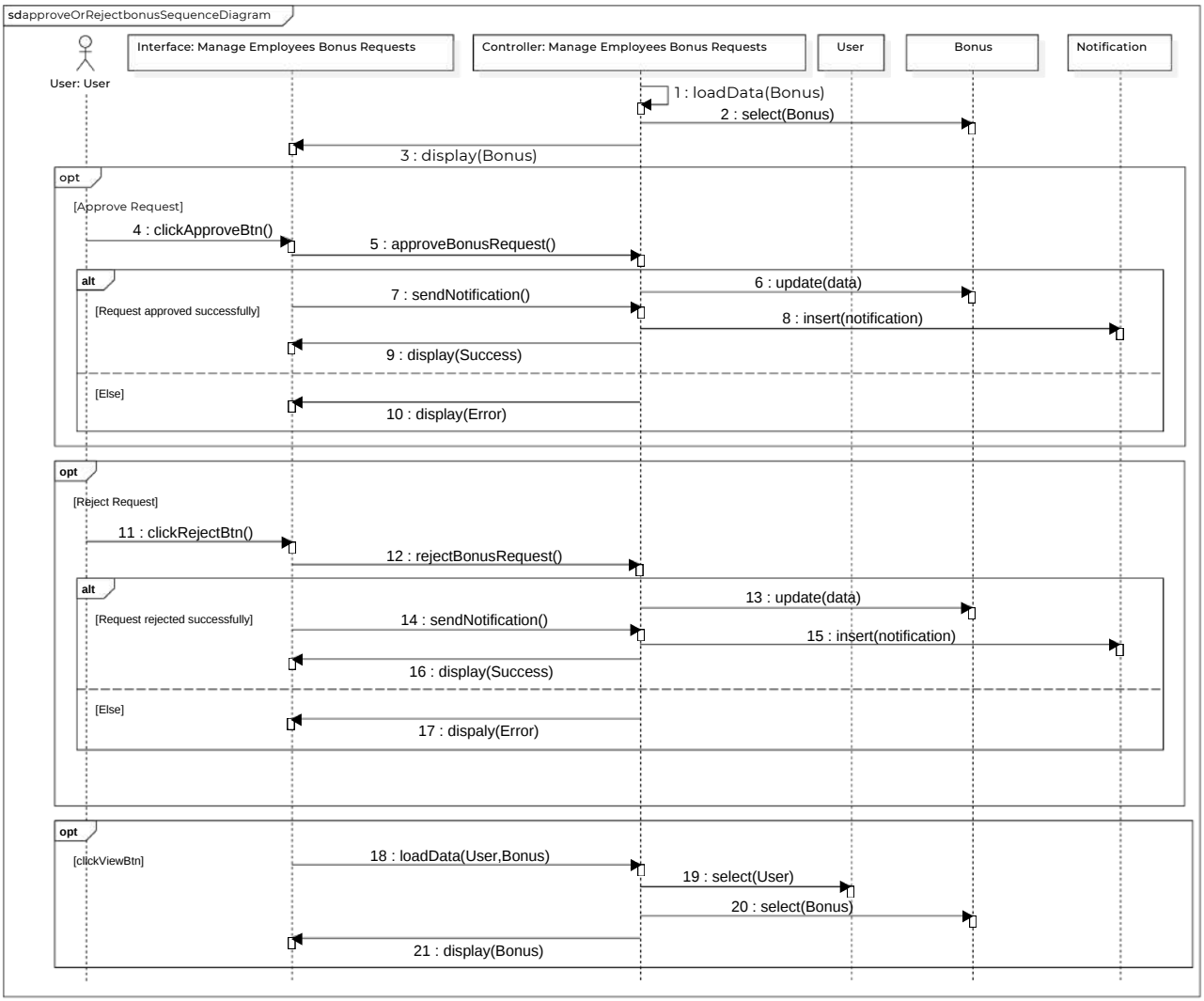


Figure 5.21: "Manage employees bonus requests" Sequence diagram

5.5 Implementation of sprint 2

In this section, we present the interfaces of each use case of this third sprint accompanied by a short description to explain how users interact with it.

5.5.1 Realization of the "Manage my leave request" use case

In this interface, the user can view the list of his leave requests, as shown in Figure 5.22.

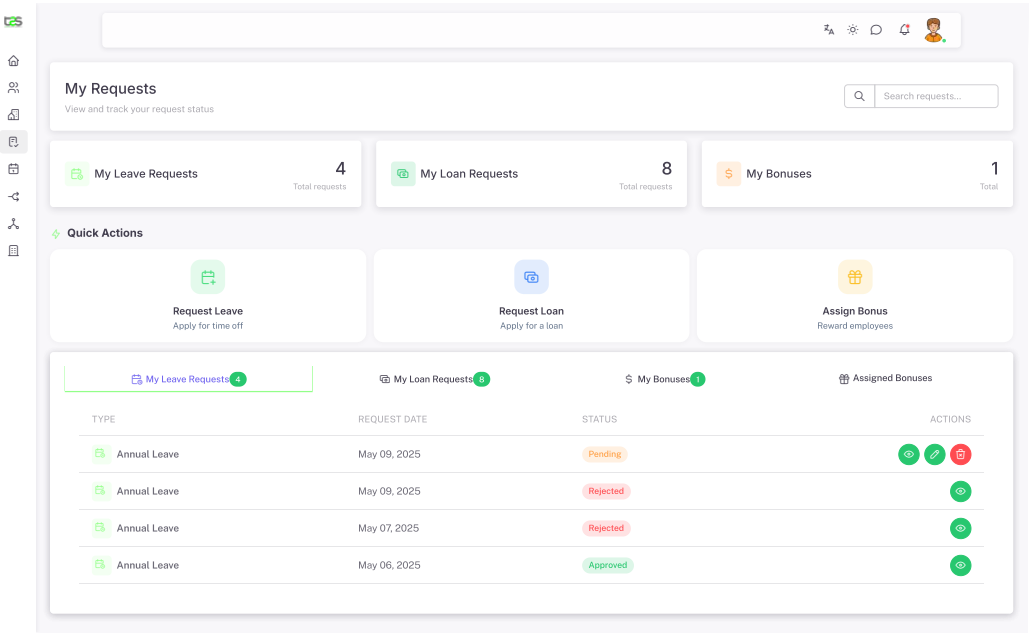


Figure 5.22: "Manage my leave request" Interface

In this interface, the user can view the data related to a specific leave request, as shown in Figure 5.23.

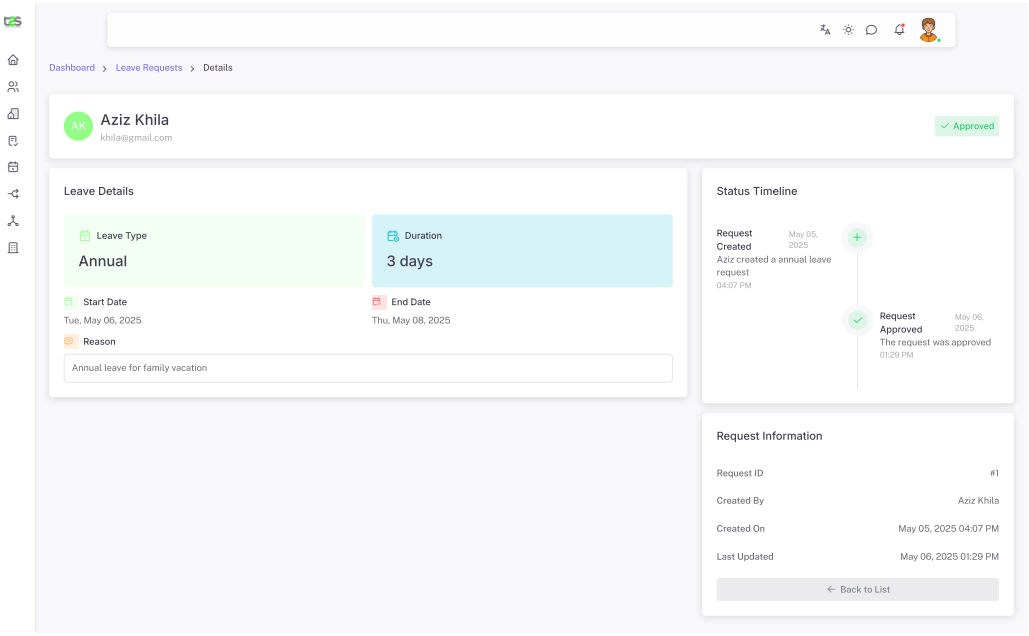


Figure 5.23: "View leave request" Interface

In this interface, the user enters the new leave request data (leave type, start date, end date ...) then clicks on "Submit Request" so the new leave request is created, as shown in Figure 5.24.

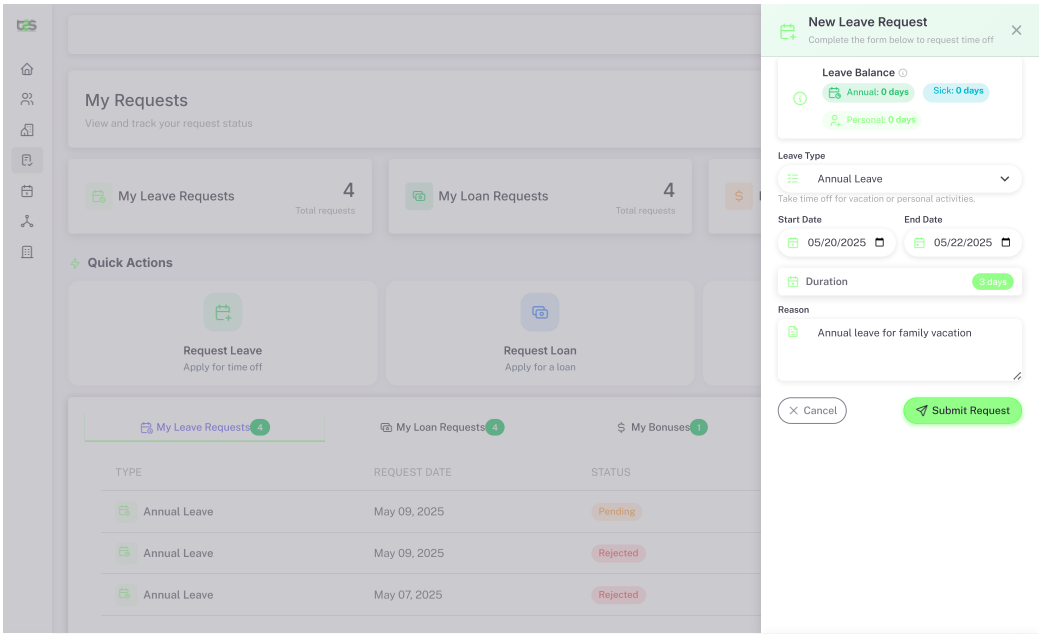


Figure 5.24: "Add leave request" Interface

The user clicks on the delete button, a confirmation modal appears, then the user clicks on the "Yes, Delete" button so the targeted leave request is deleted, as shown in Figure 5.25.

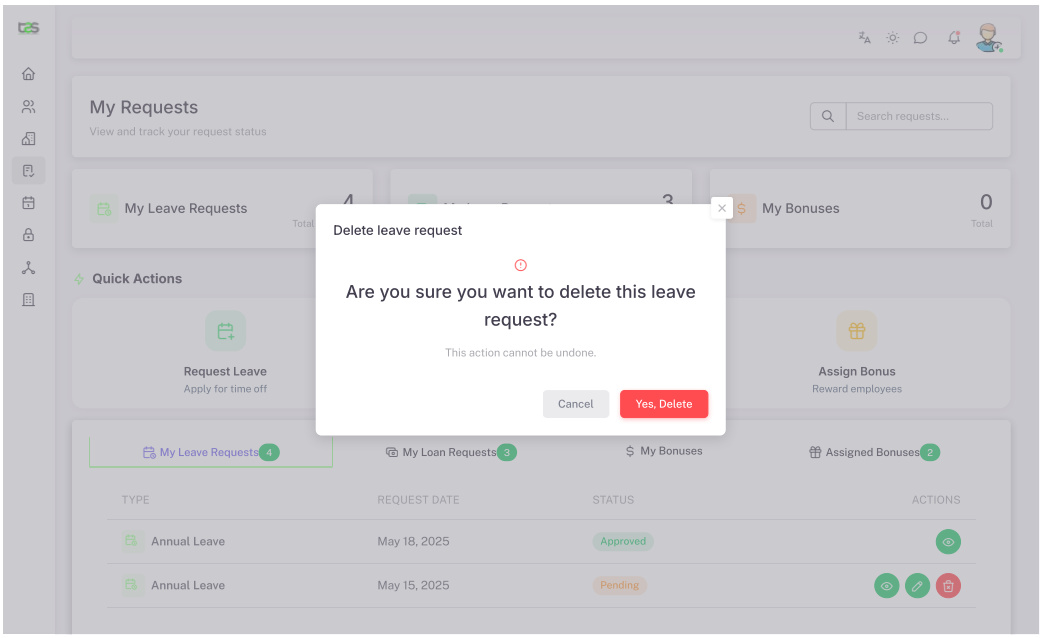


Figure 5.25: "Delete leave request" Interface

In this interface, the user updates the desired data, then clicks on the "Save Changes" button so the targeted leave request is updated, as shown in Figure 5.26.

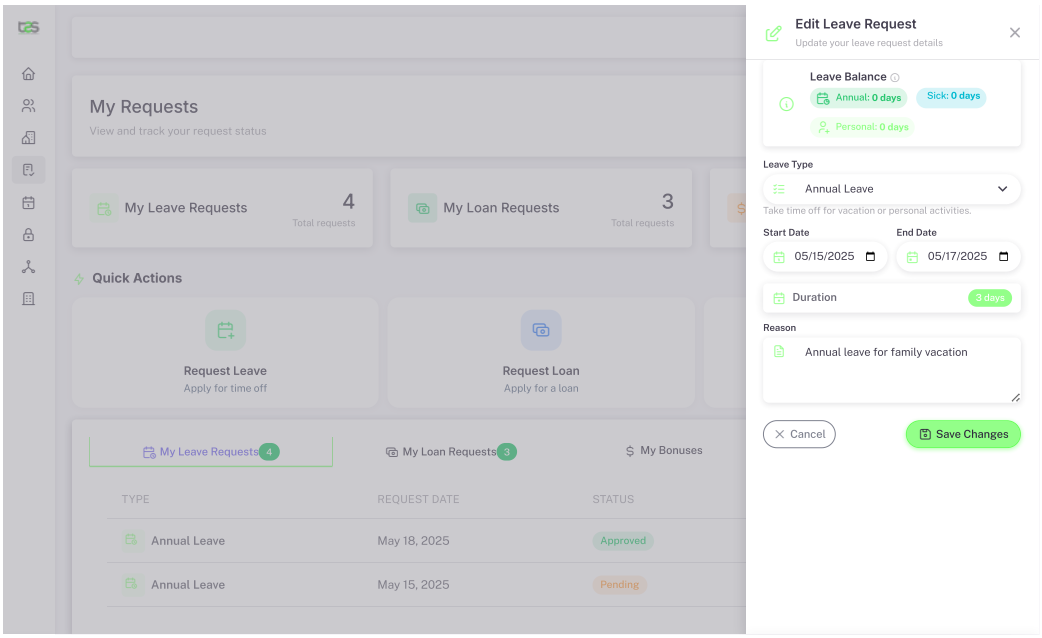


Figure 5.26: "Edit leave request" Interface

5.5.2 Realization of the "Manage my loan request" use case

In this interface, the user can view the list of his loan requests, as shown in Figure 5.27.

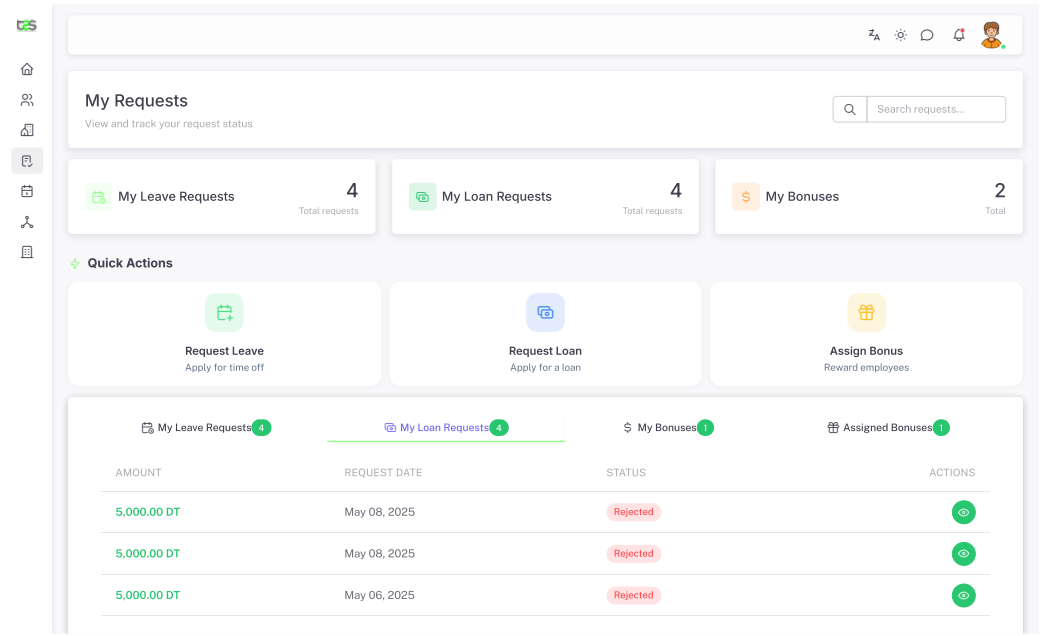


Figure 5.27: "Manage my loan request" Interface

In this interface, the user can view the data related to a specific loan request, as shown in Figure 5.28.

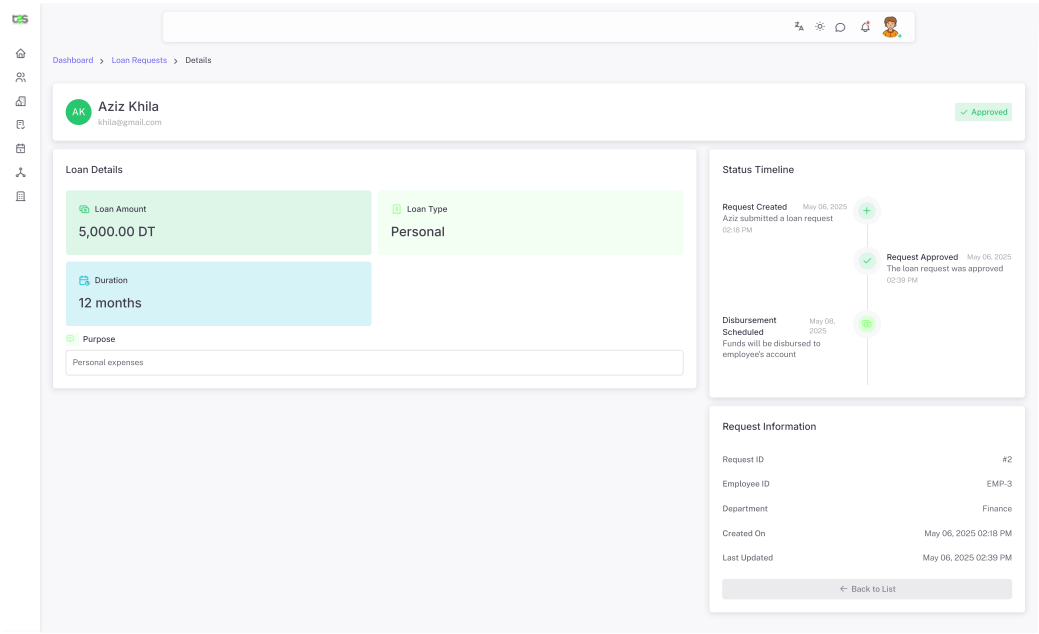


Figure 5.28: "View loan request" Interface

In this interface, the user enters the new loan request data (loan type, loan amount, duration ...) then clicks on "Submit Request" so the new loan request is created, as shown in Figure 5.29.

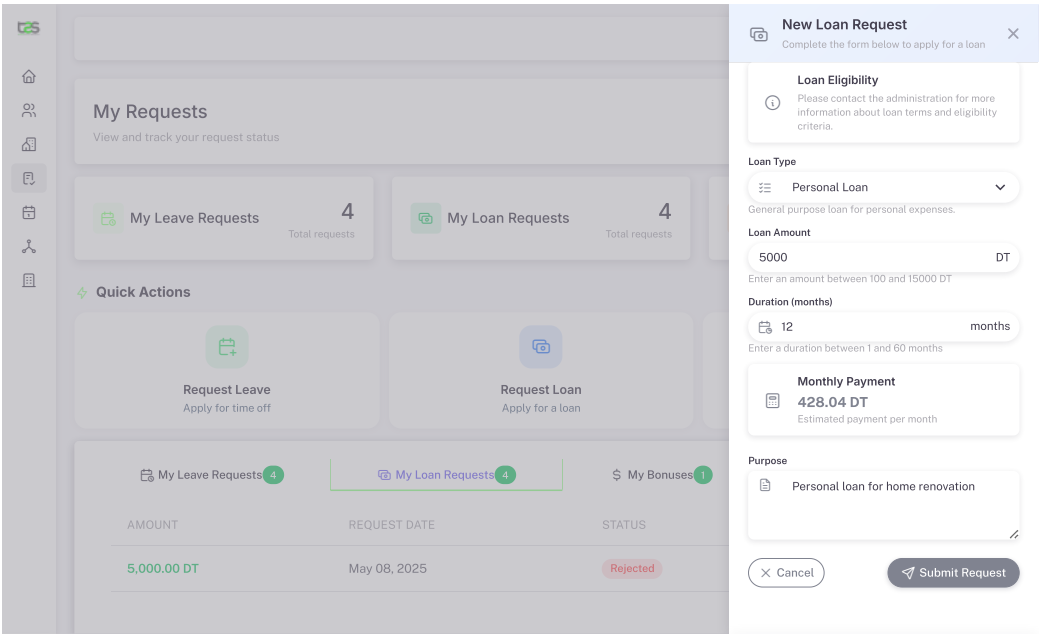


Figure 5.29: "Add loan request" Interface

The user clicks on the delete button, a confirmation modal appears, then the user clicks on the "Yes, Delete" button so the targeted loan request is deleted, as shown in Figure 5.30.

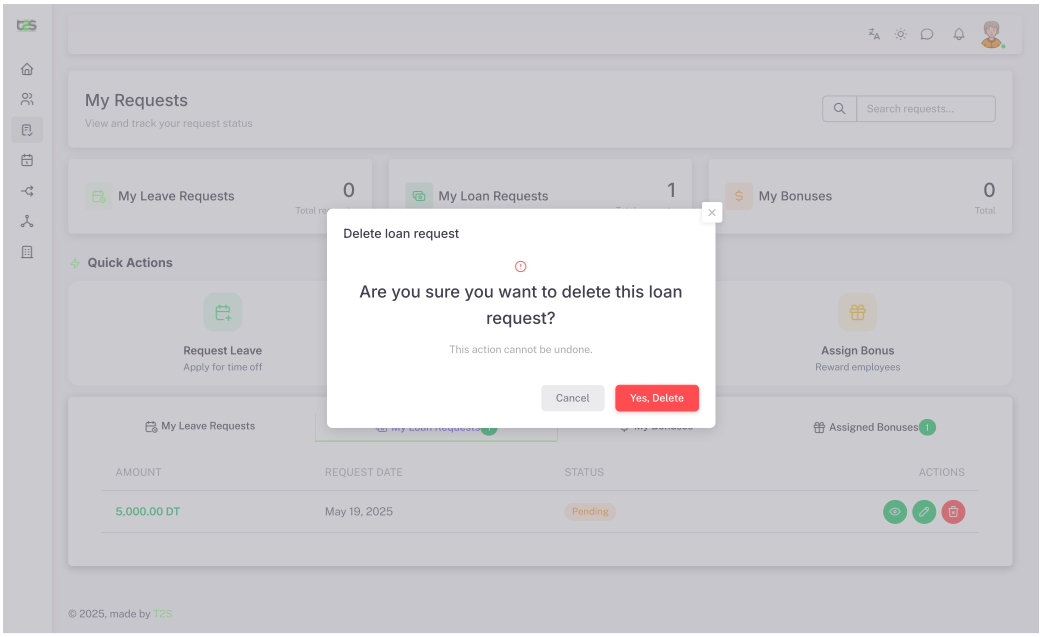


Figure 5.30: ”Delete loan request” Interface

In this interface, the user updates the desired data, then clicks on the ”Save Changes” button so the targeted loan request is updated, as shown in Figure 5.31.

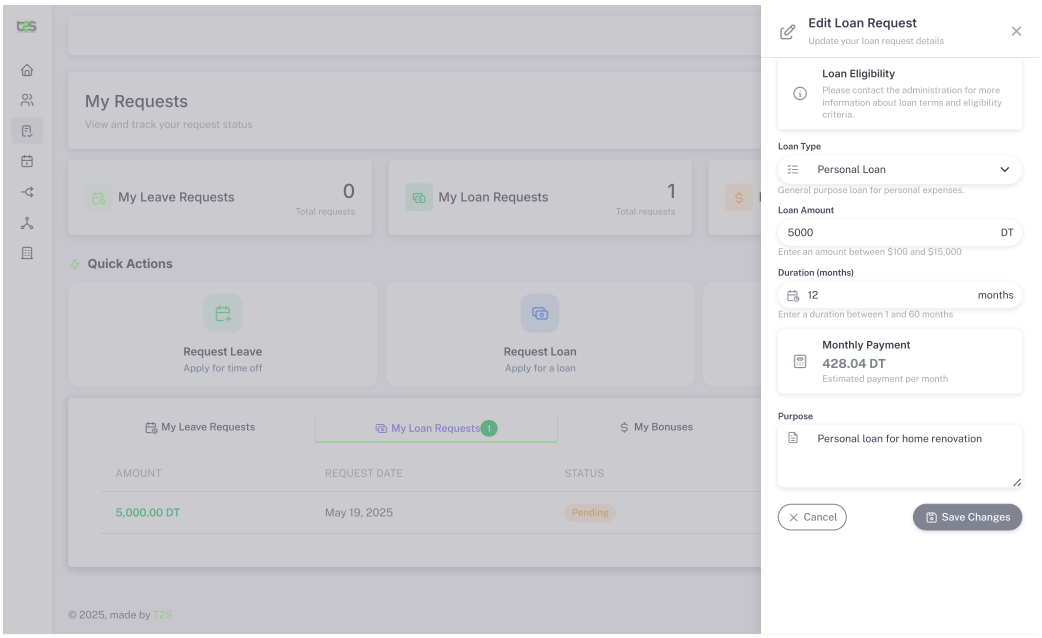


Figure 5.31: ”Edit loan request” Interface

5.5.3 Realization of the “Manage my assigned bonus request” use case

In this interface, the user can view the list of his assigned bonus requests, as shown in Figure 5.32.

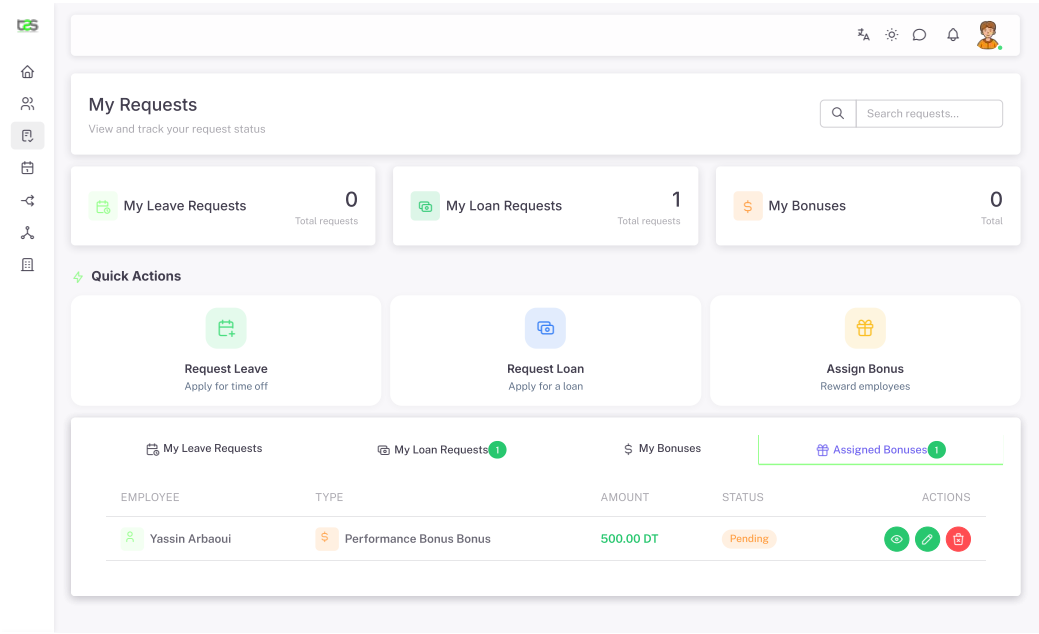


Figure 5.32: "Manage my assigned bonus requests" Interface

In this interface, the user can view the data related to a specific assigned bonus request, as shown in Figure 5.33.

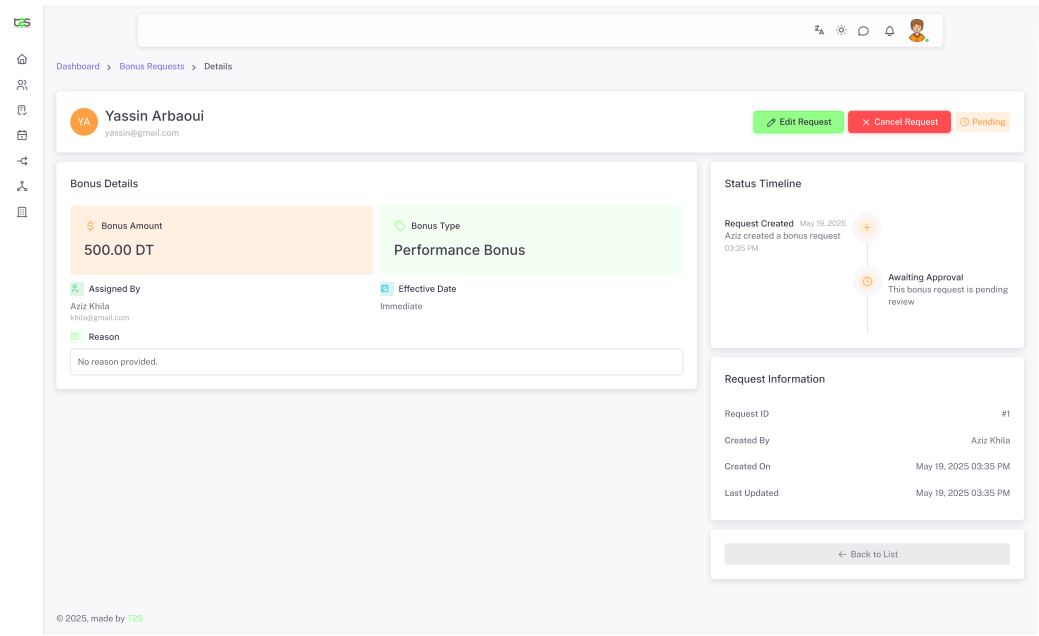


Figure 5.33: "View my assigned bonus request" Interface

In this interface, the user enters the new bonus request data (Employee, Bonus type, Amount ...) then clicks on "Submit Bonus" so the new assigned bonus request is created, as shown in Figure 5.34.

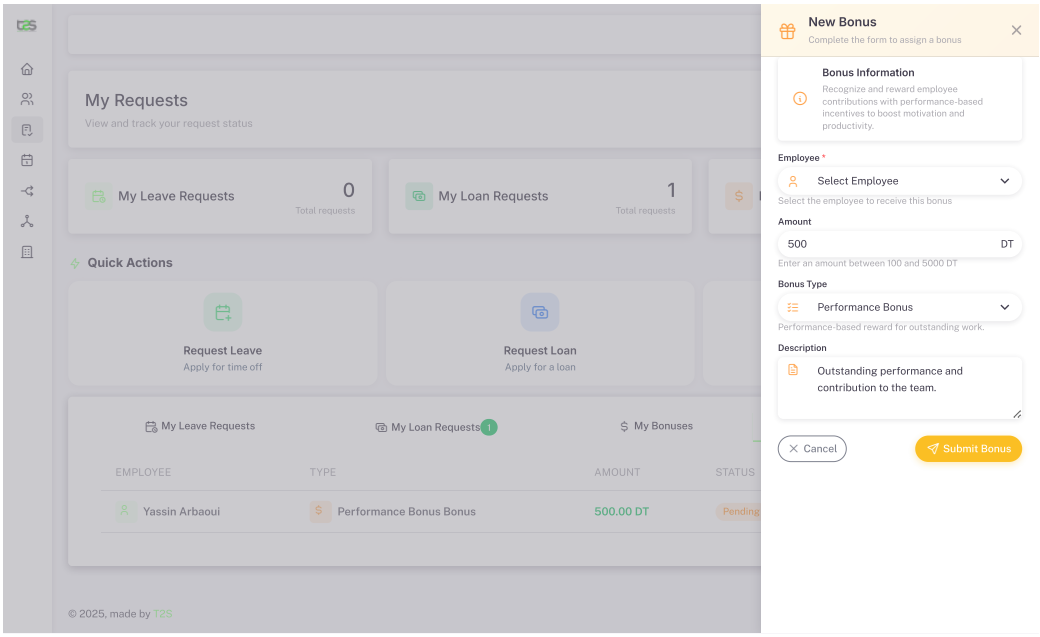


Figure 5.34: "Assign bonus to employee" Interface

The user clicks on the delete button, a confirmation modal appears, then the user clicks on the "Yes, Delete" button so the targeted assigned bonus request is deleted, as shown in Figure 5.35.

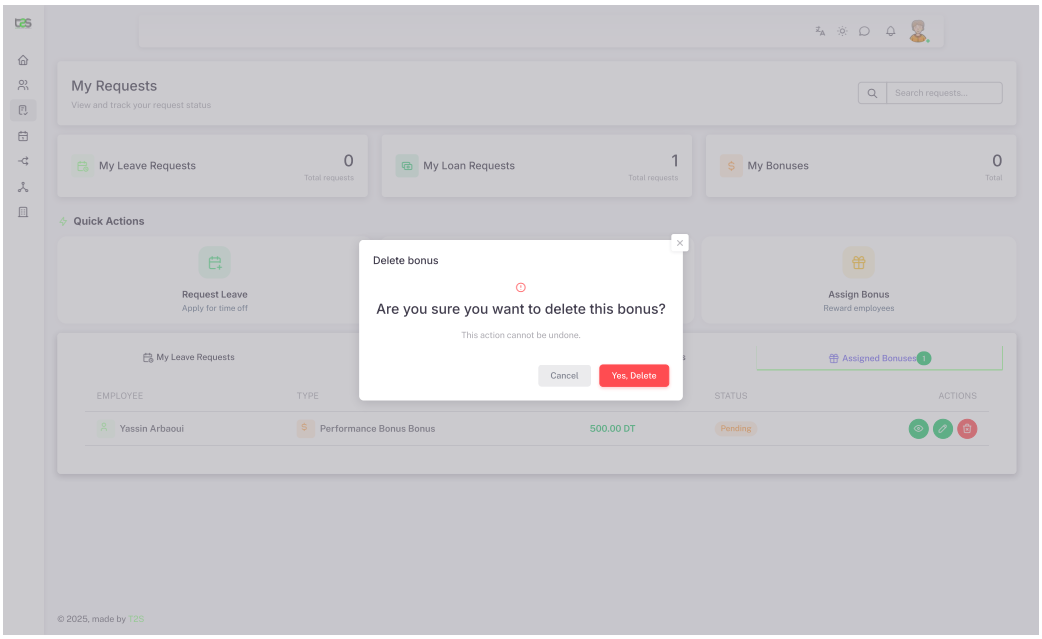


Figure 5.35: "Delete my assigned bonus request" Interface

In this interface, the user updates the desired data, then clicks on the "Save Changes" button so the targeted assigned bonus request is updated, as shown in Figure 5.36.

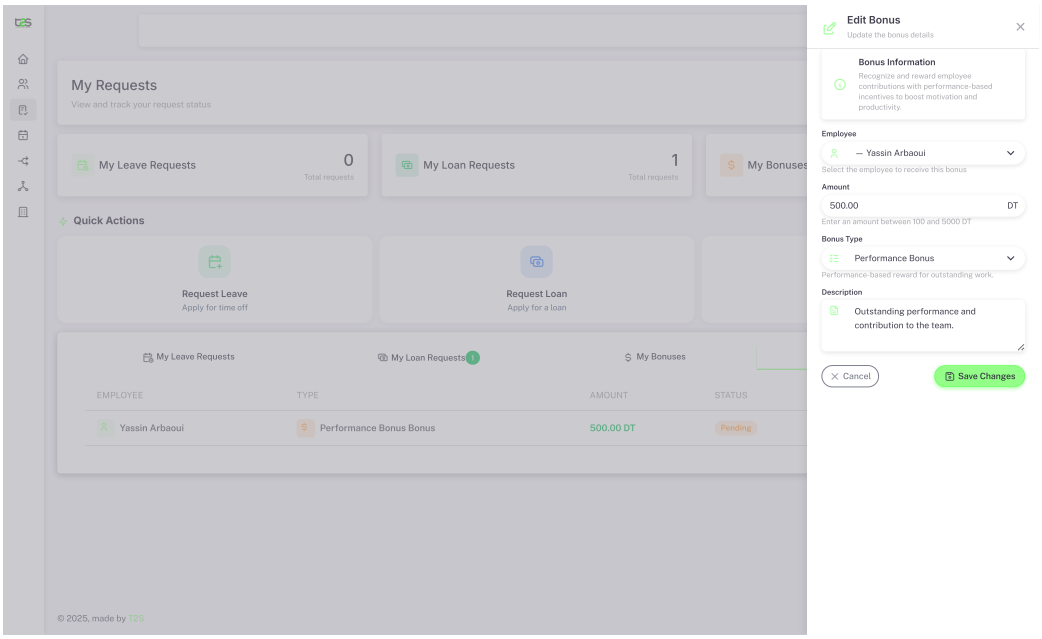


Figure 5.36: "Edit my assigned bonus request" Interface

5.5.4 Realization of the "View my received bonus" use case

In this interface, the user can view the list of his received bonuses, as shown in Figure 5.37.

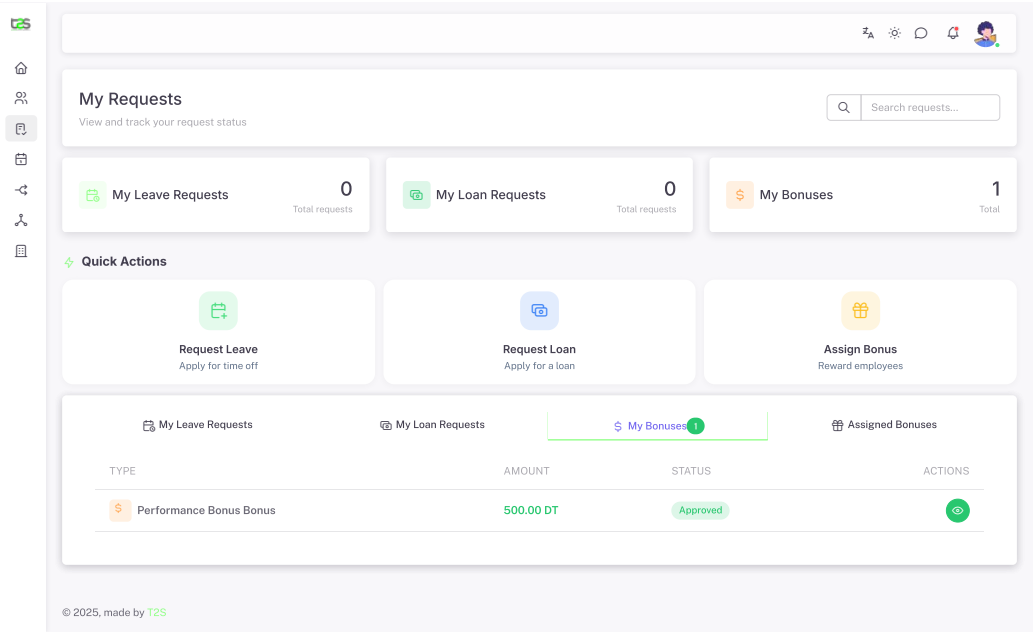


Figure 5.37: "View received bonuses" Interface

In this interface, the user can view the details related to a specific received bonus, as shown in Figure 5.38.

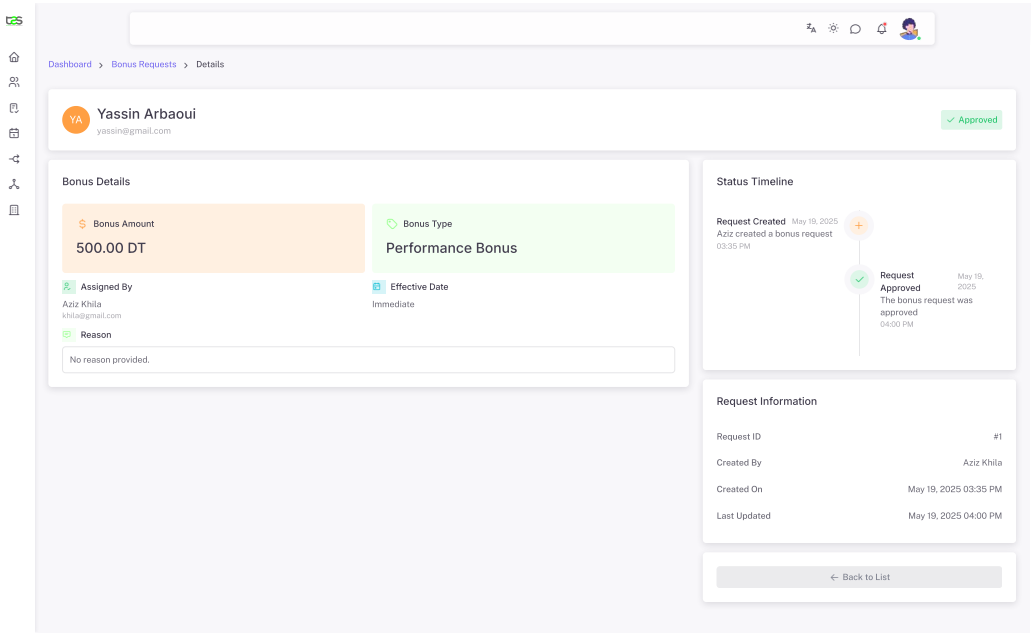


Figure 5.38: "View a received bonus" Interface

5.5.5 Realization of the "Manage employees leave requests" use case

In this interface, the user can view the list of leave requests that need his decision (approval or rejection), as shown in Figure 5.39.

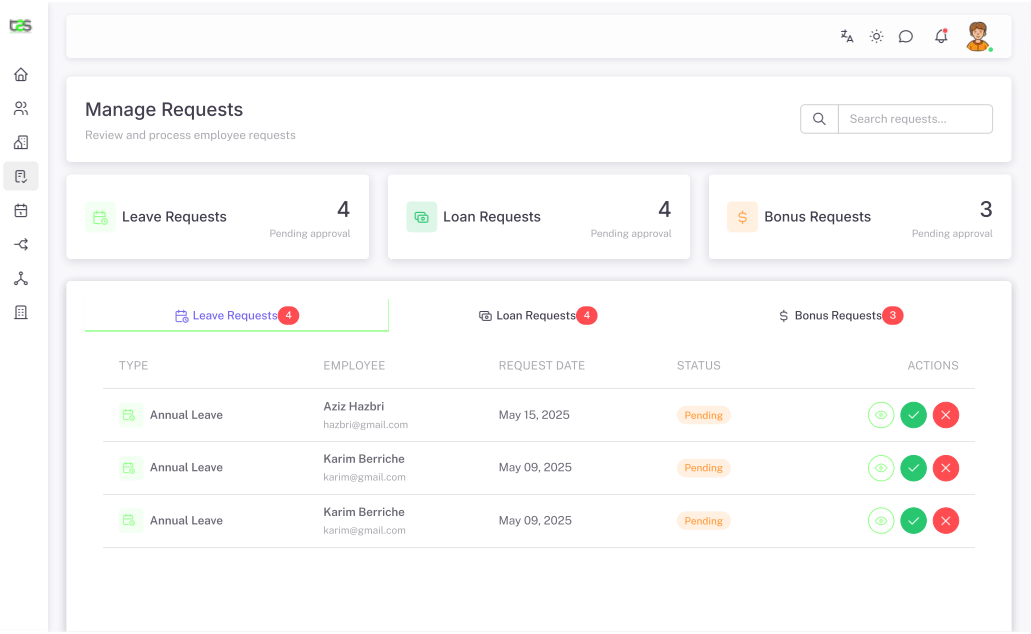


Figure 5.39: "Manage employees leave requests" Interface

In this interface, the user can view the details of a leave request, as shown in Figure 5.40.

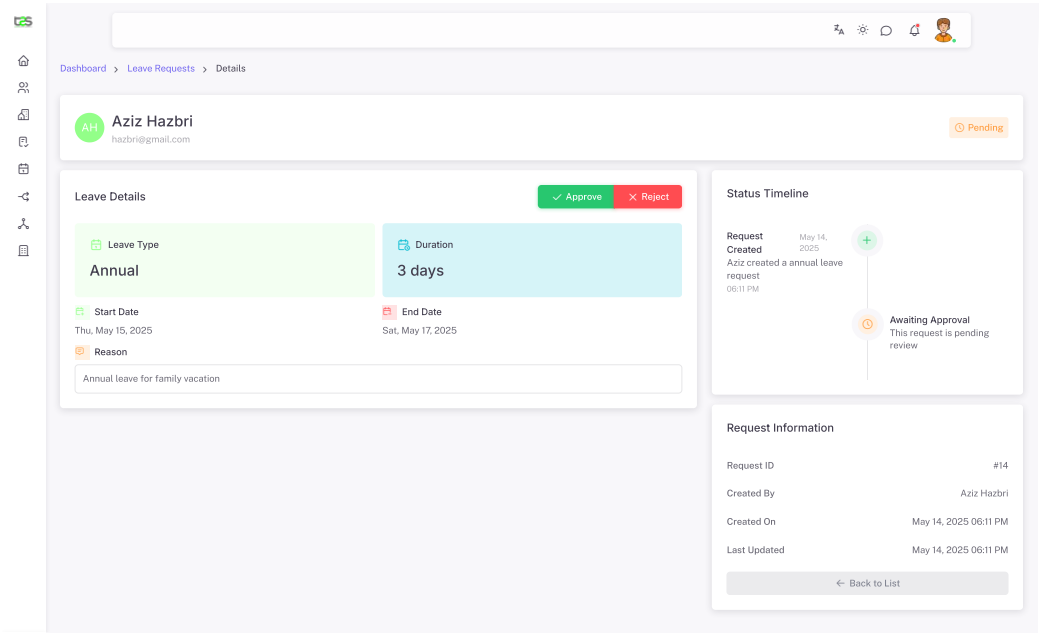


Figure 5.40: "View Leave Request" Interface

In this interface, the user approves a leave request and confirms approval in the modal, as shown in Figure 5.41.

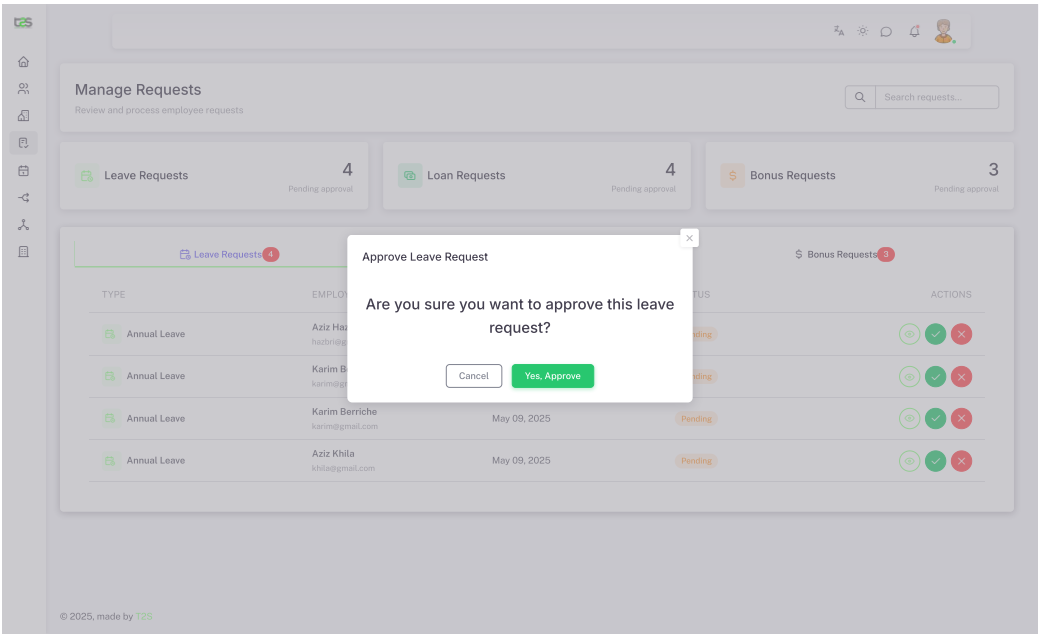


Figure 5.41: "Approve leave request" Interface

In this interface, the user rejects a leave request, writes the rejection reason and confirms rejection in the modal, as shown in Figure 5.42.

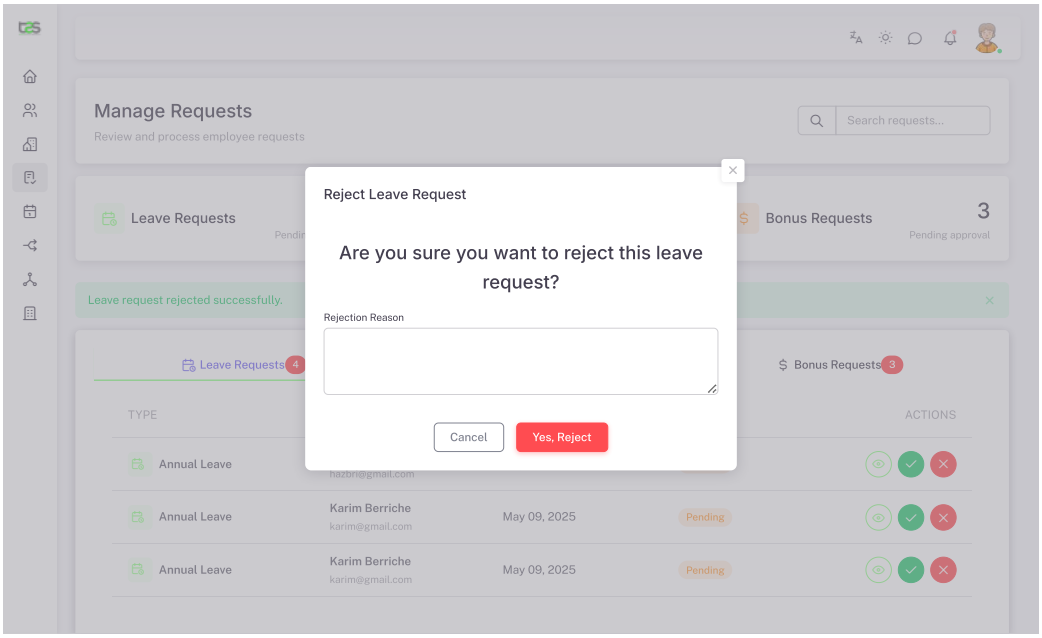


Figure 5.42: "Reject leave request" Interface

5.5.6 Realization of the "Manage employees loan requests" use case

In these interfaces, the user can view the lists of loan requests that need his decision (approval or rejection), as shown in Figure 5.43.

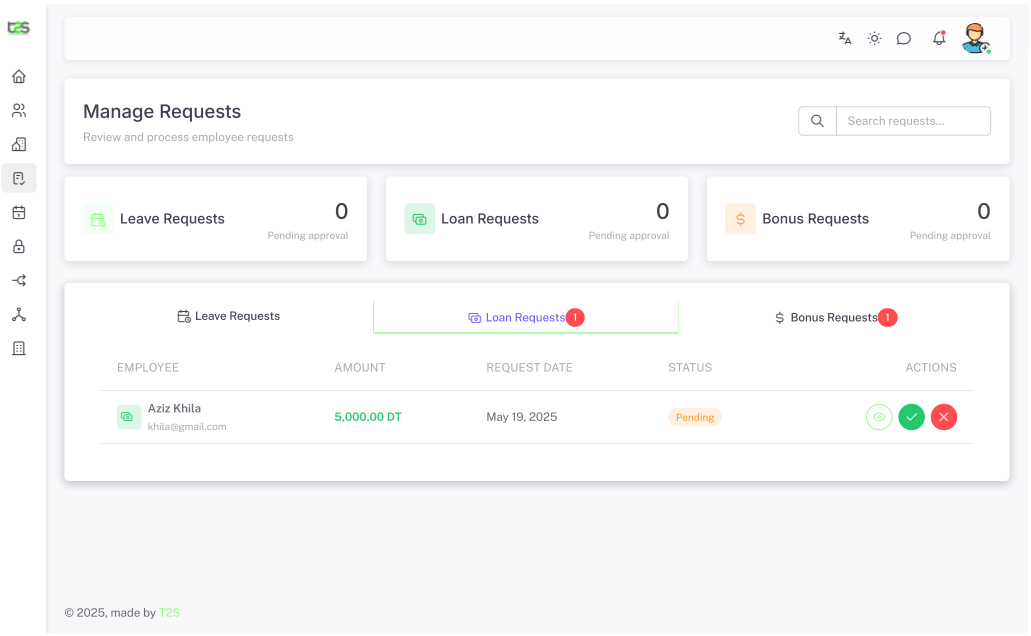


Figure 5.43: "Manage employees loan requests" Interface

In this interface, the user can view the details of a loan request, as shown in Figure 5.44.

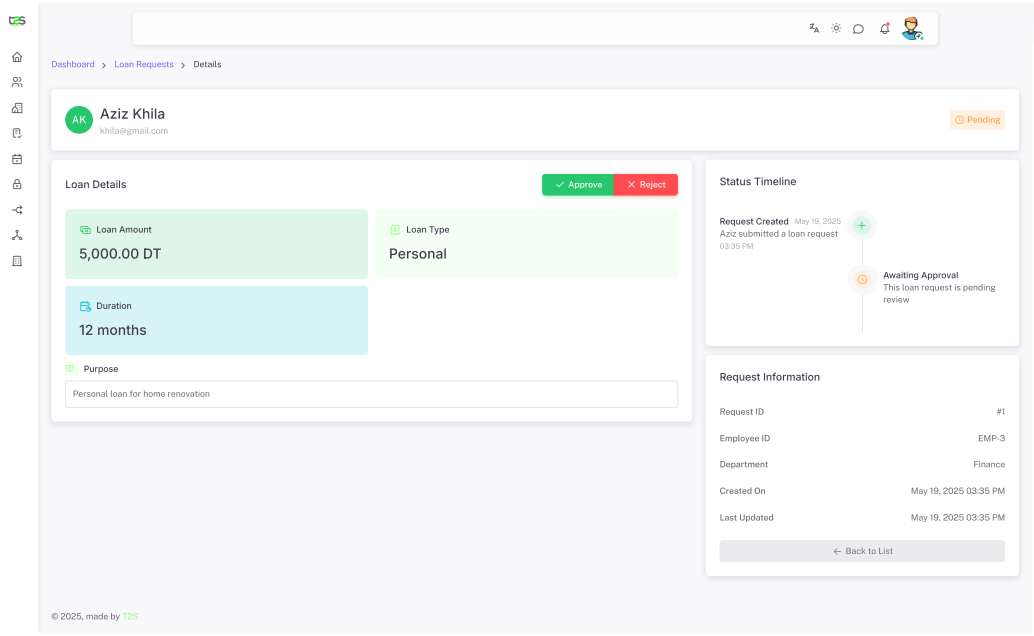


Figure 5.44: "View a loan Request" Interface

In this interface, the user approves a loan request and confirms approval in the modal, as shown in Figure 5.45.

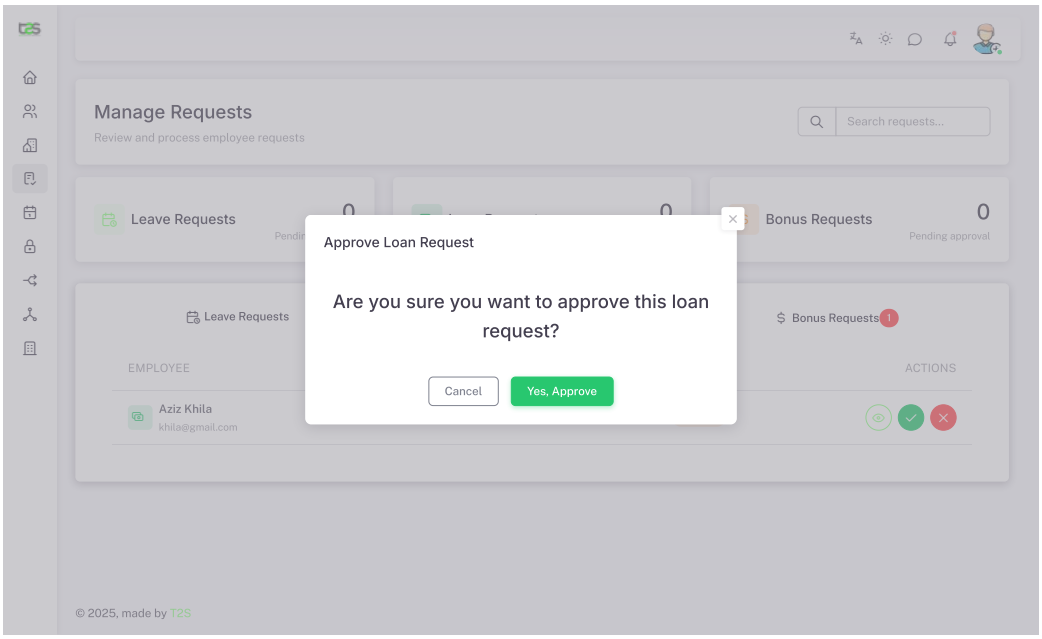


Figure 5.45: "Approve loan request" Interface

In this interface, the user rejects a loan request, writes the rejection reason and confirms rejection in the modal, as shown in Figure 5.46.

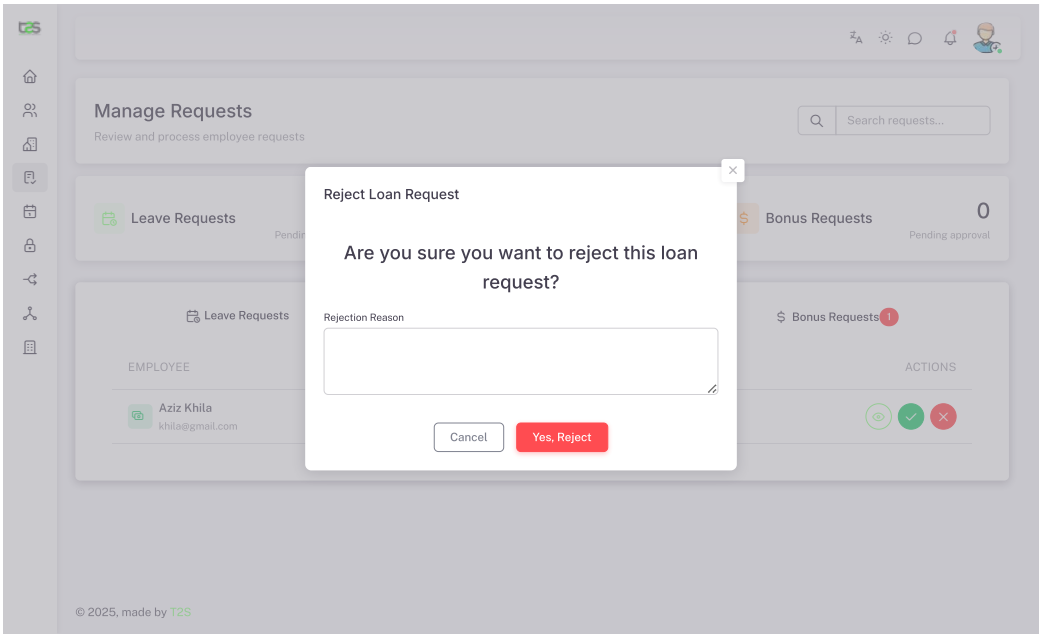


Figure 5.46: "Reject Loan request" Interface

5.5.7 Realization of the "Manage employees bonus requests" use case

In these interfaces, the user can view the lists of bonus requests that need his decision (approval or rejection), as shown in Figure 5.47.

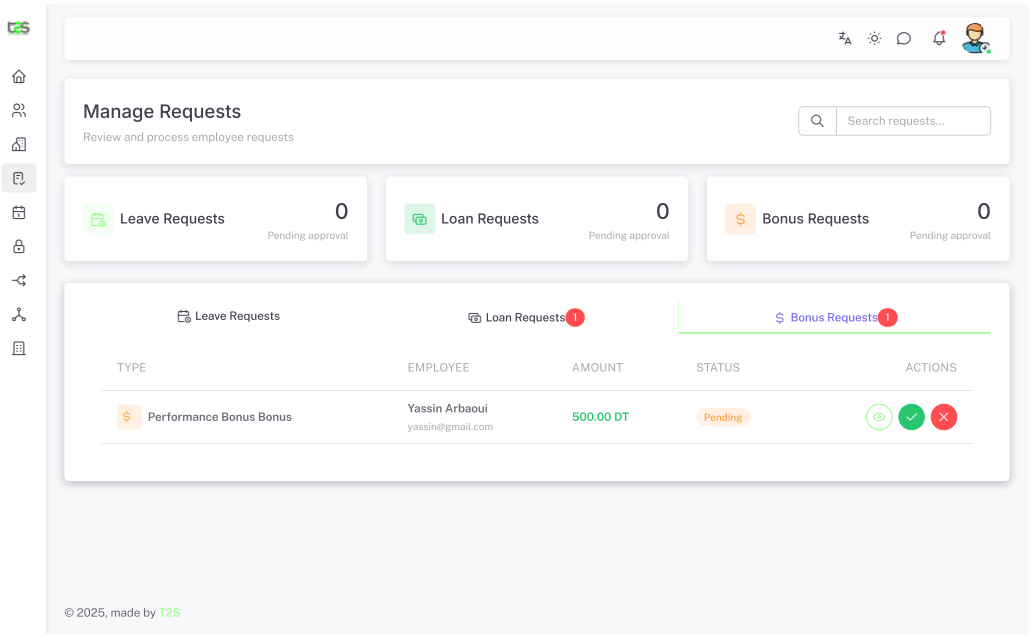


Figure 5.47: "Manage employees bonus requests" Interface

In this interface, the user can view the details of a bonus request, as shown in Figure 5.48.

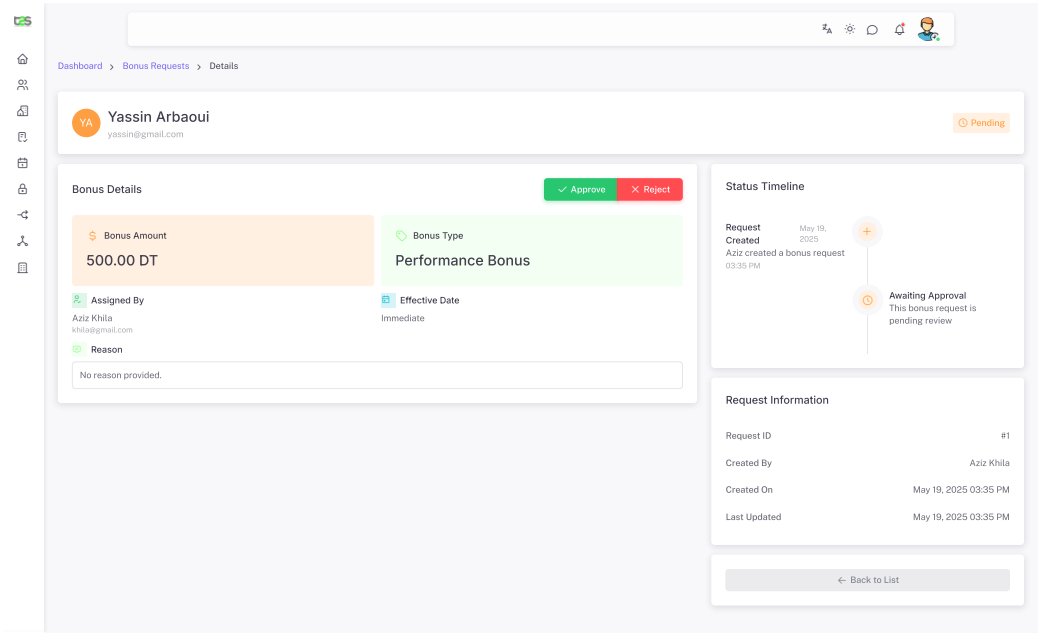


Figure 5.48: "View a Bonus Request" Interface

In this interface, the user approves a bonus request and confirms approval in the modal, as shown in Figure 5.49.

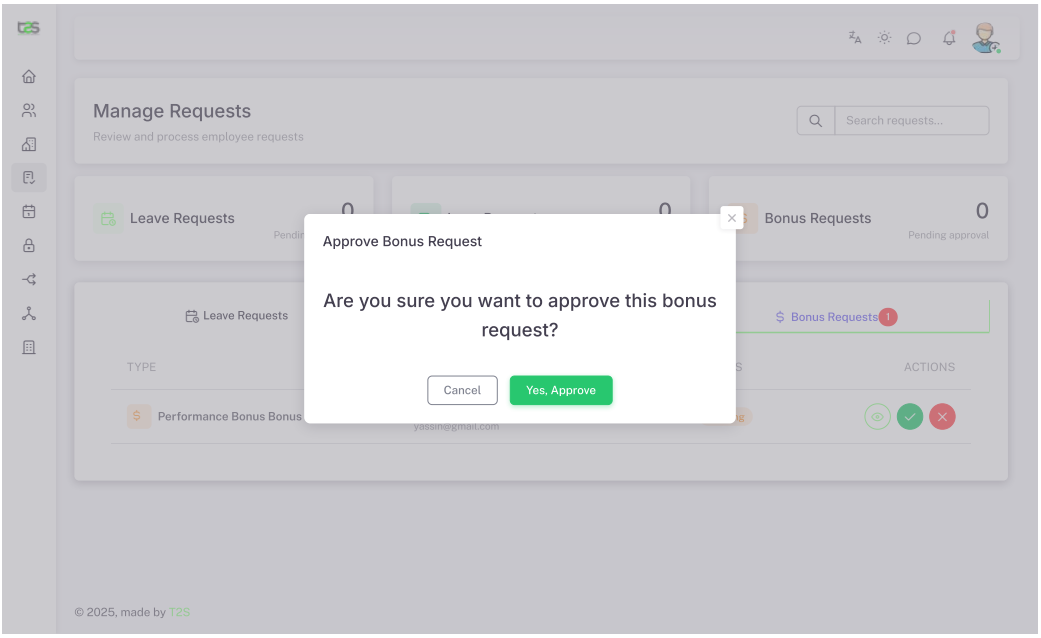


Figure 5.49: "Approve bonus request" Interface

In this interface, the user rejects a bonus request, writes the rejection reason and confirms rejection in the modal, as shown in Figure 5.50.

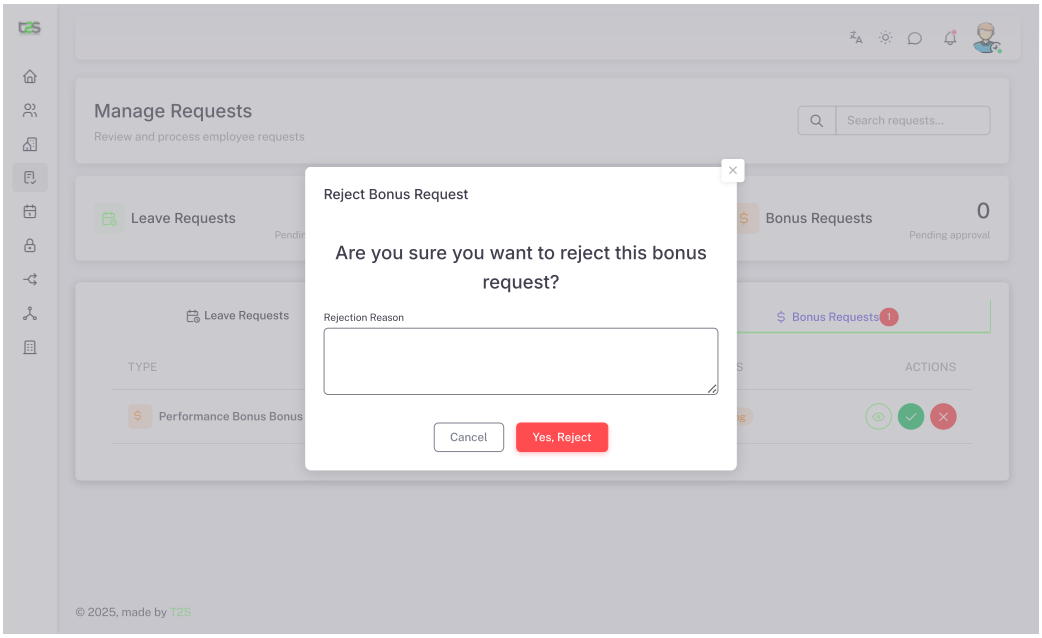


Figure 5.50: "Reject bonus request" Interface

5.6 Conclusion

In this chapter, we present the refinement, the design, and the interfaces of use cases of sprint 2.

Chapter 6

Sprint 3 : Engagement Internal Policies

6.1 Introduction

In this chapter, we present the use cases for Sprint 3. The study of this sprint includes refinement, design, and implementation.

6.2 Identifying the sprint 3 backlog

The table 6.1 contains the backlog elements that are executed during sprint 3:

Table 6.1: Product backlog sprint 3

User Story	Priority	Estimation	Sprint
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete events in the calendar.	4	Average	Sprint 3
As a user, I can view my profile.	4	Low	Sprint 3
As a user, I can chat with other users.	4	Average	Sprint 3
As a user, depending on the permissions, I can create or/and view or/and edit or/and delete company policies.	4	Average	Sprint 3

6.3 Refinement of sprint 3

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

6.3.1 Refinement of the ”Manage events” use case

The figure 6.1 showcases the use case.

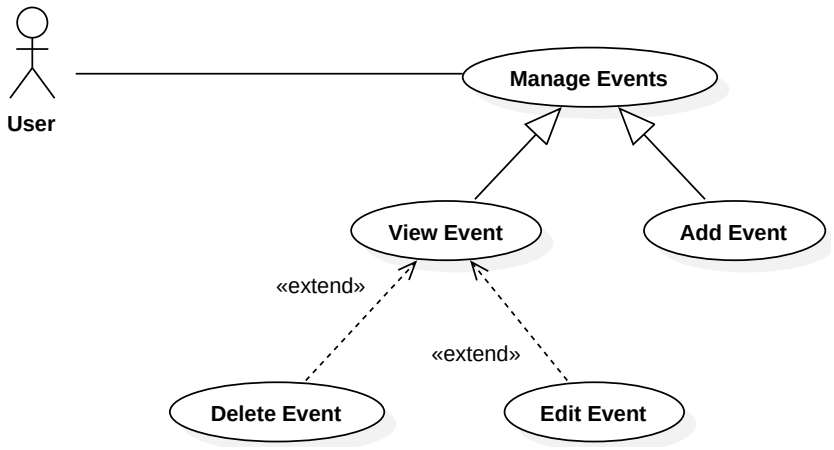


Figure 6.1: “Manage events” use case diagram

The table 6.2 elaborates on this user story with textual description.

Table 6.2: Detailed description of the ”Manage events” use case

Use Case	« Manage events »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor have permissions to manage events.
Post-condition	Events managed.
Principal scenario	- System displays calendar interface. - Actor enters information of a new event then clicks on « Add Event » button to create new event. - Actor clicks on the desired event to view its details. - Actor clicks on the desired event then on edit button to update an existing event. - Actor clicks on the desired event then on delete button to delete an existing event.
Exception	The system displays an error message if the changed information is incorrect.

6.3.2 Refinement of the ”View profile” use case

The figure 6.2 showcases the use case.

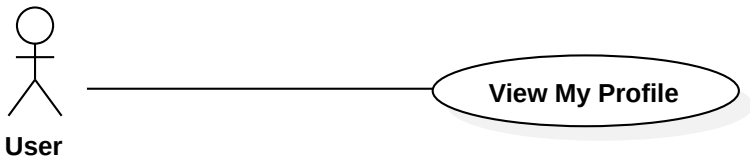


Figure 6.2: “View profile” use case diagram

The table 6.3 elaborates on this user story with textual description.

Table 6.3: Detailed description of the ”View profile” use case

Use Case	« View profile »
Actor(s)	User
Pre-condition	Actor authenticated
Post-condition	Profile consulted.
Principal scenario	- System displays profile interface. - Actor consults his profile that contains his information and his manager contacts.
Exception	No exception.

6.3.3 Refinement of the ”Chat with other users” use case

The figure 6.3 showcases the use case.

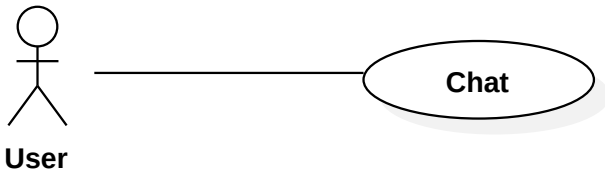


Figure 6.3: “Chat with other users” use case diagram

The table 6.4 elaborates on this user story with textual description.

Table 6.4: Detailed description of the ”Chat with other users” use case

Use Case	« Chat with other users »
Actor(s)	User
Pre-condition	Actor authenticated
Post-condition	User communicated with other users.
Principal scenario	- System displays chat interface. - Actor chooses which user to chat with, types messages and clicks on send message button. - Actor chooses the status(online, do not disturb,etc.) that would appear in the chat contact bar for others.
Exception	No exception

6.3.4 Refinement of the ”Manage company policies” use case

The figure 6.4 showcases the use case.

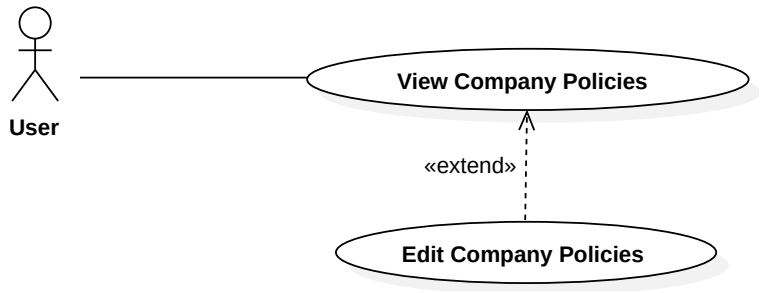


Figure 6.4: “Manage company policies” use case diagram

The table 6.5 elaborates on this user story with textual description.

Table 6.5: Detailed description of the ”Manage company policies” use case

Use Case	« Manage company policies »
Actor(s)	User
Pre-condition	- Actor authenticated - Actor have permissions to manage company policies.
Post-condition	Company policies managed.
Principal scenario	- System displays company policies interface. - Actor clicks on « add New Policy » button and fills the details of the new policy then clicks on save. - Actor updates the policy that he wants to be updated then clicks on « save » button . - Actor clicks on « Delete » button to get rid of a policy .
Exception	You cannot have the same policy name twice.

6.4 Design of sprint 3

We present now the design of the sprint 3 use cases.

6.4.1 “Manage events” Use Case Design

Class diagram for the “Manage events” use case is shown Figure 6.5.

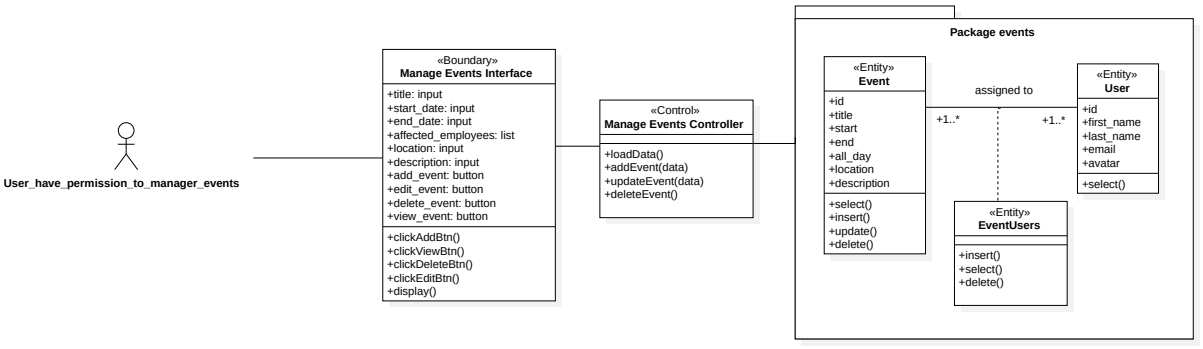


Figure 6.5: ”Manage events” Class diagram

Sequence diagram for the ”Manage events” use case is shown in Figure 6.6.

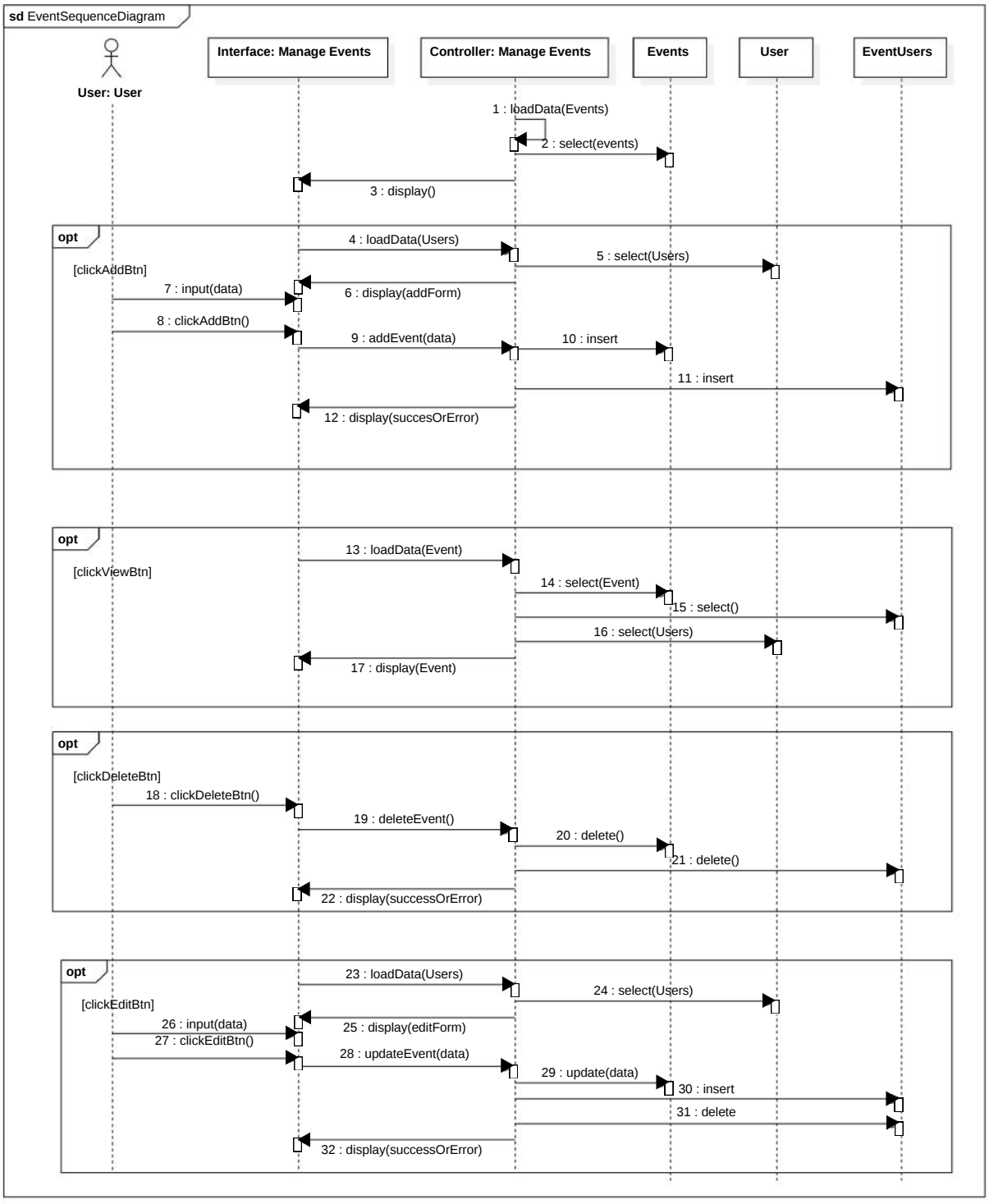


Figure 6.6: "Manage events" Sequence diagram

6.4.2 "View my profile" Use Case Design

Class diagram for the "View profile" use case is shown in Figure 6.7.

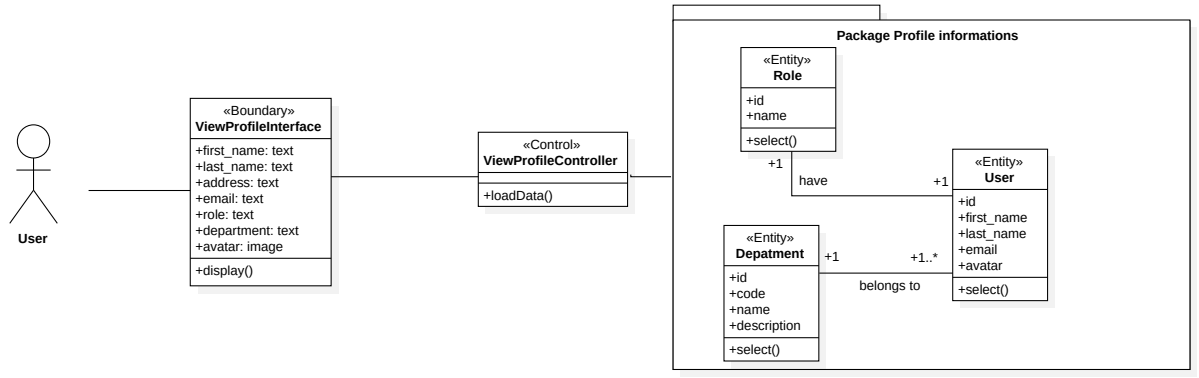


Figure 6.7: "View my profile" Class diagram

Sequence diagram for the "View my profile" use case is shown in Figure 6.8.

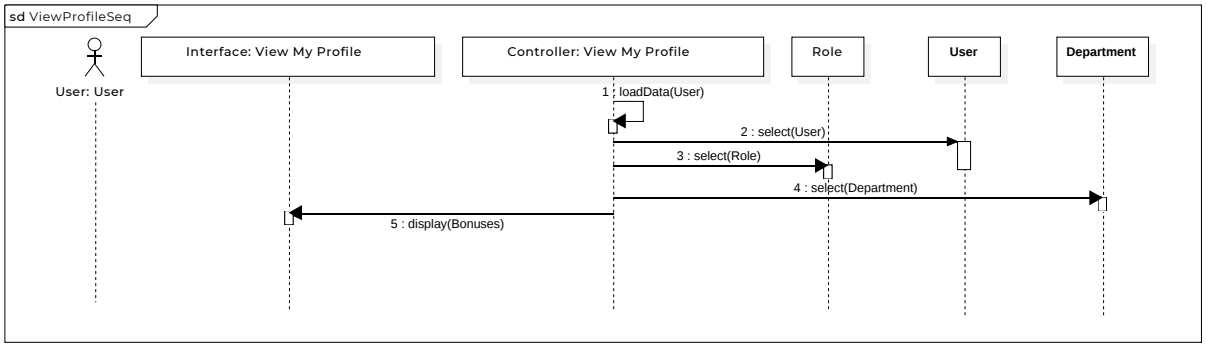


Figure 6.8: "View my profile" Sequence diagram

6.4.3 "Chat with other users" Use Case Design

Class diagram for the "Chat with other users" use case is shown Figure 6.9.

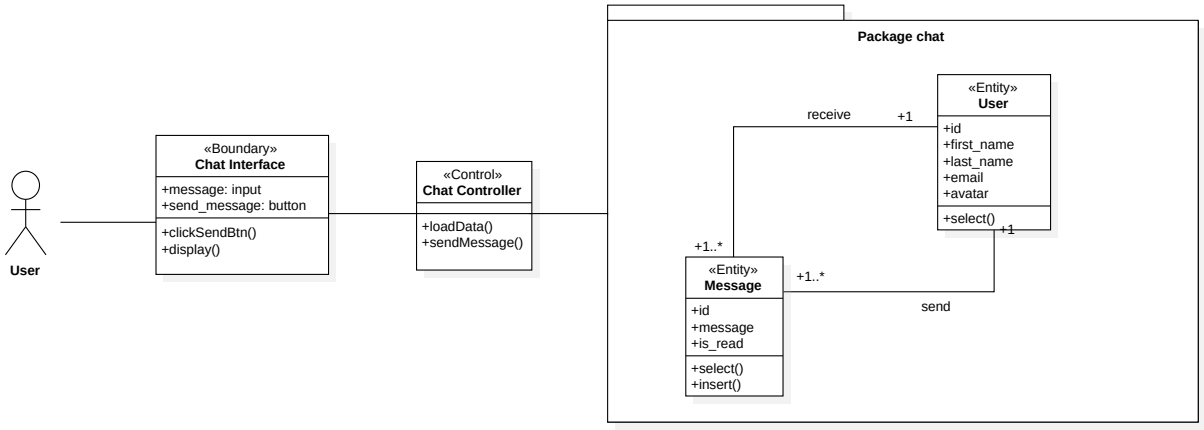


Figure 6.9: "Chat with other users" Class diagram

Sequence diagram for the ”Chat with other users” use case is shown in Figure 6.10.

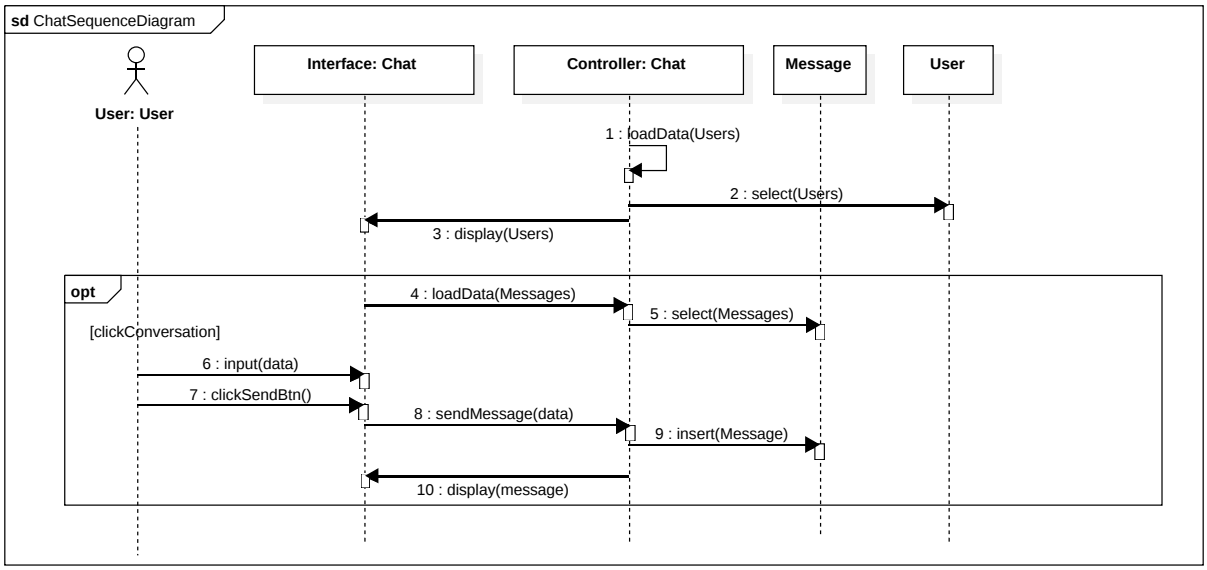


Figure 6.10: ”Chat with other users” Sequence diagram

6.4.4 “Manage company policies” Use Case Design

Class diagram for the ”Manage company policies” use case is shown in Figure 6.11.

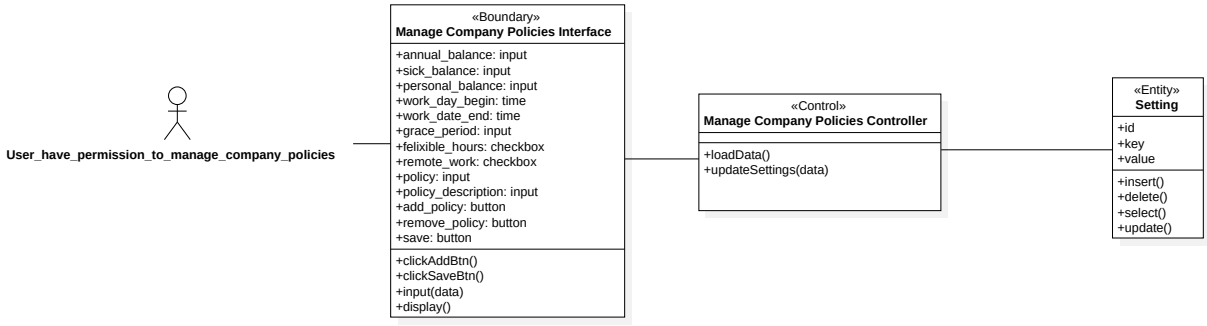


Figure 6.11: ”Manage company policies” Class diagram

Sequence diagram for the ”Manage company policies” use case is shown in Figure 6.12.

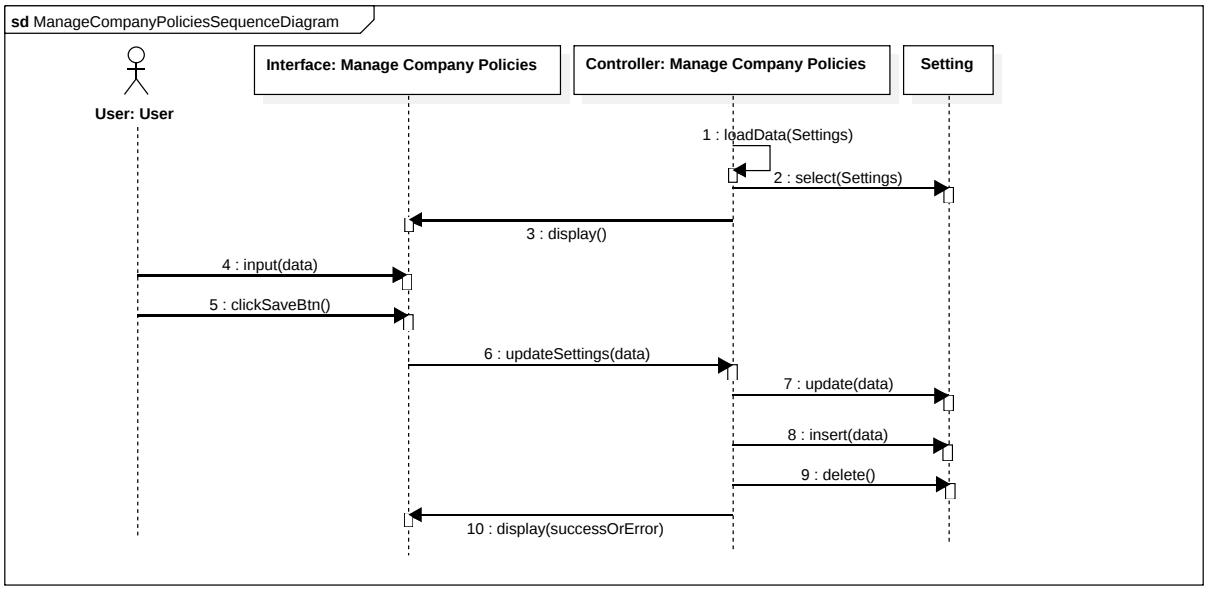


Figure 6.12: "Manage company policies" Sequence diagram

Global Class Diagram

The figure 6.13 concludes the the global class diagram of our project

- Deparments contains many users.
- Deparments may have only one user as a chief.
- Users can send their feedback.
- Users are assigned roles.
- Roles have permissions (RolePermissions class).
- Each permission have specific action and module pairs.
- Action class contains only static values (view, edit, add , delete).
- Module class contains only static values (functionalities of the website).
- Requests must be validated by users.
- Events, messages, and notifications are related to users.
- Setting can be updated but we don't keep the modifier id.

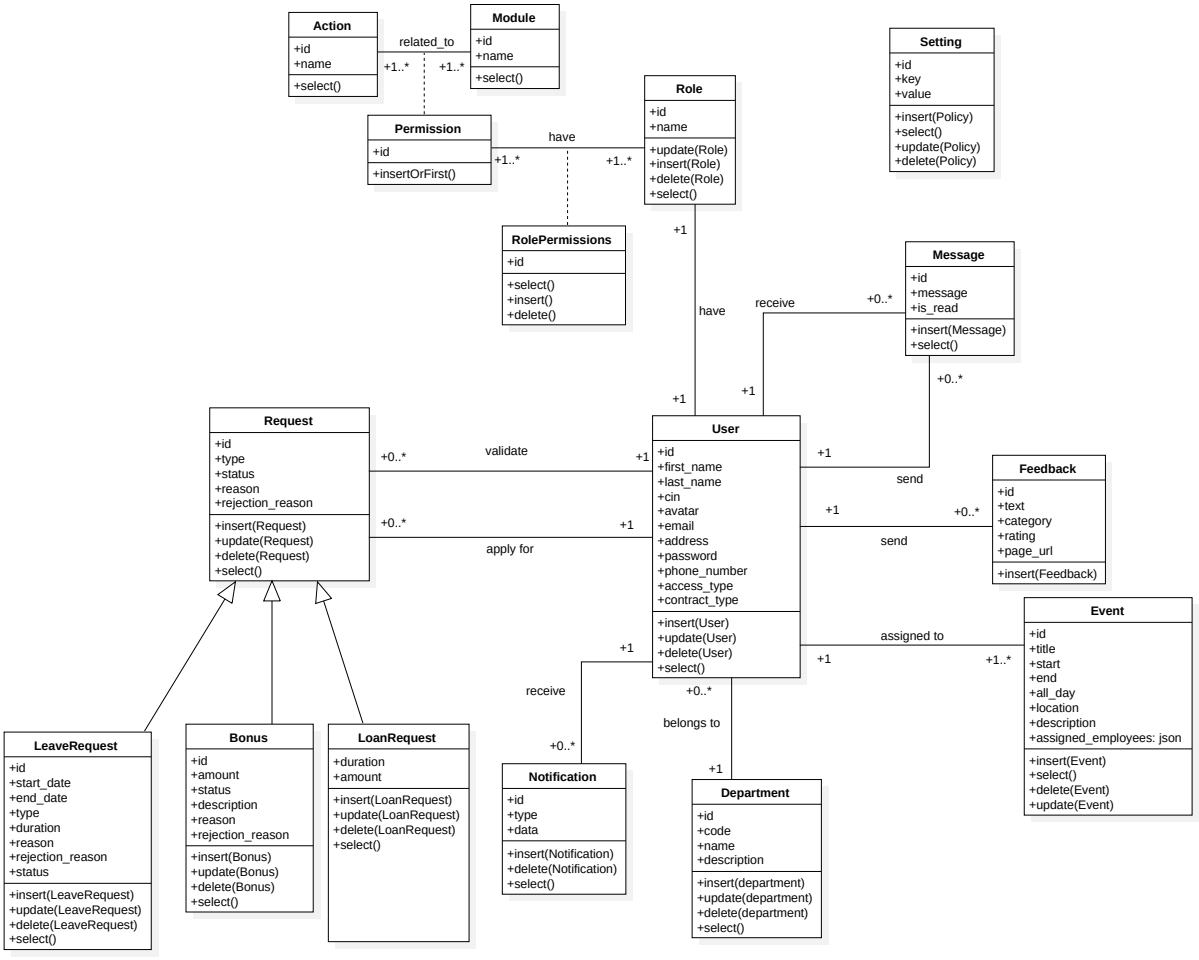


Figure 6.13: Global Class Diagram

6.5 Implementation of sprint 3

In this section, we present the interfaces of each use case of this fourth sprint accompanied by a short description to explain how users interact with it.

6.5.1 Realization of the “Manage events” use case

In this interface, the user can view the calendar that contains all the events, as shown in Figure 6.14.

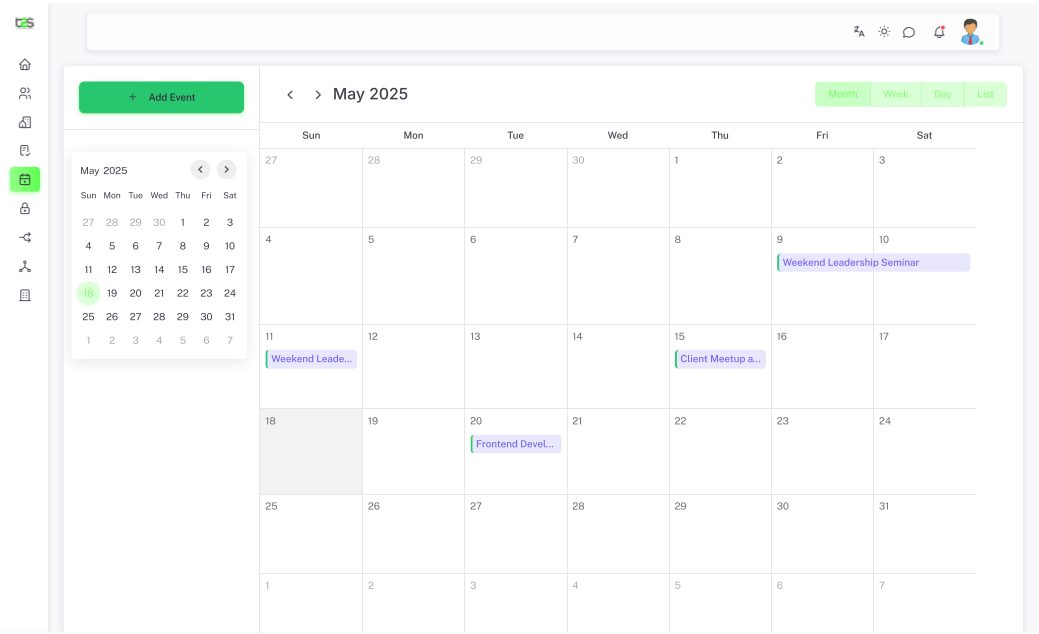


Figure 6.14: "Manage events" Interface

In this interface, the user can view the data related to a specific event, as shown in Figure 6.15.

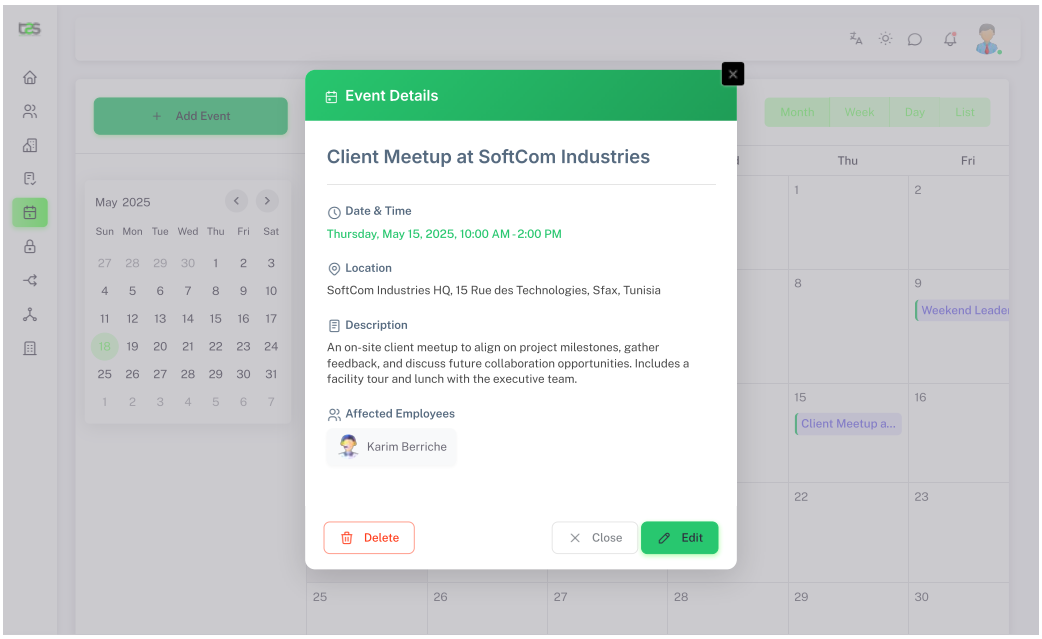


Figure 6.15: "View event" Interface

In this interface, the user enters the new event data (title, start date, end date ...) then clicks on "Add Event" so the new event is created, as shown in Figure 6.16.

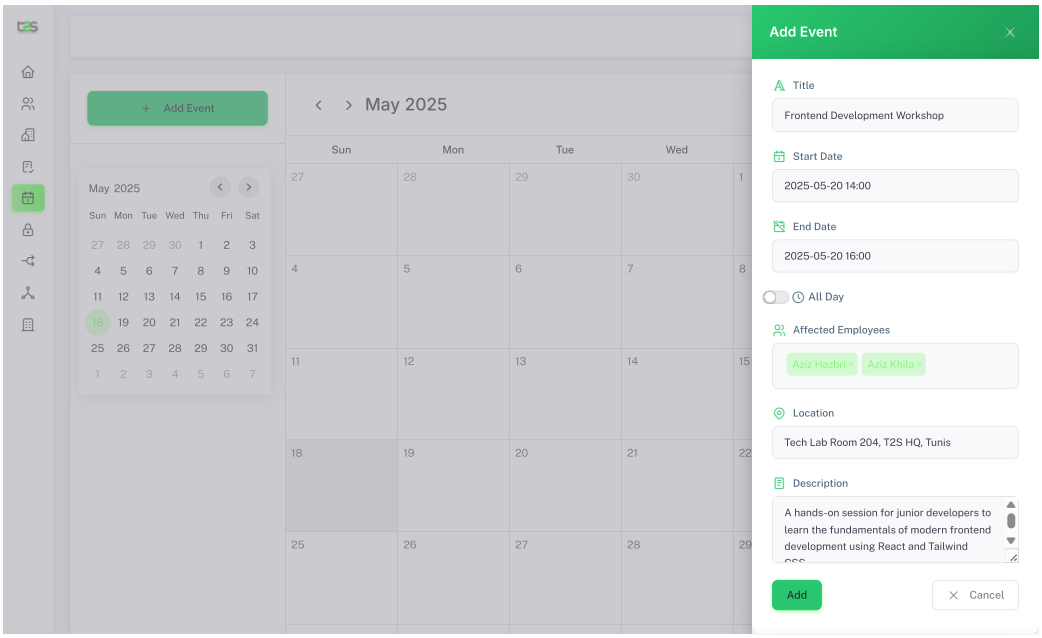


Figure 6.16: "Add event" Interface

The user clicks on the delete button, a confirmation modal appears, then the user clicks on the "Yes, Delete" button so the targeted event is deleted, as shown in Figure 6.17.

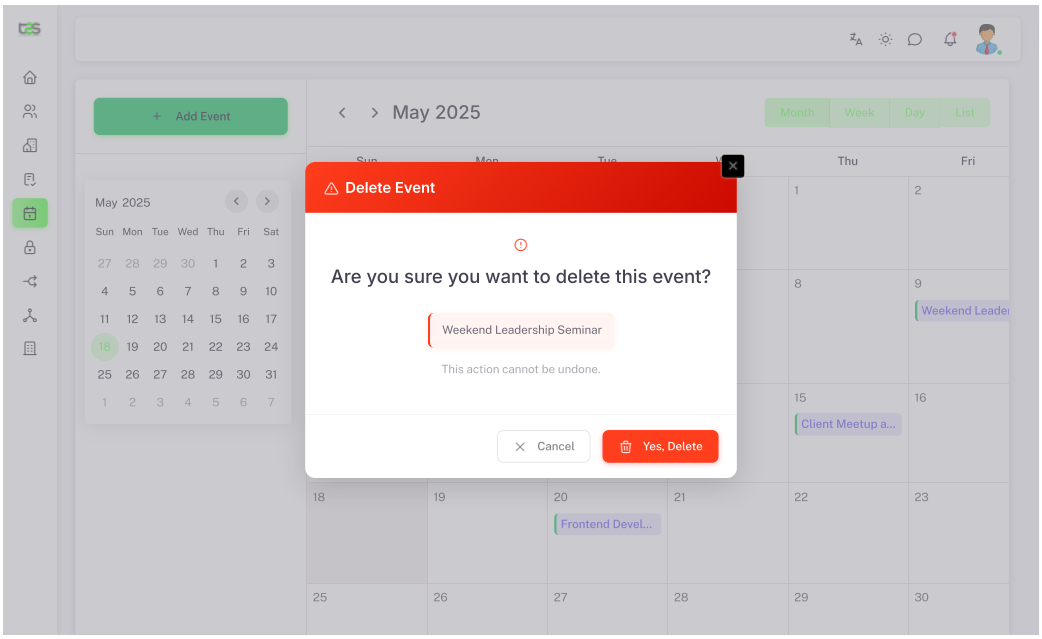


Figure 6.17: "Delete event" Interface

In this interface, the user updates the desired data, then clicks on the "Update Event" button so the targeted event is updated, as shown in Figure 6.18.

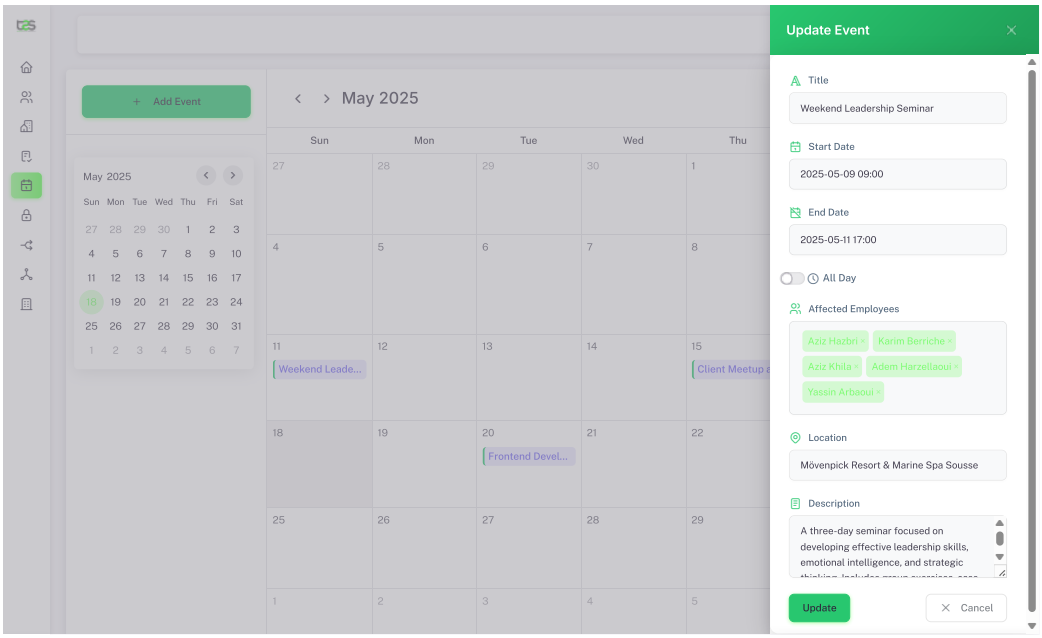


Figure 6.18: "Edit event" Interface

6.5.2 Realization of the "View my profile" use case

In this interface, the user can view his information and contact information of his manager, as shown in Figure 6.19.

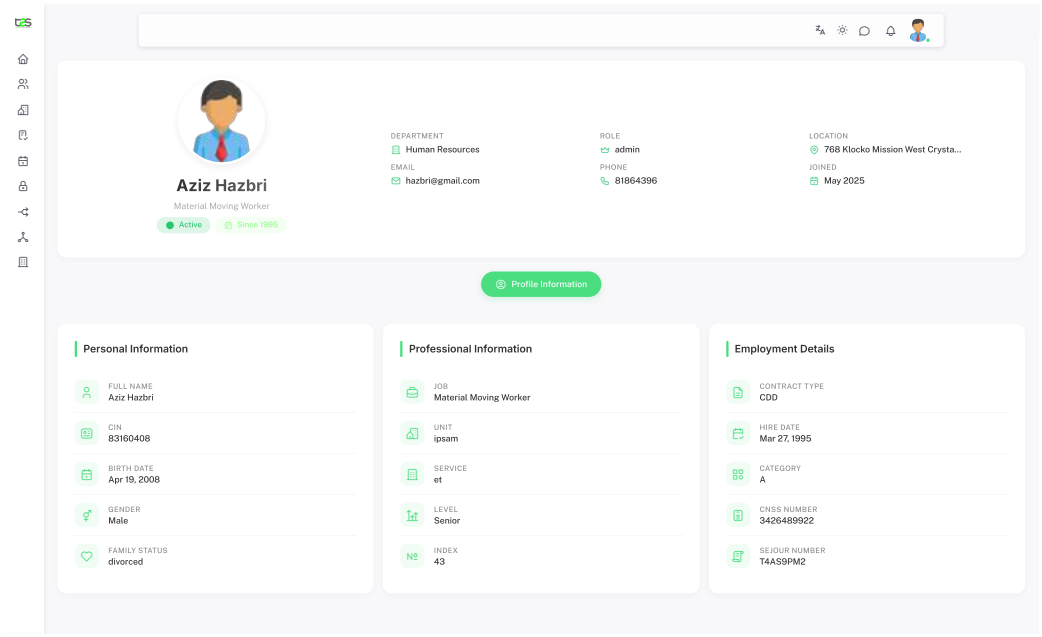


Figure 6.19: "View my profile" Interface

In this interface, the user can change his password by typing the old password and the new one, as shown in Figure 6.20.

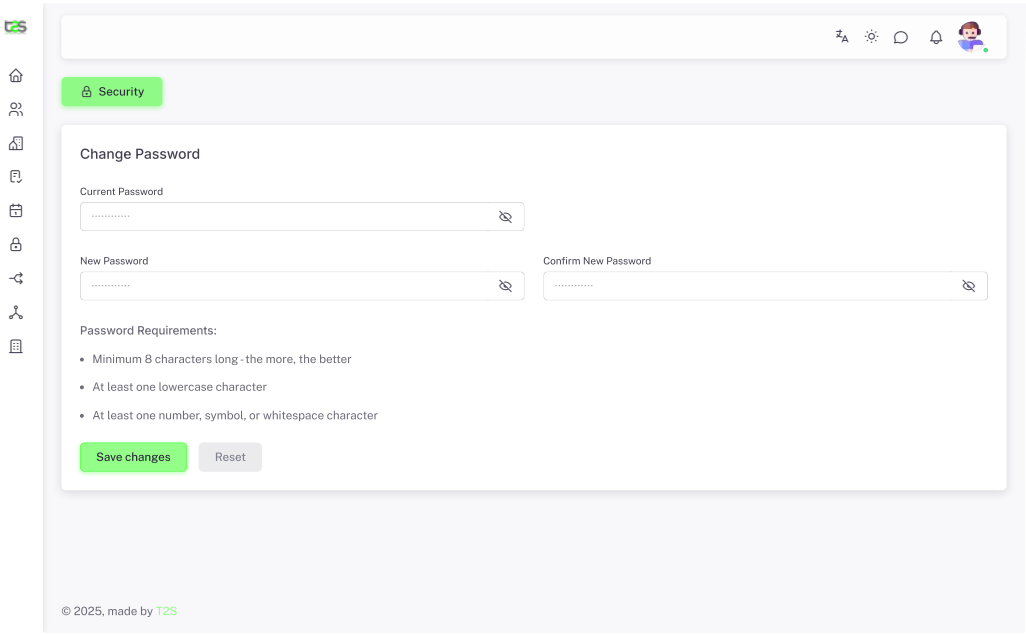


Figure 6.20: "Change password" Interface

6.5.3 Realization of the "Chat with other users" use case

In this interface, the user can select a user to chat with and send him messages, as shown in Figure 6.21.

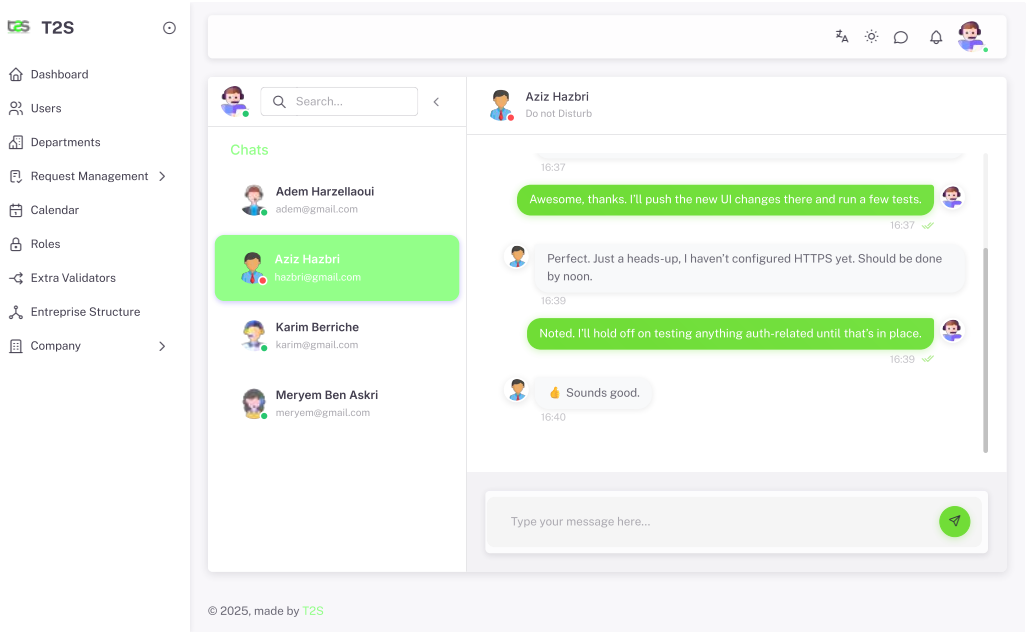


Figure 6.21: "Chat with other users" Interface

6.5.4 Realization of the "Manage company policies" use case

In this interface, the user can view the company policies information, as shown in Figure 6.22.

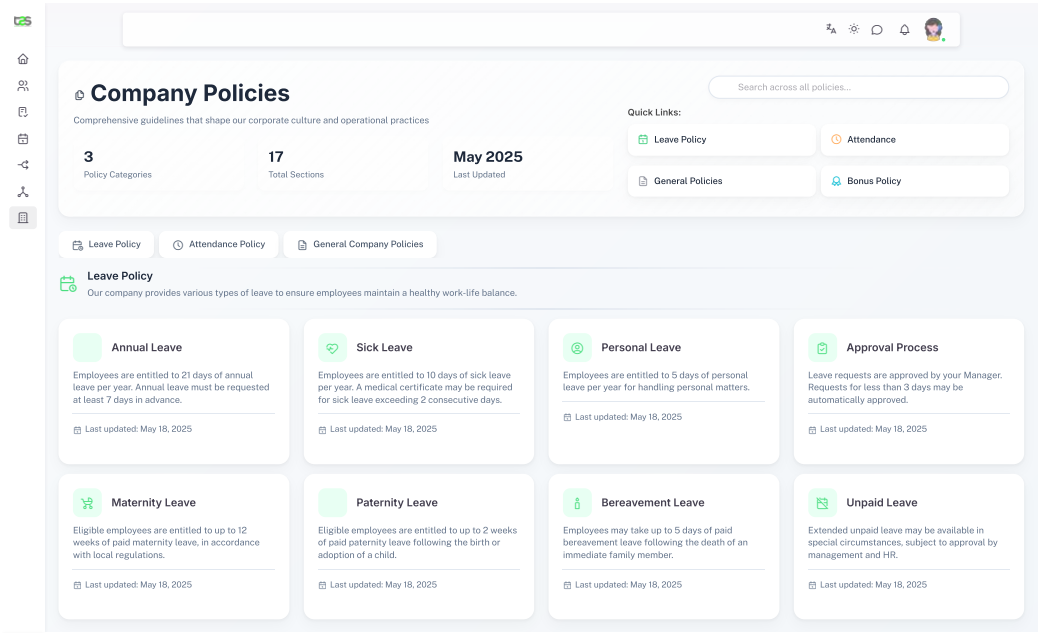


Figure 6.22: "View company policies" Interface

In this interface, the user can manage the company policies, as shown in Figure 6.23.

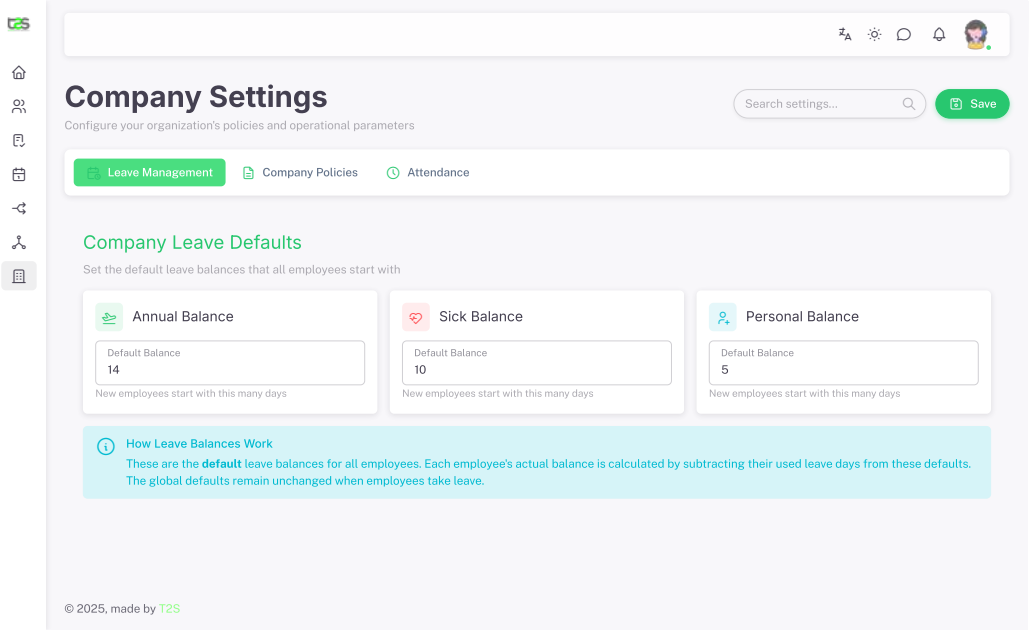


Figure 6.23: "Manage company settings" Interface

6.5.5 Realization of the "View dashboard" use case

In this interface, the user can view the dashboard, as shown in Figure 6.24.

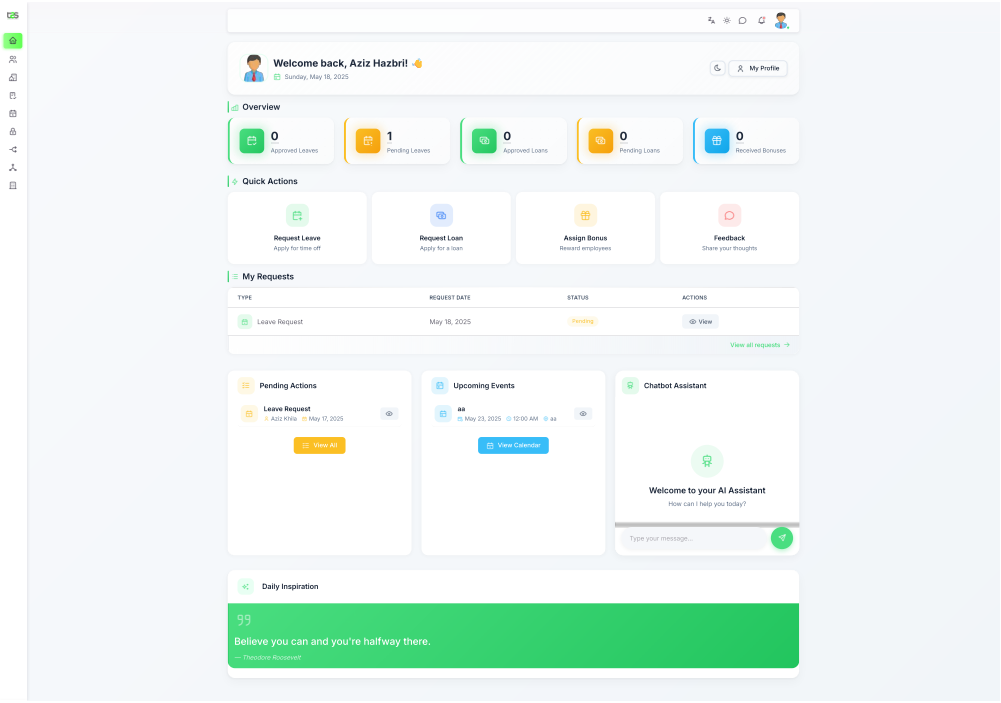


Figure 6.24: "View dashboard" Interface

In this interface, the user can share his feedback via quick actions in the dashboard, as shown in Figure 6.25.

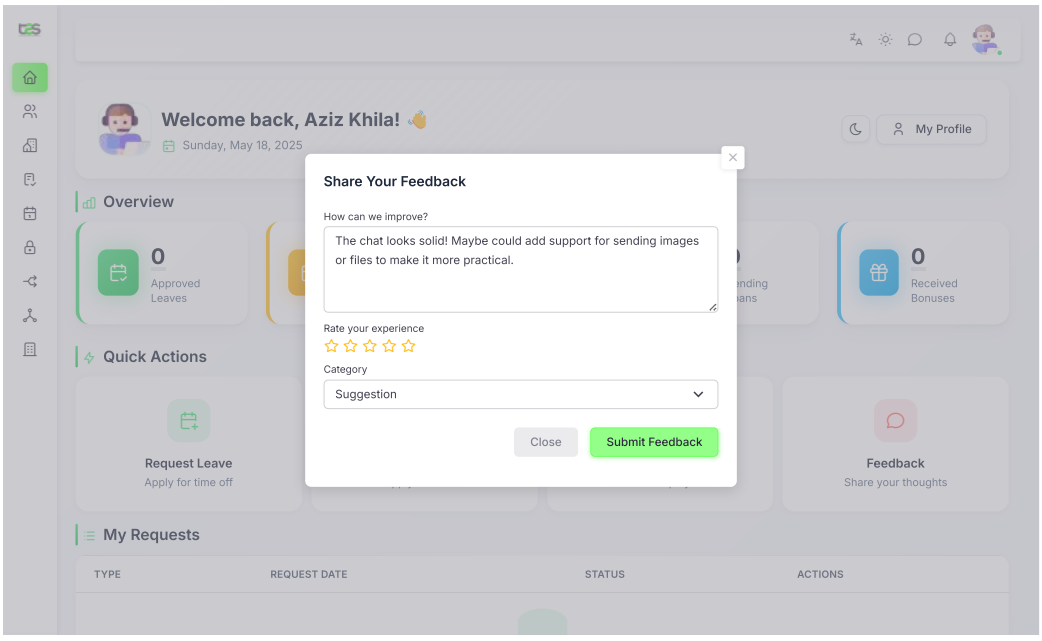


Figure 6.25: "Share feedback" Interface

6.6 Conclusion

In this chapter, we present the refinement, the design, and the interfaces of use cases of sprint 3.

General Conclusion

Throughout this end-of-studies project, we have had the opportunity to model and develop an administrative web application for the T2S company, adapted for effective human resource management. Solutions here assist with a number of pain points seen in the systems in current use, as flexible role-based access, improved validation workflows, and a user-friendly interface. Across a number of sprints, we developed major features such as user and department management, salary bonus distribution, leave and loan management, and real-time communication between employees through an integrated chat system.
We think this application, as it is and as it will continue to develop, can significantly enhance the productivity and satisfaction of employees and HR personnel at T2S.

This project allowed us to consolidate and extend our software engineering skills, from backend programming with Laravel to frontend design, with Agile Scrum methodology informing the iterative development and delivery. Beyond the technical skills, we also developed our communication, teamwork, and professional collaboration skills during our internship at T2S.

Looking ahead, we look forward to further developing this platform. The future vision will include an integration of an intelligent chatbot that will assist users with HR-related questions and navigation, greater user autonomy and system utilization, and a promotion and advancement tracking module that would leave employee development open and merit-based.

Netography

[1] T2S. https://t2s.com.tn/about-us/, consulted on February 27, 2025.

[2] Odoo. https://www.odoo.com/documentation, consulted on March 11, 2025.

[3] Workday. https://www.workday.com/content/web/documentation, consulted on March 11, 2025.

[4] Laravel. https://laravel.com/docs, consulted on March 11, 2025.

[5] HTML. https://developer.mozilla.org/docs/web/html, consulted on March 11, 2025.

[6] CSS. https://developer.mozilla.org/en-us/docs/web/css, consulted on March 11, 2025.

[7] JS. https://developer.mozilla.org/docs/web/javascript, consulted on March 11, 2025.

[8] Git. https://git-scm.com/doc, consulted on March 11, 2025.

[9] GitHub. https://docs.github.com/github, consulted on March 11, 2025.

[10] VSCode. https://code.visualstudio.com/docs, consulted on March 11, 2025.

[11] Postman. https://learning.postman.com/docs/, consulted on March 11, 2025.

[12] SSMS. https://docs.microsoft.com/sql/ssms, consulted on March 11, 2025.

[13] StarUML. https://staruml.io, consulted on March 11, 2025.